An Abductive Paraconsistent Semantics – MH_P

Mário António Abrantes[1] and Luís Moniz Pereira[2]

[1]Instituto Politécnico de Bragança, [2]CENTRIA, Portugal

INAP13' September, 2013

Outline

MHP: an instant description

MH_P is a semantics for **extended normal logic programs** whose models are **total** and **paraconsistent**.

By **total** and **partial** Models we mean (normal logic programs case):

P	Q
$b \leftarrow not\ d$	$b \leftarrow not\ d$
$a \leftarrow not\ a$	$c \leftarrow$

$$WFM(P) = \langle \{b\}^+, \boldsymbol{\{a\}^u}, \{d\}^- \rangle : \textbf{Partial Model}$$
$$WFM(Q) = \langle \{b, c\}^+, \boldsymbol{\{\}^u}, \{d\}^- \rangle : \textbf{Total Model}$$

The necessity of Explicit Negation

This is a classic example (due to John McCarthy) .

We do not want to cross the railway on basis of **lack of a proof** the train is coming.

$$cross \leftarrow not\ train$$

The adequate way, is to **make** the train is not coming: we need to be able to **assert falsity**!

$$cross \leftarrow \neg train$$

Total Models via Abductive Semantics (1/2)

Sometimes **all the information** must be squeezed from a logic program.

For example, in an **emergency** situation,

danger $\leftarrow$ *not run*
run $\leftarrow$ *not safe*
safe $\leftarrow$ *not danger*

indecision may not be acceptable. **Eliminate indecision** enforcing a 2-valued semantics.

Total Models via Abductive Semantics (2/2)

Eliminate undecision via a 2-valued semantics.

- Add to P a **minimal set of hypotheses**, H, such that $WFM^u(P \cup H) = \emptyset$.
- **Assumable set of hypotheses**: atoms that **appear default negated**: $\{run, danger, safe\}$.
- For example, $H = \{run\}$:

P $\cup\{run\}$

danger $\leftarrow$ *not run*
safe $\leftarrow$ *not danger*
run $\leftarrow$ *not safe*
run $\leftarrow$

$$\text{WFM}(\text{P} \cup \{run\}) = \{run, not\ danger, safe\}$$

The *MH* Spirit: the Holiday Problem (1/2)

Four friends are planning a holiday.

- **First friend** says "If we don't go to Germany, then we must go to Sweden"

  sweden $\leftarrow$ **not germany**.

  etc. for the first 3 friends.
- **Fourth friend** says "We must go to Denmark"

  denmark $\leftarrow$.

$$
\begin{aligned}
sweden &\leftarrow not\ germany\\
denmark &\leftarrow not\ sweden\\
germany &\leftarrow not\ denmark\\
denmark &\leftarrow
\end{aligned}
$$

The *MH* Spirit: the holiday Problem (2/2)

There is **a single** stable model solution.

SM(P)={denmark, not germany, sweden}

But on simple inspection, **another** solution is devised.

SM(P)={denmark, germany, not sweden}

Both solutions are obtained if we envisage the loop in the program as a **choice device**, by considering all the default negated atoms as assumable hypotheses.

WFM Computation via Program Transformation

The **WFM** of a logic program may be computed via the **remainder** of the program.

The **remainder** is computed by transforming the original program using 5 operations: **loop detection**, **failure**, **positive reduction**, **success**, **negative reduction**.

This reduction system is **terminating** and **confluent** for finite ground normal logic programs.

WFM Computation via Program Remainder (2/2)

Example

- Rules and literals highlighted in program Q, below, are eliminated during remainder computation
- $remainder(Q) = \widehat{Q}$

$$WFM(Q) = WFM(\widehat{(Q)}) = \{d, s\} \cup not\ \{g, k, u, w\}.$$

$u \leftarrow w$		Loop Detection
$w \leftarrow u$		Loop Detection
$k \leftarrow g$		Failure
$s \leftarrow not\ g, d$		Positive Reduction+Success
$g \leftarrow not\ d$		Negative Reduction
$d \leftarrow not\ s$	$d \leftarrow$	Negative Reduction

Layered Remainder Computation (1/2)

The **layered remainder** uses the loops as **choice devices**. The key to preserve loops is to replace **negative reduction** by . . .

layered negative reduction: Use fact **f** to eliminate a rule **h** ← **not f** iff the rule is not in loop through **not f**.

The **layered remainder** is computed by transforming the original program using 5 operations: loop detection, failure, positive reduction, success, **layered negative reduction**.

The model obtained with the layered remainder is the **layered well-founded model,** *LWFM* .

Layered Remainder Computation (2/2): example

Example of **layered remainder** computation of program Q below:

- The highlighted rules and literals are eliminated.
- Denote by $\mathring{Q}$ the **layered remainder** of Q.
- $LWFM(Q) = \{d\} \cup not\ \{u, w\}$

$\boxed{u \leftarrow w}$	Loop Detection
$\boxed{w \leftarrow u}$	Loop Detection
$k \leftarrow g$	
$s \leftarrow not\ g, \boxed{d}$	Success
$g \leftarrow not\ d$	
$d \leftarrow not\ s$	
$d \leftarrow$	

MH Models Computation

- Form the **assumable hypotheses set** of Q (**default negated atoms** that **are not facts** in $\mathring{Q}$): $\{g, s\}$.
- Compute all the 2-valued stable models of $Q \cup H$, for all **nonempty minimal hypotheses sets** $H \subseteq \{g, s\}$ **and for** $H = \emptyset$.
- MH models of Q: $\{d, not\ g, not\ k, s\}$ with **hypotheses sets** $H = \emptyset$ and $H = \{s\}$, and $\{d, g, k, not\ s\}$ with **hypotheses set** $H = \{g\}$.

$$\mathring{Q}$$

$$\begin{array}{lll} k \leftarrow g & s \leftarrow not\ g & d \leftarrow not\ s \\ g \leftarrow not\ d & d \leftarrow & \end{array}$$

$WFSX_P$ Semantics (1/4)

Extended logic programs allow two types of negation: **default negation** *not* b and **explicit negation** $\neg b$.

WFSXp: well-founded semantics for extended logic programs.

- Collapses into *WFS* for normal logic programs.
- Relates **default negation** and **explicit negation** through the **coherence principle**: if $\neg l$ holds, then *not* l also does (similarly, if l then *not* $\neg l$).
- **Detects** dependencies on contradiction.

$WFSX_P$ Semantics (2/4)

Example: WFSXp model

$$P$$

$$z \leftarrow not\ z$$

$$\neg a \leftarrow$$

$$u \leftarrow not\ a$$

$$c \leftarrow not\ d$$

WFSXp(P)=$\langle \{\neg a, c, u\}^+, \{z, \neg z\}^u, \{a, \neg c, \neg u\}^- \rangle$.

$\boldsymbol{\neg a}$ and $\boldsymbol{u \leftarrow not\ a}$ render $\boldsymbol{u}$ true via the **coherence principle**.

$WFSX_P$ Semantics (3/4)

WFSXp may be embedded into **WFS** by a simple transformation.

- Take an extended program P and compute the P^{t-o} transformed of P:

P	P^{t-o}	
$\neg a \leftarrow$	$\neg\_a \leftarrow$	$\neg\_a^o \leftarrow \textbf{not a}$
$c \leftarrow not\ b$	$c \leftarrow not\ b^o$	$c^o \leftarrow not\ b, \textbf{not } \neg\_\mathbf{c}$
$u \leftarrow \neg a$	$u \leftarrow \neg\_a$	$u^o \leftarrow \neg\_a^o, \textbf{not } \neg\_\mathbf{u}$

$\neg\_a, \neg\_a^o, \neg\_c, \neg\_u$ in P^{t-o} language are **names of atoms**, not explicit negations. **Bold** literals enforce the **coherence principle**.

- Compute the $WFM(P^{t-o})$:

$WFSX_P$ Semantics (4/4)

$$WFM(P^{t-o}) = \langle \{\neg\_a, \neg\_a^o, c, c^o, u, u^o\}^+, \{\}^u \{a, a^o, b, b^o, \neg\_b, \neg\_b^o, \neg\_c, \neg\_c^o, \neg\_u, \neg\_u^o\}^- \rangle.$$

- **Read** the $WFSXp(P)$ model **from** $WFM(P^{t-o})$

$$a \in WFMp(P) \textbf{ iff } a \in WFM(P^{t-o})$$
$$not\ a \in WFMp(P) \textbf{ iff } not\ a^o \in WFM(P^{t-o})$$
$$\neg a \in WFMp(P) \textbf{ iff } \neg\_a \in WFM(P^{t-o})$$
$$not\ \neg a \in WFMp(P) \textbf{ iff } not\ \neg\_a^o \in WFM(P^{t-o})$$

$$WFMp(P) = \langle \{\neg a, c, u\}^+, \{\}^u \{a, b, \neg b, \neg c, \neg u\}^- \rangle.$$

Computing MH_P Models (1/2)

- Take an extended normal logic program P.
- Compute the transformed P^{t-o}.
- Compute the **balanced layered remainder** bP^{t-o} (for preserving loops) by means of the **balanced reduction system**.

balanced reduction system, consists in 5 operations: loop detection, failure, positive reduction, success, **balanced layered negative reduction**.

balanced layered negative reduction: Use fact $\mathbf{f^o}$ (resp. $\mathbf{f}$) to eliminate rule $\mathbf{r} = \mathbf{h} \leftarrow \mathbf{not\ f^o}$ (resp. $\mathbf{r^o} = \mathbf{h^o} \leftarrow \mathbf{not\ f}$) iff $\mathbf{r}, \mathbf{r^o}$ are not in loop through $\mathbf{not\ f^o}, \mathbf{not\ f}$.

Computing MH_P Models (2/2)

Compute the set of **assumable hypotheses** of P, $Hyps(P)$: **all the literals** k **such that** *not* $k^o \in bP^{t-o}$ and k **is not a fact** of bP^{t-o}.

MHp models: **total** $WFSXp$ **models** of programs $P \cup H$, for all **nonempty minimal** hypotheses sets $H \subseteq Hyps(P)$ **and for** $H = \emptyset$.

Computing MH_P Models: an example (1/2)

- An extended program P and its balanced layered remainder, bP^{t-o}

P	bP^{t-o}	
$b \leftarrow h$	$b \leftarrow h$	$\cancel{b^o \leftarrow h^o, not\ \neg b}$
$h \leftarrow not\ p$	$h \leftarrow not\ p^o$	$\cancel{h^o \leftarrow not\ p, not\ \neg h}$
$p \leftarrow not\ b$	$p \leftarrow not\ b^o$	$p^o \leftarrow not\ b, \cancel{not\ \neg p}$
$b \leftarrow$	$b \leftarrow$	$b^o \leftarrow \cancel{not\ \neg b}$
$\neg h \leftarrow$	$\neg h \leftarrow$	$\neg h^o \leftarrow not\ h$

- **Assumable set of hypotheses**, $Hyps(P) = \{p\}$: $not\ p^o$ appears in bP^{t-o} and p is not a fact.

Computing MH_P Models: an example (2/2)

- MH_P models of P:

$$M_1 = \langle \{b, h, \neg h\}^+, \{\}^u, \{\neg b, h, \neg h, p, \neg p\}^- \rangle$$
$$\text{with hypotheses set} \quad H = \emptyset$$
$$M_2 = \langle \{b, p, \neg h\}^+, \{\}^u, \{h, \neg b, \neg p\}^- \rangle$$
$$\text{with hypotheses set} \quad H = \{p\}$$

- M_1 is *default inconsistent* (e.g. h and *not* h belong to M_1).
- M_2 is *consistent*: is a solution to this variant of the **holiday problem**.

Conclusion

- MH_P is a total models paraconsistent semantics that **solves any** extended normal logic program.
- MH_P models detect dependency on contradiction: objective literals L that are dependent on contradiction exhibit **default inconsistency, i.e. both** L **and** $notL$ **are in the model**.
- Computing a MH_P model is a Σ_2^P task.
- Belief revision, or contradiction removal is treated elsewhere, in MA's forthcoming PhD thesis.

THANKS!