

## Assignment 2

Instructor: Le Nguyen

**Date:** Friday, Feb 21, 2019

**Due Date:** March 15, 2019

**Total marks: 40.**

**This is an individual assignment.**

### I. Propositional Logic (8 marks)

#### a. Knowledge representation with propositional logic. (1 marks)

You are to construct a KB for the Wumpus world. Please read the book to understand the Wumpus world. (You can use propositional notations in the text book)

- i. Square (1,1) is safe.
- ii. Square (1,1) has no stench
- iii. Square (1,1) has no breeze
- iv. Square (1,1) has no Wumpus
- v. Square (1,1) is breezy if and only if a pit exists in adjacent squares
- vi. Square (1,2) has stench if and only if a Wumpus is in adjacent squares.
- vii. Square (2,1) is breezy if and only if a pit exists in adjacent squares
- viii. Square (2,2) is safe if and only if there is no pit and no Wumpus in the square.
- ix. Square(2,1) is safe if and only if there is no pit and no Wumpus in the square
- x. Square(1,2) is safe if and only if there is no pit and no Wumpus in the square.
- xi. Square (1,2) has stench
- xii. Square (1,2) is not breezy
- xiii. Square (1,2) is not breezy implies there is not pit in square (1,3)
- xiv. Square (1,2) is not breezy implies there is not pit in square (2,2)
- xv. Square (1,2) is not breezy implies there is not pit in square (1,1)
- xvi. Square (2,1) has no stench

- xvii. Square (2,1) has no stench implies no Wumpus in square(1,1)
- xviii. Square (2,1) has no stench implies no Wumpus in square (2,2)
- xix. Square (2,1) has no stench implies no Wumpus in square (3,1)
- xx. Square (2,1) is breezy
- xxi. Square (1,1) is not breezy implies Square (1,2) has no pit
- xxii. Square (1,1) is not breezy implies Square (2,1) has no pit
- xxiii. Square (1,1) has no stench implies Square (1,2) has no Wumpus
- xxiv. Square (1,1) has no stench implies Square (2,1) has no Wumpus
- xxv. Square (2,1) is breezy and Square (2,2) is safe and Square (1,1) is safe implies there is pit in Square (3,1)

**b. Consider above Wumpus Knowledge base KB: (6 marks)**

- i. Prove by Resolution  
Square(2,1) is safe
- ii. Prove by Forward Chaining  
Square(1,2) is safe
- iii. Prove by backward chaining  
Square(3,1) has a pit.

**c. Convert to CNF. (1 mark)**

$$\neg[(\neg a \leftrightarrow b) \vee d] \rightarrow [(c \wedge b) \vee a \vee d]$$

**II. First Order Logic (8 marks)**

**1. Unification (2 marks)**

For **each pair of sentences** find the unifier. Lower case letters are variables.

Sentence 1	Sentence 2
G(A, B)	G(x,y)
G(F(x), B)	G(y,z)
G(F(y),x)	G(x,F(B))
G(x,C,C)	G(A,y,z)
G(x,y)	G(y,x)
G(x)	G(A)
G(x,A,z)	G(B,y,z)
G(P(F(v)), P(u))	G(x,x)

$G(F(x), y, P(x))$	$G(F(x), x, P(x))$
$G(F(x), y, P(y))$	$G(F(x), z, P(x))$

## 2. Consider a FOL Knowledge base KB: (6 marks)

$$\forall x_1 \exists y_1 P(x_1) \rightarrow Q(y_1, x_1)$$

$$\forall x_2 \exists y_2 Q(y_2, x_2) \rightarrow P(x_2)$$

$$\forall x_3 \forall y_3 Q(x_3, y_3) \rightarrow R(x_3)$$

$$\forall x_4 \forall y_4 Q(x_4, y_4) \rightarrow R(y_4)$$

**Fact:**  $P(A)$

**Prove:**  $KB \models R(A)$

- Prove by Resolution
- Prove by Forward chaining
- Prove by Backward chaining

## III. Implementation (24 marks)

You are to implement the Forward Chaining in **propositional logic**.

Assumption:

- All sentences are in Horn forms (disjunction of literals).

There are **several classes** that are provided as a template. There comments in the template to help you understand the fields and ideas how to start your project.

- HornClause.java (4 marks)**
  - It provides the definition of Horn Clause (disjunction of literals.)
- HornKB.java (6 marks)**
  - It stores horn clauses in knowledge base.
  - It does the Forward chaining reasoning. You need to implement this algorithm. (Refer to text book page 258)
    - Tips: use **HashMaps** for **count** and **inferred**
    - Use a **stack** for **agenda**.

- **Literal.java (3 marks)**
  - It provides the definition of propositional literal.
    - A literal is either a **positive or negative** atomic sentence (single symbol).
    - Every symbol must have a **sign**.
- **Symbol.java (3 marks)**
  - It provides the definition of a propositional symbol
- **LogicalAgent.java (6 marks)**
  - It has TELL and ASK methods
    - TELL adds facts, rules to the knowledge base.
    - ASK does the query the Knowledge base for entailment and return true or false.
  - It has methods to load horn clauses from a **file** to the knowledge base and read a query file.
  - It has a main method to run the assignment.

**Test Files: There are several test files**

- A **knowledge base** file for Wumpus world.
  - **HornKB.txt**
    - This file consists of Horn clauses that are used to create the Wumpus World knowledge base.
    - Each line is a horn clause in form of **disjunction of literals**.
    - Eg.  $\neg A \vee \neg B \vee C$
    - In the format: - **A - B + C**
    - *Connector*  $\vee$  is assumed thus eliminated.
  - **Description.pdf** provides the description of literals and horn clauses.
- The **Query Files**. These are horn clauses to test the entailment of the knowledge base. Note only **the head** is used for entailment.
  - Query1.txt
  - Query2.txt
  - Query3.txt
  - Query4.txt
  - Query5.txt

**Run your program:**

**Java LogicalAgent HornKB.txt Query1.txt**

**Inferred knowledge:**

**Symbol: npt:2,2 is inferred.**

**Symbol: nwp:2,2 is inferred.**

**Symbol: nwp:3,1 is inferred.**

**Symbol: ok:2,2 is inferred.**

**Symbol: pt:3,1 is inferred.**

**Symbol: npt:2,1 is inferred.**

**Symbol: npt:1,2 is inferred.**

**Symbol: nwp:2,1 is inferred.**

**Symbol: nwp:1,2 is inferred.**

**Symbol: ok:1,2 is inferred.**

**Symbol: ok:2,1 is inferred.**

**Symbol: nwp:1,1 is inferred.**

**Symbol: wp:1,3 is inferred.**

**Query = wp:1,3**

**Answer = true**

**Submission:**

Codes that **cannot compile** will receive a zero mark. Please make sure it can compile and it should work on Windows 10 and Linux. Oracle **Java 8 SDK** is used to mark the assignment.

- Please comment your code properly.
- Please don't have any **package statement in the Java classes**.

Please submit the following files:

1. A2.pdf.
2. README.txt File
3. The Java files:
  - a. **HornClause.java**
  - b. **HornKB.java**
  - c. **Literal.java**
  - d. **Symbol.java**
  - e. **LogicalAgent.java**