Department of Computer Science  
CSC 244/444  
Fall 2021  
Assignment 2  
Due: 9:39 am, Thursday, September 23  

Some questions are marked “grads only”, but may be tried by undergrads for bonus points.

Problem 1 (Models and Truth)

A. Make up a specific model $M = (D, I)$, keeping $D$ as small as you can, such that the following sentences are true in $M$ (be sure to keep your metalanguage symbols for elements of $D$ distinct from object language symbols like Odie, Garfield):

$\text{Cat(Garfield), Likes(Odie, Garfield), } \forall x . \text{Cat}(x) \lor \neg \text{Likes(Garfield, x)}$

In supplying the model, be sure you specify the values of $I$ for all the relevant arguments. Establish the truth in $M$ of each of the sentences, justifying your steps using just the truth/satisfaction conditions of the connectives, quantifiers and predications involved. Do not perform any syntactic transformations or manipulations of the formulas, such as those you might have learned for boolean algebra – such manipulations are themselves in need of semantic justification! In the case of ground formulas you can (and should) use truth conditions not involving variable assignment functions.

B. Make up another model $M' = (D', I')$, keeping $D'$ as small as you can, such that the first and third sentences above are false in $M'$. Establish the falsity or truth of each of the sentences in this new model (again, relying only on truth/satisfaction conditions).

C. (grads only) Prove that $\models_M (x = A \land \neg P(x)) [U]$ iff $\models_M (x = A \land \neg P(A)) [U]$

D. (grads only) FOL can be used to “talk about” anything – including symbolic expressions such as those comprising Lisp (or even FOL!). Make an attempt to define (using equivalence, $\Leftrightarrow$) the notion of a basic Lisp S-expression in FOL, i.e., an expression that is either a literal atom or a list of zero or more basic S-expressions. You’re allowed to use equality and the predicate $\text{Expr}$ to say that something is a basic S-expression, $\text{Listp}$ to say that something is a list, and $\text{Atom}$ to say that something is a literal atom.

Then define $\text{Listp}$ via an equivalence as well. For this, you can add the functions $\text{car}$ (or $\text{first}$) and $\text{cdr}$ (or $\text{rest}$), and the individual constant $\text{Nil}$ to the vocabulary. Do not use a $\text{cons}$ function here.

Next write down another axiom to the effect that $\text{Nil}$ is the only atom that is also a list.

Finally, write an axiom that defines $\text{cons}$ in terms of $\text{car}$ and $\text{cdr}$.

You should be aware that the sorts of equivalence axioms suggested here don’t fully “pin down” the meanings of any of $\text{Expr}$, $\text{Listp}$, $\text{Nil}$, $\text{car}$, or $\text{cdr}$. We could add further useful axioms relating the vocabulary introduced so far (and adding $\text{append}$, etc.), but we won’t go
any further here – and indeed if we introduce lambda-expressions and their application to arguments, we cannot fully axiomatize Lisp.\(^1\)

**Problem 2 (Validity and Entailment)**

A. Show that (a) is not valid, while (b) and (c) are valid. In each case, carefully justify your answer (based only on the semantics of the symbols involved, i.e., without syntactically transforming the formulas).

a. Entity(A)  
b. \(\exists x . x = \text{God} \)  
c. \(\forall x . -(A = A) \Rightarrow P(x)\)

d. Though (b) is valid, why doesn’t it establish the actual existence of a God (as various religions might interpret this term)?

B. Which of the following entailments hold? Again justify your answers as above.

a. \(f(A) = B \models \neg(\neg B = f(A))\)

b. \(\{P(A), -(A = B)\} \models \neg P(B)\)?

c. \(\forall x \neg P(x) \models P(A) \Rightarrow P(B)\) ?

d. (grads only) \(\forall x \phi \models \exists x \phi\) ?

\(^1\)This means we can’t provide axioms that will allow us to prove any equality between lambda-expressions (and hence between S-expressions) that is true (i.e., the expressions denote the same function), a fact established by Alonzo Church (e.g., see http://www.joachim-breitner.de/various/ChurchTalk2011.pdf) and related to Gödel’s first incompleteness theorem, the undecidability (semidecidability) of logic, and the halting problem for Turing machines.