Université Libre de Bruxelles
Cours "Systèmes d'information distribués"
Année Académique 2000-2001
Examen de Première Session

1. Deductive Databases

Consider a variation of the Energy Management System seen at the course managing an electrical power distribution network. The network is composed of sites and branches connecting site pairs. The sites can be power stations, intermediate nodes, and final users. Several wires are placed along the branches. Wires are of a given type and each type carries a maximum power. A deductive database schema for this application is as follows:

- intNode (NodeId, BranchIn, CurrentPower)
- user(UserId, BranchIn, CurrentPower)
- distributor(NodeId, CurrentPower, MaxPower)
- branch (BranchId, FromSite, ToSite, CurrentPower, Length)
- wire (WireId, FromSite, ToSite, WireType, CurrentPower)
- wireType (WireType, MaxPower)

It is supposed that the configuration is correct, i.e., there are no cycles in the network, each user is connected to exactly one distributor, the current power of a node and its branches is balanced, the current power of a wire does not exceed its capacity, and the current power of a distributor does not exceed its capacity.

Define the following predicates using Datalog rules:

a) safeBranch (B) representing the fact that all wires in the branch are at most at 80% of its maximal power.
b) distrib(U,D) representing the fact that D is the distributor to which is connected a user U.
c) listIntNodes(U,L) representing the fact that L is the list of intermediate nodes connecting a user U to its distributor.
d) distanceDistrib(U,Di) representing the fact that Di is the distance from a user U to its distributor D.
e) nbUsersDistrib(D,N) representing the fact that N is the number of users connected to distributor D.

2. Temporal Databases

Consider the following TSQL2 schema belonging to a car insurance company.

- **Client**(ClientNo, FirstName, LastName, Address, TelNo)
- **Car**(RegistNo, Model, Year, ClientNo) AS VALID STATE DAY
  ClientNo references Client.ClientNo
- **Insurance**(InsNo, InsType, RegistNo, Price) AS VALID STATE DAY
  RegistNo references Car.RegistNo
- **Accident**(AccNo, RegistNo, Location, Description) AS VALID EVENT DAY
  RegistNo references Car.RegistNo
- **Repair**(RegistNo, AccNo, Description, Cost) AS VALID STATE DAY
  RegistNo references Car.RegistNo
  AccNo references Accident.AccNo

where :

- Table Client contains the persons having a car covered by an insurance policy.
- Table Car contains the information describing the cars. The temporality describes the period that a client owns the car.
- Table Insurance contains the information of the insurance policies for the cars. The temporality describes the period in which a policy was valid on a car.
- Table Accident contains the accidents occurred for a car. The temporality describes the instant at which the accident occurred.
- Table Repair contains the repairs that have been done on cars after an accident. The temporality describes the periods of time in which the repair was realized. Notice that a repair can be realized several months after the accident, even if the insurance policy has expired in the meantime.

Write the following queries

a) Give the accident number of accidents that were not covered by an insurance policy.
b) Give the first name, last name, and address of clients such that one of their insurance policies expired last month (March 2001).
c) Give the first name, last name, and address of clients such that all the cars they currently own are covered by an insurance policy.
d) Give for each insurance policy the insurance number and the total cost of all its repairs.
e) Give for the accidents occurred in year 2000 the average time that the clients waited for the corresponding repair.

3. Active Databases

Consider the following tables in a relational database of a theater company.

- Play (PlayId, Title, Author)
  This table contains the theatre plays written by an author, e.g.,
  Play (P1, 'Ménage à trois', 'Victor Hugo')
- PlayRole (PlayId, Role)
  PlayId references Play.PlayId
  This table contains the set of roles for a play, e.g.,
  PlayRole(P1, 'The Husband')
  PlayRole(P1, 'The Wife')
  PlayRole(P1, 'The Lover')
- Setting (SettingId, PlayId, Troupe, Season, Director)
  PlayId references Play.PlayId
  This table materializes plays, giving production decisions for a theatrical season, e.g.,
  Setting (S1, P1, 'Artists United', 2001, 'Peter Brooks')
- SettingRole (SettingId, Role, Actor)
  SettingId references Setting.SettingId
  This table contains the set of actors that may play each role in a play, e.g.,
  SettingRole (S1, 'The Husband', Delon)
  SettingRole (S1, 'The Husband', Shariff)
- Performance (PerfId, SettingId, Title, Date, NbAttend)
  SettingId references Setting.SettingId
  This table materializes settings, at a particular date with a number of persons attending the performance, e.g.,
  Performance (P1, S1, 'Ménage à trois', 10/01/2001, 557)
- PerfRole (PerfId, Role, Actor)
  PerfId references Performance.PerfId
  This table keeps track of the actor that played a particular role in a performance, e.g.,
  PerfRole(P1, 'The Husband', Delon)

Consider the following integrity constraints.

- The roles for a setting are those defined in the corresponding play in table PlayRole.
- The actor chosen for a role in a performance is one among those that may play that role according to table SettingRole.

Consider also the following derived rule.

- Attribute Title of Performance is a derived attribute corresponding to the title of its play.

a) For each one of the above constraints:
   – determine all the events that may cause a violation of the constraint, and
   – derive in Starburst a trigger that rollbacks the transaction when one of these events occurs.
b) Determine all the events that may cause an update of the derived attribute. Derive in Starburst a trigger that computes the attribute when one of these events occurs.

4. Object Databases

Consider the following ODMG schema used by the Internet provider "LibertyWorld" to manage the information about its clients and their connections.

```
Struct Address { string street; string number; string zip; string town; }
class Client (extent Clients) {
        attribute String clientNo;  attribute String firstName;
        attribute String lastName; attribute Address address;
        attribute SubscriptionType subscription;
        relationship list<Account> accounts inverse Account::client;
 }
class SubscriptionType (extent Subscriptions) {
        attribute String subsCode;
        attribute Integer maxNbAddresses; attribute Integer maxMB
 }
 class Account (extent Accounts) {
        attribute String login; attribute Integer currentMB;
        relationship Client client inverse Client::accounts;
        relationship list<Connection> connections inverse Connection::account;
 }
 class Connection (extent Connections) {
        attribute Date conDate; attribute Time conTime;
        attribute Integer Duration;
        relationship Account account inverse Account::connections;
        relationship list<Site> sitesVisited inverse Site::connections;
 }
 class Site (extent Sites) {
        attribute String name; attribute String httpAddress;
        attribute String category;
        relationship list<Connection> connections inverse Connection:: sitesVisited;
 }
```

Notice that

- In class SubscriptionType each type is identified by a code, and to each type corresponds a maximum number of email addresses and disk space (e.g., for subsCode = 'WeekEnd' maxNbAddresses = 10 and maxMB = 5).
- In class Account, attribute currentMB keeps track of the current disk space used by each account.
- Attribute Duration in class Connection stores the number of minutes that the connection took.

Write in OQL the following queries.

a) Give the name of clients that realized a connection in March 2001 and that one of their accounts uses more disk space than that authorized by its subscription type
b) Give the logins that have not realized a connection since March 1st 2001
c) Give the login and the duration of the longest connection since January 1st 2001
d) For each client, give the client number and a structure showing for each of his logins the sum of durations of his connections during the December 2001 using that login, if this sum is greater than 10h.
e) Give the client number of clients that all sites they visited are of category "Adult"