Project Part 6: Morning Dress-up

This is where we will discuss in detail our implementation guide.

----------Representing the Beliefs

The belief set will consist of at least 4 facts. The machine's goal is to take these facts and create a solution under which the user can be satisfied with the outfit chosen for them on that day. The set of weather conditions as well as other sets available will vary in the final version of the project. There will also be a set consisting of moods that a person can wake up feeling on a certain day, the set containing activities for the day, as well as different sets for certain types of clothing, ie, tops, pants/bottoms, shoes, head accessories, face accessories, and items that you can hold adjacent like bags or potentially more items. Certain kinds of clothing have special relationships with weather, mood, and activity conditions.

%The Belief Set of a person dressing themselves

weather(rainy).
weather(sunny, cloudy, snowy, highWind, humid)
mood(happy, tired, sad, sexy, indifferent)
activity(regularDay, busyDay, funDay, bumDay)
tops(t-shirt, buttonedTop, tankTop, hoodie, niceJacket, coat)
bottoms(jeans, shorts, suitPants, sweatpants, khakis, skirt, leggings)
shoes(sneakers, dressShoes, boots, sandals, heels, slippers)
headAccessories(baseballCap, beanie, scarf, topHat, noHat)
faceAccessories(earrings, sunGlasses, seeingGlasses, makeup, noFace)
bag(backpack, handbag, noBag, smallBag)

%Types of clothing will only be available under certain conditions

t-shirt(sunny/cloudy, happy/tired/indifferent, regularDay/funDay/bumDay)
buttonedTop(sunny/cloudy/highWind, happy/sexy, regularDay/busyDay)
tankTop(sunny/humid, tired/indifferent, regularDay/bumDay)
hoodie(sunny/rainy/cloudy, tired/sad/indifferent, regularDay/funDay/bumDay)
niceJacket(sunny, happy/sexy, regularDay/busyDay)
coat(rainy/cloudy/snowy/highWind, tired/sad/happy/indifferent, regularDay/bumDay)
jeans(sunny/cloudy/highWind, happy/indifferent, regularDay/funDay)

shorts(sunny/humid, happy/tired/indifferent, regularDay/funDay/bumDay)
suitPants(sunny/cloudy, happy/sexy, busyDay/funDay)
sweatpants(rainy/cloudy/highWind/snowy, tired/sad/indifferent, regularDay/bumDay)
khakis(sunny/cloudy, happy/sexy, busyDay/funDay)
skirt(sunny/cloudy/humid, happy/sexy, regularDay/funDay)
leggings(cloudy/rainy/snowy/highWind, tired/sad/indifferent, regularDay/bumDay)
sneakers(sunny/cloudy/rainy, happy/tired/indifferent, regularDay/funDay/bumDay)
dressShoes(sunny/cloudy, happy/sexy, busyDay/funDay)
boots(rainy/snowy/highWind, tired/indifferent, regularDay/bumDay)
sandals(sunny/humid/cloudy, happy/indifferent, regularDay/funDay)
heels(sunny/cloudy, happy/sexy, busyDay/funDay)
slippers(sunny/humid/cloudy, happy/tired/indifferent, regularDay/funDay/bumDay)

----------Reasoning with the Beliefs

As mentioned above, the goal of the machine is to orchestrate an outfit for the day which will be provided to the user to double check whether or not the outfit presented was a good option. The machine will then dive back into the belief set and make revisions based on the input given from the user and try to formulate another outfit for the user under the given conditions. The system is not terribly intricate since the facts of the system are few in number and reasoning with them should not require much effort. This project lines up and operates very similarly to the first prolog assignment in this course which is fortunate for us as we will be able to work by analogy to some degree.

Here is a representation of the bare minimum necessary for going out during the day. A top, bottom, and shoes and other accessories need to be chosen before leaving while the rest of the accessories can be conditional and either included or not included. The user will be prompted with a condition as well as an option.

condition([weather], [mood], [activity])
solve:- option([top], [pants], [shoes], _, _, _).
option([top], [pants], [shoes], [headAccessories], [faceAccessories], [bag])

Listed above is some of the pseudo code for how the system will operate itself when it attempts to solve for an outfit based on the condition. The condition should always yield a different output to the console while options will always vary based on the condition as well as from user input into the console.

----------Conditions for Revising Beliefs

The beliefs will be revised under a single condition which will be based off the user's input. Once the user is prompted with whether or not they like an outfit, they can simply reply with a confirmation or a declination. Should the user accept the outfit, the program will exit and be finished with a statement of what clothes were chosen. Should the user decline, the program will go back through the system and offer another option for an outfit. This will act as the primary function for belief revision unless we decide to incorporate some way of allowing one of the conditions to change by itself for whatever reason or perhaps one of the conditions is in direct contradiction with another and then we will have to set a means of choosing precedence over one condition like activity over weather for the day and allowing the system to create a new outfit based on more conditions than others.The user will be allowed the option to choose one thing to choose from the outfit and be prompted with a replacement item.

Below we can see an example of user interaction with the system:

option(t-shirt, jeans, sneakers, _, _, _).
"Decline"
option(t-shirt, jeans, sandals, _, _, _).
"Confirm"
'You chose to wear a t-shirt, jeans, and sandals today. Good choice!'

----------Techniques for Revising Beliefs

The techniques that will be utilized are mostly from prolog list processing like regular list traversal and accumulation of facts. Weather, mood, and activity should vary with each new startup of the system and should yield unique combinations of clothes for a given situation.

The system will perform loops for every declination of a given outfit so the system can parse through all the options again and offer a new option for outfit of the day. If we decide to go the route of one condition taking precedence over others, then we will have to use belief revision similar to the techniques showcased in the second and third assignments where the state of the model is in an almost constant state of change. We'll need to add parameters to each conditional fact that decides based on a given situation or combination which fact will be more important than the rest. In order to get the one item replaced from an outfit option, we'll need to have a conditional added onto our loop which allows for switching a single item with another.