CSC 416

Question Set 3a

Jacob Peck

A set of questions based on chapter 3 of Coppin's Artificial Intelligence Illuminated. This chapter deals with knowledge representation, and how it pertains to the field and study of AI.

GRADE = (X + N + LF + A + R) = (50 + 14 + 5 + 4 + 22) = 95

1. With respect to AI, what is a **representation**? (1)
   - A representation is a model of how to represent knowledge within a computer, so that a problem may be solved given the right methods.

   Points: 1

2. Why are representations significant with respect to AI? (1)
   - Representations are significant to AI because the representation chosen for a particular set of data directly affects the methods used in tackling the problem.

   Points: 1

3. In a semantic network, what do the nodes represent? What do the arcs represent? (2)
   - In a semantic network, the nodes represent objects (either instances or classes) and the arcs represent relationships between the objects.

   Points: 2

4. Consider the following property list approach to representing semantic nets in LISP. If C1 is related to C2 by R, then the pair R/C2 is placed on the property list of C1. Write a sequence of LISP instructions to create the semantic network depicted in Figure 3.1 using this representation. (Write “is a” as “isa”.) (5)

```
(setf (get ‘BOB ‘ISA) ‘BUILDER)
(setf (get ‘BOB ‘OWNS) ‘FIDO)
(setf (get ‘BOB ‘EATS) ‘CHEESE)
(setf (get ‘MICE ‘EATS) ‘CHEESE)
(setf (get ‘FIDO ‘ISA) ‘DOG)
(setf (get ‘FIDO ‘CHASES) ‘FANG)
(setf (get ‘FANG ‘ISA) ‘CAT)
(setf (get ‘FANG ‘CHASES) ‘MICE)
```

   Points: 5

5. In the context of data modeling, what is **inheritance**? (1)
   - Inheritance is the idea that one form inherits all of the properties of a “parent” form.

   Points: 1

6. What is a **frame system**? In answering this question, be sure to mention *slots* and *frame relations*. (2)
   - A frame system is a collection of frames with relationships between them. A frame consists of slots which hold values, and these values might be references to other frames, in order to create a sort of relationship.

   Points: 2

7. In the context of frames, indicate the more formal names given to the “isa” relation and the “partof” relation. How are all other relations classified? (3)

- The “isa” relation is referred to as either the “instance-of” or “subclass-of” relation, depending on the context. The “partof” relation is the “property-of” relation. Other relations are defined in terms of these two “atomic” relations.

Points: 2

8. Give three examples of “frame system programming languages” – programming languages which can be viewed as a particular interpretation of the frame system knowledge representation. Simply name the languages. (1)
   - Java, C++, APL2

Points: 1

9. In Java, how do you create a “frame” that is an *instance frame*? How do you create a “frame” that is a *class frame*? (2)
   - To create an instance frame in Java, you instantiate a new object with the `new` keyword, in a manner like: `Animal animal1 = new Animal();`. To create a class frame in Java, you define a new class, using the `class` keyword, in a manner like: `public class Monkey extends Animal { … }`.

Points: 2

10. What is the difference between an ordinary *procedure* and a *demon*? (2)
    - An ordinary procedure is a method/function/subroutine that is called explicitly, while a demon is a procedure that is automatically called whenever a particular piece of data is acted upon. There are two major classes of demons, WHEN-READ and WHEN-CHANGED.

Points: 2

11. What is the difference between *procedural semantics* and *declarative semantics*? (1)
    - Procedural semantics have the side effect that a system will behave based upon the order that data and rules are received, whereas declarative semantics are deterministic as far as order of input goes.

Points: 1

12. In “abstract” frame systems, the “isa” relation is sometimes used a bit too casually for some people and purposes. In particular, it is used both to relate objects to classes and classes to other classes. In Java, clarity is achieved through the mechanisms of construction which result in the sometimes overused “isa” being replaced by well-defined “instance-of” and “subclass-of” relations – which, respectively, relate objects to classes and classes to other classes. How does one model “instance-of” in Java? How does one model “subclass-of” in Java? (2)
    - The model for “instance-of” is the instantiation keyword `new`. The “subclass-of” model in Java is represented by the `extend` keyword.

Points: 2

13. In the section on object-oriented programming your author mentions that it is possible to achieve the same results using single inheritance as it is with multiple inheritance. Show that this is so in Java by sketching code to define a class called A which inherits from classes B an C. (Note: do not use the "interface" mechanism). (3)
    - One way is to have B inherit from A, and C inherit from B, in a manner such as:

      ```
      public class A {…}

      public class B extends A {…}

      public class C extends B {// contains functionality of both B and
      A}
      ```

      Points: 2

14. Does Java have a built-in demon mechanism? (1)
    - As far as I can tell, the Java language does not have a demon mechanism built in, but it is relatively simple to code to account for call-back functionality in a manner that would resemble a demon. The JVM, however, must have a demon, in order to perform efficient garbage collection.

      Points: 1