

Some thoughts on a fixed point operator for rSM and rWFS

Luís Rodrigues Soares

November 29th, 2005

1 Intro

My first attempts with the definition of a fixed point for rSM (and hence for rWFS) were very naive as all I tried to do was check for the intersections of interpretations and the application of Γ operator on those interpretations, to see where did I get the models. I took me only a week to realize this wasn't a good approach because it only worked on small programs (although it *did* work for small programs...).

So the next approach, and the best one so far, was to try to look for the *essence* of a RAA so that the operator (which I named Γ^r) could preprocess the logic program on some way, and after that the Γ operator could be used. If this resulted well, it might be possible to give the same use to Γ^r as it is given to Γ for the SMs and on the WFS.

I now start by showing the reasoning I followed, and then present some examples, good ones (that work) and bad ones (that *still* don't work). I end up with some ideas for the bad examples.

2 Searching for the essence of a RAA

As I stated before, my idea was to do some sort of preprocessing to the logic program so that when I applied the Γ operator it would not remove the rules belonging to RAAs. So for example, for program P_1 , what would be P_{r1} ?

$$P_1 = \{a \leftarrow \sim a\}$$

$$P_{r1} = \left\{ \begin{array}{l} a \leftarrow \sim a \\ a \leftarrow \end{array} \right\}$$

If I now applied $\Gamma_{P_{r1}}(\{a\})$ I would get a , the fixed point I initially intended to get.

I also used this idea on the following:

$$P_2 = \left\{ \begin{array}{l} a \leftarrow \sim b \\ b \leftarrow \sim c \\ c \leftarrow \sim a \end{array} \right\}$$

which has three models, $M_1 = \{a, b\}$, $M_2 = \{b, c\}$ and $M_3 = \{c, a\}$. So if we were trying to test for M_1 , it would suffice to have the following program before applying $\Gamma_{P_{r2}}(\{a, b\})$:

$$P_{r2} = \left\{ \begin{array}{l} a \leftarrow \sim b \\ b \leftarrow \sim c \\ c \leftarrow \sim a \\ a \leftarrow \end{array} \right\}$$

And equivalently to

$$P_3 = \left\{ \begin{array}{l} a \leftarrow x \\ x \leftarrow \sim a \end{array} \right\}$$

and

$$P_{r3} = \left\{ \begin{array}{l} a \leftarrow x \\ x \leftarrow \sim a \\ a \leftarrow \end{array} \right\}$$

where $\Gamma_{P_{r3}}(a)$ I would give me a , the fixed point I initially intended to get.

These observations eventually got me to the following definition:

Definition 1 (Γ^r Operator) *Let P be a NLP and I an interpretation.*

P_r is the logic program obtained by performing the following operation on P :

- *For each literal $X \in I$, if there is a rule $H \leftarrow A_1, \dots, A_m, \sim B_1, \dots, \sim B_n$ such that $X \in \{B_1, \dots, B_n\}$ and $H \notin I \setminus \{X\}$, add to P the rule $X \leftarrow A_1, \dots, A_m, \sim B_1, \dots, \sim B_k$, with $\{B_1, \dots, B_k\} = \{B_1, \dots, B_n\} \setminus \{X\}$.*
- *No other rule is to be added to P_r .*

On program P_r apply $\Gamma(I)$ operator.

Definition 2 (Γ^r -based rSM definition) *Given any 2-valued interpretation I and a NLP P , I is a rSM of P if and only if $I = \Gamma^r(I)$*

Definition 3 (Γ^r -based rWFS definition) *Given a NLP P , the rWFM of P is the tuple $\langle T, TU \rangle$ where T is the least fixed point of Γ^{r^2} , starting with the empty interpretation, and TU is the next iteration of Γ^{r^2} , after reaching a fixed point.*

The intuition here is to add certain literals present in the model to the program. Which ones should we add? Those that are negated in some rule in the program and the head of that rule does not belong to the rest of the model (i.e., the model except the literal we are considering). This means that only either heads equal to the negated literal, either heads not at all present in the model will be considered when adding literals to the program. We'll see in the following section how this behaves with other programs.

3 Good examples

I'll now show the several tested programs with the good results first. I'm not presenting all the steps to all programs as some of them take more time than others. But I believe the idea will be easily grasped.

3.1 Program 1

$$P_1 = \left\{ \begin{array}{l} a \leftarrow \sim b \\ b \leftarrow \sim c \\ c \leftarrow \sim a \end{array} \right\}$$

Starting with the rWFM, we have:

$$\begin{aligned} \Gamma^{r^2}(\{\}) &= \Gamma^r(\Gamma^r(\{\})) = \\ &= \Gamma^r(\Gamma_{P_{r1}}(\{\})) = \end{aligned}$$

$$\begin{aligned}
&= \Gamma^r(\{a, b, c\}) = \\
&= \Gamma_{P_{r_2}}(\{a, b, c\}) = \{\}
\end{aligned}$$

where $P_{r_1} = P \cup \{\}$ because as the initial interpretation is empty, no rule is added to P and $P_{r_2} = P \cup \{\}$ because all heads belong to the model except the tested literal. Therefore no literal is ever added. We reach a fixed point with the first iteration of Γ^{r^2} and the next iteration will give us the set $\{a, b, c\}$ which allows us to say that $rWFM(P_1) = \langle \{\}, \{a, b, c\} \rangle$, which is correct according to what we know from the rSMs of this program. Let's now test if $rSM_1 = (\{a, b\})$.

$$\Gamma^r(\{a, b\}) = \Gamma_{P_r}(\{a, b\}) = \{a, b\}$$

with $P_r = P \cup \{a \leftarrow\}$. This would give the equivalent results to $rSM_2 = (\{b, c\})$ and $rSM_3 = (\{c, a\})$.

3.2 Program 2

Program 2 is an enhanced version of program 1:

$$P_2 = \left\{ \begin{array}{l} a \leftarrow \sim b, x \\ b \leftarrow \sim c, x \\ c \leftarrow \sim a \\ x \leftarrow \sim x \\ y \leftarrow d, \sim y \end{array} \right\}$$

Let's start by calculating the $rWFM(P_2)$:

First iteration of Γ^{r^2} :

$$\begin{aligned}
\Gamma^{r^2}(\{\}) &= \Gamma^r(\Gamma^r(\{\})) = \\
&= \Gamma^r(\Gamma(\{\})) = \\
&= \Gamma^r(\{a, c, x\}) = \\
&= \Gamma_{P_{r_1}}(\{a, c, x\}) = \{a, x\}
\end{aligned}$$

with

$$P_{r1} = P \cup \left\{ \begin{array}{l} c \leftarrow y \\ x \leftarrow \end{array} \right\}$$

As we still haven't reached a fixed point, we try the next iteration:

$$\begin{aligned} \Gamma^{r2}(\{a, x\}) &= \Gamma^r(\Gamma^r(\{a, x\})) = \\ &= \Gamma^r(\Gamma_{P_{r2}}(\{a, x\})) = \\ &= \Gamma^r(\{a, x\}) = \{a, x\} \end{aligned}$$

$$P_{r2} = P \cup \left\{ \begin{array}{l} a \leftarrow \\ x \leftarrow \end{array} \right\}$$

As we reached a fixed point we can conclude $rWFM(P_2) = \langle \{a, x\}, \{a, x\} \rangle$

Let's now check if this is really a rSM of P_2 :

$$\Gamma^r(\{a, x\}) = \Gamma_{P_{r3}}(\{a, x\}) = \{a, x\}$$

with

$$P_{r3} = P \cup \left\{ \begin{array}{l} a \leftarrow \\ x \leftarrow \end{array} \right\}$$

Apparently all went ok. Let's now check to see if other interpretations are models of P_2 , as they shouldn't be.

Starting with $I = (\{a, c, x\})$, we get:

$$\Gamma^r(\{a, c, x\}) = \Gamma_{P_{r4}}(\{a, c, x\}) = \{a, x\}$$

with

$$P_{r4} = P \cup \left\{ \begin{array}{l} c \leftarrow y \\ x \leftarrow \end{array} \right\}$$

Now, for $I = (\{c, x\})$ we get:

$$\Gamma^r(\{c, x\}) = \Gamma_{P_{r5}}(\{a, c, x\}) = \{c, a, x\}$$

with

$$P_{r5} = P \cup \left\{ \begin{array}{l} c \leftarrow y \\ x \leftarrow \end{array} \right\}$$

3.3 Program 3

$$P_3 = \left\{ \begin{array}{l} a \leftarrow \sim b, \sim a \\ b \leftarrow \sim a, \sim b \end{array} \right\}$$

If we start by calculating the $rWFM(P_3)$ we get:

$$\begin{aligned} \Gamma^{r^2}(\{\}) &= \Gamma^r(\Gamma^r(\{\})) = \\ &= \Gamma^r(\Gamma(\{\})) = \\ &= \Gamma^r(\{a, b\}) = \\ &= \Gamma_{P_{r1}}(\{a, b\}) = \\ &= \{\} \end{aligned}$$

with

$$P_{r1} = P \cup \left\{ \begin{array}{l} a \leftarrow \sim b \\ b \leftarrow \sim a \end{array} \right\}$$

which results in $rWFM(P_2) = \langle \{\}, \{a, b\} \rangle$. This result suggests that there are two rSMs, $\{a\}$ and $\{b\}$. Let's test to see if this is the case:

$$\Gamma^r(\{a\}) = \Gamma_{P_{r2}}(\{a\}) = \{a\}$$

with

$$P_{r2} = P \cup \{a \leftarrow \sim b\}$$

and

$$\Gamma^r(\{b\}) = \Gamma_{P_{r3}}(\{b\}) = \{b\}$$

with

$$P_{r3} = P \cup \{b \leftarrow \sim a\}$$

3.4 Program 4

Another program I tested was the following:

$$P_4 = \left\{ \begin{array}{l} a \leftarrow \sim b \\ b \leftarrow \sim a \\ c \leftarrow a, d, \sim c \\ e \leftarrow f \\ g \leftarrow \sim h \\ i \leftarrow d, \sim i \\ d \leftarrow \sim d \end{array} \right\}$$

Starting with the $rWFM(P_4)$, we have the first iteration of Γ^{r^2}

$$\begin{aligned} \Gamma^{r^2}(\{\}) &= \Gamma^r(\Gamma^r(\{\})) = \\ &= \Gamma^r(\Gamma(\{\})) = \\ &= \Gamma^r(\{a, b, c, d, g, i\}) = \\ &= \Gamma_{P_{r1}}(\{a, b, c, d, g, i\}) = \\ &= \{g, d, i\} \end{aligned}$$

with

$$P_{r1} = P \cup \left\{ \begin{array}{l} c \leftarrow a, d \\ d \leftarrow \\ i \leftarrow d \end{array} \right\}$$

and the second one,

$$\begin{aligned} \Gamma^{r^2}(\{g, d, i\}) &= \Gamma^r(\Gamma^r(\{g, d, i\})) = \\ &= \Gamma^r(\Gamma_{P_{r2}}(\{g, d, i\})) = \\ &= \Gamma^r(\{a, b, c, d, g, i\}) = \\ &= \{g, d, i\} \end{aligned}$$

with

$$P_{r2} = P \cup \left\{ \begin{array}{l} i \leftarrow d \\ d \leftarrow \end{array} \right\}$$

As a result we get $rWFM(P_4) = \langle \{g, d, i\}, \{a, b, c, d, g, i\} \rangle$.

Let's now test for $rSM_1 = \{a, c, d, g, i\}$ and $rSM_2 = \{b, d, g, i\}$:

$$\Gamma^r(\{a, c, d, g, i\}) = \Gamma_{P_{r3}}(\{a, c, d, g, i\}) = \{a, c, d, g, i\}$$

with

$$P_{r3} = P \cup \left\{ \begin{array}{l} a \leftarrow \\ c \leftarrow a, d \\ d \leftarrow \\ i \leftarrow d \end{array} \right\}$$

and

$$\Gamma^r(\{b, d, g, i\}) = \Gamma_{P_{r4}}(\{b, d, g, i\}) = \{b, d, g, i\}$$

with

$$P_{r4} = P \cup \left\{ \begin{array}{l} b \leftarrow \\ d \leftarrow \\ i \leftarrow d \end{array} \right\}$$

3.5 Program 5

Finally, here's the last program I tried:

$$P_5 = \left\{ \begin{array}{l} p(X) \leftarrow p(s(X)) \\ p(X) \leftarrow \sim p(s(X)) \end{array} \right\}$$

We start with the first iteration Γ^{r^2} , assuming the grounding $X = 0$:

$$\begin{aligned} \Gamma^{r^2}(\{\}) &= \Gamma^r(\Gamma^r(\{\})) = \\ &= \Gamma^r(\Gamma(\{\})) = \\ &= \Gamma^r(\{p(0)\}) = \\ &= \Gamma_{P_{r1}}(\{p(0)\}) = \\ &= \{p(0)\} \end{aligned}$$

where

$$P_{r1} = P \cup \{\}$$

and then the second iteration,

$$\begin{aligned} \Gamma^{r^2}(\{p(0)\}) &= \Gamma^r(\Gamma^r(\{p(0)\})) = \\ &= \Gamma^r(p(0)) = \\ &= \{p(0)\} \end{aligned}$$

leaving us with $rWFM(P_5) = \langle \{p(X)\}, \{p(X)\} \rangle$.

If we try $rSM = \{p(0)\}$, as we already seen from the $rWFM$ it will give a fixed point on $p(0)$. Just to be sure, let's try $rSM = \{p(s(0))\}$ and $rSM = \{p(s(0)), p(0)\}$:

$$\Gamma^r(\{p(s(0))\}) = \Gamma_{P_{r2}}(\{p(s(0))\}) = \{p(s(0)), p(0)\}$$

with

$$P_{r2} = P \cup \{p(s(0))\}$$

and

$$\Gamma^r(\{p(s(0)), p(0)\}) = \Gamma_{P_{r3}}(\{p(s(0)), p(0)\}) = \{\}$$

with

$$P_{r3} = P \cup \{\}$$

As no fixed point is reached those are not rSMs of P_5 . We'll see in the next section the problems encountered so far.

4 Bad example and some ideas

Let P be the following program:

$$P = \left\{ \begin{array}{l} a \leftarrow x \\ x \leftarrow \sim a \end{array} \right\}$$

If we try $rSM_P = \{a\}$ we get:

$$\Gamma^r(\{a\}) = \Gamma_{P_r}(\{a\}) = \{a\}$$

with

$$P_r = P \cup \{a \leftarrow\}$$

This is easy to see because $x \leftarrow \sim a$ is the only rule where a appears negated and $x \notin \{a\} \setminus \{a\}$. So we add $a \leftarrow$ to P and apply Γ operator. Let's now see what happens if we try to calculate $rWFM(P)$:

$$\begin{aligned} \Gamma^{r^2}(\{\}) &= \Gamma^r(\Gamma^r(\{\})) = \\ &= \Gamma^r(\Gamma(\{\})) = \\ &= \Gamma^r(\{a, x\}) = \\ &= \Gamma_{P_{r1}}(\{a, x\}) = \\ &= \{\} \end{aligned}$$

where

$$P_{r1} = P \cup \{\}$$

which would give us $rWFM(P) = \langle \{\}, \{a, x\} \rangle$. This is obviously wrong and goes against the result we got when we calculated the $rSM(P)$. So where is the problem?

When we calculate $\Gamma^r(\{a, x\})$, we seek for rules where either a or x is negated. There is only one and it is $x \leftarrow \sim a$. Now, when $I = \{a, x\}$, and the literal we're testing is a , we cannot add $a \leftarrow$ to P because $x \in I \setminus \{a\}$. And that is why we get $\Gamma^r(\{a, x\}) = \{\}$.

So where exactly is the problem? Note that we would get the same results if P were

$$P = \left\{ \begin{array}{l} a \leftarrow y \\ y \leftarrow x \\ x \leftarrow \sim a \end{array} \right\}$$

In fact, whatever number of clauses we add between the first and the last one, if they are in the form $x \leftarrow y$, with x eventually being in the tail of some literal A and y being in the head of some literal $\sim A$, we would always get this problem.

$$P = \left\{ \begin{array}{l} a \leftarrow x \\ x \leftarrow \dots \\ \dots \leftarrow y \\ y \leftarrow \sim a \end{array} \right\}$$

So what was my idea so far? I think the definition is not incorrect. Actually, when we are searching for the head of the rule $x \leftarrow \sim a$, we shouldn't get x but a , because a is directly dependent on x ($a \leftarrow x$). So one possible solution would be to preprocess the program, and replace such rules. For example, for P , the equivalent program would be $a \leftarrow \sim a$, and its models (rSM and rWFM) would not even differ between the two forms. The problem is that we can have an infinite number of rules in that form between the first and the last rule. And how can we find those rules even if they appear in a finite number?

These are the problems so far...