Bafna Gold
Date: 19/11/24 Page:

Lab-07

A. Formalize the following by writing truth tables for the premises & conclusion by determining valid arguments. Also provide truth table.

"Either John isn't stupid and he is lazy, or he's stupid. John is stupid. Therefore, John isn't lazy."

- Symbols:
  S : John is stupid
  L : John is lazy.

Premises and conclusion

Premise 1: $(\neg S \wedge L) \vee S$ → Either John isn't stupid or he is lazy and he is lazy or he is stupid.

Premise 2: S → John is stupid

Conclusion: $\neg L$ (John isn't lazy)

Truth table.

S	L	KB	Query
T	T	T	F
T	F	T	T
F	T	F	F
F	F	F	T

KB does not entail hence argument is not valid.

B Translate the natural language sentences into (FOL) first order logic

1. John is a human : Human (John)
2. Every Human is mortal : forall x (Human(x) → Mortal(x))
3. John loves mary : Loves(John, mary)
4. There is someone who loves mary : exists x (Loves(x, mary))
5. All dogs are animal : forall x (Dog(x) → Animal(x))
6. Some dogs are brown : exists x (Dog(x) and Brown(x))

function translate_to_FOL (Sentence):
Sentence = Sentence.lower()

if "is" in sentence:
subject = extract_subject (Sentence)
predicate = extract_predicate (Sentence)
return predicate (subject)

else if "every" in sentence:
subject = extract-subject (sentence)
predicate = extract-predicate (sentence)
return "for all x, Predicate(x) → Predicate(x)"

else if "there" in sentence:
subject = extract_subject (sentence)
predicate = extract-predicate (sentence)
object = extract_object (sentence)
return "There exists x, Predicate (x, object)".

```
else:
    return "Sentence not recognised"

function extract_subject(sentence):
    return sentence.split()[0]

function extract_predicate(sentence):
    return sentence.split()[1]

function extract_object(sentence):
    return sentence.split()[-1]
```

Output:

Enter a sentence to translate:
Mary has a pet dog

FOL: HasPet(Mary, Dog)

19-11-24