# Homework 5

Due Thursday, 02/25/2016, by 11:55 p.m.

Your programs should be in a single file that can be loaded into ghci and tested. Problem numbers and any discussion of the programs should be set off as comments (using –) to make the Haskell compiler ignore them. Your program file should be turned in on Sakai like last week.

I should be able to load your file into Haskell (ghci) and have it compile without error. If there are programs with compilation errors, please comment them out so they don't get in the way of your other programs.

Please include good comments and use good variable names with your programs. Otherwise it is very difficult to read your code because Haskell is so concise.

**Important:** Your programs should have the exact names and types specified in the problem statement as your code will be automatically tested. Code that uses the incorrect name or expects different types (i.e., won't correctly execute my tests) will be counted as incorrect. Notice that all the functions to be written are in curried form.

Use comments to explain what each function does, especially helper functions.

Problems that are not programs should be written using LaTeX. When completed, please put the program file and the pdf generated from LaTeX into a folder, zip it up, and then turn it in as usual on Sakai.

1. In class we talked through the definitions in the files Model.hs, FSynF.hs, and MCWPL.hs. In this problem I'd like you to use these definitions to determine the truth of sentences in English. However, I've modified the above files so they define all the pieces used in these examples. Thus please use files HW5Model.hs, HW5FSynF.hs, and HW5MCWPL.hs in this and the other problems in this assignment.

   Consider the following sentences:

   - "Some boy admired Goldilocks"
   - "Every boy admired some girl that laughed"

   (a) (4 points) For each of these sentences, first write them in Haskell as an expression of type Sent.

   (b) (4 points) For each of these sentences, use the function lfSent to translate them into terms of type LF (= Formula Term).

   (c) (4 points) Interpret each of these in the model (provided in HW5Model.hs) using the function evl. In applying evl, use the interpretation of function symbols given by a function `fint3` with type given by `fint3:: String -> [Entity] -> Entity` and definition given in file HW5MCWPL.hs. This tells us the meaning of all constant functions (there are no function symbols in this language).

You can also use the variable assignment function (usually written as g) given by ass0, which assigns every variable to A. This is again fine because the sentence above has no free variables.

2. In this question you will add several adjectives to our language and be able to determine whether sentences using the new terms are true or false in a model.

   (a) (6 points) Modify the file HW5FSynF.hs to update the data type ADJ to include the words Red and Dangerous (along with Fake, which is there now).

   Notice that the type RCN has a clause that includes the use of ADJ. The definition given for RCN corresponds to a set of productions of a context free grammar of the form

   ```
   RCN -> CN That VP | CN That NP TV | ADJ CN
   ```

   In the data type definition, the first right side is tagged with RCN1, the second with RCN2, and the last with RCN3. Finally, the data type RCN itself is on the right side of the rules for NP.

   (b) (6 points) Many adjectives can be interpreted as unary relations, just like nouns. These include red and dangerous. Please add interpretations of the primitives red and dangerous to the file HW5Model.hs. Please assume that the red items in the universe are B, R, and X, and the dangerous items are F, G, X, W, and V. (We won't worry about interpreting Fake for now.)

   (c) (6 points) In the file HW5MCWPL.hs, you will notice that there is a function lfRCN defined. While there are definitions for terms with tags RCN1 and RCN2, they omitted the definition for terms with tag RCN3. Your job for this problem is to interpret those terms. The interpretation is very similar to that of terms with tag RCN1.

   For example, the interpretation of "princess that helped Alice" (written more formally as `RCN1 Princess That (VP1 Helped Alice)` is interpreted as a unary relation that is true of all the objects in the domain that are a princess and that helped Alice. The definition of lfRCN (RCN1 cn _ vp) specifies exactly this. It gives a characteristic function that takes a parameter t, and returns the conjunction of the meaning of cn t (i.e., whether t is a cn) and the meaning of vp t (i.e., whether t makes the vp true). (Notice that the _ is in the definition because the argument in that place is always `THAT` and we don't need it to find the meaning of the common noun phrase.

   It is similar when you interpret "red dwarf". A red dwarf is something that is red and is a dwarf. Similarly for a dangerous dagger. It is both dangerous and a dagger. (*Note: Not all adjectives can be interpreted this way. For example, adjectives like large and small must be interpreted differently. E.g, a small elephant is only small*

*with respect to the set of all elephants, and a big mouse may be much smaller than a small elephant. But we won't worry about this kind of adjective right now.*

Extend the definition of lfRCN so that it captures this meaning for terms of the form `ADJ CN`.

(d) (6 points) Finally you are to extend the interpretation of the model (also in HW5MCWPL.hs) so that it can understand the meaning of the new adjectives. You can do this by adding the interpretation of red and dangerous to the function `int0`.