CS221 Fall 2018 Homework 8

SUNet ID: 05794739
Name: Luis Perez
Collaborators:

By turning in this assignment, I agree by the Stanford honor code and declare that all of this is my own work.

Problem 1

(a) In "submission.py".

(b) In "submission.py".

(c) In "submission.py".

Problem 2

(a) In "submission.py".

(b) In "submission.py".

(c) In "submission.py".

(d) In "submission.py".

Problem 3

(a) In "submission.py".

Problem 4

(a) We begin with a KB $= \{A \vee B) \to C, A\}$. We can conver this into conjunctive normal form (following the hint) to arrive at:

$$\begin{aligned}
\text{KB} &= [\neg(A \vee B) \vee C] \wedge A && \text{(apply hint)} \\
&= (\neg A \vee C) \wedge A \wedge (\neg B \vee C) && \text{(distribute the not)} \\
&= (\neg A \vee C) \wedge A \wedge C \wedge (\neg B \vee C) \\
&&& \text{(application of Modus ponens to } (\neg A \vee C) \wedge A)
\end{aligned}$$

By Modus ponens, we therefore have the formula C.

(b) We follow a similar approach to before by first converting the database to CNF.

$$
\begin{aligned}
\text{KB} &= (A \vee B) \wedge (\neg B \vee C) \wedge [\neg(A \vee C) \vee D] && \text{(hint from previous problem)} \\
&= (A \vee B) \wedge (\neg B \vee C) \wedge (\neg A \vee D) \wedge (\neg C \vee D) && \text{(distributing the not)} \\
&= (A \vee B) \wedge (\neg B \vee C) \wedge (C \vee A) \wedge (\neg A \vee D) \wedge (\neg C \vee D) \\
& && \text{(resolution rule applied to } (A \vee B) \wedge (\neg B \vee C)\text{)} \\
&= (A \vee B) \wedge (\neg B \vee C) \wedge (C \vee A) \wedge (\neg A \vee D) \wedge (D \vee C) \wedge (\neg C \vee D) \\
& && \text{(resolution rule applied to } (C \vee A) \wedge (\neg A \vee D)\text{)} \\
&= (A \vee B) \wedge (\neg B \vee C) \wedge (C \vee A) \wedge (\neg A \vee D) \wedge (D \vee C) \wedge (\neg C \vee D) \wedge D \\
& && \text{(resolution rule applied to } (D \vee C) \wedge (\neg C \vee D)\text{)}
\end{aligned}
$$

We have now derived D in our databse.

Problem 5

(a) In "submission.py".

(b) We seek to prove by contradiction. Suppose we have a finite, non-empty model (set of assignments $X = \{x_1, \cdots, x_n\}$) which satisfies all of the 7 contraints. By the newly added constraint, we must have that a number is not larger than itself. Combined with the transitive property of "larger", this induces an absolute ordering on our finite, non-empty model. In other words, these two constraints imply that there is an x_{LARGEST} which has no value which is larger. However, this contradicts the fact that each x has exactly one successor, and that this succesor is larger.

Problem 6

(a) In "submission.py".

(b) In "submission.py".

(c) In "submission.py".