A Declarative Characterisation of Disjunctive Paraconsistent Answer Sets

João Alcântara and **Carlos Viegas Damásio** and **Luís Moniz Pereira** [1]

Abstract.
In this work, paraconsistent answer sets for extended disjunctive logic programs are presented as a fully declarative approach. In order to do so, we introduce a frame-based semantics. Frames are a powerful and elegant tool which have been used to characterise and study substructural logics. Unlike the original definition, no kind of syntactic transformation is employed. Indeed, paraconsistent answer sets are defined by minimising models satisfying some conditions. Considering that paraconsistent answer sets embed both answer sets and stable models, these semantics are also captured via frames.

1 INTRODUCTION

Paraconsistent answer [4] sets can be used to represent Rough Knowledge Bases supporting reasoning with Rough Sets, namely for medical applications. Nonetheless, as the authors define their semantics by resorting to syntactic transformations, the aim of characterising logic programming semantics by following only standard logical definition is not totally accomplished.

In [2], Pearce characterised Stable Models and Answer Sets as specific minimal models under Heyting's monotonic logic of *here-and-there*. Afterwards, Cabalar [1] resorted to a two-dimensional version of the logic of here-and-there to capture the Well-founded Semantics. For Paraconsistent Answer Sets, however, until now, there is no fully declarative proposal.

Our general aim in the present work is to fill this gap by providing a definition of paraconsistent answer sets grounded only in logical terms. Unlike the original one, this is attained without employing any kind of syntactic transformation. Like Pearce we also resort to the logic here-and-there; like Cabalar, we also introduce a two-dimensional version of this logic. However, we employ the additional dimension to capture explicit negation information instead of incomplete information as Cabalar does.

Although preserving Pearce's original motivation, we exhibit our approach by sticking to frame-based semantics as presented by Greg Restall [3]. Frames are a powerful and elegant tool which have been used to characterise and integrate substructural logics. Furthermore, since paraconsistent answer sets embed both answer sets and stable models, frames are also suited to capture them.

In next section we show our main contributions: the definition of paraconsistent answer sets as a frame-based semantics, and how we can easily adjust it to grasp answer sets and stable models. Then, in Section 3 we draw conclusions and mention future work.

[1] Centro de Inteligência Artificial (CENTRIA) 2829-516 Caparica, Portugal. email: `jfla|cd|lmp@di.fct.unl.pt`

2 A frame-based semantics for PAS

Now we present an alternative entirely logical definition for PAS based on frames [3] instead of requiring a syntactic operation (like the program reduct) as in [4]. Frames are characterised by tuples constituted by point sets, accessibility relations, and truth sets:

Definition 1 (Point Set) *[3] A point set $\mathcal{P} = \langle Q, \sqsubseteq \rangle$ is a set Q together with a partial order $\sqsubseteq$ on Q. The set $Prop(\mathcal{P})$ of* propositions *on $\mathcal{P}$ is the set of all subsets X of Q* closed upwards, *that is, if $x \in X$ and $x \sqsubseteq x'$ then $x' \in X$.*

We shall employ accessibility relations to evaluate intensional connectives:

Definition 2 (Accessibility Relations) *[3]*

- *A relation C is a* plump negative 2-place accessibility relation *on the point set $\mathcal{P} = \langle Q, \sqsubseteq \rangle$ if and only if for any $x, y, x', y' \in Q$, in which xCy, $x' \sqsubseteq x$ and $y' \sqsubseteq y$ it follows that $x'Cy'$.*
- *A relation R is a* plump 3-place accessibility relation *on the point set $\mathcal{P} = \langle Q, \sqsubseteq \rangle$ if and only if for any $x, y, z, x', y', z' \in Q$, in which $Rxyz$, $x' \sqsubseteq x$, $y' \sqsubseteq y$ and $z \sqsubseteq z'$ then $Rx'y'z'$.*

Plump negative 2-place accessibility relations will be associated with negations, whilst plump 3-place accessibility relations with the rule symbol "$\rightarrow$". Below we define truth sets, whose interpretation makes them eligible to define the truth constant $\mathtt{t}$.

Definition 3 (Truth Sets) *[3] If R is a (plump) 3-place accessibility relation on a point set $\mathcal{P} = \langle Q, \sqsubseteq \rangle$ then for any subset $T \in Prop(\mathcal{P})$, T is a* right truth set *for R if and only if for each $x, y \in Q$, $x \sqsubseteq y$ if and only if for some $z \in T$, $Rxzy$.*

Now that we have defined a point set, accessibility relations and truth sets, the notion of frame can be introduced straightforwardly:

Definition 4 (Frame) *[3] A frame is a point set $\mathcal{P}$ together with any number of accessibility relations and truth sets on $\mathcal{P}$.*

Below we present the frame we shall use in the definition of PAS. We reserve $\mathcal{F}$ to denote the following frame:

1. The point set $\mathcal{P} = \langle Q, \sqsubseteq \rangle$ such that $Q = \{hp, hn, tp, tn\}$, $hp \sqsubseteq hp$, $tp \sqsubseteq tp$, $hn \sqsubseteq hn$, $tn \sqsubseteq tn$, $hp \sqsubseteq tp$, and $tn \sqsubseteq hn$. In Figure 1 we exhibit a graphical representation of $\mathcal{P}$ in which the relation $\sqsubseteq$ is presented through the sense pointed by the arrow. The elements of Q are intended to represent different levels of positive truth values expounded in the Table 1.

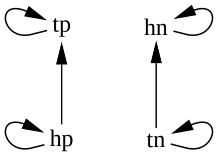

Figure 1. Point Set for *PAS*

Point	Intended Truth Value
hp	Necessarily true
hn	Necessarily not explicitly false
tp	True
tn	Not explicitly false

Table 1. Intended Truth Values in $\mathcal{P}$

2. The accessibility relations defined on $\mathcal{P}$: $R, R_\neg, R_{not}$, in which R is exhibited in Tables 2, and $R_\neg$ and R_{not} respectively in Figures 2, 3, such that an arrow from x to y denotes that there is an accessibility relation from x to y. As the reader can check, R is a plump 3-place accessibility relation, whilst $R_\neg$ and R_{not} are plump negative 2-place accessibility relations.

$R\ hp\ hp\ hp$	$R\ tn\ tp\ tn$
$R\ hp\ hp\ tp$	$R\ tn\ tp\ hn$
$R\ hp\ tp\ tp$	$R\ tn\ hp\ tn$
$R\ tp\ tp\ tp$	$R\ tn\ hp\ hn$
$R\ tp\ hp\ tp$	$R\ hn\ hp\ hn$

Table 2. Accessibility Relation R

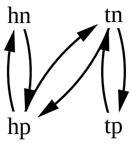

Figure 2. Accessibility Relation $R_\neg$

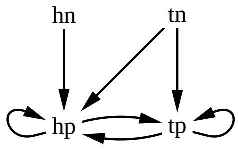

Figure 3. Accessibility Relation R_{not}

3. The (right) truth set of R is $\{hp, tp\}$, and corresponds to the truth constant $\mathbf{t}$ (see Definition 7).

The accessibility relations R, $R_\neg$ and R_{not} are obtained by adapting N-valuations used in [2] to present respectively the operators $\rightarrow$, $\neg$ and *not* into our frame to capture PAS^2. Alternatively, explicit negation could be treated in terms of a partial Kripke-style semantics as Pearce did. However, guided by Greg Restall's directions, we prefer to preserve the two-valued assessment applied to a two-dimensional frame. In the sequel some definitions handle the semantical part.

Definition 5 *By belief set, we mean (S^p, S^n), in which S^p and S^n are sets of atoms. Thus, an atom A is true in S^p (resp. S^n) iff $A \in S^p$ (resp. $A \in S^n$); otherwise, A is false in S^p (resp. A is false in S^n).*

The knowledge ordering $\leq_k$ (see [4]) can be mimicked in a relationship involving belief sets as follows: let $B_1 = (S_1^p, S_1^n)$ and $B_2 = (S_2^p, S_2^n)$ be belief sets. The knowledge ordering $\leq_k$ among them is defined by $B_1 \leq_k B_2$ iff $S_1^p \subseteq S_2^p, S_2^n \subseteq S_1^n$. The mechanism behind knowledge ordering between belief sets is crucial to guarantee the expected definition of PAS. Pursuant to this aim, firstly we use $\leq_k$ to define HT^2-interpretations:

Definition 6 *A HT^2-interpretation is the pair $[B^h, B^t]$, in which B^h and B^t are belief sets satisfying $B^h \leq_k B^t$.*

Recalling frame $\mathcal{F}$, for each atom, a HT^2-interpretation can assign nine possible (truth) values corresponding isomorphically to the values found in the nine-valued logic IX (see [4]). To expunge any misunderstanding, we shall reserve the letters B and S to respectively denote belief sets and sets of atoms, using the notation $B^h = (S^{hp}, S^{hn})$ and $B^t = (S^{tp}, S^{tn})$. Now we are going to associate each S^x in $[B^h, B^t]$ to a $x \in Q$:

Definition 7 (HT^2-model) *Let $w \in \{hn, hp, tn, tp\}$ be a point of $\mathcal{F}$, $M = [B^h, B^t]$ be a HT^2-interpretation,"A" be an atom, and both ϕ and ψ be formulae. We say that ϕ is satisfied by M in w, written $(M, w) \Vdash \phi$, iff*

1. *$(M, w) \Vdash A$ iff $A \in S^{wp}$*
2. *$(M, w) \Vdash \mathbf{t}$ for all w in $\{hp, tp\}$*
3. *$(M, w) \Vdash \phi \wedge \psi$ iff $(M, w) \Vdash \phi$ and $(M, w) \Vdash \psi$*
4. *$(M, w) \Vdash \phi \vee \psi$ iff $(M, w) \Vdash \phi$ or $(M, w) \Vdash \psi$*
5. *$(M, w) \Vdash \neg\phi$ iff for each w' in $\mathcal{F}$ s.t. $wR_\neg w'$, $(M, w') \nVdash \phi$*
6. *$(M, w) \Vdash$ not ϕ iff for each w' in $\mathcal{F}$ s.t. $wR_{not} w'$, $(M, w') \nVdash \phi$*
7. *$(M, w) \Vdash \psi \rightarrow \phi$ iff for each w', w'' in $\mathcal{F}$ s.t. $R\ w\ w'\ w''$, if $(M, w') \Vdash \psi$, then $(M, w'') \Vdash \phi$*

M is a HT^2-model of a theory T iff $(M, hp) \Vdash \phi$ for each ϕ in T.

We say a HT^2-model $[B^h, B^t]$ of a program[3] P is *p-minimal* if there is no belief set $B' <_k B$ such that $[B', B^t]$ is a HT^2-model of P. The main result of this paper is shown below:

Theorem 1 *The p-minimal HT^2-models $[B^t, B^t]$ of a program P are exactly its paraconsistent answer sets.*

Considering paraconsistent answer sets embed both answer sets and stable models, our proposal is obviously eligible to deal with them. Answer sets can be defined by adding $hp \sqsubseteq hn$ and $tp \sqsubseteq tn$ to the point set of $\mathcal{F}$, and because of the conditions in Definition 2, we should also add new instances to $R, R_\neg$, and R_{not}. Similarly, stable models can be seen as answer sets versions free of explicit negation. The resulting frame is isomorphic to the one presented by Pearce [2] for answer sets (stable models). The main difference is that Pearce resorts to partial Kripke models to characterise answer sets, whilst we preserve the two-valued evaluation in each point (world).

3 Conclusion

We have defined a fully declarative approach for paraconsistent answer sets, by resorting to a frame based semantics. This is the first time a complete declarative characterisation is presented for paraconsistent answer sets, no syntactic transformation is used. Indeed paraconsistent answer sets are obtained by minimising models satisfying some conditions. Our proposal not only captures paraconsistent answer sets for extended disjunctive logic programs, but also models for any theory composed by formulae recursively definable for all program connectives. We have shown how one embed answer sets and stable models via frames.

Motivated by preliminary results, a general frame-based semantics to simultaneously capture both stable models and well-founded semantics families is expected in a following work. Finally, our proposal permits us to explore questions involving the role of logic programming semantics in the context of substructural logics.

João Alcântara is a PhD student supported by CAPES - Brasília, Brazil. The authors also thank the TARDE and FLUX projects, sponsored by FCT - Portugal.

REFERENCES

[1] P. Cabalar, 'Well founded semantics as two-dimensional here-and-there', in *Answer Set Programming*, pp. 15–20. AAAI Press, (2001).

[2] D. Pearce, 'From here to there: stable negation in logic programming', in *What is Negation?*, eds., D. M. Gabbay and H. Wansing, 161–181, Kluwer Academic Publishers, (1999).

[3] Greg Restall, *An Introduction to Substructural Logics*, Routledge, 2002.

[4] C. Sakama and K. Inoue, 'Paraconsistent stable semantics for extended disjunctive programs', *J. Logic and Computation*, **5**(3), 265–285, (1995).

[2] In [2] N-valuations are used to determine Answer Sets.

[3] A program is a set of rules $L_1 \wedge \ldots \wedge L_l \wedge$ *not* $L_{l+1} \wedge \ldots \wedge$ *not* $L_m \rightarrow L_{m+1} \vee \ldots \vee L_n$, where each L_i is an atom or its explicit negation.