Intelligent System - HW3

Vinh T. Nguyen. Email: vinh.nguyen@ttu.edu

October 12, 2018

1 Question 1

Given the signature $O = \{a\}, F = \{f\}, P = \{p\}$ and $V = \{X\}$, specify which of the following are terms, atom, literals, or none.

(a) a.	(b) $\neg a$	(c) X	(d) Y
(e) $f(X)$	(f) $f(f(X))$	(g) $p(a)$	(h) $p(\neg a)$
(i) $\neg p(a)$			

Table 1: Specifying which of the following are terms, atoms, literals, and none

Answer:
Terms:= $\{a, X, f(X), f(f(X))\}$
Atoms:= $\{p(a)\}$
Literals:= $\{p(a), \neg p(a)\}$
None:= $\{\neg a, Y, p(\neg a)\}$

2 Question 2

Given the signature $O = \{a, b\}, F = \{\}, P = \{p, q\}$ and $V = \{X, Y\}$, what is the grounding of the following program.
$p(X, Y) : -q(Y, X)$.
Answer:
$p(a, a) : -q(a, a)$. *% when X = a, Y = a %*
$p(a, b) : -q(b, a)$. *% when X = a, Y = b %*
$p(b, a) : -q(a, b)$. *% when X = b, Y = a %*
$p(b, b) : -q(b, b)$. *% when X = b, Y = b %*

3 Question 3

(Satisfaction of rules) Given a rule r and S where
r is $q(c)$ or $q(a) : -p(a), \neg s(b)$, not $s(a)$, and
$S = \{\neg q(c), p(a), \neg s(b)\}$,
does S satisfy r? Give detailed explanation.
Answer:
First we check the body of the rule if it is satisfied by S. The body consists of three literals:

$p(a), \neg s(b)$, not $s(a)$.

The first two ($p(a), \neg s(b)$) are satisfied by clause 1 (l if $l \in S$).

The later *not* $s(a)$ is satisfied by clause 2 (not l if $l \notin S$).

Thus the body of the rule is satisfied by S. By clause 5 (rule r if, whenever S satisfies $r's$ body, it satisfies $r's$ head), the head must also be satisfied. However, neither $q(c)$ or $q(a)$ is in S by clause 3 (l_1 or ... or l_n **if** for some $1 \leq i \leq n$, $li \in S$). Hence, the rule r is not satisfied by S.

4 Question 4

(Answer set definitions) Explain, in a precise manner, that $S = \{p(b)\}$ is an answer set of the following program:

$p(a) : -$not $p(b)$. % If $p(b)$ does not belong to your set of beliefs, then $p(a)$ must.%

$p(b) : -$not $p(a)$. % If $p(a)$ does not belong to your set of beliefs, then $p(b)$ must.%

This answer set supports the second rule, since $p(a)$ is not in the set then $p(b)$ must be in the set.

5 Question 5

State the definition of a program **entails** a literal. *A program* Π *entails a literal* l *(*$\Pi \mid = l$*) if* l *belongs to all answer sets of* Π

State the definition of **answer** to a disjunctive query to a program: The answer to a ground disjunctive query, l_1 or ... or l_n,where $n \geq 1$, is

- The answer is **yes** if there exists at least i such that $\Pi \mid = l_i$. In other words, there is at least one literal i belongs to all answer set of Π.
- The answer is **no** if Π entails all negation of literals ($\Pi \mid = \{\overline{l_1}, ..., \overline{l_n}\}$). In other words, all negation of literals belong to all answer set of Π.
- **unknown** otherwise

Answer:

6 Question 6

(Query Answering) Assume a program Π has one answer set $\{p(a), \neg q(a)\}$. What is the answer to the following queries?.

- (1) ? $p(a)$. The answer is **Yes**.
- (2) ? $q(a)$. The answer is **No**.
- (3) ? $p(a) \wedge q(a)$. The answer is **No**.
- (4) ? $p(a)$ or $q(a)$. The answer is **Yes**
- (5) ? $p(X)$. The answer is $X = a$.

If Π has one more answer set $\{p(a), q(a)\}$, what would be the answer for the queries above?

- (1) ? $p(a)$. The answer is **Yes**.
- (2) ? $q(a)$. The answer is **Yes**.
- (3) ? $p(a) \wedge q(a)$. The answer is **Yes**.
- (4) ? $p(a)$ or $q(a)$. The answer is **Yes**
- (5) ? $p(X)$. The answer is $X = a$.

7 Question 7

Compute the answer sets of the following program.

$p(a) :-$ not $p(b)$.
$p(b) :-$ not $p(a)$.
$q(a)$.
$\neg q(b) :- p(X)$, not $r(X)$.

Recall that you can figure the the signature of the program from the program itself. Remember to ground the rules with variables.

Answer:

After grounding the program Π we have
$p(a) :-$ not $p(b)$.
$p(b) :-$ not $p(a)$.
$q(a)$.
$\neg q(b) :- p(a)$, not $r(a)$.
$\neg q(b) :- p(b)$, not $r(b)$.

From the first rule and second rule, we have two answer set $\{p(a)\}$ and $\{p(b)\}$. The third rule is the fact so that it should be in every set. Therefore, we have two answer sets $\{p(a), q(a)\}$ and $\{p(b), q(a)\}$. The program has no rule that supports $r(a)$ or $r(b)$, so every answer set of the program satisfies *not* $r(a)$ and *not* $r(b)$. This means that every answer set that contains $p(a)$ must also contain $\neg q(b)$, and every answer set that contains $p(b)$ must also contain $\neg q(b)$. Thus we have two answer sets

$\{p(a), q(a), \neg q(b)\}$ and
$\{p(b), q(a), \neg q(b)\}$