CSCU9T6 Spring 2019
Tutorial on Inference and Rule-Based Systems (Solutions)

1 **Forward chaining** is also called *data-driven reasoning*. The idea is to generate all possible conclusions from the available rules and evidence. The system goes through the rules in turn, attempting to match the *antecedent* (the bit that appears after IF) of each rule with the facts in the database. Whenever a rule is matched, it "fires" and its *conclusion* (the bit after the THEN) is added as a new fact to the database. The rules are examined repeatedly, until a point is reached where a full pass through all remaining un-fired rules fails to yield any new matches. At this point, all the facts that can be inferred are now present in the database, and the process stops.

Here is how the process works when applied t o the example in the question:

Initially, the data base contains the following facts, which the user inputs after observing the dinosaur specimen.

[weight] = 3

[appearance of face] = "front facing nostrils and eyes"

The system then makes its first pass through all the rules in the knowledge base:

Rule 1 does not match the facts in the database.

Rule 2 does not match the facts in the database.

Rule 3 matches, so its conclusion [diet] = "carnivorous" is added to the database.

Rule 4 does not match the facts in the database.

The database has been updated with a new fact, so the system makes a second pass through the rules:

Rule 1 does not match the facts in the database.

Rule 2 matches, so [dinosaur species] = "compsognathus" is added to the database.

Rule 3 has already fired so is not re-examined.

Rule 4 does not match the facts in the database.

The database has been updated with a new fact, so the system makes a third pass through the rules:

Rule 1 does not match the facts in the database.

Rule 2 has already fired so is not re-examined.

Rule 3 has already fired so is not re-examined.

Rule 4 does not match the facts in the database.

No new facts were added to the database this time, so the inference process stops. The final set of facts in the database is:

[weight] = 3

[appearance of face] = “front facing nostrils and eyes”

[diet] = “carnivorous”

[dinosaur species] = “compsognathus”

Backward chaining is also called *goal-driven reasoning*. Unlike forward chaining, here we start off with a particular goal hypothesis in mind, and then attempt to use the rules and evidence to prove this hypothesis. The system first attempts to prove the hypothesis by matching it against the facts in the database. If there is a matching fact, then the hypothesis has been proved. If no facts match, then the system looks through the rules in turn, attempting to find a rule whose *conclusion* matches the goal hypothesis. If such a rule is found, the *antecedent* of that rule then becomes the new (sub)goal, and the reasoning process starts again in an attempt to prove this new subgoal. The process ends in one of two ways: (1) all subgoals have been successfully proved, and the system then concludes that the original goal hypothesis is true; or (2) the system tries all possible routes to proving the original goal, without success, and then concludes that the original goal hypothesis is not supported by the evidence.

Let us now apply this process to the given example. The data base contains the following facts:

[weight] = 3

[appearance of face] = “front facing nostrils and eyes”

We start by making a guess about the species of the dinosaur. Cheating slightly (because we already know the answer) we guess that it is a compsognathus.

Our initial goal hypothesis is then

Goal: [dinosaur species] = “compsognathus”

We first try to match the goal with the facts in the database. None of them match. So we next go through the rules in turn to try to find whose conclusion matches the goal.

Rule 1 does not match the goal.

Rule 2 matches the goal, and its antecedent becomes two new goals:

Goal 1: [diet] = “carnivorous”

Goal 2: [weight] < 5

The system now starts trying to prove Goal 1. It is first compared with the facts in the database. None of them match. So the system then tries to find a matching rule.

Rule 1 does not match the goal.

Rule 2 does not match the goal.

Rule 3 does match the goal. Because its antecedent contains OR, this rule gives us two possible routes to prove the goal. These are tried in turn. Trying the first route gives us:

Goal 1.1: [appearance of teeth] = “sharp, pointed, blade-like”

The system attempts to prove Goal 1.1, first by trying to match it against the facts in the database (there is no match) and then by trying to match it against the rules (which also fails). The system then gives up on trying to prove Goal 1.1, and moves on to the alternative possibility:

Goal 1.2: [appearance of face] = “front facing nostrils and eyes”

Goal 1.2 matches one of the facts in the database, so it can be proved. This means that Goal 1 has now been proved.

The system then attempts to prove Goal 2. This also matches a fact present in the database, so Goal 2 is proved.

As Goal 1 and Goal 2 have now been proved, the system concludes that the original Goal is proved.

[As an exercise, trace through the sequence of events that would take place if the initial guess had been [dinosaur species] = “tyrannosaurus rex”. You should find that you are able to prove the first subgoal but reach a dead end when attempting to prove the second subgoal.]

2 This is an exercise in using the rules given in the lecture notes on “Dealing with uncertainty in RBS” (see specifically slide 8).

From Rule 5, CF(E) = min (CF(K), CF(F)) * 100% = min(50, 75) = 50%

From Rule 4, CF(D) = min (CF(G), CF(H)) * 100% = min(90, 80) = 80%

From Rule 3, CF(C) = max (CF(D), CF(E)) * 60% = max(80, 50) * 60% = 48%

From Rule 3, CF(B) = max (CF(J), CF(E)) * 100% = max(70, 50) = 70%

From Rule 1, CF(A) = min (CF(B), CF(C)) * 90% = min(70, 48) * 90% = 43.2%

3 This requires use of the formula for combining certainty factors obtained from two independent rules. See slide 12 of the lecture notes on “Dealing with uncertainty in RBS”.

CF1(A) = 43.2%

CF2(A) = 50%

CF(A) = 43.2 + 50(100 - 43.2) / 100 = 71.6