CSCI-UA 9472 - Assignment 2 (Part II)
Logic and artificial intelligence
Solutions

Given date: October 12
Due date: October 29

Augustin Cosse

July 2020

In this second assignment, we consider the paper of Nilsson, *Logic and artificial intelligence*. Some of the questions below are slightly more involved and as a result will be weighted on 2 (in which case those will be indicated with a single *) or 3 (in which case those will be indicated with two **) points instead of 1.

1. What are the three theses of Nilsson ?

    - Intelligent Machines will have knowledge of their environments
    - The most versatile intelligent machines will represent much of their knowledge about their environment declaratively
    - For the most versatile machines, the language in which declarative knowledge is represented must be at least as expressive as first-order predicate calculus

2. How does Nilsson make the distinction between declarative and procedural knowledge ? How does he suggest to improve the first simpler distinction that he introduces ? Declarative knowledge is encoded explicitly in the machine in the form of sentences in some language, while procedural knowledge is manifested in programs in the the machine. According to Nilsson, a more precise distinction can be made by taking into account the notion of level of knowledge. To illustrate this idea, Nilsson takes the example of LISP which albeit it is considered as a program at one level, can be regarded as a declarative structure (interpreted by another program) at another level.

3. According to McCarthy and Smolensky, what are some advantages of declarative knowledge? What is your opinion on those?

    - Versatility (it could be used by the machine, even for purposes unforeseen by the machine's designer and could more easily be modified than than could knowledge emodied in programs. Moreover, it facilitates communication between the machine and other machines or humans

- Public access (Declarative knowledge is accessible to many people)
- Reliability (Different people can reliably check whether conclusions have been validly reached)
- Formality, Bootstrapping, Universality (The inferential operations require very little experience with the domain to which the symbols refer)

4. Why is English not a good declarative language candidate? and what language does Nilsson advocate as a good declarative language?

   According to Nilsson, the problem with English (although if we could use it it would make all the knowledge already compiled in books immediately available for use by computers) is that too ambiguous a representational medium for present-day computers. The meanings of English sentences depend too much on the contexts in which they are uttered and understood.

   As a better alternative, Nilsson suggest *First order First-Order Predicate calculus*.

5. What should be, according to Nilsson, the key elements of a machine interacting with the world?

   Nilsson considers a conceptualization made of 3 main components:

   - A function *mem* that maps the set of pairs (inputs, states), or $(\mathcal{S}, \mathcal{M})$ onto $\mathcal{M}$ which models the machine's memory behavior (The machine state at any instant being a function of the machine's input and its previous state)
   - A function *act* which describes how the machine acts on the world (which maps $(\mathcal{S}, \mathcal{M})$ onto $\mathcal{A}$ the set of actions).
   - A function *effect* which describes the effects of the actions on the world (hence maps $(\mathcal{A}, \mathcal{W})$ onto $\mathcal{W}$)
   - Finally, a function *see* can be used to model the fact that a machine is not sensitive to every aspect of the world but partitions the world into classes whose members, as far as the machine is concerned,a re equivalent.

6. In what sense is the machine designer comparable to a scientist ?

   According to Nilsson, the machine designer is in the same predicament as is the scientist. Scientists invent descriptions of the world and gradually refine them until they are more useful

7. How does Nilsson describes the construction of first order predicate calculus ? and the sentences in this language ? Within the framework of first order predicate calculus, how can one summarize the task of the designer?

   According to Nilsson, sentences from First Order predicate calculus are constructed as follows: for every world object in the conceptualization we create an object constant. For every world relation, we create a relation constant and for every world function, we create a function constant. This is then combined with the syntax of predicate calculus. When a designer cannot specify which of two relations holds, he uses a disjunction such as $\mathrm{Box(Ob1)} \wedge [\mathrm{Blue(Ob1)} \vee \mathrm{Green(Ob1)}]$. Or he may use an existentially quantified statement $(x)\mathrm{Box}(x) \wedge \mathrm{Green}(x)$ or he might know that all boxes are green $(\forall x)\mathrm{Box}(x) \Rightarrow \mathrm{Green}(x)$.

8. What is the difference between state and interpretation?

   A single state Δ (as described in the machine) can be satisfied by a set of interpretations

9. How does Nilsson describe *reasoning* (in particular the *reasoning* aspect of the function *mem*)? What does he describe as *sentence manipulation*?

   Because the actions emitted by the agent depend on the syntactic form of the sentences in Δ, it is necessary for *mem* to be able to rewrite these sentences in the form appropriate to the task at hand. This aspect of the function *mem* is what Nilsson calls reasoning. If we consider a robot designed to paint boxes green. Its sentence-to-action process *act* may include a production rule like "if Δ includes the sentence Box(η) for some value of η, paint the object denoted by η green." but not $Box(G17)$ explicitly. We might expect that correct behavior for this robot would be to paint the object denoted by $G17$ green, but there is no sentence-to-action rule to accomplish that unless Box(G17) occurs explicitly in Δ. Constructing the sentence Box(G17) from the sentences $(\forall x)\text{Blue}(x) \Rightarrow \text{Box}(x)$ and Blue(G17) is an example of one kind of sentence manipulation, or *inference* that we want mem to to.

10. What is the approach suggested by Nilsson as an alternative to the (expensive) verification that all models of Δ are also models of the new sentence ϕ (explain in your own words) As indicated by Nilsson, in practice we don't have to check that all the models of Δ are also models of ϕ, but we can instead rely on strictly syntactic operations on Δ that are able to compute logically entailed formulas. We use the phrase *rule of inference* to refer to any computation on a set of sentences that produces new sentences. If ψ can be derived from Δ by a sequence of applications of rules of inference, we say that ψ can be deduced from Δ and write $\Delta \vdash \psi$. An example is the rule of inference called *modus ponens*: From any sentence of the form $\rho \Rightarrow \sigma$, and ρ, we can deduce the sentence σ by modus ponens. The process of logical deduction involves using a set of rules of inference to deduce additional sentences from a set of sentences.

11. What rule can be considered *sound* according to Nilsson ? In particular, what is the difference between a *rule of inference* and a *sound rule of inference*? Explain the difference between the symbols $\Vdash$ and $\vdash$

    It happens that there are rules of inference, modus ponens is an example, that have the property that if $\Delta \vdash \phi$ then $\Delta \Vdash \phi$. Such rules of inference are called *sound*.

    In this case the notation $\Delta \Vdash \phi$ indicates that the sentence ϕ is entailed by the knowledge base Δ.

12. How do we call a set of inference rules that will ultimately be able to prove any entailed sentence ϕ?

    such such a set is called complete

13. Is it always good to limit an agent to sound inferences? (try to be as exhaustive as possible)

    As Nilsson indicates, although deduction (through sound inference) is important, much human thought involves leap of intuition, inductive inference, and other guessing strategies that lie outside the realm of sound inference.

14. To describe the connection between the real environment and the representation of this environment through the knowledge base, Nilsson cites the American essayist Edward Abbey. What is Abbey's point ?

    Nilsson cites Abbey to indicate that lose track of the real world by focusing our attention exlusively on the internal representation of the agent.

15. According to Nilsson, what is the problem with logical inference as the main source of deduction and reasoning? In particular, what is the difference between induction and deduction? Give an example of a inductive inference. How does Nilsson argue one might sometimes rewrite an unsound inference (or induction) as a sound one (or deduction)?

According to Nilsson, it could happen that the designer has some subset of the models of Δ in mind and if (for some reason or another) he could not specify this subset by enlarging the knowledge base, then there are circumstances under which unsound inference might be appropriate.

Deduction, according to Nilsson involves using a set of rules of inference to deduce additional sentences from a set of sentences. Induction corresponds to deriving a general property from a set of examples. As indicated by Nilsson in his example, we could consider the premises

$$\begin{aligned}&\text{Emerald(Ob1)} \wedge \text{Color(Ob1, Green)},\\&\text{Emerald(Ob2)} \wedge \text{Color(Ob2, Green)},\\&\ldots\\&\text{Emerald(Obn)} \wedge \text{Color(Obn, Green)}\end{aligned}$$

For some adequately large value of n, we may want to inductively infer (unsoundly but reasonably) that

$$(\forall x)\text{Emerald}(x) \Rightarrow \text{Color}(x, \text{Green})$$

provided there is no η mentioned in Δ such that Δ entails the sentence

$$\text{Emerald}(\eta) \wedge \neg\text{Color}(\eta, \text{Green})$$

Finally, according to Nilsson, in the context of sufficient additional information, sound conclusions can be drawn that might have seemed to have required unsound inference without the additional information. An example of this is as follows. Assume that the knowledge base Δ contains the sentence

$$(\exists y)(\forall x)\text{Emerald }(x) \Rightarrow \text{Color}(x, y)$$

(i.e. there is a color such that all emerald have that color.) as well as

$$(\forall x, y, z)(\text{Color}(x, y)) \wedge \text{Color}(x, z) \Rightarrow (y = z)$$

(i.e. a thing can have only one color)

From those statements, we can deduce that if a thing is an emerald, it has a unique color. Then if we subsequently learn the sentence

$$\text{Color(Ob1, Green)} \wedge \text{Emerald(Ob1)}$$

one can deduce soundly

$$(\forall x)\text{Emerald}(x) \Rightarrow \text{Color}(x, \text{Green})$$

16. How does Nilsson define the term *reification*? Can you give an illustration? When and why could *reification* be useful? What is a metatheory in this framework?

    Nilsson uses the term reification to denote the bestowal of an existence on what was originally an abstract concept

17. What does Nilsson mean by *qualifications* and what example does he use to illustrate a problem that can arise due to the infinite number of those qualifications?

    By qualifications, Nilsson denotes the whole set of proposition needed to guarantee with perfect confidence that an event will occur. To illustrate this idea, Nilsson mentions an example from John McCarthy that consists in formulating a sentence which would be used to indicate that under certain conditions, a car will start. An example of such a sentence might be "If the fuel tank is not empty and your turn the ignition key, the car will start" yet this simple sentence is not true in a world in which the carburetor is broken, or in which the fuel tank is full of water,...

18. What is a *theory that is not inaccurate*? How can inference be performed in the framework of a theory that is not inaccurate according to Nilsson? In particular, explain the notion of *defeasible inference* and *non monotonic reasoning*.

    Nilsson defines a theory that is not inaccurate as a theory whose models include the world as conceived by the designer. Nilsson takes as an example a machine that has to decide whether or not an apple is edible. if Δ is to be not inaccurate, and our description of the world include the exceptions Wormy(x) and Rotten(x) we cannot include in it the statement

    $$(\forall x)\text{Apple}(x) \wedge \text{Ripe}(x) \Rightarrow \text{Edible}(x)$$

    In this framework it might still be possible to do inference. Indeed, if we want to encode the fact that an apple denoted by 'apple1', for which we could not conclude that it was rotten or wormy, is edible, we may do this through the sentence 'Edible(Apple1)'. If the agent later learns the sentence 'Rotten(Apple1)', we must then withdraw the earlier conclusion 'Edible(Apple1)'.

    Such an inference is called *defeasible* because it can be defeated by additional information. Non monotonic reasoning is the process of making defeasible inferences.

19. ** In his paper *Applications of Circumscription to Formalizing Common-Sense Knowledge*, John McCarthy uses the following example to illustrate the use of circumscription to tackle the qualification problem. We consider a world with at least 3 blocks, A, B and C. This information is stored in the first order predicate sentence

    $$\textit{is block}(A) \wedge \textit{is block}(B) \wedge \textit{is block}(C) \tag{1}$$

    From this we may want to express that unless something abnormal happens, if an object is a block, then it must necessarily be A, B or C. This can be done by circumscription. Circumscription relies on (1) replacing an original predicate P (from a sentence S such as (1)) with a second more restrictive predicate $\Phi(x)$ and then (2) applying the circumscription rule

    $$S(\Phi) \vee \forall x\,(\Phi(x) \Rightarrow P(x)) \Rightarrow \forall y\,(P(y) \Rightarrow \Phi(y)) \tag{2}$$

    In the case of the block world, we can thus take the predicate $\Phi(x)$ to be given (for example) by

    $$\Phi(x) \equiv (x = A \vee x = B \vee x = C) \tag{3}$$

Applying the circumscription rule to $\Phi(x)$, we get the implication

$$\Phi(A) \wedge \Phi(B) \wedge \Phi(C) \wedge \forall x.\,(\Phi(x) \Rightarrow \textit{is block } x) \Rightarrow \forall x.(\textit{is block } x \Rightarrow \Phi(x)) \tag{4}$$

which, given that the predicates are true, provides the conclusion

$$\forall x.(\textit{is block } x \Rightarrow (x = A \vee x = B \vee x = C)) \tag{5}$$

If we then learn that $is; block\ D$, the proposition (1) becomes

$$\textit{is block}(A) \wedge \textit{is block}(B) \wedge \textit{is block}(C) \wedge \textit{is block}(D) \tag{6}$$

which invalidates the conclusion (5) (1) Describe in words, and through the example provided above, how circumscription provides a solution to the qualification problem? (2) Is circumsription sound or unsound?

Using non monotonic reasoning makes it possible to do inference even in the absence of a perfect characterization of the world (a full listing of the all qualifications). Since the conclusions are drawn from incomplete data they are logically unsound.

20. How does Nilsson describe the frame problem?

    Nilsson describes the *frame* problem as the problem of specifying which of the aspects of the world that do not change.

21. How can the notion of *abnormality* be used to tackle the qualification problem? (limit yourself to Nilsson's paper. No need to read the references that appear in the bibliography)

    If we consider the general rule

$$(\forall x)Q(x) \Rightarrow P(x)$$

It might be the case that this rule is not strictly correct without additional qualifications. We might however want to encode the fact that "typically, all objects that satisfy property Q also satisfy property P". One solution is to rely on the notion of abnormality represented by the relation "Ab". We then say that all objects that are not abnormal and that satisfy property Q also satisfy property P,

$$(\forall x)Q(s) \wedge \neg \mathrm{Ab}(x) \Rightarrow P(x)$$