CS345A Midterm Solutions

Problem 1: (40 points) Consider the following rules:

```
p(X) :- int(X) & X >= 2 & not c(X)
c(X) :- int(X) & p(Y) & divides(X,Y) & X != Y
```

Think of $p(X)$ as meaning "X is a prime" and $c(X)$ as "X is composite." The EDB predicate $int(X)$ says that X is a positive integer, and in practice it will hold a finite set of integers. The EDB predicate $divides(X, Y)$ means that Y evenly divides X.

Suppose that $int = \{1, 2, 3, 4\}$, and *divides* is the expected relation on these four integers; that is, $divides = \{(1,1), (2,1), (3,1), (4,1), (2,2), (4,2), (3,3), (4,4)\}$. If we instantiate these rules in all possible ways, eliminate rules with a known false subgoal and then eliminate known true subgoals from the remaining rules, we are left with the following:

```
p(2) :- not c(2)     c(2) :- p(1)
p(3) :- not c(3)     c(3) :- p(1)
p(4) :- not c(4)     c(4) :- p(1)
                     c(4) :- p(2)
```

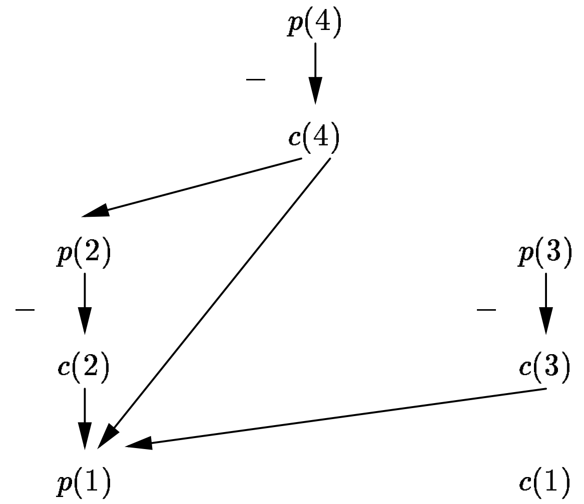
a) In the table below, construct the well-founded model by the definition (not the alternating fixedpoint). Indicate what inferences are made about the truth or falsehood of $p(i)$ and $c(i)$, for $1 \leq i \leq 4$ at each round.

Round	Positive Inferences	Negative Inferences
1		$\neg c(1)$, $\neg c(2)$, $\neg c(3)$, $\neg p(1)$
2	$p(2)$, $p(3)$, $c(4)$	$\neg p(4)$

NOTE: Many people did not get the maximal unfounded set at a round or make all possible inferences at a round. We gave you some leeway on the midterm.

NOTE: The instructor was embarrassed to realize that he did not explain the alternating fixedpoint method properly. On each round, you start with the information about negative subgoals from the previous round, but you make *all* possible positive inferences, including those that require several successive rule applications. It didn't make a difference in the class example, but it does for the question on the midterm (which is why I decided not to ask for the alternating fixedpoint). If you aren't aggressive about making positive inferences, you don't get the nice alternation between upper and lower bounds, although you do converge eventually.

b) In the space below, draw the dependency graph for the instantiated atoms $p(i)$ and $c(i)$ for $1 \leq i \leq 4$.



c) Are the rules with the given EDB locally stratified? Ans: Yes. If so, tell what the strata are; if not, describe an infinite negative path.

Stratum 0: $p(1)$, $c(1)$, $c(2)$, $c(3)$
Stratum 1: $p(2)$, $p(3)$, $p(4)$
Stratum 2: $p(4)$

NOTE: Compare the strata to the answer for (a). Is that a coincidence?

d) Suppose *int* contains the integers from 1 to n, and *divides* contains all those pairs (i, j) such that j divides i and i and j are integers between 1 and n. For what values of n will the rules and EDB be locally stratified? Explain briefly.

Ans: For all values of n. The c's only depend positively on smaller p's, while p's depend only negatively on the same c. Thus, all paths have arguments that alternatingly decrease and stay the same, and arguments can never go below 1. All paths are therefore finite. Certainly the number of negative steps on paths are finite; so are strata.

Problem 2: (30 points) Consider the following conjunctive queries:

Q_1: `h(X) :- red(X,Y) & blue(Y,Y) & red(Y,Z)`
Q_2: `h(X) :- red(X,Y) & blue(Y,Z) & red(Z,W)`
Q_3: `h(X) :- red(X,Y) & red(Y,Z)`

In the table below, for each containment, either give a containment mapping that proves it or give a database that serves as a counterexample.

Containment	Counterexample or Containment Mapping
$Q_1 \subseteq Q_2$	$X \to X,\ Y \to Y,\ Z \to Y,\ W \to Z$
$Q_1 \subseteq Q_3$	$X \to X,\ Y \to Y,\ Z \to Z$
$Q_2 \subseteq Q_1$	$red(x,y),\ blue(y,z),\ red(z,w)$
$Q_2 \subseteq Q_3$	Same
$Q_3 \subseteq Q_1$	$red(x,y),\ red(y,z)$
$Q_3 \subseteq Q_2$	Same

Problem 3: (20 points) Below are two conjunctive queries with arithmetic comparisons.

Q_1': **panic** `:- arc(X,Y) & arc(Y,Z) & X <= Y & Y <= Z`
Q_2': **panic** `:- arc(X,Y) & arc(Y,Z) & X <= Z`

To prove that $Q_1' \subseteq Q_2'$ we begin by splitting the double use of Y in each of the conjunctive queries:

Q_1: **panic** `:- arc(X,Y) & arc(W,Z) & W=Y & X <= Y & Y <= Z`
Q_2: **panic** `:- arc(X,Y) & arc(W,Z) & W=Y & X <= Z`

a) What are all the containment mappings from the ordinary subgoals of Q_2 to the ordinary subgoals of Q_1? Name them and list them here:

μ	$\mu(W)$	$\mu(X)$	$\mu(Y)$	$\mu(Z)$
μ_1	W	X	Y	Z
μ_2	X	X	Y	Y
μ_3	W	W	Z	Z
μ_4	X	W	Z	Y

b) The "Gupta test" for $Q_1 \subseteq Q_2$ requires that we write down a logical expression. Write that expression here:

$$\begin{aligned} W = Y \ \wedge\ X \le Y \ \wedge\ Y \le Z \ \Rightarrow & \\ & W = Y \ \wedge\ X \le Z \ \vee \\ & X = Y \ \wedge\ X \le Y \ \vee \\ & W = Z \ \wedge\ W \le Z \ \vee \\ & X = Z \ \wedge\ W \le Y \end{aligned}$$

NOTE: Common error: forgetting that two subgoals can be mapped to the same subgoal.

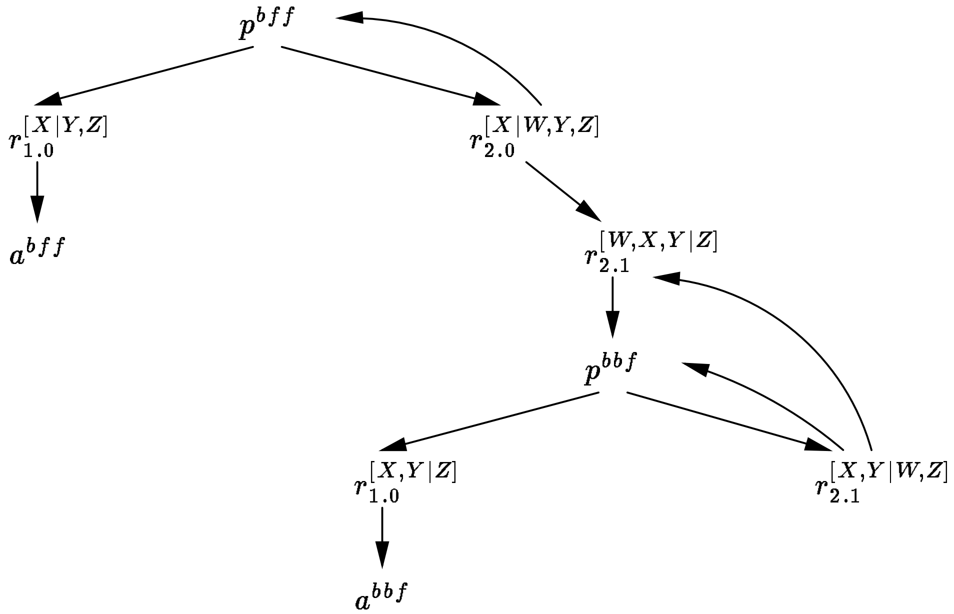
c) Briefly explain why your expression from (b) holds. *Hint*: The implication of disjunction (logical OR) can be shown true by showing that the hypothesis implies any *one* of the disjuncts.

Ans: The hypothesis implies the first of the disjuncts. That is, $W = Y$ surely implies $W = Y$, and the combination of $X \le Y$ and $Y \le Z$ implies $X \le Z$ by transitivity of $\le$.

Problem 4: (20 points) For the rules

```
p(X,Y,Z) :- a(X,Y,Z)
p(X,Y,Z) :- p(X,Y,W) & p(Y,W,Z)
```

draw the rule goal graph starting with the goal p^{bff} in the space below.



NOTE the example of "reuse" involving a rule node.

Problem 5: (20 points) Convert the following rules:

```
p(X,Y) :- a(X,Y)
p(X,Y) :- p(X,Z) & p(Z,Z) & p(Z,Y)
```

into equivalent rules with rectified subgoals.

```
p(X,Y) :- a(X,Y)
p(X,Y) :- p(X,Z) & q(Z) & p(Z,Y)

q(Z) :- a(Z,Z)
q(Z) :- p(Z,W) & q(W) & p(W,Z)
```

Problem 6: (20 points) For the query $p(0, Z)$, apply a magic-sets transformation to the following rules:

r_1: `p(X,Y) :- c(X,Y)`
r_2: `p(X,Y) :- a(X,U) & p(U,V) & b(V,W) & p(W,Y)`

You may use either the basic transformation or a simplified transformation that eliminates some supplementary predicates.

Group I

```
m_p(U) :- sup2.1(X,U)
m_p(W) :- sup2.3(X,W)
```

Group II

```
sup1.0(X) :- m_p(X)
sup2.0(X) :- m_p(X)
```

Group III

```
sup2.1(X,U) :- sup2.0(X) & a(X,U)
sup2.2(X,V) :- sup2.1(X,U) & p(U,V)
sup2.3(X,W) :- sup2.2(X,V) & b(V,W)
```

Group IV

```
p(X,V) :- sup1.0(X) & c(X,Y)
p(X,Y) :- sup2.3(X,W) & p(W,Y)
```

Group V

```
m_p(0)
```

NOTE: Common error: putting useless variables into supplementary relations.