Semantics of Normal Programs

Negative information from definite programs

1. Consider the program P:
   a. p(X) ←q(Y), r(Y).
   b. q(h(Y)) ←q(Y).
   c. r(g(Y)) ←.
   d. r(a) ←.
   Draw an infinite and a finite SLD-tree for the query ← p(a). Are they fair?
   Is $\neg p(a)$ provable by CWA? By NAF?

   (a) Infinite SLD-tree:

Goal	**Clause**	**Substitution**
←p(a).	a.	{X/a}
←q(Y),r(Y).	b.	{Y/h(Y1)}
←q(Y1),r(h(Y1)).	b.	{Y1/h(Y2)}
←q(Y2),r(h(h(Y2))).	b.	{Y2/h(Y3)}
⋮		

   It is not a *fair* tree because the atom $r(\ldots)$ is never selected.

   (b) Finite SLD-tree:

Goal		**Clause**	**Substitution**
←p(a).		a.	{X/a}
←q(Y),r(Y).		c. & d.	{Y/a}&Y/g(Y1)
←q(a).	←q(g(Y1)).		
fail	fail		

   It is a *fair* tree because it is finite.

   The fair tree fails finitely, hence $\neg p(a) \in FFS(P)$ and $\neg p(a)$ is provable by NAF;
   $p(a) \notin M_P$ hence $\neg p(A) \in CWA(P)$ i.e. it is provable by CWA.
   (Another argument: $FFS(P) \subseteq CWA(P)$, $\neg p(a)$ is provable by NAF, hence it is provable by CWA)

2. Consider the following program P:
   a. parent(a,b) ←.
   b. parent(b,c) ←.
   c. anc(X,Y) ←parent(X,Y).
   d. anc(X,Y) ←anc(X,Z), anc(Z,Y).

   (a) Construct for the query ← anc(a,c): (i) a finite SLD-refutation, (ii) an unfair infinite SLD-derivation, (iii) a fair infinite SLD-derivation.
   Is $\neg anc(a, c)$ provable by CWA? By NAF?

   i. Finite SLD-refutation:

Goal	**Clause**	**Substitution**
←anc(a,c).	d.	{X/a, Y/c}
←anc(a,Z),anc(Z,c).	c.	{X/a, Y/Z}
←parent(a,Z),anc(Z,c).	a.	{Z/b}
←anc(b,c).	c.	{X/b, Y/c}
←parent(b,c).	b.	
□		

ii. Unfair infinite SLD-derivation:

Goal	Clause	Substitution
←anc(a,c).	d.	{X/a, Y/c}
←anc(a,Z),anc(Z,c).	d.	{X1/a, Y1/Z}
←anc(a,Z1),anc(Z1,Z),anc(Z,c).	d.	{X2/a, Y2/Z1}
←anc(a,Z2),anc(Z2,Z1),anc(Z1,Z),anc(Z,c).	d.	{X3/a, Y3/Z2}

⋮

select leftmost atom

iii. Fair infinite SLD-derivation:

Goal	Clause	Substitution
←anc(a,c).	d.	{X/a, Y/c}
←anc(a,Z),anc(Z,c).	d.	{X1/a, Y1/Z}
←anc(a,Z1),anc(Z1,Z),anc(Z,c).	d.	{X2/Z, Y2/c}
←anc(a,Z1),anc(Z1,Z),anc(Z,Z2),anc(Z2,c).	d.	{X3/a,Y3/Z1}
←anc(a,Z3),anc(Z3,Z1),anc(Z1,Z),anc(Z,Z2),anc(Z2,c).		

⋮

select least recent atom

There exists a refutation hence $\neg anc(a,c)$ is neither provable by CWA nor by NAF

(b) Construct for the query ← anc(c,a): (i) a failed SLD-derivation, (ii) a fair SLD-tree, (iii) an unfair SLD-tree.

Is $\neg anc(c,a)$ provable by CWA? By NAF?

i. Failed SLD-derivation:

Goal	Clause	Substitution
←anc(c,a).	c.	{X/c, Y/a}
←parent(c,a).		
failure		

ii. Fair SLD-tree:

Goal		Clause	Substitution
←anc(c,a).		d.&c.	{X/c,Y/a}& {X/c,Y/a}
	←parent(c,a).		
←anc(c,Z),anc(Z,a).	failure	d. &c.	{Z/Y1}& {. . . }
←parent(c,Y1), anc(Z,a).	. . .		

⋮

Always select the least recent atom

iii. Unfair SLD-tree:

similar but always select the leftmost atom.

The fair SLD-tree is infinite, hence $\neg anc(c,a)$ cannot be proven by NAF.

$anc(c,a)$ is not in the least Herbrand model (and there exists no refutation for the goal $\leftarrow anc(c,a)$) hence $\neg anc(c,a)$ is provable by CWA.

3. Choose a language and construct an interpretation such that the equality axioms 1,2 and 3 are true but neither 4 nor 5 are true.

Functors: $a/0$, $b/0$ and $f/1$; predicate $p/1$

Domain: $\{0, 1, 2, 3\}$

Consider the following interpretation: $a_I = 0, b_I = 1, f_I(0) = 2, f_I(1) = 3, f_I(2) = 2, f_I(3) = 3$;
$I(=)$ (or $=_I$) $= \{\langle 0,0\rangle, \langle 1,1\rangle, \langle 2,2\rangle, \langle 3,3\rangle, \langle 0,1\rangle\langle 1,0\rangle\}$;
$p_I = \{\langle 0\rangle\}$
One can easily verify that axioms 1,2 and 3 are true in I
The formula $x = y \rightarrow f(x) = f(y)$ is false for the case $x = 0$, $y = 1$ ($\langle 0,1\rangle \in =_I$, $\langle 2,3\rangle \notin =_I$), hence axiom 4 is false in I.
The formula $x = y \rightarrow p(x) \leftrightarrow p(y)$ is false for the case $x = 0$, $y = 1$ ($\langle 0,1\rangle \in =_I$, $p(0) \in p_I$, $p(1) \notin p_I$), hence axiom 5 is false in I.

4. Let $D = \{1, 2\}$, Choose a language and construct an interpretation I over domain D such that axioms 6,7,and 8 evaluate to true but not all of 1,2,3,4,5 evaluate to true.

   Functors: $a/0$, $b/0$.
   Interpretation: $a_I = 1$, $b_I = 2$.
   $I(=) = \emptyset$.
   The axioms 6,7, and 8 evaluate to true (the condition is always false)
   axiom 1 evaluates to false.

5. The program:
   $p(f(x)) \leftarrow p(x) \quad q(a) \leftarrow p(x)$
   $gfp(T_P) = \emptyset$

   The interpretation:
   Domain $= Z \cup N$ (integers + natural numbers)
   $J(a) = 0_N, J(f(n)) = n + 1, J(f(z)) = z + 1$
   $I(p/1) = \{\langle z\rangle \mid z \in Z\}, I(q/1) = \{\langle 0_N\rangle\}, I(=)\{\langle k,k\rangle | k \in Z \cup N\}$
   Show that I is a model of the completion.

   The IFF part of the completion has two formulas:

   (a) $\forall x : p(x) \leftrightarrow \exists y : x = f(y) \wedge p(y)$
   For arbitrary $n \in N$, one obtains:
   $p(n) \leftrightarrow \exists y : n = f(y) \wedge p(y)$
   $p(n)$ is false for all n; for $y \in N$, $p(y)$ is false;
   for $y \in Z$, $p(y)$ is true, but $n = f(y)$ is false, hence no y exists and we obtain $false \leftrightarrow false$ which is true.
   For abritrary $z \in Z$ one obtains:
   $p(z) \leftrightarrow \exists y : z = f(y) \wedge p(y)$
   which (for $y = z - 1$) reduces to $true \leftrightarrow z = (z - 1) + 1 \wedge true$ which is true.

   (b) $\forall x : q(x) \leftrightarrow \exists y : x = a \wedge p(y)$
   For $x = 0_N$, this gives $true \leftrightarrow \exists y : 0_N = 0_N \wedge p(y)$ which is true for a $y \in Z$
   for $x \neq 0_N$, this gives $false \leftrightarrow \exists y : false \wedge p(y)$ which is true.

   The pre-interpretation J maps each distinct term to a distinct domain element and $=$ is interpreted as the identity, hence I is also a model of the free equality axioms.

6. For the same program: Show that with $D = \{0\}$, $J(a) = 0$, $J(f(0)) = 0$, $I(p/1) = I(q/1) = \{\langle 0\rangle\}$, $I(=)$ the identity relation, I is not a model of the completion.

   Consider $a = f(a) \rightarrow false$ (an instance of equality axiom 7). It reduces to $0 = 0 \rightarrow false$ which is false.

7. For the above program, is $\neg q(a)$ is provable by the closed world assumption? by negation as finite failure? from the completion of the program?

$\neg q(a)$ is provable by CWA ($M_P = \emptyset$ hence $q(a) \in B_P \setminus M_P$); it is not provable by finite failure (fair SLD-tree is infinite) and not from the completion (as shown above, the completion has a model in which $q(a)$ is true).

8. Consider the following program P:
   parent(a,b) $\leftarrow$.
   parent(b,c) $\leftarrow$.
   anc(X,Y) $\leftarrow$parent(X,Y).
   anc(X,Y) $\leftarrow$anc(X,Z), anc(Z,Y).

   (a) i. Give the completion $Comp(P)$ of P.
   The completion consists if the Free Equality Axioms and the following $IFF(P)$ formulas:
   $\forall X,Y\ parent(X,Y) \leftrightarrow (X = a \wedge Y = b) \vee (X = b \wedge Y = c)$
   $\forall X,Y\ anc(X,Y) \leftrightarrow parent(X,Y) \vee \exists Z(anc(X,Z) \wedge anc(Z,Y))$

   ii. Give 2 different Herbrand models of $Comp(P)$.
   ("parent" is abbreviated as "p")
   $T_P \uparrow 0 = \emptyset$
   $T_P \uparrow 1 = \{p(a,b), p(b,c)\}$
   $T_P \uparrow 2 = \{p(a,b), p(b,c), anc(a,b), anc(b,c)\}$
   $T_P \uparrow 3 = \{p(a,b), p(b,c), anc(a,b), anc(b,c), anc(a,c)\}$
   $T_P \uparrow 4 = \{p(a,b), p(b,c), anc(a,b), anc(b,c), anc(a,c)\}$ (least fix point hence $= T_P \uparrow \omega$
   Hence, $M_1 = \{p(a,b), p(b,c), anc(a,b), anc(b,c), anc(a,c)\}$ is a Herbrand model.
   $T_P \downarrow 0 = B_P$
   $T_P \downarrow 1 = \{p(a,b), p(b,c)\} \cup \{anc(X,Y) | X,Y \in \{a,b,c\}\}$
   $T_P \downarrow 2 = \{p(a,b), p(b,c)\} \cup \{anc(X,Y) | X,Y \in \{a,b,c\}\}$ (greatest fix point and $T_P \downarrow \omega$) hence
   $M_2 = \{p(a,b), p(b,c)\} \cup \{anc(X,Y) | X,Y \in \{a,b,c\}\}$ is a Herbrand model.

   iii. Give a Herbrand interpretation which is a model of P but not of $Comp(P)$.
   $M_3 = \{p(a,b), p(b,c), anc(a,b), anc(b,c), anc(a,c), p(a,c)\}$
   is a model of the original program but not of $Comp(P)$.

   iv. Which of the following are true?
   $Comp(P) \models anc(a,c)$?
   Yes, it is true in M_1, the least Herbrand model of P hence $P \models anc(a,c)$ and by monotonicity of FOL, $Comp(P) \models anc(a,c)$
   $Comp(P) \models anc(c,a)$?
   No, it is not true in all models (e.g. in M_1, the least fixpoint).
   $Comp(P) \models \neg anc(c,a)$?
   No, it is not true in all models (e.g. in M_2, the greatest fixpoint).

   (b) Is $\neg anc(c,a)$ provable by:
   i. CWA?
   $anc(c,a) \notin T_P \uparrow \omega = M_1$, hence provable by CWA.
   ii. NAF?
   No, any SLD-derivation using second clause for solving $anc/2$ atoms is infinite.
   iii. Completion?
   No, because $anc(c,a)$ is true in $gfp(T_P) = M_2$, a model of the completion.

iv. Prolog?
No, infinite SLD-tree for Prolog's computation rule.

9. Consider the following program P:
parent(a,b) $\leftarrow$.
parent(b,c) $\leftarrow$.
anc(X,Y) $\leftarrow$parent(X,Y).
anc(X,Y) $\leftarrow$anc(X,Z), parent(Z,Y).

(a) Is $\neg anc(c, a)$ provable by:

i. CWA?
$T_P \uparrow 0 = \emptyset$
$T_P \uparrow 1 = \{p(a, b), p(b, c)\}$
$T_P \uparrow 2 = \{p(a, b), p(b, c), anc(a, b), anc(b, c)\}$
$T_P \uparrow 3 = \{p(a, b), p(b, c), anc(a, b), anc(b, c), anc(a, c)\}$
$T_P \uparrow 4 = \{p(a, b), p(b, c), anc(a, b), anc(b, c), anc(a, c)\}$ (fix point)
$anc(c, a) \notin T_P \uparrow 4 = T_P \uparrow \omega$ hence provable.

ii. NAF?
Yes, any fair SLD-tree fails finitely:
$\leftarrow anc(c, a)$
$\leftarrow anc(c, Z), \underline{p(Z, a)}$ $\quad\quad$ $\leftarrow p(c, a)$
failure $\quad\quad$ failure

iii. Completion - Yes
$T_P \downarrow 0 = B_P$
$T_P \downarrow 1 = \{p(a, b), p(b, c)\} \cup \{anc(X, Y)|X, Y \in \{a, b, c\}\}$
$T_P \downarrow 2 = \{p(a, b), p(b, c), anc(a, b), anc(b, b), anc(c, b), anc(a, c), anc(b, c), anc(c, c)\}$
$T_P \downarrow 3 = \{p(a, b), p(b, c), anc(a, b), anc(a, c), anc(b, c), anc(c, c)\}$
$T_P \downarrow 4 = \{p(a, b), p(b, c), anc(a, b), anc(a, c), anc(b, c)\}$
$T_P \downarrow 5 = \{p(a, b), p(b, c), anc(a, b), anc(a, c), anc(b, c)\}$ fixpoint hence $= T_P \downarrow \omega$
hence provable because $anc(c, a)$ is not in $T_P \downarrow \omega$

iv. Prolog?
No, infinite SLD-tree under Prolog's computation rule (one derivation where always $anc/2$ atom is selected).

(b) For $P1 = P \cup \{parent(a, a)\}$, is $\neg anc(c, a)$ provable by:

i. CWA?
$T_P \uparrow 0 = \emptyset$
$T_P \uparrow 1 = \{p(a, a), p(a, b), p(b, c)\}$
$T_P \uparrow 2 = \{p(a, a), p(a, b), p(b, c), anc(a, a), anc(a, b), anc(b, c)\}$
$T_P \uparrow 3 = \{p(a, a), p(a, b), p(b, c), anc(a, a), anc(a, b), anc(b, c), anc(a, c)\}$
$T_P \uparrow 4 = \{p(a, a), p(a, b), p(b, c), anc(a, a), anc(a, b), anc(b, c), anc(a, c)\}$ (fix point hence $= T_P \uparrow \omega$)
Provable because $anc(c, a) \notin T_P \uparrow \omega$.

ii. NAF? No because there is a fair infinite derivation:
$\leftarrow anc(c, a)$
$\leftarrow anc(c, Z), \underline{parent(Z, a)}$
$\leftarrow anc(c, a)$
...

iii. Completion?
$T_P \downarrow 0 = B_P$

$T_P \downarrow 1 = \{p(a,a), p(a,b), p(b,c)\} \cup \{anc(X,Y)|X,Y \in \{a,b,c\}\}$
$T_P \downarrow 2 = \{p(a,a), p(a,b), p(b,c)\} \cup \{anc(X,Y)|X,Y \in \{a,b,c\}\}$ (fix point hence $= T_P \downarrow \omega$)
and $anc(c,a) \in T_P \downarrow \omega$ hence not provable.

iv. Prolog?
No, infinite SLD-tree under Prolog computation rule.

10. Consider the program P:
p(f(X))←p(X).
q(a)←p(X).
Is $\neg q(a)$ true according to:

(a) CWA?
Yes because $q(a) \notin T_P \uparrow \omega = \emptyset$.

(b) NAF?
No because infinite fair SLD-derivation: $\leftarrow q(a)$, $\leftarrow p(x_0)$, $\leftarrow p(x_1)$, ...

(c) Completion?
No, because $q(a) \in T_P \downarrow \omega = \{q(a)\}$. ($\{q(a)\}$ is not a fixpoint)

(d) Prolog?
No, infinite SLD-tree under Prolog computation rule.

11. Consider the following program P:
r(X) ←p(X), q(X).
p(a) ←.
p(X) ←p(f(X)).
q(b) ←.

(a) Give the completion of P.
The completion consists of the Free Equality Axioms and of:
$\forall X\ r(X) \leftrightarrow p(X) \wedge q(X)$
$\forall X\ p(X) \leftrightarrow X = a \vee p(f(X))$
$\forall X\ q(X) \leftrightarrow X = b$

(b) Which of $\neg r(a)$, $\neg r(b)$, and $\neg r(c)$ are provable by (i) CWA, (ii) NAF, (iii) Completion, (iv) Prolog?
$T_P \uparrow 0 = \emptyset$
$T_P \uparrow 1 = \{p(a), q(b)\}$
$T_P \uparrow 2 = \{p(a), q(b)\}$ ($= T_P \uparrow \omega$)
$T_P \downarrow 0 = B_P$
$T_P \downarrow 1 = \{r(a), r(f(a)), \ldots\} \cup \{r(b), r(f(b)), \ldots\} \cup \{p(a), p(f(a)), \ldots\} \cup \{p(b), p(f(b)), \ldots\} \cup \{q(b)\}$
$T_P \downarrow 2 = \{r(b), q(b)\} \cup \{p(a), p(f(a)), \ldots\} \cup \{p(b), p(f(b)), \ldots\}$
$T_P \downarrow 3 = \{r(b), q(b)\} \cup \{p(a), p(f(a)), \ldots\} \cup \{p(b), p(f(b)), \ldots\}$ ($= T_P \downarrow \omega$)

i. $\neg r(a)$:
Provable by CWA because $r(a) \notin T_P \uparrow \omega$
Provable by NAF and Completion because fair SLD-tree fails finitely and $r(a) \notin T_P \downarrow \omega$
Not provable by Prolog because infinite SLD-tree under Prolog computation rule.

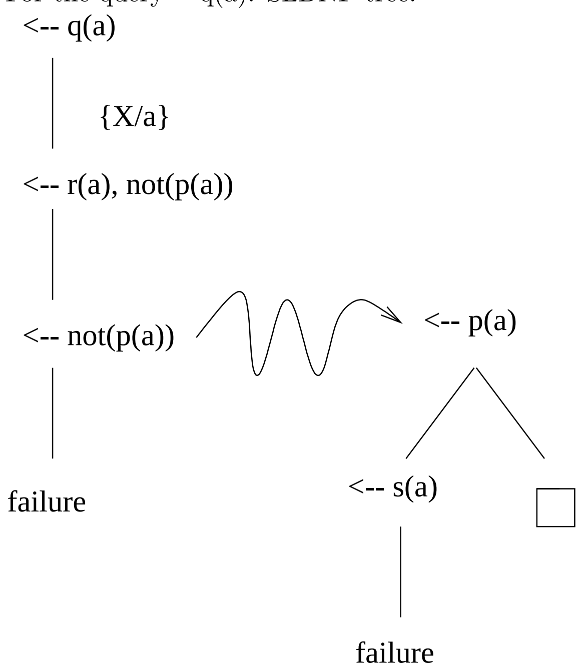
ii. $\neg r(b)$:
Provable by CWA because $r(b) \notin T_P \uparrow \omega$
Not provable by NAF and Completion because fair SLD-tree is infinite and $r(b) \in T_P \downarrow \omega$
Not provable by Prolog, also infinite tree.

iii. $\neg r(c)$:
Provable by CWA because $r(c) \notin T_P \uparrow \omega$
Provable by NAF and Completion because fair SLD-tree fails finitely and $r(c) \notin T_P \downarrow \omega$
Not provable by Prolog, infinite SLD-tree.

SLDNF

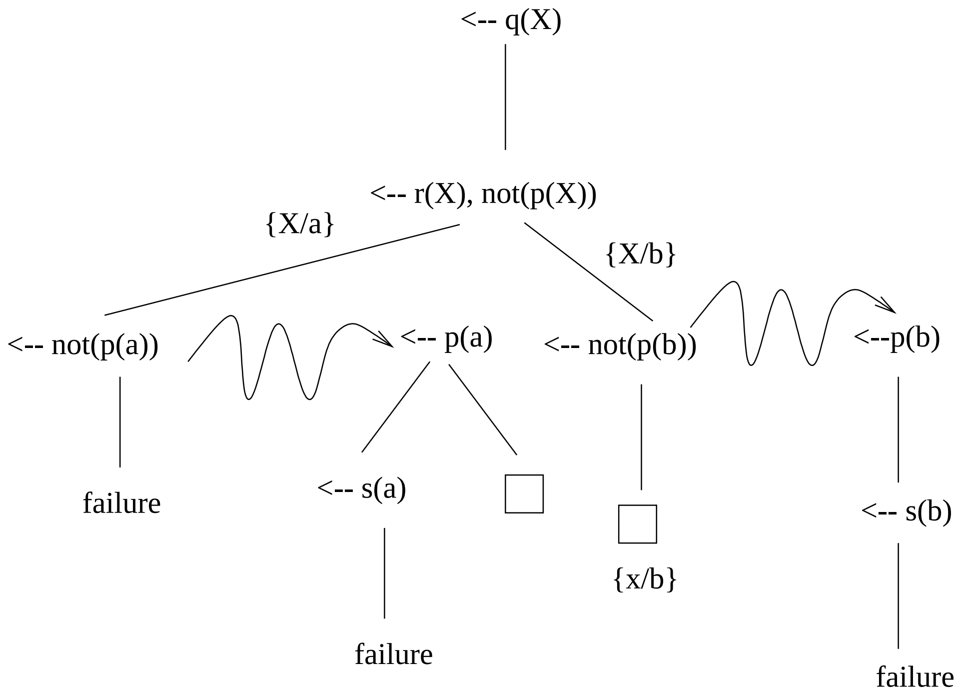
1. Consider P:
q(X) ←r(X), not p(X).
p(X) ←s(X).
p(a).
r(a).
r(b).
s(c).
Construct SLDNF-trees for the queries ←q(a). and ←q(X).

   (a) For the query ←q(a): SLDNF-tree:



   The query fails.

   (b) For the query ←q(X): SLDNF-tree:



The query succeeds with cas {X/b}.

2. Add the clause
p(b) ←p(b).
to the previous program. What changes to the SLDNF-tree of the query ←q(X)?

There is now an extra infinite branch for $p(b)$, hence $p(b)$ does not fail finitely and so, $q(X)$ does not succeed (neither does it fail finitely).

The Fitting operator Φ_P

1. Consider P:
q(X) ←r(X), not p(X).
p(X) ←s(X).
p(b) ←p(b).
p(a).
r(a).
r(b).
s(c).
Compute the least fixpoint of the Fitting operator.

$\Phi \uparrow 1 = (\{p(a), r(a), r(b), s(c)\}, \{r(c), s(a), s(b)\})$
$\Phi \uparrow 2 = (\{\ldots, p(c)\}, \{\ldots, q(c), q(a)\})$
Is fixpoint, $p(b)$ and $q(b)$ remain undefined, SLDNF does not terminate for ←`p(b)` and ←`q(b)`.

2. Consider P:
shaves(b,X) ←citizen(X), not shaves(X,X).
citizen(a).

citizen(b).
Compute the least fixpoint of the Fitting operator.

$\Phi \uparrow 1 = (\{citizen(a), citizen(b)\}, \{shaves(a,a), shaves(a,b)\})$
$\Phi \uparrow 2 = (\{\ldots, shaves(b,a)\}, \{\ldots\})\}$
Is fixpoint, $shaves(b,b)$ remains undefined, SLDNF does not terminate for $\leftarrow$`shaves(b,b)`.

3. Consider P:
r(a).
r(b) ←not q.
q ←p(X).
p(f(X)) ←p(X).
Compute the least fixpoint of the Fitting operator.

$\Phi \uparrow 1 = (\{r(a)\}, \{p(a), p(b), r(f^n(a))(n>0), r(f^n(b))(n>0)\})$
$\Phi \uparrow 2 = (\{\ldots\}, \{\ldots, p(f(a)), p(f(b))\})$
$\Phi \uparrow n+1 (n>1) = (\{\ldots\}, \{\ldots, p(f^n(a)), p(f^n(b))\})(n>0)$

$\Phi \uparrow \omega = (\{r(a)\}, \{p(a), p(b), r(f^n(a)), r(f^n(b)), p(f^n(a)), p(f^n(b))\})$ (now all p-atoms are false)
$\Phi \uparrow \omega+1 = (\{\ldots,\}, \{\ldots, q\})$ discontinuity
$\Phi \uparrow \omega+1 = (\{\ldots, r(b)\}, \{\ldots\})$ fixpoint

Stratification, Perfect Models

1. Consider the program P:
p(X) ←r(X), not q(X).
r(X) ←s(X).
q(X) ←s(X), q(X).
r(a).
s(b).

(a) Give the dependency graph.
$D_P = \{(p,r)^+, (p,q)^-, (r,s)^+, (q,s)^+, (q,q)^+\}$.

(b) Give stratifications:
$S_1 : P_1 = \{s\}, P_2 = \{r\}, P_3 = \{q\}, P_4 = \{p\}$
$S_2 : P_1 = \{s\}, P_2 = \{q\}, P_3 = \{r\}, P_4 = \{p\}$
$S_3 : P_1 = \{s\}, P_2 = \{q,r\}, P_3 = \{p\}$
$S_4 : P_1 = \{s\}, P_2 = \{q\}, P_3 = \{r,p\}$
$S_5 : P_1 = \{s,r\}, P_2 = \{q\}, P_3 = \{p\}$
$S_6 : P_1 = \{s,q\}, P_2 = \{r\}, P_3 = \{p\}$
$S_7 : P_1 = \{s,q\}, P_2 = \{r,p\}$
$S_8 : P_1 = \{s,q,r\}, P_2 = \{p\}$

(c) Compute M_P the perfect model.
Using the stratification $S_8 : P_1 = \{s,q,r\}, P_2 = \{p\}$.
$T_{P_1} \Uparrow 1(\emptyset) = \{r(a), s(b)\}$
$T_{P_1} \Uparrow 2(\emptyset) = \{r(a), r(b), s(b)\}$
$T_{P_1} \Uparrow 3(\emptyset) = \{r(a), r(b), s(b)\}$ (fix point)
$M_1 = \{r(a), r(b), s(b)\}$

$T_{P_2} \Uparrow 1(M_1) = \{r(a), r(b), s(b), p(a), p(b)\}$
$T_{P_2} \Uparrow 2(M_1) = \{r(a), r(b), s(b), p(a), p(b)\}$ (fix point)
$M_P = M_2 = \{r(a), r(b), s(b), p(a), p(b)\}$

(d) Is M_P a fix-point of T_P?
$T_P(M_P) = \{r(a), r(b), s(b), p(a), p(b)\} = M_P$: yes

(e) Is M_P equal to $T_P \uparrow \omega$?
$T_P \uparrow 1 = \{r(a), s(b)\}$
$T_P \uparrow 2 = \{r(a), r(b), s(b), p(a)\}$
$T_P \uparrow 3 = \{r(a), r(b), s(b), p(a), p(b)\}$
$T_P \uparrow 4 = \{r(a), r(b), s(b), p(a), p(b)\}$ (fix point)
$T_P \uparrow \omega = M_P$

2. Program
p(X) ←not q(X).
p(X) ←p(X).
r(X) ←s(X).
q(X) ←s(X), r(X).
r(a).
s(b).

(a) Give the dependency graph.
$D_P = \{(p, q)^-, (p, p)^+, (r, s)^+, (q, s)^+, (q, r)^+\}$.

(b) All possible stratifications:
$S_1 : P_1 = \{s\}, P_2 = \{r\}, P_3 = \{q\}, P_4 = \{p\}$
$S_2 : P_1 = \{s\}, P_2 = \{r, q\}, P_3 = \{p\}$
$S_3 : P_1 = \{s, r\}, P_2 = \{q\}, P_3 = \{p\}$
$S_4 : P_1 = \{s, r, q\}, P_2 = \{p\}$

(c) Compute M_P the perfect model.
Using the stratification $S_4 : P_1 = \{s, r, q\}, P_2 = \{p\}$.
$T_{P_1} \Uparrow 1(\emptyset) = \{r(a), s(b)\}$
$T_{P_1} \Uparrow 2(\emptyset) = \{r(a), r(b), s(b)\}$
$T_{P_1} \Uparrow 3(\emptyset) = \{r(a), r(b), s(b), q(b)\}$
$T_{P_1} \Uparrow 4(\emptyset) = \{r(a), r(b), s(b), q(b)\}$ (fix point)
$M_1 = \{r(a), r(b), s(b), q(b)\}$

$T_{P_2} \Uparrow 1(M_1) = \{r(a), r(b), s(b), q(b), p(a)\}$
$T_{P_2} \Uparrow 2(M_1) = \{r(a), r(b), s(b), q(b), p(a)\}$ (fix point)
$M_P = M_2 = \{r(a), r(b), s(b), q(b), p(a)\}$

(d) Is M_P a fix-point of T_P?
$T_P(M_P) = \{r(a), r(b), s(b), q(b), p(a)\} = M_P$: yes

(e) Is M_P equal to $T_P \uparrow \omega$?
$T_P \uparrow 1 = \{r(a), s(b), p(a), p(b)\}$
$T_P \uparrow 2 = \{r(a), r(b), s(b), p(a), p(b)\}$
$T_P \uparrow 3 = \{r(a), r(b), s(b), q(b), p(a), p(b)\}$
$T_P \uparrow 4 = \{r(a), r(b), s(b), q(b), p(a), p(b)\}$ (fix point)
$T_P \uparrow \omega \neq M_P$

Well-founded semantics

1. Consider P:
   q(X) ←r(X), not p(X).
   p(X) ←s(X).
   p(b) ←p(b).
   p(a).
   r(a).
   r(b).
   s(c).
   Compute the well-founded model.

   First iteration:
   The "conservative" program:
   p(X) ←s(X). ($X \in \{a, b, c\}$)
   p(b) ←p(b).
   p(a).
   r(a).
   r(b).
   s(c).
   The true atoms: its least fixpoint: $\{s(c), r(b), r(a), p(a), p(c)\}$
   The "liberal" program:
   q(X) ←r(X). ($X \in \{a, b, c\}$)
   p(X) ←s(X). ($X \in \{a, b, c\}$)
   p(b) ←p(b).
   p(a).
   r(a).
   r(b).
   s(c).
   Its fixpoint: $\{s(c), r(b), r(a), p(a), p(c), q(a), q(b)\}$
   The false atoms: its complement: $\{s(a), s(b), r(c), p(b), q(c)\}$

   $\mathcal{PI} \uparrow 1 = \mathcal{GL} \uparrow 1 = (\{s(c), r(b), r(a), p(a), p(c)\}, \{s(a), s(b), r(c), p(b), q(c)\})$

   Second iteration:
   The "conservative" program = "liberal" program:
   q(X) ←r(X). ($X \in \{b\}$)
   p(X) ←s(X). ($X \in \{a, b, c\}$)
   p(b) ←p(b).
   p(a).
   r(a).
   r(b).
   s(c).
   Its fixpoint: $\{s(c), r(b), r(a), p(a), p(c), q(b)\}$
   Its complement: $\{s(a), s(b), r(c), p(b), q(a), q(c)\}$

   $\mathcal{PI} \uparrow 2 = \mathcal{GL} \uparrow 2 = (\{s(c), r(b), r(a), p(a), p(c), q(b)\}, \{s(a), s(b), r(c), p(b), q(a), q(c)\})$

   It is a total model.
   Notice the difference with the fixpoint of the Fitting operator Φ

2. Consider P:
   shaves(b,X) ←citizen(X), not shaves(X,X).

citizen(a).
citizen(b).
Compute the well-founded model

First iteration:
The "conservative" program:
citizen(a).
citizen(b).
Its fixpoint: $\{citizen(a), citizen(b)\}$

The "liberal" program:
shaves(b,X) ←citizen(X). ($X \in \{a, b\}$)
citizen(a).
citizen(b).
Its fixpoint: $\{citizen(a), citizen(b), shaves(b, b), shaves(b, a)\}$
Its complement: $\{shaves(a, a), shaves(a, b)\}$

$\mathcal{PI} \uparrow 1 = \mathcal{GL} \uparrow 1 = (\{citizen(a), citizen(b)\}, \{shaves(a, a), shaves(a, b)\})$

Second iteration:
The "conservative" program:
shaves(b,X) ←citizen(X). ($X \in \{a\}$)
citizen(a).
citizen(b).
Its fixpoint: $\{citizen(a), citizen(b), shaves(b, a)\}$

The "liberal" program:
shaves(b,X) ←citizen(X). ($X \in \{a, b\}$)
citizen(a).
citizen(b).
Its fixpoint: $\{citizen(a), citizen(b), shaves(b, b), shaves(b, a)\}$
Its complement: $\{shaves(a, a), shaves(a, b)\}$

$\mathcal{PI} \uparrow 2 = \mathcal{GL} \uparrow 2 = (\{citizen(a), citizen(b), shaves(b, a)\}, \{shaves(a, a), shaves(a, b)\})$

The conservative and liberal program remain the same, hence fixpoint. The model is NOT total, $shaves(b, b)$ remains undefined.

Incomplete Knowledge ID-Logic

1. Consider the following program P:

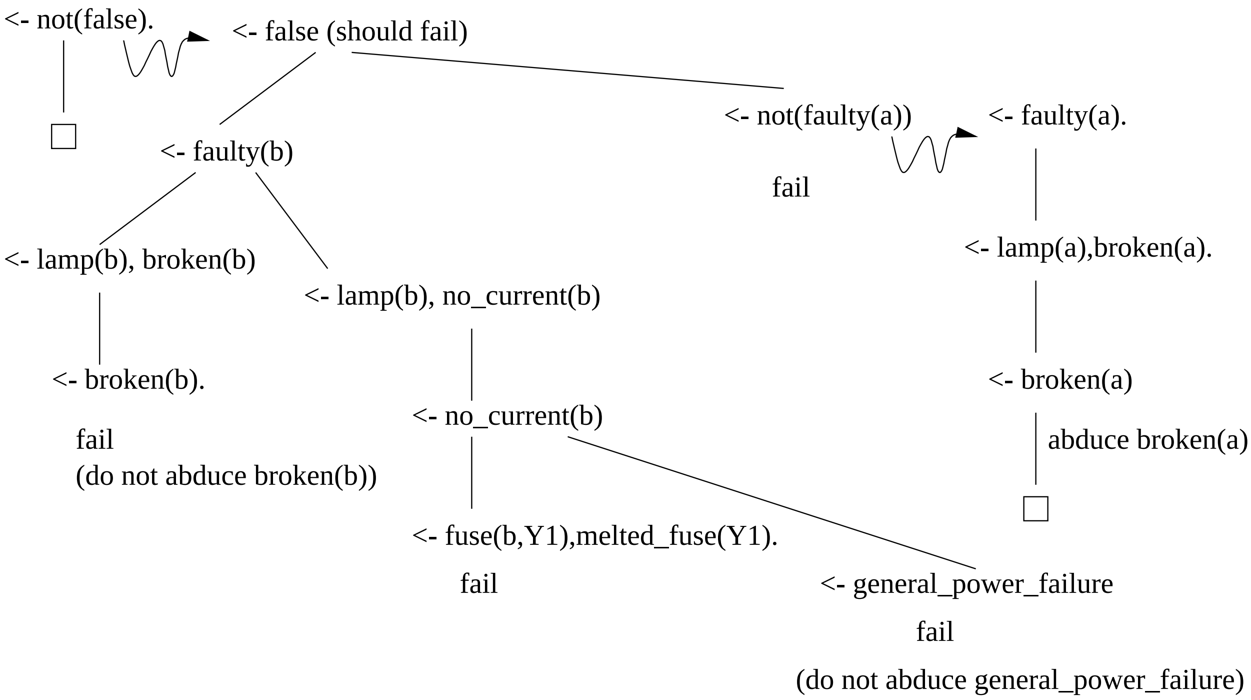
```
open predicates broken/1, melted_fuse/1, general_power_failure/0.
%definitions
faulty(X) :- lamp(X), broken(X).
faulty(X) :- lamp(X), no_current(X).
no_current(X) :- fuse(X,Y), melted_fuse(Y).
no_current(X) :- general_power_failure.
lamp(a).
lamp(b).
%constraints
false <- faulty(b).
faulty(a) <- true.
```

Using SLDNFA, construct an abductive solution.

Add the clauses:

```
false :- faulty(b).
false :- not(faulty(a)).
```

and execute the query $\leftarrow not(false)$):



Abductive solution: `broken(a)`.

Answer Set Programming

1. $\{posit(1,2), posit(2,4), posit(3,1), posit(4,2)\}$ is a solution for the 4-queens problem. For the program on p. 91, construct a stable set that contains this solution (and verify that it indeed is a stable set).

   Note that the definitions of $p/1, row/1$, and $col/1$ do not depend on the "unfounded" predicates $posit/2$ and $neg\_posit/2$ hence every stable model must include the least Herbrand model of these definitions, i.e. $S_1 = \{p(1), p(2), p(3), p(4),\ row(1), row(2), row(3), row(4), col(1), col(2), col(3), col(4)\}$

   The stable set must include the given elements $S_2 = \{posit(1,2), posit(2,4), posit(3,1), posit(4,3)\}$.

   It follows from the constraints that no other $posit/2$ atoms can be part from a stable set, hence, from rule $neg\_posit(R,C) \leftarrow not\ posit(R,C), row(R), col(C)$. it follows that ($neg\_posit(R,C)$ has to be true whenever the body is true), $S_3 = \{neg\_posit(1,1), neg\_posit(1,3), neg\_posit(1,4), neg\_posit(2,1), neg\_posit(2,2), neg\_posit(2,3), neg\_posit(3,2), neg\_posit(3,3), neg\_posit(3,4), neg\_posit(4,1), neg\_posit(4,2), neg\_posit(4,4)\}$ must be in the stable set.

   Having chosen a model for $posit/2$, now also $row\_has\_queen/1$ is defined, its least model is $S_4 = \{row\_has\_queen(1), row\_has\_queen(2), row\_has\_queen(3), row\_has\_queen(4)\}$.

$S_1 \cup S_2 \cup S_3 \cup S_4$ is a stable set as it is the least model of the reduct and it satisfies all the constraints. Note that the reduct contains the (grounding of) the definite clauses as well as the clauses

```
posit(1,2) <- row(1), col(2).
..
posit(4,3) <- row(4), col(3).
neg_posit(1,1) <- row(1), col(1).
...
neg_posit(4,4) <- row(4), col(4).
```

2. $\{in(1,2), in(2,3), in(3,1)\}$ is a Hamiltonian cycle for the graph of the program on p. 92. Construct a stable set that contains this solution (verify that it indeed is a stable set).

   The least model of the definite part contains:
   $S_1 = \{node(1), node(2), node(3),\ edge(1,2), edge(2,3), edge(2,1), edge(3,1)\}$.

   The unfounded atoms are $in(1,2), in(2,3), in(2,1), in(3,1)$, $out(1,2), out(2,3), out(2,1)$, and $out(3,1)$.

   With $S_2 = \{in(1,2), in(2,3), in(3,1)\}$ only the choice between $out(2,1)$ and $in(2,1)$ remains to be made. The latter violates a constraint, hence $S_3 = \{out(2,1)\}$.

   Now the least model of the reachable predicate can be computed:
   $S_4 = \{reachabel(2), reachabel(3), reachabel(1)\}$.

   $S_1 \cup S_2 \cup S_3 \cup S_4$ is a stable set as it is the least model of the reduct and it satisfies all the constraints. Note that the reduct contains the (grounding of) the definite clauses as well as the clauses

```
in(1,2) :- edge(1,2).
in(2,3) :- edge(2,3).
in(3,1) :- edge(3,1).
out(2,1) :- edge(2,1).
```