CS135B (Fall 2024) Homework 2

Due October 15, 2024

1. Predicate Logic

Use the predicates:

$$\begin{aligned} A(x,y) &: \text{x admires y} \\ B(x,y) &: \text{x attended y} \\ P(x) &: \text{x is a professor} \\ S(x) &: \text{x is a student} \\ L(x) &: \text{x is a lecture} \end{aligned}$$

And the constant:

$$m : \text{Mary}$$

to translate the following into predicate logic:

(a) Mary admires every professor.

(b) No lecture was attended by every student.

(c) No lecture was attended by any student.

2. Model Checking

Let M be the model pictured in Figure 5.5 of the van Eijck and Unger book. (In other words, let $D = \{1,2,3\}$, $I(P) = \{1,3\}$, and $I(R) = \{(1,1),(1,2),(1,3),(2,2),(3,1),(3,2)\}$.) Which of the following statements are true in the model?

(a) $\exists x(P(x) \land R(x,x))$

(b) $\forall x(P(x) \to \exists y(R(x,y)))$

(c) $\forall x(\exists y(R(y,x)) \to R(x,x))$

3. Tree Derivation

Draw a binary tree structure for the sentence ”Every student bought a gift” with the following rules. Use Quantifier Raising and Quantifier Substitution to derive the meaning of the two readings of the sentence.

1. $[\![every]\!] = \lambda P \lambda Q \forall x.(P(x) \to Q(x))$
2. $[\![student]\!] = \lambda x.(Student(x))$
3. $[\![bought]\!] = \lambda y \lambda x.(Bought(x, y))$
4. $[\![a]\!] = \lambda P \exists x.P(x)$
5. $[\![gift]\!] = \lambda x.(Gift(x))$

4. Functors and Monoids

1. **Implement Monoid for SumInt**
   Write the Monoid instance for the `SumInt` data type given in HW2.hs.
2. **Implement Functor for Tree**
   Write the Functor instance for the `Tree` data type given in HW2.hs.
3. **Practice Using Apply**
   Fill in `f2`, `f3`, and `result`.
4. **List of Functions**
   Call `words` and `pwords` (from HW1) on sentences using `fs` and `<*>`. Store the list of Strings that appear in both lists in `preserved`.

5. Subtypes of Entity

This problem considers ”subtypes” of Entity, and how we can use type classes and instances to constrain the kinds of arguments certain verbs can take.

In HW2.hs you are given code similar to that in Model.hs. Instead of a single Entity type, though, we define separate types Human, Vehicle, and Animal. We then define `Catch` as a type class, and instances for combinations of arguments that make sense for the semantics of the verb ”catch”, i.e., a human can catch a bus (vehicle), and a cat (animal) can catch a mouse (animal), but other combinations are not valid.

Fill in the instance declarations for `Catch Human Vehicle` and `Catch Animal Animal`, such that `catch john bus` and `catch cat mouse` are True, while `catch mouse cat` is False. Other combinations of arguments (not covered by the instances) should output an error.

6. Transitive Verb

Once you've accomplished this, do the same for one more transitive verb whose sense changes depending on the semantics of its arguments (for example, "play", as in "Babe Ruth plays baseball" vs. "Yo-Yo Ma plays the cello", or "serve", as in "Elected officials serve the people" vs. "Olive Garden serves food"). What datatypes do you need to define to make sense of the different senses of the verb? (Feel free to create new types.) As with problem 5, you should be able to allow and disallow certain combinations of arguments for the verb you choose to implement. In your PDF document, please explain what datatypes your chosen verb can take and what restrictions you've placed on them in your code.

Turning in Your Assignment

Submit two files to LATTE: a PDF document containing your answers to problems 1-3 and a description of the verb you implemented in problem 6, and HW2.hs, containing your code for problems 4, 5, and 6.