Thomas Mintun
Krishn Mishra - CS4250 – HW3

1. John is the father if Susan. → father(john,susan).
   John is the husband of Martha → husband(john,martha).
   Susan eats oranges. → eats(susan,oranges).
   John bought pizza for Martha. → bought(john,pizza,martha).
   Susan is tired. → tired(susan).
   John eats a pepperoni pizza. → eats(john,pepperoni pizza).
   X and Y are parents of Z if X is the mother of Z and Y is the father of Z. → parent(X,Y,Z) :- mother(X,Z) , father(Y,Z).
   Sparky barks at anyone who walks quickly. →barks(sparky,anyone who walks quickly).
   A equals B. → equals(a,b).

2. How prolog satisfies goals? How do you write rules in Prolog?

   Prolog searches the list of facts you gave it from the top with the goal of satisfying the question. If a question is satisfied by finding it as a fact, Prolog will report Yes or True. If a sentence is satisfied by finding facts about someone who eats pizza, Prolog responds with name of the person that satisfies it. Note the computer only knows what you've told it. If it cannot find a match for information if reports No. It reports no even if fact is true in real world IFF not given the fact earlier. Prolog also backtracks so sometimes it starts search at top of list and sometimes starts search at where it last satisfied goal. Programmers can introduce "cut" ("!") to prevent backtracking beyond that point.

   Concerning conditional statements: the first statement is interpreted as rule to the effect that the second statement succeeds. The ":-" symbols tell Prolog how to read the rule. in Prolog we give the computer facts and rules about facts and then we can ask it questions. All these facts and rules we give it are stored in an internal database, and we can save this database in the form of a file that we can use in a later session. If we don't save it, the computer will not remember it if we turn off the computer and run Prolog later.

3. Represent following programs in prolog:
   a. eats(john, quickly) :- feels(john,hungry).
      gets(john,heartburn) :- eats(john,quickly).
      takes(john,medicine) :- gets(john,heartburn).
      feels(john,hungry).

      Ask Prolog "takes(john,medicine)." The answer should be "yes".

   b. dates(john,Nancy).
      takes(john,vacation) :- (after(irs,john) ; is(john,doomed)).
      dates(john,susan).
      knows(Nancy, (dates(john,nancy), dates(john,susan))) , knows(susan, (dates(john,nancy) , dates(john,susan)))).
      leaves(john,country) :- takes(john,country)

Ask Prolog “leaves(john,country).” The answer should be “no” because the computer hasn’t been told that Susan knows that john is dating both Nancy and Susan, and so it interprets this statement as false (unproved) and so the inference doesn’t go through.