

Problem assignment 3
Due: Friday, February 21, 2024

Problem 1. Propositional logic

Part a. [3 points] Translate the following english sentences into *propositional logic*. If symbols are not already provided, you should define your own and state their meaning.

1. A and B are both true.
?
2. If A is true, then B must be true as well.
?
3. If a student studies for a test, they will do well on it. We can also tell that if a student did well on a test, then they must have studied for it.
?

Part b. [8 points] Using the truth-table approach, prove the following logical equivalencies. I recommend using a \LaTeX table generator (<https://www.tablesgenerator.com/>) if you need to make changes.

1. $\neg(A \vee B) \iff \neg A \wedge \neg B$ *De Morgan's Law*

A	B						
T	T						
T	F						
F	T						
F	F						

2. $A \Rightarrow B \iff \neg A \vee B$

A	B						
T	T						
T	F						
F	T						
F	F						

3. $A \Rightarrow B \iff \neg B \Rightarrow \neg A$ *Contrapositive*

A	B						
T	T						
T	F						
F	T						
F	F						

4. $A \wedge (B \vee C) \iff (A \wedge B) \vee (A \wedge C)$ *Distributive Property*

A	B	C					
T	T	T					
T	T	F					
T	F	T					
T	F	F					
F	T	T					
F	T	F					
F	F	T					
F	F	F					

Problem 2. Inference with propositional rules.

Assume a simplified animal identification problem. 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 and 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 $A_1 \wedge A_2 \wedge A_3 \cdots \Rightarrow B$, 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.

Part a. [4 points] Assume a set of initial facts:

- a) the animal has feathers
- b) the animal does not fly
- c) the animal swims
- d) the animal is black and white

Construct a *forward-chaining* proof of the theorem T1: *the animal is a penguin*. That is, show inferences that can be made from the initial facts and KB, chaining them together until the conditions are met to prove the theorem. For each step, give the number/label of the rules that are used.

Part b. [4 points] Assume a different set of initial facts from Part a.:

- a) the animal gives milk
- b) it chews cud
- c) it has long legs and a long neck
- d) it has a tawny color and dark spots

Construct a *backward-chaining* proof of the theorem T2: *the animal is a giraffe*. Begin with the rule that would prove the theorem, and work backwards to show that its conditions are satisfied by the KB and given facts.

Problem 3. Theorem Proving

Let KB consist of the following sentences:

$$R1 : \neg B \vee R$$

$$R2 : \neg Q \Rightarrow A \vee R$$

$$R3 : A \vee \neg R \Rightarrow B \wedge \neg Q$$

$$R4 : \neg(\neg A \vee Q)$$

Part a. [3 points] Prove the following theorem via a forward-chaining proof. T1: *R* At each step, state which logical equivalencies and inference rules you are applying.

Part b. [6 points] Next, convert the KB above into conjunctive normal form (CNF). See page 227 (Section 7.5) of the textbook for the necessary steps. *hint: R2 will be converted to a single clause, R3 into four clauses, and R4 into two clauses.*

Part c. [4 points] Now that your knowledge base has been converted to CNF, a single inference rule is sufficient to derive all true consequents from this KB: resolution with refutation (aka proof by contradiction). This process involves first adding the **negated** theorem you wish to prove to your KB, then repeatedly applying the resolution rule until a

contradiction is reached, i.e. $B \wedge \neg B$. See pgs 226-229 of the textbook for details of this procedure.

Here is how I would like you to write a step where the resolution rule is applied. Assume, for example, that we have the rule $A \vee B \vee C$ and also know the fact $\neg C$. Since this literal is complementary to one of the terms in the disjunction, we can infer that one of the other two variables must be true. The expression below the line is a new rule added to the KB:

$$\frac{\text{R1} : A \vee B \vee C, \quad \text{F1} : \neg C}{\text{R2} : A \vee B}$$

Prove the following theorem via resolution with refutation T2: B

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 a. [8 points] 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 b. [8 points] 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 inferring 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 successful the new fact being derived)
- 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.