Branwin DuBose & Josh Echols

CS 401 – Prolog Assignment

Statement of Work:

We are required to write two prolog programs. One program takes in inputs in the form of “Y is a Z.”, “A Y is a Z.”, “Is Y a Z?”, and “Is a Y a Z?”. The other program simulates a therapist.

Discription of design:

We split the work between us per the two parts of the assignment:

Here is the general gist of the part 1 program: The predicate "main" is the main entry point to this program. Main calls itself recursively without a base case to create the main program loop. The predicate "is\_a" is the rule that the users will assert. The predicate "check\_is\_a" is called for the users' queries This allows chains like "A is a B. B is a C. Is A a C?" to work correctly. "stmt" and "quer" are the delegate predicates that main calls after figuring out the type of the user input. “type\_is\_query" will check the first letter of the input (via the "is\_I" predicates) and return true if the type is query. "deleteLastElement" is a helper predicate used for removing the punctuation.

Part two of the program was designed with iterating through a list in mind. The user enters in input, which is then split at whitespaces and/or punctuation and made into a list. Once the list is created, it is then passed to a predicate called read\_stmts. This is where the majority of the work is done. Prolog has a built-in method called “nth0”. This method pretty much takes the 0 index in a list and assigns it to a variable. It also would return the rest of the list without that leading variable which is used later. So with the variable in hand, the predicate begins to run through a bunch of if else statements trying to match against keywords specified. This behavior continues until the variable is either matched against any of the keywords, or until all search results have been exhausted. If either of these cases exists, then the program would ask the user to be more specific in their replies.

Difficult:

Prolog was a difficult language due to its declarative nature. This made the way we made our programs change dramatically. It was also difficult due to the odd ways of reading user input. The built in “read” predicate doesn’t simply read in a string as one would expect.

Liked about Prolog:

Prolog seems to be a good language for making databases and queries.

Dislike about Prolog:

Reading in user input (like mentioned above) is difficult.