CS322 Fall 1999
Module 7 (Knowledge Representation Issues)
Assignment 7

Solution.

The aim of this assignment is to learn about some knowledge representation issues that will be used later in the course.

Question 1

Consider the relations:

- *course*(*Id*, *Year*, *Inst*, *Room*, *Limit*) that is true if the course with identifier *Id*, in year *Year* has instructor *Inst*, is held in room *Room* and has a limit of *Limit* students.
- *limit*(*Id*, *Year*, *Limit*) that is true if the course with identifier *Id*, in year *Year* has a limit of *Limit* students.

Consider the knowledge base:

```
limit(Id,Year,Limit) <-
   course(Id,Year,Inst,Room,Limit).
course(cs322,1999,david,cicsr208,120).
course(cs322,1998,craig,cicsr208,100).
course(cs327,1999,jim,cicsr202,50).
```

(a) Give the knowledge base where we represent the *course* relation using the object-attribute-value representation. (I.e., specify the above three facts for *course* in terms of the *prop* relation.)

(b) Define *limit* in terms of this new representation for the course information. (The *limit* relation should have the same semantics as before.)

(c) Explain why it may be advantageous to use the object-attribute-value representation for course information.

Check that your representation works with CIlog. Your new axiomatization should be able to answer exactly the same *limit* queries as the original version.

Solution

(a) Here is a CILog axiomatization of the *course* relation using the object-attribute-value representation. I have tried to give meaningful names for the individuals I created:

```
prop(cs322_1999,identifier,cs322).
prop(cs322_1999,year,1999).
prop(cs322_1999,instructor,david).
prop(cs322_1999,room,cicsr208).
prop(cs322_1999,limit,120).
prop(cs322_1998,identifier,cs322).
prop(cs322_1998,year,1999).
prop(cs322_1998,instructor,craig).
prop(cs322_1998,room,cicsr208).
prop(cs322_1998,limit,100).
prop(cs327_1999,identifier,cs327).
prop(cs327_1999,year,1999).
prop(cs327_1999,instructor,jim).
prop(cs327_1999,room,cicsr202).
prop(cs327_1999,limit,50).
```

(b) *limit* cen be defined using:

```
limit(Id,Year,Limit) <-
   prop(Obj,intentifier,Id) &
   prop(Obj,year,Year) &
   prop(Obj,limit,Limit).
```

Question 2

Suppose a conditional expression is either:

- a value, where a value is either a number or is a Boolean value (*true* or *false*); or
- is of the form *if* (*Att*, *Then*, *Else*) where *Att* is a Boolean attribute, and *Then* and *Else* are conditional expressions.

You are to write a relation

- *ceval*(*Obj*, *CE*, *Val*) where *Obj* is an object, *CE* is a conditional expression and *Val* is the resulting value of evaluating the conditional expression for the individual *Obj*.

To evaluate a conditional expression for an individual is simple: if the conditional expression is a value, then that value is returned. If the conditional expression is of the form *if* (*Att*, *Then*, *Else*), then if the value of the attribute *Att* for the individual is *true*, you evaluate the *Then* conditional expression, otherwise evaluate the *Else* conditional expression.

For example, given the knowledge base:

```
prop(cs322,fun,true).
prop(cs322,easy,false).
prop(cs322,interesting,true).
prop(cs322,confusing,true).
```

Then *ceval*(*cs*322, *if* (*fun*, *if* (*easy*, 99, 80), *if* (*confusing*, 55, 70)), *Val*) should return $Val = 80$.

Solution

Here is the CILog axiomatization:

```
ceval(Obj,V,V) <- value(V).
ceval(Obj, if(Att,Then,Else), V) <-
   prop(Obj, Att, true) &
   ceval(Obj, Then, V).
ceval(Obj, if(Att,Then,Else), V) <-
   prop(Obj, Att, false) &
   ceval(Obj, Else, V).
value(N) <-
   number(N).
value(true).
value(false).
```