National University of Singapore
School of Computing
CS3243 Introduction to AI

Tutorial 7: Inference in First-Order Logic

Issued: March 23, 2020 Due: Week 10, in the tutorial class

Important Instructions:

- *This tutorial is ungraded, but please bring a copy to class for the purpose of following the class discussion.*
- *You may discuss the content of the questions with your classmates. But work out and write up ALL the solutions by yourself.*

A knowledge base KB is a set of logical rules that model what the agent knows. These rules are written using a certain language (or *syntax*) and use a certain truth model (or *semantics* which say when a certain statement is true or false). In propositional logic sentences are defined as follows

1. Atomic Boolean variables are sentences.
2. If S is a sentence, then so is $\neg S$.
3. If S_1 and S_2 are sentences, then so is:
   (a) $S_1 \wedge S_2$ "S_1 and S_2"
   (b) $S_1 \vee S_2$ "S_1 or S_2"
   (c) $S_1 \Rightarrow S_2$ "S_1 implies S_2"
   (d) $S_1 \Leftrightarrow S_2$ "S_1 holds if and only if S_2 holds"

We say that a logical statement a models b ($a \models b$) if b holds whenever a holds. In other words, if $M(q)$ is the set of all value assignments to variables in a for which a holds true, then $M(a) \subseteq M(b)$.

An inference algorithm $\mathcal{A}$ is one that takes as input a knowledge base KB and a query α and decides whether α is derived from KB, written as $KB \vdash_{\mathcal{A}} \alpha$. $\mathcal{A}$ is sound if $KB \vdash_{\mathcal{A}} \alpha$ implies that $KB \models \alpha$; $\mathcal{A}$ is complete if $KB \models \alpha$ implies that $KB \vdash_{\mathcal{A}} \alpha$.

1. Given the following logical statements, use truth-table enumeration to show that $KB \models \alpha$. In other words, write down all possible true/false assignments to the variables, the ones for which KB is true and the one for which α is true, and see whether one is a subset of the other.

(a)

$$\begin{aligned} KB =&(x_1 \vee x_2) \wedge (x_1 \Rightarrow x_3) \wedge \neg x_2 \\ \alpha =&x_3 \vee x_2 \end{aligned}$$

(b)

$$\begin{aligned} KB =&(x_1 \vee x_3) \wedge (x_1 \Rightarrow \neg x_2) \\ \alpha =&\neg x_2 \end{aligned}$$

2. Given a graph $G = \langle V, E \rangle$, we say that a subset of vertices $X \subseteq V$ is an *independent set* if no two vertices in X share an edge.

   (a) Given a set of vertices $X \subseteq V$, write the constraint "no two vertices in X share an edge". You may only use Boolean variables of the form $x_v \in \{0, 1\}$ which indicate that x_v is part of the independent set. You may not refer to the set X in your solution, only the set V, and the edges in E. You can use basic arithmetic operators and basic logical operators.

   (b) Consider the graph in Figure 1. Write down the independent set constraints explicitly for this graph. In addition write down the constraint that the independent set size must be exactly 2. Then, run the AC3 algorithm assuming that we set $x_1 = 1$, and when we set $x_1 = 0$. Ignore the size constraint for this part (you can't write it in binary constraints when the size $k \geq 2$).

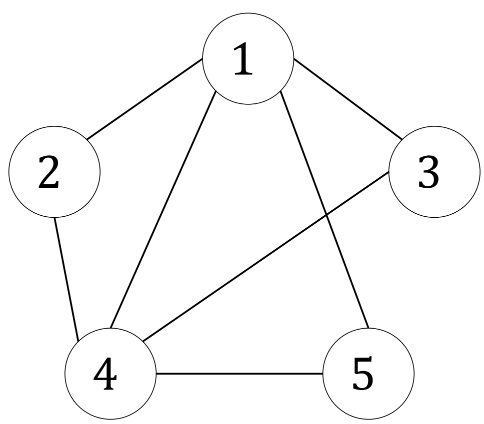

Figure 1: Independent set graph example

   (c) Write down the independent set constraints in Figure 1 in the language of propositional logic. Run inferences according to

i. Resolution
ii. Forward chaining
iii. Backwards chaining

to decide the following queries

i. The vertex 1 is not in the independent set (assume independent set size is $k = 2$). More formally, you need to show that if $x_1 = \mathit{True}$ then $x_2, x_3, x_4 = \mathit{False}$.
ii. The vertex 4 is not in the independent set (assume independent set size is $k = 3$). More formally, you need to show that if $x_4 \wedge x_j$ for some j, then the rest of the variables must be false.
iii. Feel free to think of other queries!

3. What is the problem in each the following first order logic statements? Suggest how these statements can be corrected.

   (a) $\forall x : \mathit{Boy}(x) \wedge \mathit{Tall}(x)$
   (Intended meaning: all boys are tall)

   (b) $\exists x : \mathit{Boy}(x) \Rightarrow \mathit{Tall}(x)$
   (Intended meaning: some boy is tall)

4. (Slightly modified from Question 9.24 of AIMA 3rd edition) Here are two sentences in the language of first-order logic:

   (a) $\forall x : \exists y : (x \geq y)$

   (b) $\exists y : \forall x : (x \geq y)$

   (i) Assume that the variables range over all the natural numbers $0, 1, 2, \ldots$ and that the "$\geq$" predicate means "is greater than or equal to". Under this interpretation, translate (a) and (a) into English.

   (ii) Is (a) true under this interpretation? Is (b) true under this interpretation?

   (iii) Does (a) logically entail (b)? Does (b) logically entail (a)? Justify your answers.

5. Suppose that we are maintaining a knowledge base with propositional logical statements involving Boolean variables $x_1, \ldots, x_n$. Given a logical formula q, let $M(q)$ be the set of all truth assignments to variables for which q is true. Recall that an inference algorithm $\mathcal{A}$ is sound if whenever a statement q is inferred by a knowledge base KB, it must be the case that $M(KB) \subseteq M(q)$. An inference algorithm $\mathcal{A}$ is *complete* if whenever $M(KB) \subseteq M(q)$, then eventually q will be inferred by $\mathcal{A}$. Formally prove that the resolution algorithm is sound (you've seen a sketch in class). You may also try to show completeness, but this is a longer proof (hint: you can use induction).

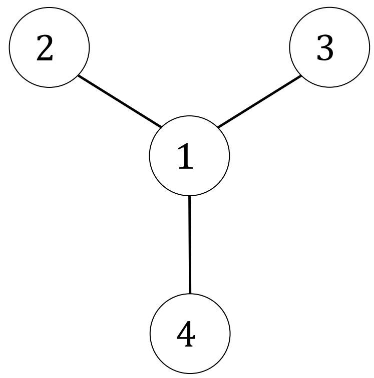

Figure 2: Graph for Vertex Cover CSP part (b)

6. Consider an instance of the VERTEXCOVER problem given in Figure 2. In the VERTEXCOVER problem we are given a graph $G = \langle V, E\rangle$. We say that a vertex v *covers* an edge $e \in E$ if v is incident on the edge e. We are interested in finding a *vertex cover*; this is a set of vertices $V' \subseteq V$ such that every edge is covered by some vertex in V'. In what follows you may **only** use variables of the form x_v where $x_v = 1$ if v is part of the vertex cover, and is 0 otherwise. When writing the constraints you may **only use**

    - Standard logical and set operators ($\forall, \exists, \vee, \wedge$ and $x \in X, X \subseteq Y$)

    (i) Write down the vertex cover constraints as logical statements, as well as the size constraints in the case that the vertex cover is of size $k = 1$.

    (ii) Apply the resolution algorithm in order to prove that the vertex 1 must be part of the vertex cover; again, assume that the cover in Figure 2 must be of size $k = 1$.

7. Consider the no regret learning setting we have studied in class. We saw that in order to achieve a low regret, we need a sufficiently large number of rounds. What happens when the number of rounds is small? Prove that if there are n experts and T time steps, such that $T < \lfloor \log_2 n \rfloor$, then for any randomized algorithm $\mathcal{A}$, there is some randomized assignment of losses[1] in $\{0, 1\}$ to the experts (which can depend on what $\mathcal{A}$ does) such that

    (a) $\mathcal{A}$ incurs a loss of at least $\frac{T}{2}$.

    (b) There exists at least one action that has a loss of 0.

8. Recall that the RANDGREEDY algoritm picks at time t an action uniformly at random from S_t, the set of best-performing actions at time t. Formally prove that the randomized greedy algorithm discussed in class achieves a $\mathcal{O}(\log n)$ regret when compared to the best action.

[1]Here we consider losses instead of rewards, but the idea is similar.

In other words, that for every loss of the best action, RANDGREEDY loses at most $\mathcal{O}(\log n)$ times. Recall the setup: we look at the best action, call it b, the specific time-steps that it incurred a loss. Suppose that after T time steps, b loses at steps $t_1, \dots, t_k$ for a total loss of k. Formally prove that the expected loss of RANDGREEDY in between consecutive losses is $\mathcal{O}(\log n)$. We have gone over most of the steps in class, but not written down a formal proof. The general idea is this: assume that the for every $t \in t_j + 1, \dots, t_{j+1}$, S_t is the set of best actions at time t, and that $r(t)$ is the number of actions in S_t that incurred a loss at time t. What happens to the size of S_t? What is the expected loss of RANDGREEDY? How can we bound the loss of RANDGREEDY between t_j and t_{j+1}?