Homework 2

Due: Tues, Feb 6, 2024

1. Given the following program Π:

$$\begin{array}{l} p \leftarrow r,\ not\ s. \\ r. \end{array}$$

   (a) Given candidate set $S_1 = \{r\}$, compute Π^{S_1}.

   (b) Is S_1 an answer set of Π? Justify your answer by using the definition of answer set.

   (c) Given candidate set $S_2 = \{p, r\}$, compute Π^{S_2}.

   (d) Is S_2 and answer set of Π? Justify your answer.

   (e) In considering candidate sets for program Π, we could exclude ones which contain any of the following literals: s, $\neg p$, $\neg r$, $\neg s$. Why?

2. Given the following program Π:

$$\begin{array}{l} \neg p\ or\ r. \\ \neg p \leftarrow\ not\ p. \end{array}$$

whose signature is limited to the predicates used in the program.

   (a) List *all* possible sets that can be made from the signature of this program. Hint: there are 16; many of them can be ruled out immediately in your search for an answer set, but I want you to see the difference between random beliefs and those limited by the definition of answer set.

   (b) Compute the answer set(s) of the program if they exist. Show at least one relevant candidate set and reduct.

3. Given the following program Π:

$$\begin{array}{l} p \leftarrow r,\ not\ s. \\ s \leftarrow\ not\ p. \end{array}$$

Compute the answer set(s) of the program if they exist. Show at least one candidate set and reduct relevant to your answer.

4. Suppose a program has the following answer sets:

$$\begin{array}{l} A_1 = \{p(a), \neg p(b), r, s, \neg t\} \\ A_2 = \{p(a), \neg s, \neg t\} \end{array}$$

How would it answer the following queries? (Make sure you use the definition of query from the slides, not the book, or see the update on the book webpage under Errata.)

(a) ? $\neg p(a)$

(b) ? $\neg p(b)$

(c) ? s

(d) ? $t \wedge \neg s$

(e) ? $\neg t \vee s$

5. Consider one of Jordan Peterson's rules for life:
"Pursue what is meaningful, (not what is expedient)."
A possible direct translation of this statement might look like this:

$$\begin{array}{ll} pursue(X) \leftarrow meaningful(X). & \%\ rule1 \\ -pursue(X) \leftarrow expedient(X). & \%\ rule2 \end{array}$$

However, if we add two lines to represent a thing which is both meaningful and expedient,

$$\begin{array}{l} meaningful(thing). \\ expedient(thing). \end{array}$$

we will get no answer set.

Thus, the program does not capture the desired meaning. Consider the implicit, commonsense knowledge that is missing, and rewrite rule 2 to fix the problem.