Washington State University
School of Electrical Engineering and Computer Science
Fall 2019

CptS 440/540 Artificial Intelligence
Homework 6 - Solution
Due: October 10, 2019 (11:59pm)

General Instructions: Put your answers to the following problems into a PDF document and submit as an attachment under Content → Homework 6 for the course CptS 440 Pullman (all sections of CptS 440 and 540 are merged under the CptS 440 Pullman section) on the Blackboard Learn system by the above deadline. Note that you may submit multiple times, but we will only grade the most recent entry submitted before the above deadline.

1. Consider the following logic problem:

   All people who like computers also like coding.
   All people who like coding and like chess will learn AI.
   For all people that learn AI, there is at least one company that will hire them.
   All people who are hired by some company will be rich and famous.

   a. We will solve this problem using first-order logic. First, show one first-order logic sentence for each of the first four sentences in the above problem. You may only draw from the following first-order predicates.
      - Like(x, Chess)
      - Like(x, Computers)
      - Like(x, Coding)
      - Learn(x, AI)
      - Hire(c, x) – which means company c hires person x
      - Rich(x)
      - Famous(x)

      Solution:
      i. $\forall x$ (Like(x, Computers) $\Rightarrow$ Like(x, Coding))
      ii. $\forall x$ (Like(x, Coding) $\wedge$ Like(x, Chess) $\Rightarrow$ Learn(x, AI))
      iii. $\forall x$ (Learn(x, AI) $\Rightarrow \exists c$ Hire(c, x))
      iv. $\forall x$ ($\exists c$ Hire(c, x) $\Rightarrow$ (Rich(x) $\wedge$ Famous(x))

   b. Convert each of the four sentences from part (a) into Conjunctive Normal Form (CNF). You may just show the final result for each sentence; no need to show the intermediate steps. Number each clause. We will refer to these clauses as the knowledge base (KB) below.

Solution:

CNF:

i. $\neg$Like(x, Computers) $\vee$ Like(x, Coding)
ii. $\neg$Like(x, Coding) $\vee$ $\neg$Like(x, Chess) $\vee$ Learn(x, AI)
iii. $\neg$Learn(x, AI) $\vee$ Hire(SK(x), x)
iv. ($\neg$Hire(c, x) $\vee$ Rich(x)) $\wedge$ ($\neg$Hire(c, x) $\vee$ Famous(x))

Clauses:

1. $\neg$Like(x, Computers) $\vee$ Like(x, Coding)
2. $\neg$Like(x, Coding) $\vee$ $\neg$Like(x, Chess) $\vee$ Learn(x, AI)
3. $\neg$Learn(x, AI) $\vee$ Hire(SK(x), x)
4. $\neg$Hire(c, x) $\vee$ Rich(x)
5. $\neg$Hire(c, x) $\vee$ Famous(x)

c. To the KB from part (b), add the two facts: "Larry likes computers" and "Larry likes chess". Using this augmented KB, perform a resolution proof by refutation to prove "Rich(Larry)". In your proof, be sure to do the following:
   - For each resolution step, show the numbers of the two clauses used, the resulting clause, any variable substitutions resulting from unifying the complementary literals, and then number the resulting clause.
   - Be sure to use unique variables (e.g., x_1, x_2, c_1, etc.) for each new use of a clause from the KB.
   - Remember: each resolution step can only be done with two clauses at a time and can eliminate only one literal from each clause.

Solution:

Additional clauses:

6. Like(Larry, Computers)
7. Like(Larry, Chess)

Negated query:

8. $\neg$Rich(Larry)

Proof:

Resolve 4 ($\neg$Hire(c_1, x_1) $\vee$ Rich(x_1)) and 8 ($\neg$Rich(Larry)) with {x_1 / Larry}:

9. $\neg$Hire(c_1, Larry)

Resolve 9 and 3 ($\neg$Learn(x_2, AI) $\vee$ Hire(SK(x_2), x_2)) with {c_1 / SK(x_2), x_2 / Larry}:

10. $\neg$Learn(Larry, AI)

Resolve 10 and 2 ($\neg$Like(x_3, Coding) $\vee$ $\neg$Like(x_3, Chess) $\vee$ Learn(x_3, AI)) with {x_3 / Larry}:

11. $\neg$Like(Larry, Coding) $\vee$ $\neg$Like(Larry, Chess)

Resolve 11 and 7:

12. $\neg$Like(Larry, Coding)

Resolve 12 and 1 ($\neg$Like(x_4, Computers) $\vee$ Like(x_4, Coding)) with {x_4 / Larry}:

13. $\neg$Like(Larry, Computers)

Resolve 13 and 6:

14. Empty clause

Thus, $\neg$Rich(Larry) must be false, and therefore, Rich(Larry) is true.

d. To the KB from part (b), add the fact: “There exists someone who learns AI.” Be sure to first convert this fact to CNF. Using this augmented KB, perform a resolution proof by refutation to prove “$\exists x$ Famous(x)”, following the guidelines in part (c).

Solution:

Additional clause:

6. Learn(SK0, AI)

Negated query:

7. $\neg$Famous(x)

Proof:

Resolve 7 ($\neg$Famous(x_1)) and 5 ($\neg$Hire(c_1, x_2) $\vee$ Famous(x_2)) with {x_1 / x_2}:

8. $\neg$Hire(c_1, x_2)

Resolve 8 and 3 ($\neg$Learn(x_3, AI) $\vee$ Hire(SK(x_3), x_3)) with {c_1 / SK(x_3), x_2 / x_3}:

9. $\neg$Learn(x_3, AI)

Resolve 9 and 6 with {x_3 / SK0}:

10. Empty clause

Thus, $\neg$Famous(x) must be false, and therefore, $\exists x$ Famous(x) is true.

2. *CptS 540 Students Only*. Create an input file for the Vampire theorem prover that can be used to solve Problem 1c. Include your input file and the Vampire output in your Homework 6 submission. You can include them as separate text files or cut-and-paste them into the document containing the rest of your solution. You can download the Vampire theorem prover from https://vprover.github.io/. There is a Linux binary available there that runs on the ssh1-ssh10 servers that all grad students have access to. You can run it on your own machine if you prefer or build it from source.

   Solution:

   Vampire Input:

```
fof(a1, axiom,
    ! [X] : (like(X,computers) => like(X,coding))).

fof(a2, axiom,
    ! [X] : ((like(X,coding) & like(X,chess)) => learn(X,ai))).

fof(a3, axiom,
    ! [X] : (learn(X,ai) => (? [C] : hire(C,X)))).

fof(a4, axiom,
    ! [X] : ((? [C] : hire(C,X)) => (rich(X) & famous(X)))).

fof(a5, axiom, like(larry,computers)).

fof(a6, axiom, like(larry,chess)).

fof(c1, conjecture, rich(larry)).
```

Vampire Output:

```
% Refutation found. Thanks to Tanya!
% SZS status Theorem for
% SZS output start Proof for
1. ! [X0] : (like(X0,computers) => like(X0,coding)) [input]
2. ! [X0] : ((like(X0,chess) & like(X0,coding)) => learn(X0,ai)) [input]
3. ! [X0] : (learn(X0,ai) => ? [X1] : hire(X1,X0)) [input]
4. ! [X0] : (? [X1] : hire(X1,X0) => (famous(X0) & rich(X0))) [input]
5. like(larry,computers) [input]
6. like(larry,chess) [input]
7. rich(larry) [input]
8. ~rich(larry) [negated conjecture 7]
9. ~rich(larry) [flattening 8]
10. ! [X0] : (? [X1] : hire(X1,X0) => rich(X0)) [pure predicate removal 4]
11. ! [X0] : (like(X0,coding) | ~like(X0,computers)) [ennf transformation 1]
12. ! [X0] : (learn(X0,ai) | (~like(X0,chess) | ~like(X0,coding))) [ennf
transformation 2]
13. ! [X0] : (learn(X0,ai) | ~like(X0,chess) | ~like(X0,coding)) [flattening
12]
14. ! [X0] : (? [X1] : hire(X1,X0) | ~learn(X0,ai)) [ennf transformation 3]
15. ! [X0] : (rich(X0) | ! [X1] : ~hire(X1,X0)) [ennf transformation 10]
16. ! [X0] : (? [X1] : hire(X1,X0) => hire(sK0(X0),X0)) [choice axiom]
17. ! [X0] : (hire(sK0(X0),X0) | ~learn(X0,ai)) [skolemisation 14,16]
18. ~like(X0,computers) | like(X0,coding) [cnf transformation 11]
19. ~like(X0,chess) | learn(X0,ai) | ~like(X0,coding) [cnf transformation 13]
20. hire(sK0(X0),X0) | ~learn(X0,ai) [cnf transformation 17]
21. ~hire(X1,X0) | rich(X0) [cnf transformation 15]
22. like(larry,computers) [cnf transformation 5]
23. like(larry,chess) [cnf transformation 6]
24. ~rich(larry) [cnf transformation 9]
25. like(larry,coding) [resolution 18,22]
26. ~learn(X0,ai) | rich(X0) [resolution 20,21]
27. learn(larry,ai) | ~like(larry,coding) [resolution 19,23]
28. learn(larry,ai) [subsumption resolution 27,25]
29. rich(larry) [resolution 26,28]
30. $false [subsumption resolution 29,24]
% SZS output end Proof for
% ------------------------------
% Version: Vampire 4.2.2 (commit e1949dd on 2017-12-14 18:39:21 +0000)
% Termination reason: Refutation


% Memory used [KB]: 4733
% Time elapsed: 0.312 s
% ------------------------------
% ------------------------------
```