Contradiction: when Avoidance equals Removal

José Júlio Alferes          Luís Moniz Pereira

AI Centre, Uninova and DCS U. Nova Lisboa
2825 Monte da Caparica
Portugal

Partially supported by ESPRIT BRA Compulog 2

WELP’93             March 1993
Recently several authors have illustrated the importance of including $\neg$ in logic programs.

Proposals for extended logic programs semantics has been advanced:

- Answer Sets semantics [GL90], shown to be an extension of the Stable Model semantics of normal logic programs.
- [KS90] proposes a similar extension.
- [Prz90] proposes an extension of WFS.
- [PA92] defines a semantics based on WFS and the coherence principle:

\[
\text{Whenever } \neg L \text{ holds not } L \text{ holds too}
\]
Contradiction

Once $\neg$ is introduced contradiction may arise (e.g. when $L$ and $\neg L$ both hold) and no meaning is assigned.

The program $\left\{ \begin{array}{c} a \\ \neg a \end{array} \right\}$ has no semantics.

Consider the statements:

*Birds not shown to be abnormal fly;*
*Tweety is a bird and does not fly;*
*Socrates is a man.*

\[
\begin{align*}
\text{fly}(X) & \leftarrow \text{bird}(X), \text{not abnormal}(X) \\
\neg \text{fly(tweety)} & \\
\text{bird(tweety)} & \\
\text{man(socrates)} & 
\end{align*}
\]

None of the above mentioned semantics assign a meaning to this program.
Dealing with contradiction

Two possible solutions:

- Define a sceptical semantics which *avoids* contradiction caused by acceptance of hypotheses.
  
  *Do not accept hypotheses leading to contradiction*

- Use a less sceptical semantics and provide a contradiction *removal* method.
  
  *If a program is contradictory then its revision is in order*
• Contradiction Avoidance
  – Scenaria Semantics
  – Control over acceptance of hypotheses
  – Avoidance

• Contradiction Removal
  – Revision of a program
  – Declarative revisions
  – A revision procedure

• Conclusions
The scenario semantics paradigm of logic programs has been recently expanded to encompass extended logic programs, including WFSX.

A scenario of $P$ is the first order theory $P \cup H$, where the default literals in $H$ are the hypotheses.

$\neg L$ is mandatory wrt $P \cup H$ iff

$$P \cup H \cup \{\neg L \leftarrow \neg L \mid L \in H\} \vdash \neg L.$$  

$\text{Mand}(H)$ denotes the set of all $\neg L$ mandatory wrt $P \cup H$.

$P \cup H$ is consistent iff for no $L$:

$$P \cup H \cup \text{Mand}(H) \vdash \{L, \neg L\}$$

A program is consistent if it has a consistent scenario.
not L is acceptable in $P \cup H$ iff every evidence to the contrary is defeated by it.

The contrary of not L is L.

$E$ is evidence to L iff $P \cup E \cup \text{Mand}(E) \vdash L$.

$P \cup H$ defeats $E$ iff: 
\[ \exists \neg L \in E \mid P \cup H \cup \text{Mand}(H) \vdash L \]

\[
P: \quad a \leftarrow \neg b, \neg c \quad \neg c \leftarrow \quad b \leftarrow \neg d
\]

In $P \cup \{\neg c, \neg d, \neg a\}$:

- not c is mandatory.
- not d is acceptable.
- not a is acceptable: 
  \[ P \cup \{\neg b, \neg c\} \vdash a \text{ but } P \cup H \nvdash b \]
Admissible Scenaria

$P \cup H$ is admissible iff all hypotheses are either acceptable or mandatory, and all mandatories are in $H$:

$$\text{Mand}(H) \subseteq H \subseteq \text{Mand}(H) \cup \text{Acc}(H)$$

The semantics of admissible scenaria is the most sceptical one for extended logic programs: it contains no hypotheses except for mandatory ones.

$H$ is the ideal sceptical semantics, $WFS0 = P \cup H$, if it is the greatest set satisfying the condition:

For each admissible scenario $P \cup K$,

$P \cup K \cup H$ is again admissible.
Complete Scenaria

$P : \ a \leftarrow \text{not } p \quad c \leftarrow \text{not } r$
\[\neg a \leftarrow \text{not } q\]

$WFS0 = P \cup \{\text{not } r\}$. Hence we conclude $c$, despite the inconsistency potentially caused by the other rules.

$P \cup H$ is complete iff it is admissible and contains every acceptable hypothesis:

$$H = \text{Mand}(H) \cup \text{Acc}(H)$$

The complete scenaria semantics is equivalent to $WFSX$. Total complete scenaria correspond to answer–sets.

$P : \ a \leftarrow \text{not } b, \text{not } c \quad \neg c \leftarrow \ b \leftarrow \text{not } d$

$P \cup \{\text{not } c, \text{not } d, \text{not } a, \text{not } \neg a, \text{not } \neg b, \text{not } \neg d\}$ is the only complete scenario. It is also total.
Contradiction and diagnosis

CSS assigns no meaning to some reasonable programs.
There is motivation to consider semantics more sceptical than WFS0.

\[ \neg \text{wobbly\_wheel} \leftarrow \text{not flat\_tyre}, \text{not br\_spokes} \]
\[ \text{flat\_tyre} \leftarrow \text{leaky\_valve} \]
\[ \text{flat\_tyre} \leftarrow \text{punctured\_tube} \]
\[ \neg \text{no\_light} \leftarrow \text{not faulty\_dynamo} \]
\[ \text{wobbly\_wheel} \]

WFS0 assigns the meaning:
\[ \{ \text{ww, not fd, } \neg \text{nl, not nl, not lv, not pt} \} \]
neither accepting \text{not ft} nor \text{not bs}.

One would like the semantics to delve deeper into the bycicle model and, again being sceptical, accept neither \text{not lv} nor \text{not pt} as well.
Optatives

WFS0 allows no control over which acceptable hypotheses are not accepted. Conceivably, any acceptable hypothesis may or may not be accepted, in some discretionary way.

In the wobbly wheel example we wish that only not bs, not lv, not pt, and not fd, may be optative.

Optatives might or might not be accepted even if acceptable. Non–optatives must be accepted if acceptable.

Opt is a set of hypotheses provided by the user along with the program.

In the paper we identify a special class of optatives, governed by:

Exactly the hypotheses not depending on any other are optatives.
Complete wrt $Opt$

$P \cup H$ is a complete scenario wrt $Opt$ iff it is consistent, and for each $not \ L$:

(i) $not \ L \in H \Rightarrow not \ L \in Acc_{Opt}(H) \lor not \ L \in Mand(H)$

(ii) $not \ L \in Mand(H) \Rightarrow not \ L \in H$

(iii) $not \ L \in Acc_{Opt}(H)$ and $not \ L \notin Opt \Rightarrow not \ L \in H$

In the wobbly wheel example if $Opt = \{not \ bs, not \ lv, not \ pt, not \ fd\}$

complete scenarios wrt $Opt$ are:

\[
\begin{align*}
\{not \ \neg ww\} & \quad \{not \ \neg ww, not \ lv\} \\
\{not \ \neg ww, not \ fd\} & \quad \{not \ \neg ww, not \ pt\} \\
\{not \ \neg ww, not \ bs\} & \quad \{not \ \neg ww, not \ fd, not \ bs\} \\
\{not \ \neg ww, not \ fd, not \ lv\} & \quad \{not \ \neg ww, not \ lv, not \ pt, not \ ft\} \\
\ldots & \quad \ldots
\end{align*}
\]

Some of these scenarios are over–sceptical.
In the first scenario above, in order to avoid contradiction, no optative is accepted.

A maximallity condition must be enforced.
Avoidance set of $P \cup H$: all optatives that though acceptable were not accepted, i.e.:

$$(Opt \cap Acc(H)) - H$$

The avoidance set of the first scenario in the wobbly wheel example is $\{\text{not lv, not pt, not fd}\}$ and of the second one is $\{\text{not lv, not pt}\}$.

\[
\begin{align*}
a & \leftarrow \text{not } b & c & \leftarrow \text{not } d \\
b & \leftarrow \text{not } a & \neg c
\end{align*}
\]

Complete scenaria wrt $Opt = \{\text{not } d\}$ are:

$P \cup \emptyset \quad P \cup \{\text{not } a\} \quad P \cup \{\text{not } b\}$

All have avoidance set $\{\text{not } d\}$.

In keeping with the sceptical vocation of WFSX: *Base scenaria* are minimal complete scenaria wrt $Opt$ for some avoidance set.
Well–founded wrt $Opt$

Quasi–complete scenaria wrt $Opt$ are base scenaria with minimal avoidance set.

In the wobbly wheel example, quasi–complete are:

$$P \cup \{\text{not } \neg ww, \text{not } fd, \text{not } bs, \text{not } lv\}$$
$$P \cup \{\text{not } \neg ww, \text{not } fd, \text{not } bs, \text{not } pt\}$$
$$P \cup \{\text{not } \neg ww, \text{not } fd, \text{not } lv, \text{not } pt, \text{not } ft\}$$

The well-founded model wrt $Opt$ ($WFS_{Opt}$), being sceptical, is the meet of all quasi–complete scenaria in the semi–lattice of base scenaria, so that its avoidance set is the union of their avoidance sets.

Above, $WFS_{Opt} = \{\text{not } \neg ww, \text{not } fd\}$.

One can conclude:

$$\{ww, \neg nl, \text{not } \neg ww, \text{not } fd\}$$

i.e. no other hypothesis can be assumed for certain; everything is sceptically assumed faulty except for $fd$. 
Partial Scenaria

To obtain less sceptical complete scenaria wrt Opt, and in the spirit of partial stable models we introduce partial scenaria.

Let the well–founded semantics of $P$ wrt Opt be $P \cup H$. $P \cup K$ is a partial scenario of $P$ wrt Opt iff it is a base scenario wrt Opt and $H \subseteq K$.

In the wobbly wheel example, partial scenaria are the union of $P$ with each of:

- $\{not \neg ww, not \neg fd\}$
- $\{not \neg ww, not \neg fd, not bs\}$
- $\{not \neg ww, not \neg fd, not \neg lv\}$
- $\{not \neg ww, not \neg fd, not \neg pt\}$
- $\{not \neg ww, not \neg fd, not bs, not \neg lv\}$
- $\{not \neg ww, not \neg fd, not bs, not \neg pt\}$
- $\{not \neg ww, not \neg fd, not \neg lv, not \neg pt, not \neg ft\}$

The first is the $WFS_{Opt}$. The other represent all other alternative hypothetical presences and absences of faults still compatible with the wobbly wheel observation.
Avoidance Example

Consider the statements:

Let’s go hiking if it is not known to rain;
Let’s go swimming if it is not known to rain;
Let’s go swimming if the water is not known to be cold;
We cannot go both swimming and hiking.

\[
\begin{align*}
\neg\text{hiking} & \leftarrow \text{swimming} \\
\neg\text{swimming} & \leftarrow \text{hiking} \\
\text{hiking} & \leftarrow \neg\text{rain} \\
\text{swimming} & \leftarrow \neg\text{rain} \\
\text{swimming} & \leftarrow \neg\text{cold\_water}
\end{align*}
\]

and let \( Opt = \{ \neg\text{rain}, \neg\text{cold\_water} \} \).

Complete scenaria wrt \( Opt \) are:

\[ P \cup \{\} \text{, and } P \cup \{ \neg\text{cold\_water}\} \]

where the latter is the \( WFS_{Opt} \). It entails that \( \text{swimming} \) is true.

Note that \( \neg\text{rain} \) is not assumed because it is optative to do so, and by assuming it contradiction would be unavoidable.
Removing Contradiction

Instead of defining more sceptical semantics, one can rely on a less sceptical semantics and provide a revision process.

Less sceptical semantics model a reasoner confident in his acceptability criterium. If a program is contradictory, then the problem is with the program, and its revision is in order.

We revise a program by adding to it rules preventing the assumption of some revisables.

Revisables are default literal provided by the user along with the program.
In order to revise contradictions we need to identify those contradictory sets implied by a program under a paraconsistent WFSX.

\[
\begin{align*}
  a & \leftarrow \text{not } b \quad \text{(i)} & d & \leftarrow \text{not } a \quad \text{(iii)} \\
  \neg a & \leftarrow \text{not } c \quad \text{(ii)} & e & \leftarrow \text{not } \neg a \quad \text{(iv)}
\end{align*}
\]

1. not \(b\) and not \(c\) hold since there are no rules for either \(b\) or \(c\)
2. \(\neg a\) and \(a\) hold from 1 and rules (i) and (ii)
3. not \(a\) and not \(\neg a\) hold from 2 and the coherence principle
4. \(d\) and \(e\) hold from 3 and rules (iii) and (iv)
5. not \(d\) and not \(e\) hold from 2 and rules (iii) and (iv), as they are the only rules for \(d\) and \(e\)
6. not \(\neg d\) and not \(\neg e\) hold from 4 and the coherence principle.
How to revise

Adding rules of the form $L \leftarrow \text{not } L$ (inhibition rules) force default literals, otherwise true, to become undefined.

Changing the truth value from to true to undefined is less committing than changing it to false.

To declaratively define revision we start by considering all possible ways of adding to $P$ inhibition rules for revisables.

Several different revisions might be equivalent, from the standpoint of their consequences.
Indissociables

\[
p \leftarrow \text{not } a \quad a \leftarrow b \quad a \leftarrow c \\
\neg p \quad b \leftarrow a
\]

with \( Rev = \{\text{not } a, \text{not } b, \text{not } c\} \).

Adding \( a \leftarrow \text{not } a, b \leftarrow \text{not } b \), or both, has the same consequences, since undefining \( a \) leads to the undefinedness of \( b \) and vice-versa. Considering all three as distinct can be misleading because it appears that the program has three different revisions.

Revisables \( \text{not } a \) and \( \text{not } b \) are said indissociable, and it is indifferent to introduce inhibition rules for one, the other, or both.

\( Ind(S) \supseteq S \) is the set of indissociables of default literals \( S \) iff \( Ind(S) \subseteq Rev \),

\[
Ind(S) \subseteq WFSX_p(P) \quad \text{and} \quad WFSX_p(P \cup IR(S)) \cap Ind(S) = \emptyset
\]
A submodel of $P$ is a pair $\langle M, R \rangle$ where:

**Submodel Revision:** $R \subseteq Rev$ and is closed under indissociables

**Consequences:** $M = WFSX_p(P \cup IR(R))$

where $IR(R) = \{L \leftarrow not \ L \mid not \ L \in R\}$.

$\langle M, R \rangle$ is contradictory iff $M$ is contradictory.

We are interested in revising contradiction in a minimal way. Thus we care about submodels that are non–contradictory and, among these, about those that are minimal in the submodels lattice.

Submodels in these conditions are minimal non–contradictory submodels or MNSs.
Declarative Revisions

Let \( \langle M, R \rangle \) be some MNS of \( P \). A *minimally revised program* MRP of \( P \) is: \( P \cup IR(R) \).

When different MRPs exist, and having no reason to prefer one over the others, a sceptical revision undefines all literals in some MNS.

The *sceptical submodel* of \( P \) is the join \( \langle M_J, R_J \rangle \) of all MNSs of \( P \). The *sceptical revised program* of \( P \) is \( P \cup IR(R_J) \).

Program \( P \) :

\[
\begin{align*}
a & \leftarrow \text{not } b & \neg a \\
b & \leftarrow \text{not } c & c
\end{align*}
\]

with \( Rev = \{ \text{not } c \} \) has no revisions.

This would not be the case if \( \text{not } b \in Rev \).
If \( \text{not } b \) were absent from the 1st rule, then \( P \) would have no revision no matter what the revisables.
A procedure for finding the minimal and the more sceptical submodels can hardly be based on their declarative definition: one would have to generate all the possible revisions to select those intended. In the paper we define a revision procedure, and show that it concurs with the declaratively intended revisions.

*Contradiction supports* are sets of revisable literals present in the $WFSX_p$ which are sufficient to support contradiction: from their truth contradiction inevitably follows.

*Contradiction removal sets* are minimal sets of literals chosen from those supports having at least one literal from each.
Definition of Support

Given $P$ and $Rev$, the supports of $L \in WFSX_p$ (represented as $SS(L)$) are:

1. If $L$ is an objective literal:
   (a) If there is a fact for $L$ then a support of $L$ is $SS(L) = \emptyset$.
   (b) For each rule $L \leftarrow B_1, \ldots, B_n \ (n \geq 1)$ in $P$ such that $\{B_1, \ldots, B_n\} \subseteq WFSX_p(P)$, there is a support $SS(L) = \bigcup_i SS_{j(i)}(B_i)$, for each combination of one $j(i)$ for each $i$.

2. If $L = not A$ (where $A$ is an objective literal):
   (a) If $L \in Rev$ then a support of $L$ is $SS(L) = \{L\}$.
   (b) If $L \notin Rev$ and there are no rules for $A$ then a support of $L$ is $SS(L) = \emptyset$.
   (c) If $L \notin Rev$ and there are rules for $A$, choose from each rule with non-empty body for $A$, a literal such that its default complement belongs to $WFSX_p(P)$. For each such multiple choice there exist several $SS(L)$; each contains one support of each default complement of the chosen literals.
   (d) If $\neg A \in WFSX_p(P)$ there are, additionally, supports $SS(L) = SS_k(\neg A)$ for each $k$. 
$p \leftarrow \neg q \quad q \leftarrow \neg r \quad \neg a \leftarrow \neg b$
$
\neg p \leftarrow \neg a \quad r \leftarrow \neg s$

with $Rev = \{\neg q, \neg a, \neg b\}$.

- Supports of $p$ are the supports of $\neg q$.
- Since $\neg q \in Rev$ it has a support $\{\neg q\}$.
- As $\neg q \notin WFSX_p(P)$, there are no other supports.

The only support of $p$ is $\{\neg q\}$.

- Supports of $\neg p$ are the supports of $\neg a$.
- Since $\neg a \in Rev$ it has a support $\{\neg a\}$.
- Supports of $\neg a$ are the supports of $\neg b$.
- The only support of $\neg b$ is $\{\neg b\}$.

$\neg p$ has two supports: $\{\neg a\}$ and $\{\neg b\}$.
Removal Sets

Contradiction supports of $P$ are supports of $\bot$ in: $P \cup \{ \bot \leftarrow L, \neg L \mid L \in \mathcal{H} \}$.

A pre–removal set of $L \in WFSX_p(P)$ is formed by the union of some non–empty subset of each $SS(L)$.

A removal set of $L$ is the closure under indissociables of some pre–removal set. If $\emptyset$ is a support of $L$ then the only $RS(L)$ is $\emptyset$.

A Contradiction removal set CRS is a minimal removal set of $\bot$.

Above, CRSs are $\{not\ q\}$ and $\{not\ a, not\ b\}$. 
CRSs and Revisions

**Soundness:** If \( R \) is a nonempty CRS of \( P \) then \( \langle M, R \rangle \) is a MNS of \( P \).

**Completeness:** If \( \langle M, R \rangle \), where \( R \neq \emptyset \), is a MNS of \( P \) then \( R \) is a CRS of \( P \).

In order to compute the minimal and sceptical submodels:

- One starts by computing all supports of \( \bot \).
- If \( \emptyset \) is a support of \( \bot \) then the program is unrevisable.
- Otherwise, after having all supports of \( \bot \), the rest follows by operations on these sets.
- Finally, a minimal revised program is obtained by adding to \( P \) one inhibition rule for each element of a CRS, and the sceptical revision is obtained as the union of all such minimally revised programs.
Conclusions

We’ve defined a sceptical semantics based on an abductive approach, which avoids contradiction.

We’ve defined a revision process that removes contradiction for programs under WFSX.

In the paper it is shown that avoidance is equivalent to removal, and optatives are tantamount to revisables.

The need for semantics more sceptical than WFSX, can be seen as showing the inadequacy of the latter. These results show that this is not the case.

Advantages of using WFSX reside mainly on:

- its simplicity

- the existence of top–down procedures for it.