



## Knowledge Representation

**Source:** <https://cs50.harvard.edu/ai/2020/weeks/1/>

The aim of this laboratory is to implement ideas of knowledge representation using Python. Traditionally, such techniques are implemented in Prolog, but for the sake of simplification, we choose to implement it using the one chosen programming language for the course, Python. Students are optionally recommended to implement the same tasks in Prolog.

### A. Propositional logic

What does “reasoning based on knowledge to draw a conclusion” mean?

Let’s start answering this with a Harry Potter example. Consider the following sentences:

1. If it didn’t rain, Harry visited Hagrid today.
2. Harry visited Hagrid or Dumbledore today, but not both.
3. Harry visited Dumbledore today.

Based on these three sentences, we can answer the question “did it rain today?”, even though none of the individual sentences tells us anything about whether it is raining today. Here is how we can go about it: looking at sentence 3, we know that Harry visited Dumbledore. Looking at sentence 2, we know that Harry visited either Dumbledore or Hagrid, and thus we can conclude

4. Harry did not visit Hagrid.

Now, looking at sentence 1, we understand that if it didn’t rain, Harry would have visited Hagrid. However, knowing sentence 4, we know that this is not the case. Therefore, we can conclude

5. It rained today.

```
from logic import *
rain = Symbol("rain")
hagrid = Symbol("hagrid")
dumbledore = Symbol("dumbledore")

knowledge = And(
    Implication(Not(rain), hagrid),
    Or(hagrid, dumbledore),
    Not(And(hagrid, dumbledore)),
    dumbledore
)

print(model_check(knowledge, rain))
knowledge = And(
    Implication(Not(rain), hagrid)
)
print(knowledge.formula())
```

## B. First order logic

The law says that it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West, who is American. Prove that Colonel West is a criminal

American(x): x is an American

Weapon(x): x is a weapon

Hostile(x): x is a hostile nation

Criminal(x): x is a criminal

Missile(x): x is a missile

Owns(x, y): x owns y

Sells(x, y, z): x sells y to z

Enemy(x, y): x is an enemy of y

Constants: America, Nono, West

```
# Import libraries
import aimutils
import aimlogic
# The main entry point for this module
def main():
    # Create an array to hold clauses
    clauses = []
    # Add first-order logic clauses (rules and fact)
    clauses.append(aimutils.expr("(American(x) & Weapon(y) & Sells(x, y, z)
& Hostile(z)) ==> Criminal(x)"))
    clauses.append(aimutils.expr("Enemy(Nono, America)"))
    clauses.append(aimutils.expr("Owns(Nono, M1)"))
    clauses.append(aimutils.expr("Missile(M1)"))
    clauses.append(aimutils.expr("(Missile(x) & Owns(Nono, x)) ==>
Sells(West, x, Nono)"))
    clauses.append(aimutils.expr("American(West)"))
    clauses.append(aimutils.expr("Missile(x) ==> Weapon(x)"))
    # Create a first-order logic knowledge base (KB) with clauses
    KB = aimlogic.FolKB(clauses)
    # Add rules and facts with tell
    KB.tell(aimutils.expr('Enemy(Coco, America)'))
    KB.tell(aimutils.expr('Enemy(Jojo, America)'))
    KB.tell(aimutils.expr("Enemy(x, America) ==> Hostile(x)"))
    # Get information from the knowledge base with ask
    hostile = KB.ask(aimutils.expr('Hostile(x)'))
    criminal = KB.ask(aimutils.expr('Criminal(x)'))
    # Print answers
    print('Hostile?')
    print(hostile)
    print('\nCriminal?')
    print(criminal)
```

```

    print()
# Tell python to run main method
if __name__ == "__main__": main()
*****
*****

# Import libraries
import aima.utils
import aima.logic
# The main entry point for this module
def main():
    # Create an array to hold clauses
    clauses = []
    # Add first-order logic clauses (rules and fact)
    clauses.append(aima.utils.expr("(American(x) & Weapon(y) & Sells(x, y, z)
& Hostile(z)) ==> Criminal(x)"))
    clauses.append(aima.utils.expr("Enemy(Nono, America)"))
    clauses.append(aima.utils.expr("Owns(Nono, M1)"))
    clauses.append(aima.utils.expr("Missile(M1)"))
    clauses.append(aima.utils.expr("(Missile(x) & Owns(Nono, x)) ==>
Sells(West, x, Nono)"))
    clauses.append(aima.utils.expr("American(West)"))
    clauses.append(aima.utils.expr("Missile(x) ==> Weapon(x)"))
    # Create a first-order logic knowledge base (KB) with clauses
    KB = aima.logic.FolKB(clauses)
    # Add rules and facts with tell
    KB.tell(aima.utils.expr('Enemy(Coco, America)'))
    KB.tell(aima.utils.expr('Enemy(Jojo, America)'))
    KB.tell(aima.utils.expr("Enemy(x, America) ==> Hostile(x)"))
    # Get information from the knowledge base with ask
    hostile = aima.logic.fol_fc_ask(KB, aima.utils.expr('Hostile(x)'))
    criminal = aima.logic.fol_fc_ask(KB, aima.utils.expr('Criminal(x)'))
    # Print answers
    print('Hostile?')
    print(list(hostile))
    print('\nCriminal?')
    print(list(criminal))
    print()
# Tell python to run main method
if __name__ == "__main__": main()

```

# Assignment

Source:

<http://athena.ecs.csus.edu/~mei/logicp/exercises.html#prolog>

- 1- Write python code, accompanied with equivalent English statements, that represent the following facts:
  - (1) color(carrots, orange).
  - (2) likes(Person, carrots):-vegetarian(Person).
  - (3) pass(Student) :- study\_hard(Student).
  - (4) ?- pass(Who).
  - (5) ?- teaches(professor, Course).
  - (6) enemies(X, Y) :- hates(X, Y), fights(X, Y).
- 2- For below English sentences write applicable Python code that represents facts, rules & goals:
  - (1) Maria reads logic programming book by author peter lucas.
  - (2) Anyone likes shopping if she is a girl.
  - (3) Who likes shopping?
  - (4) kirke hates any city if it is big and crowded.
- 3- There are flaws in the following clauses, find them, correct them, and write python code that implements them
  - (1) hates(X,Y), hates(Y,X) :- enemies(X, Y)
  - (2) p(X):- (q(X):-r(X)).
- 4- For given English statements write a python program, - Facts & Rules:
  - (1) jia is a woman.
  - (2) john is a man.
  - (3) john is healthy.
  - (4) jia is healthy.
  - (5) john is wealthy.
  - (6) anyone is a traveler if he is healthy and wealthy.
  - (7) anyone can travel if he is a traveler.- Goals.
  - (1) Who can travel?
  - (2) Who is healthy and wealthy?