CIS*3700 (Winter 2020): Introduction to Intelligent Systems

ASSIGNMENT 2

Instructor: Y. Xiang
Assigned date: Tue., Feb. 11, 2020. Due Date: Thur., Mar. 12, 2020

Problems (Total marks: 40)

1. (3 marks) (Soundness and completeness of inference)

   Agent Ag uses inference algorithm to deduce sentences from knowledge base KB that is true in task environment TE. Ag has 3 alternative algorithms, Inf1, Inf2, and Inf3. Inf1 is sound but not complete; Inf2 is complete but not sound; and Inf3 is both sound and complete.

   (a) If Inf1 yields sentence S_1, will S_1 be true in TE?

   (b) Let S_2 be a sentence entailed by KB. Can Inf1 derive S_2?

   (c) If Inf2 yields sentence S_3, will S_3 be true in TE?

   (d) Let S_4 be a sentence entailed by KB. Can Inf2 derive S_4?

   (e) If Inf3 yields sentence S_5, will S_5 be true in TE?

   (f) Let S_6 be a sentence that is true in TE. Can Inf3 derive S_6?

2. (3 marks) (Knowledge representation) Suppose that Wumpus World is a 4 x 4 grid. Represent following assertions as propositional logic sentences. Use propositional symbols that convey the meaning, and define meaning of each symbol.

   (a) Room (2,3) has a pit whenever adjacent rooms are breezy, and vice versa.

   (b) When Wumpus is in (1,4) and agent is in (1,2) facing north, if agent shoots, Wumpus will die.

   (c) Exactly one Wumpus exists.

   You may use multiple sentences such that their conjunction expresses this assertion. For each unique sentence pattern, list at least 3 sentences if more are needed, and indicate the number of sentences under this pattern.

3. (2 marks) (Model checking on entailment) Consider a 3 x 2 Wumpus World with rooms (1,1), (2,1), (3,1), (1,2), (2,2) and (3,2). The agent knows that Wumpus only moves among rooms (2,2), (3,1) and (3,2). When agent starts at (1,1), its KB contains the following sentences, where $W22$ stands for whether Wumpus is in room (2,2), and $S22$ stands for whether room (2,2) is smelly.

   $$S_1 : W22 \Rightarrow (S12 \wedge S21 \wedge S32), \quad S_2 : W31 \Rightarrow (S21 \wedge S32), \quad S_3 : W32 \Rightarrow (S22 \wedge S31).$$

   When agent enters the cave from room (1,1), Wumpus is in room (3,1), which is unobservable to agent. After moving from (1,1) to (2,1), agent senses smell, and performs $Tell(S21)$.

   (a) If agent calls $isEntailed(KB, W31)$, how many models are checked in the worst case?

   (b) If agent calls $isEntailed(KB, W31)$, what is the outcome and why?

4. (3 marks) (Convertion to CNF) Convert the following sentence to conjunctive normal form.

   $$(X \wedge \neg Y) \Leftrightarrow W$$

(a) Trace execution of $getCnf()$ algorithm with an execution tree in the format illustrated in lectures. To simply labels of nodes, show each call of $getCnf(s)$ as $gc(s)$, where s is the sentence to be converted. Show the sentence returned by each call.

(b) Show the final CNF by removing each clause that is equivalent to $true$.

5. (2 marks) (Resolution closure)

   Resolution algorithm takes as arguments a knowledge base and a query sentence. If the knowledge base and query contain 10 propositional symbols, what is the size of the resolution closure in the worst case, measured by the total number of clauses?

6. (10 marks) (Implementing CNF conversion)

   [**Task**] Develop a Java program *ExpToCnf* that converts propositional logic sentences to CNF by $getCnf()$ algorithm from lecture. Input sentences are given in a file $ExpFile$ and the result is stored in a file $CnfFile$.

   [**Compile, run, and submit**] Program should compile by placing source code in directory \3700A2FstName, where FstName is your first name, and issuing the command below at \3700A2FstName.

```
javac ExpToCnf.java
```

   It should execute by the following command issued at \3700A2FstName, and there should be no restriction to paths $ExpFile$ and $CnfFile$.

```
java ExpToCnf ExpFile CnfFile
```

   Compress Java source code in directory \3700A2FstName (including code and subdir from the next two problems) into file 3700A2FstName.zip and submit the file.

   [**ExpFile**] It is text file with one or more logic sentences (one per line). Each sentence consists of propositional symbols, logic operators, and (), formatted as follows:

   (a) Operators must be ~ for NOT, ^ for AND, v for OR, => for imply, and <-> for iff. They cannot appear in symbols. Spaces must surround ^, v, =>, and <->. A literal is as W12 or ~W12 (no space between ~ and W12).

   (b) When multiple ^ or v are in sequence, such as A^B^C, indicate order by (), e.g., (A^B)^C or A^(B^C). For A ^ B v C, indicate order as (A^B) v C or A^(B v C).

   (c) Input sentence is not be enclosed by (), but each complex sub-sentence is enclosed by (). Negative literals may or may not be enclosed by (), e.g., `R <-> (P v ~Q)` and `~(~P)`.
   For example, sentence $A \Leftrightarrow (\neg(B \wedge C) \vee (D \wedge E \wedge F) \vee (G \wedge H \wedge I))$ is formatted as (not unique):

```
A <-> (((~(B ^ C)) v ((D ^ E) ^ F)) v ((G ^ H) ^ I))
```

   [**CnfFile**] It is text file of CNF clauses for all sentences from $ExpFile$. Each line is one clause. When $ExpFile$ has sentence `R <-> (P v ~Q)`, the following is the $CnfFile$, where syntax (a) above is followed:

```
R v ~P
R v Q
P v ~Q v ~R
```

   [**Program echo**] The program should echo main steps and the output CNF for each input sentence in the following format. Note that the input sentence is echoed at the start and at the end with the final CNF.

```
Convert (R <-> (P v ~Q))
Convert ((R ^ (P v ~Q)) v (~R ^ ~(P v ~Q)))
Convert (R ^ (P v ~Q))
Convert R
   Return R
Convert (P v ~Q)
Convert P
   Return P
Convert ~Q
   Return ~Q
   Return (P v ~Q)
   Return (R) ^ (P v ~Q)
Convert (~R ^ ~(P v ~Q))
Convert ~R
   Return ~R
Convert ~(P v ~Q)
Convert (~P ^ Q)
Convert ~P
   Return ~P
Convert Q
   Return Q
   Return (~P) ^ (Q)
   Return (~P) ^ (Q)
   Return (~R) ^ (~P) ^ (Q)
   Return (R v ~R)^(R v ~P)^(R v Q)^(P v ~Q v ~R)^(P v ~Q v ~P)^(P v ~Q v Q)
   Return (R v ~R)^(R v ~P)^(R v Q)^(P v ~Q v ~R)^(P v ~Q v ~P)^(P v ~Q v Q)

Exp: R <-> (P v ~Q)
CNF:(R v ~P) ^ (R v Q) ^ (P v ~Q v ~R)
```

7. (10 marks) (Implementing resolution)

   [**Task**] Develop Java program *Resolution* for propositional logic inference, by resolution algorithm in lecture.

   [**Compile, run, and submit**] Program should compile by placing source code in directory \3700A2FstName and issuing command below at \3700A2FstName.

   ```
   javac Resolution.java
   ```

   It should execute by the following command issued at \3700A2FstName, without restriction (full or relative) to paths of input files $KbCnfFile$, $QueryFile$, and $PerceptFile$.

   ```
   java Resolution KbCnfFile QueryFile PerceptFile
   ```

   Compress Java source code in dir \3700A2FstName (including code and subdir from the last and next problems) into file 3700A2FstName.zip and submit the file.

   [**Input files**] Each input files is a list of clauses (one per line), formatted as $CnfFile$ of last problem. $KbCnfFile$ is knowledge base in CNF. Though it may be manually edited, it should be output of $ExpToCnf$ when knowledge base has many sentences. $PerceptFile$ describes agent observations. $QueryFile$ contains **negation** of the query sentence in CNF, and should be output of $ExpToCnf$ if the query is complex.

   [**Java classes**] Implement the following Java classes.

   (a) Literal: It maintains a literal as a symbol and a sign. The symbol is a Java String. The sign is Boolean, indicating whether the literal is positive.

   (b) Clause: It maintains a list of Literals in the clause and represents their disjunction.

(c) ClauseBase: It maintains a set of clauses and represents their conjunction. It should contain methods to load clauses from file, to join two ClauseBases into one, and to perform resolution when the clauses represent $KB \wedge \neg\alpha$, where α is the query sentence.

[**Echo**] After loading input files, echo their clauses. Echo at the end of each round of resolution, and indicate the number of clauses in the current clause base $Clau$ as well as the number in new. At end of resolution, indicate whether KB entails α by "KB $\models$ alpha" or "KB !$\models$ alpha".

8. (7 marks) (Knowledge representation and inference)

(a) Construct KB on Wumpus World with the following assertions. Save the KB in file kb1.txt. Convert kb1.txt to CNF by ExpToCnf, and save outcome in file kb1cnf.txt.

i. Room (1,1) is pit-free.
ii. There is no Wumpus in room (1,1).
iii. Room (1,1) is smelly if and only if Wumpus is in adjacent rooms.
iv. Room (1,1) is breezy if and only if a pit exists in adjacent rooms.
v. Room (1,2) is smelly if and only if Wumpus is in adjacent rooms.
vi. Room (1,2) is breezy if and only if a pit exists in adjacent rooms.
vii. Room (2,1) is smelly if and only if Wumpus is in adjacent rooms.
viii. Room (2,1) is breezy if and only if a pit exists in adjacent rooms.

(b) Encode the following agent percepts in file per1.txt. Convert per1.txt to CNF by ExpToCnf, and save outcome in file per1cnf.txt.

i. There is no smell in room (1,1).
ii. There is no breeze in room (1,1).
iii. There is no smell in room (2,1).
iv. There is breeze in room (2,1).
v. There is smell in room (1,2).
vi. There is no breeze in room (1,2).

(c) Write query sentence α for each query below, and save sentence $\neg\alpha$ to file as specified.

i. Room (1,2) is pit-free. Save to file nq1.txt.
ii. Room (2,2) is Wumpus-free. Save to file nq2.txt.
iii. Either room (2,2) or (3,1) has pit. Save to file nq3.txt.
iv. There is Wumpus in room (1,2). Save to file nq4.txt.

Convert each nqx.txt ($x = 1, 2, 3, 4$) to CNF by ExpToCnf, and save outcome to file nqxcnf.txt.

(d) Run Resolution using kb1cnf.txt, per1cnf.txt, and each nqxcnf.txt. For each execution, record and submit the following in your short answer assignment submission.

i. How many clauses are in $Clau$ before the first round of resolution?
ii. How many clauses are in $Clau \cup new$ when resolution terminates?
iii. Does KB entail the query sentence α?

(e) Place all .txt files mentioned above in subdir `\3700A2FirstName\Q8`. Compress all files and subdir \Q8 in dir \3700A2FirstName (including code from the last two problems) into file 3700A2FirstName.zip and submit the file.