University of Pittsburgh
CS 2710 Foundations of Artificial Intelligence — Handout 6
Professor Milos Hauskrecht — October 2, 2019

Problem assignment 5

Due: October 9, 2019

Propositional Logic

Problem 1.

Let KB consists of the following sentences:

$$\begin{aligned}\neg(P \wedge \neg Q) \vee \neg(\neg S \wedge \neg T),\\ \neg(T \vee Q),\\ U \rightarrow (\neg T \rightarrow (\neg S \wedge P)).\end{aligned}$$

Prove that $\neg U$ holds using:

- **Part a.** Truth-table approach.
- **Part b.** Inference rule approach.
- **Part c.** Resolution with refutation.

Note: When using the inference rule approach in addition to inference rules you are allowed to use standard logical equivalences to rewrite the sentences. Examples are equivalences defining De Morgan laws or the double negation.

Problem 2

Assume the following set of facts:

If the unicorn is mythical, then it is immortal, but if it is not mythical, then it is a mortal mammal. If the unicorn is either immortal or a mammal, then it is horned. The unicorn is magical if it is horned.

Part a. Express the above knowledge in the propositional logic.

Part b. Can you prove that the unicorn is mythical? Give a proof if provable.

Part c. Can you prove the fact that the unicorn is magical? Give a proof if provable.

Part d. Can you prove the fact that the unicorn is horned? Give a proof if provable.

Problem 3. Inference with propositional rules.

Assume a simplified animal identification problem due to P. Winston. The knowledge needed for the problem consists of the following set of rules:

1. If the animal has hair then it is a mammal
2. If the animal gives milk then it is a mammal
3. If the animal has feathers then it is a bird
4. If the animal flies and it lays eggs then it is a bird
5. If the animal is a mammal and it eats meat then it is a carnivore
6. If the animal is a mammal and it has pointed teeth and it has claws and its eyes point forward then it is a carnivore
7. If the animal is a mammal it has hoofs then it is an ungulate
8. If the animal is a mammal and it chews cud then it is an ungulate
9. If the animal is a mammal and it chews cud then it is even-toed
10. If the animal is a carnivore and it has a tawny color and it has dark spots then it is a cheetah
11. If the animal is a carnivore and it has a tawny color and it has black strips then it is a tiger
12. If the animal is an ungulate and it has long legs and it has a long neck and it has a tawny color and it has dark spots then it is a giraffe
13. If the animal is an ungulate and it has a white color and it has black stripes then it is a zebra
14. If the animal is a bird and it does not fly and it has long legs and it has a long neck and it is black and white then it is an ostrich,
15. If the animal is a bird and it does not fly and it swims and it is black and white then it is a penguin
16. If the animal is a bird and it is a good flyer then it is an albatross.

The above set of rules can be represented in the propositional logic using implications of the form $A1 \wedge A2 \wedge \cdots \wedge Ak \rightarrow B$, that is, all the statements are in the Horn form. Recall that inferences with modus ponens for KB in the Horn normal form are both sound and complete.

Assume a set of initial facts: the animal gives milk, it chews cud, it has long legs, long neck, tawny cloor and dark spots are all TRUE for the animal we want to identify. Assume the following set of theorems:

- Theorem1: the animal is a giraffe;
- Theorem2: the animal is a penguin;
- Theorem3: the animal is a mammal.

Decide using the repeated application of the modus ponens inference rule whether Theorems 1-3 hold. For every theorem proved give a sequence of rules (their numbers) used to derive the conclusion.

Problem 4. Implementation of logical inference with propositional rules

The inferences with propositional rules are not that hard to implement. Two procedures for making logical inferences with propositional rules are: forward and backward chaining.

For the sake of simplicity we assume that the knowledge base is represented using two knowledge components:

- rule-base (RB) that consists of a set of rules;
- fact base (FB) that consists of a set of statements (facts) that are known to be true in the actual world.

The definition of a knowledge base and its components: rule base (RB) and fact base (FB) can be found in *Propositional_KB_agent.py* given to you. Both the rule and the fact base are represented using list structures. Facts are represented as strings of characters. Rules are more complex and consist of:

- a rule identifier (or name)
- an antecedent (if part) represented by a list of facts that need to be satisfied;
- a consequent (then part) that corresponds to a simple fact that becomes true whenever the antecedent is true.

Please note that file *Propositional_KB_agent.py* also defines the animal identification problem we will experiment with in this assignment.

Part b. Forward chaining of propositional rules

Write and submit a $forwardchain(KB, theorem)$ function that takes a knowledge base (with rules and initial facts) and the theorem to be proved. It returns true if the theorem can be proved and false if it cannot. Your forward chaining procedure should:

- repeatedly scan rules with consequents that are not in the fact base;
- check whether the antecedent of a rule is satisfied: if yes, add the fact in the consequent of the rule to the fact base and print the rule and the new fact added to the knowledge base;
- report success if the theorem is in the fact base;
- report failure if no new fact was added during the last scan cycle.
- print the facts in the fact base after the procedure stops

Include the forward chaining procedure in file *forward_chain.py*. In the same file, run the procedure on the animal identification problem and five theorems defined in variables theorem1, theorem2, theorem3, theorem4 and theorem5. Note: Please remember to reset the fact base to its initial state prior to running the algorithm on five different theorems.

Part c. Backward chaining of propositional rules

The forward chaining procedure scans the rules "blindly" in the order in which they were defined. For more complex knowledge bases it may keep infering facts that are not helping in proving the target theorem. An alternative inference technique, called backward chaining, alleviates this difficulty.

The backward chaining works on the proof backwards: it starts from the theorem. If it is not satisfied, it checks all rules with consequents (then parts) equal to the theorem to see whether their antecedents (if parts) are satisfied. If at least one of the rules is satisfied the fact is proved and added to the fact base. If not, the facts that are not known in the rule antecedent become new theorems to be proved and the backward chaining procedure is called recursively on new theorems. Note that there may be many rules with the same antecedent so before saying that the fact cannot be proved make sure that all rules are exhausted and cannot be used to prove the theorem.

Write and submit a $backwardchain(KB, theorem)$ function that takes a knowledge base and the theorem to be proved. It returns true if the theorem can be proved and false if it cannot. The function should print a trace of:

- every theorem (even the ones set-up internally) we are trying to prove
- every rule we attempt to use to prove a theorem
- failure or success of using the above rule (and if succesful the new fact being derivied)
- success or failure in proving the initial theorem

- the fact base at the end

Include the backward chaining procedure in the file *backward_chain.py*. In the same file, run the procedure on the animal identification problem and five theorems defined in variables theorem1, theorem2, theorem3, theorem4 and theorem5. Report the results. Compare the two procedures for each theorem (theorem1-theorem5), in terms of the number of items in the fact base at the end. Discuss and analyze the results.