Northeastern University
CS5100 Foundations of Artificial Intelligence
Summer 2025
Professor Sarita Singh
Erdun E
June 04, 2025

Assignment 3 Answer

Question 1

The Symbol Grounding Problem (SGP) is about how a system, like a robot or computer can give real meaning to the symbols it uses. As Harnad (Harnad, 2003) explained, most computers just process symbols based on their shape or structure, without knowing what they mean. For example, a system might recognize the word "apple," but unless it can connect that symbol to something in the real world (like the actual fruit through vision or touch), it doesn't really understand it. Similarly, a translation program might convert "cat" to "chat" in French, but it has no idea what a cat actually is, its soft fur, its meowing sound, or its behavior. The system is just matching symbols to other symbols, like using a dictionary without ever seeing the real objects. That's what makes this a problem: without grounding, systems are only manipulating symbols that refer to other symbols, leading to a kind of circular definition.

This problem matters in AI because if we want machines to reason, learn, or communicate like humans, they need to go beyond just symbol manipulation. Without grounding, AI systems might give correct answers but for wrong reasons. For instance, a medical AI might recommend the right treatment by pattern matching, but it doesn't understand what "pain" or "healing" actually means. This makes such systems unreliable in new situations and unable to explain their reasoning in meaningful ways. They need to experience the world in some way so that the symbols they use are tied to real things. Without grounding, we cannot say a machine really understands language or concepts, even if it performs well on tasks like translation or answering questions.

One proposed solution is the Praxical approach by Taddeo and Floridi (Taddeo & Floridi, 2007). Their idea is called Action based Semantics (AbS). It suggests that an agent can ground symbols by linking them to its own actions and the internal states those actions produce. For example, if a robot turns left and enters a specific internal state, that state can become the "meaning" of a symbol like "left." The robot doesn't need a programmer to tell it what "left" means, it learns this through its own physical experience. Over time, the robot can build up a vocabulary of grounded symbols, each connected to real actions and sensations. This creates what Taddeo and Floridi call "embodied meaning", meaning that comes from the robot's own body and experiences, not from external definitions. This avoids needing any pre installed knowledge or external help, which makes it meet the Zero Semantical Commitment Condition, a key requirement for solving the SGP.

This approach has strengths. It is fully agent centered, doesn't assume built in meanings, and allows symbols to grow from the agent's own experience. However, it also has challenges. It is still hard to implement in large systems, and some critics argue that even defining these internal states might involve hidden assumptions or biases. So while the idea is promising, we're still far from a complete practical solution.

In summary, the Symbol Grounding Problem is a deep issue in AI, and the Praxical solution gives a good direction. But real world progress still needs more work.

Question 2

First- Each correct Order Logic Statements (FOL) sentence

1. Plastic things are smooth.

$$\forall x\ (plastic(x) \to smooth(x))$$

2. Varnished things are smooth.

$$\forall x\ (varnished(x) \to smooth(x))$$

3. Blocks are ceramics.

$$\forall x\ (block(x) \to ceramic(x))$$

4. White things are ceramics.

$$\forall x\ (white(x) \to ceramic(x))$$

5. Ceramic things are polished.

$$\forall x\ (ceramic(x) \to polished(x))$$

6. Plastic things are airy.

$$\forall x\ (plastic(x) \to airy(x))$$

7. X is a ball.

$$ball(x)$$

8. X is plastic.

$$plastic(x)$$

9. Y is a block.

$$block(y)$$

10. Z is green.

$$green(z)$$

11. Z is plastic.

$$plastic(z)$$

12. W is varnished.

$$varnished(w)$$

Prolog Program - Correct syntax and structure of Prolog clauses

```
smooth(X) :- plastic(X).
smooth(X) :- varnished(X).
ceramic(X) :- block(X).
ceramic(X) :- white(X).
polished(X) :- ceramic(X).
airy(X) :- plastic(X).
ball(x).
plastic(x).
block(y).
green(z).
plastic(z).
varnished(w).
```

Query Results

```
?- ["/Users/erdune/Desktop/NortheasternMiami/CS5100FoundationsOfAI/Assignment
true.

?- smooth(w).
true.

?- smooth(x).
true.

?- smooth(y).
false.

?- smooth(z).
true.
```

Question 3

a, Explain in your own words what is meant by the following terms, in the context of knowledge representation and reasoning:

A model and Entailment

- A model is a specific interpretation or assignment of truth values that makes all sentences in a knowledge base true. It represents a possible world where all the statements hold.
- Entailment means that a sentence logically follows from the knowledge base - if the knowledge base is true, then the entailed sentence must also be true in every possible model.

Monotonicity, soundness and completeness

- Monotonicity is a logic system is monotonic if adding new knowledge or premises never invalidates or contradicts previous conclusions. Once something is proven, it remains true even when new information is added.
- Soundness is a reasoning system is sound if everything it derives or concludes is actually true and logically valid. Sound systems never produce false conclusions from true premises.
- Completeness is a reasoning system is complete if it can derive every conclusion that is logically entailed by the knowledge base. If something logically follows, a complete system will eventually prove it.

False negative and False positive

- False negative means a reasoning system, this occurs when the system fails to recognize or derive a conclusion that is actually true. The system misses a valid inference.
- False positive means a reasoning system, this occurs when the system incorrectly concludes something is true when it is actually false. The system makes an invalid inference.

Prior probability and posterior probability.

- Prior probability is the initial probability assigned to an event or hypothesis before observing any new evidence. It represents our background belief based on existing knowledge.
- Posterior probability is the updated probability of an event or hypothesis after considering new evidence. It is calculated using Bayes' theorem to incorporate the observed data.

b, Represent the following statements as formulas of Predicate Calculus, stating the intended interpretation of any predicates and constants that you use. You should give two versions of each formula, one using the existential quantifier $ and one using the universal ".

1. All panthers are cats and all cats are mammals.

- panther(x): x is a panther
- cat(x): x is a cat
- mammal(x): x is a mammal

Universal quantifier version:

$$\forall x\ (panther(x) \rightarrow cat(x)) \land \forall x\ (cat(x) \rightarrow mammal(x))$$

Existential quantifier version:

$$\neg\exists x\ (panther(x) \land \neg cat(x)) \land \neg\exists x\ (cat(x) \land \neg mammal(x))$$

2. All elephants are African or Asian.

- elephant(x): x is an elephant
- african(x): x is African
- asian(x): x is Asian

Universal quantifier version:

$$\forall x\ (elephant(x) \rightarrow (african(x) \vee asian(x)))$$

Existential quantifier version:

$$\neg\exists x\ (elephant(x) \wedge \neg(african(x) \vee asian(x)))$$

3. No elephants are ungulates.

- elephant(x): x is an elephant
- ungulate(x): x is an ungulate

Universal quantifier version:

$$\forall x\ (elephant(x) \rightarrow \neg ungulate(x))$$

Existential quantifier version:

$$\neg\exists x\ (elephant(x) \wedge ungulate(x))$$

4. Some cats are not nocturnal.

- cat(x): x is a cat
- nocturnal(x): x is nocturnal

Existential quantifier version:

$$\exists x\ (cat(x) \wedge \neg nocturnal(x))$$

Universal quantifier version:

$$\neg\forall x\ (cat(x) \rightarrow nocturnal(x))$$

5. Not all mammals are carnivorous.

- mammal(x): x is a mammal
- carnivorous(x): x is carnivorous

Existential quantifier version:

$$\exists x\ (mammal(x) \wedge \neg carnivorous(x))$$

Universal quantifier version:

$$\neg\forall x\ (mammal(x) \rightarrow carnivorous(x))$$

6. The barber shaves every man who does not shave himself.

- man(x): x is a man
- shaves(x, y): x shaves y
- barber: a constant representing the barber

Universal quantifier version:

$$\forall x\ (man(x) \land \neg shaves(x, x) \to shaves(barber, x))$$

Existential quantifier version:

$$\neg \exists x\ (man(x) \land \neg shaves(x, x) \land \neg shaves(barber, x))$$

c, Encode the above knowledge base (KB) on as a series of Prolog clauses. Note that as Prolog does not allow disjunction in the head of a clause, you will have to formulate logically equivalent versions of some of the rules.

```
:- discontiguous cat/1.
:- discontiguous mammal/1.
:- discontiguous non_nocturnal/1.

panther(panther1).
elephant(dumbo).
african(dumbo).
man(john).
man(bob).
cat(fluffy).
mammal(rabbit).
herbivorous(rabbit).

% 1. All panthers are cats and all cats are mammals
cat(X) :- panther(X).
mammal(X) :- cat(X).

% 2. All elephants are African or Asian
african_or_asian(X) :- elephant(X), african(X).
african_or_asian(X) :- elephant(X), asian(X).

% 3. No elephants are ungulates
not_ungulate(X) :- elephant(X).

% 4. Some cats are not nocturnal
non_nocturnal(fluffy).

% 5. Not all mammals are carnivorous
```

```
% We have mammal(rabbit) and herbivorous(rabbit) defined above
% This shows not all mammals are carnivorous

% 6. The barber shaves every man who does not shave himself
shaves(barber, X) :- man(X), \+ shaves(X, X).

% Helper predicates
non_nocturnal(X) :- \+ nocturnal(X).
does_not_shave_self(X) :- \+ shaves(X, X).
```

Figure 1:

Successfully implemented and tested all knowledge base rules in SWI-Prolog. All queries return expected results, demonstrating correct logical reasoning.

Question 4

```
% Facts from the knowledge base
apple(a).
apple(b).
banana(c).
mango(d).
carrot(e).
onion(f).
potato(g).

% Rule 3: Apple(x)  Banana(x)  Mango(x)  Fruit(x)
```

```
fruit(X) :- apple(X).
fruit(X) :- banana(X).
fruit(X) :- mango(X).

% Rule 4: Carrot(x)  Potato(x)  Vegetable(x)
vegetable(X) :- carrot(X).
vegetable(X) :- potato(X).

% Rule 1: Fruit(x)  Vegetable(x) - onion must be classified
vegetable(X) :- onion(X).

% Rule 2: Apple(x)  Red(x)  Green(x)
red(a).     % Apple A is red
green(b).   % Apple B is green

% Rule 5: Fruit(x)  Tasty(x)
tasty(X) :- fruit(X).

% Rule 6: Carrot(x)  Tasty(x)
tasty(X) :- carrot(X).

% Rule 7: Onion(x)  ¬Tasty(x)
not_tasty(X) :- onion(X).

% Helper predicates
colored(X) :- red(X).
colored(X) :- green(X).
```

```
?- ['/Users/erdune/Desktop/NortheasternMiami/CS5100FoundationsOfAI/Assignment3/assignment3_q4.pl'].
true.

?- fruit(X), red(X).
X = a .

?- onion(X), tasty(X).
false.

?- colored(X).
X = a ;
X = b.

?- vegetable(X), tasty(X).
X = e .

?-
```

Query 1: What fruits are red?

```
?- fruit(X), red(X).
X = a.
```

Apple A is the only red fruit.

Query 2: Is some onion tasty?

```
?- onion(X), tasty(X).
false.
```

No onions are tasty according to rule 7.

Query 3: What things are colored?

```
?- colored(X).
X = a ;
X = b.
```

Apple A (red) and Apple B (green) are the colored items.

Query 4: Is there a tasty vegetable?

```
?- vegetable(X), tasty(X).
X = e.
```

Yes, carrot E is a tasty vegetable.

Explanation

- Onions were classified as vegetables since rule 1 requires everything to be either fruit or vegetable, and they're not fruits.
- Apple A was set as red and apple B as green to satisfy rule 2's requirement.
- Disjunctive rules were split into multiple clauses for Prolog compatibility.

Unintuitive Results

- Only apples have colors, bananas and mangoes appear "colorless" because no color rules exist for them.
- Onions are not tasty according to rule 7, even though many people enjoy them.
- Potatoes have no taste property, unlike carrots which are explicitly made tasty by rule 6.

These results follow the knowledge base strictly rather than real-world knowledge.

References

- Harnad, S. (2003). *The Symbol Grounding Problem.* Encyclopedia of Cognitive Science. Retrieved from https://web-archive.southampton.ac.uk/cogprints.org/3018/
- Taddeo, M., & Floridi, L. (2007). *A Praxical Solution of the Symbol Grounding Problem.* Minds and Machines, 17(4), 369–389. https://ssrn.com/abstract=3844446