CS 246 Exercise Set #3

In this assignment, you will explore different data formats and learn the basics of Infomaster. Your group will present your work on Exercise 9 in class on Thursday, April 15. Please also email your answers to Exercises 7 and 8 mkassoff@cs before class that day.

Exercise #7

A few written questions.

Background for exercises 7.1 and 7.2:

Consider the following schema and extension for a linked list:

node.instance	node.next	lastnode.instance
(node.instance n1)	(node.next n1 n2)	
(node.instance n2)	(node.next n2 n3)	
(node.instance n3)	(node.next n3 n4)	
(node.instance n4)		

That is, node n4 is the last node in the list, because it does not have a next node. Therefore (lastnode.instance n4) should be computed in this case. It is your job for questions 7.1 and 7.2 to write rules that populate lastnode.instance (for arbitrary linked lists). The point of this exercise is for you to realize why negation-as-failure can always be simulated using semipostive negation. Your rules must be safe, i.e. (<= (lastnode.instance ?x) (node.instance ?x) (not (node.next ?x ?y)) is not a solution!

7.1) (5 points) Write rules to compute lastnode.instance assuming negation as failure semantics for negation. You can introduce new predicates/attributes if you like.

7.2) (5 points) Do the same as in 8.1, but assume semipostive semantics for negation. Again, you can introduce new predicates/attributes if you like.

7.3) (5 points) What are the inverse rules for the following resource definitions:

(<= (greatgrandparent ?w ?z) (parent ?w ?x) (parent ?x ?y) (parent ?y ?z))

(<= (clothes ?y) (wears ?x ?y))
(<= (clothes ?z) (= ?z pants))

Exercise #8 (50 points)

In this exercise you will integrate two sources of university data. One source of data is the university dataweb, which you will find in your examples directory. (The data is in the University Agent, and the rules are in the Library agent.) The other source is the Stanford Computer Science Department data, which you can find on the website. Your pivot schema is the following:

Class	**Attributes**
Thing	<none>
Person	person.firstname, person.lastname, person.image, person.affiliation, person.office, person.telephone, person.email, person.url
Faculty	person attributes + faculty.rank, faculty.administrator
Staff	person attributes
Administrator	staff attributes + administrator.faculty
Janitor	staff attributes
Technician	staff attributes
Student	person attributes + student.year, student.major
Location	(none)
Building	building.name, building.office, building.classroom
Room	room.building, room.number, room.floor
Office	room attributes + office.occupant
Classroom	room attributes + classroom.feature, classroom.capacity, classroom.telephone
Floor	(none)
Appliance	appliance.classroom
Organization	<none>
University	university.name, university.school
School	school.name, school.dean, school.department, school.university
Department	department.school, department.chair, department.group
Group	group.department, group.name, group.director
Rank	rank.faculty
Year	year.student

You can make the following assumptions:

1) The name of the example university is "Harvard." The name of the other university is "Stanford."

2) All of the Stanford offices and classrooms are in the Gates building.

3) All groups in the Stanford data are part of the Computer Science Department.

4) All of the people in the Stanford data are affiliated with the Computer Science Department, which is in the School of Engineering. All Stanford students are Computer Science majors.

5) You can assume that all of the Stanford staff are administrators. Secretaries in the University example are administrators in the pivot schema.

6) The dean of the Stanford engineering school is James Plummer.

7) The chair of the Stanford Computer Science department is Hector Garcia-Molina.

8) The class hierarchy is as follows:

```
Thing
    Person
        Faculty
        Staff
            Administrator
            Janitor
            Technician
        Student
    Location
        Building
        Room
            Office
            Classroom
        Floor
    Appliance
    Organization
        University
        School
        Department
        Group
    Rank
    Year
```

You cannot change the pivot schema in any way. You can change the given data, but only to disambiguate the attribute names (i.e. change person.instance to

stanford.person.instance and harvard.person.instance in the two cases). You should then write query rules to define the pivot schema in terms of the other two schemas.

Exercise #9 (60 points)

Populate your datawebs with real data. You should find a subset of the data that is complete. For instance if your dataweb is about movies and actors then "all movies since 2002 and actors that acted in those movies" would be complete, whereas "some movies and the actors that acted in them" would not be complete. Your data should also be sound- for example, in the above example don't include any movies from before 2002.

You should put your data into a form that is loadable by the system- for example, XML. Then you should load it into the repository and write rules to populate your dataweb. For those of you who have data in HTML, one way to get it into the system is to add close tags and make it into XHTML (or at least well-formed XML).