Properties of Conservative Extensions to Stable Model Semantics: a Structural Approach

Mário Abrantes¹ and Luís Moniz Pereira²

¹Departamento de Matemática, Escola Superior de Tecnologia e de Gestão, Instituto Politécnico de Bragança, 5300-253 Bragança, Portugal
²Centro de Inteligência Artificial (CENTRIA), Departamento de Informática, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal

Abstract

The stable model (SM) semantics lacks the properties of existence, relevance and cumulativity. It is thus reasonable to look for conservative extensions of this semantics (i.e., semantics that for each normal logic program P retrieve a superset of the set of stable models of P) that may enjoy some or all of the above properties. In this work we define a large class of conservative extensions of the SM, dubbed affix stable model semantics, ASM, and study the above referred properties into two non-disjoint subfamilies, ASMᵇ and ASMᵐ, of the class ASM. Among the results obtained, the following should be emphasized: (1) We present a refined definition of cumulativity for semantics of the set ASMᵇ ∪ ASMᵐ, that turns into an easier job the dismissal of this property by resorting to counter examples; (2) We divide the sets of rules of normal logic programs into layers and use the decomposition of models into that layered structure to define three new (structural) properties, defectivity, excessiveness and irregularity, which allow to state a number of relations between the properties of existence, relevance and cumulativity for semantics of the ASMᵇ ∪ ASMᵐ class; (3) As a consequence of our work, we show that for the SM semantics case, the properties of (lack of) existence and (lack of) cautious monotony are equivalent, which opposes statements on this issue that may be found in the literature.

Keywords: Stable model semantics, Conservative extensions to stable model semantics, Existence, Relevance, Cumulativity, Defectivity, Excessiveness, Irregularity, 2-valued semantics for logic programs.

1 Introduction

The SM semantics is generally accepted by the scientific community working on logic programs semantics as the de facto standard 2-valued semantics. Nevertheless there are some advantageous properties the SM semantics lacks such as (1) model existence for every normal logic program, (2) relevance, and (3) cumulativity (Pinto and Pereira 2011).

It is known that all logic programs that lack stable models contain odd loops over negation (Costantini 1995). It seems thus reasonable to cater for models that afford odd loops, in order to accomplish a 2-valued semantics for every normal logic program, whilst maintaining all the stable models. This goal can be achieved by considering 2-valued conservative extensions of the SM semantics (Pinto and Pereira 2011), where conservative extension means a semantics that for each normal logic program retrieves all the stable models, possibly together with some additional ones. In this paper we define a family of 2-valued conservative extensions of the SM semantics, the affix stable model semantics family, ASM. The purpose of establishing this family is to present a general enough definition of 2-valued model conservative extensions of the SM semantics, which eventually represents both a large and interesting number of semantics under this designation. We restrict the study presented in this paper to two non-disjoint classes of semantics, ASMᵇ ∩ ASM and ASMᵐ ∩ ASM. Semantics of the ASMᵇ family are minimal (positive) hypotheses generated abductive semantics (though the ensuing models might not be minimal), whilst the semantics of the ASMᵐ family are minimal models abductive semantics. We show that a semantics SEM of any of these two families is cumulative iff for any normal logic program P and any subset S ⊆ ∩ M⁺ M ∈ SEM(P) we have SEM(P) = SEM(P ∪ S), i.e., P and P ∪ S have exactly the same 2-valued models, where M⁺ represents the set of positive literals in model M. That is, when dealing with semantics of the class ASMᵇ ∪ ASMᵐ, cumulativity definition may be refined as a relation among sets of models, instead of a relation among sets of atoms, as stated by the usual definition (Dix 1995a).

By resorting to three semantics structural properties, first defined in this paper, defectivity, excessiveness and irregularity, the following relations are established for any semantics SEM ∈ ASMᵇ ∪ ASMᵐ:

1. Defectivity ⇔ ¬ Existence ⇔ ¬ Global to Local Relevance;
2. Defectivity ⇒ ¬ Cautious Monotony;
3. Excessiveness ⇒ ¬ Cut;
4. Irregularity ⇔ ¬ Local to Global Relevance.

¹mar@ipb.pt
²lmp@fct.unl.pt
Copyright © 2013, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
where the properties global to local relevance and local to global relevance, arise from splitting relevance into its two separate logic entailments. The results referred above may turn into an easier job both the defeating, as well as the proof, of the properties of existence, relevance and cumulativity, making it a matter of dealing with the decomposition of models into the structure of a program (i.e., over the layers of normal logic programs (Pereira and Pinto 2009)), thus avoiding in some cases proofs that may not be easy to obtain using a direct proof strategy.

The structural approach taken in this paper to establish the profile of $ASM^h$ and $ASM^m$ families semantics, concerning in particular the aforementioned properties, relies heavily on the concept of layering of normal logic programs (Pereira and Pinto 2009), and is thus in line with a syntactic tackling of the semantics of normal logic programs.

The results presented in this paper are enounced for the universe of finite ground normal logic programs, and are either proved in (Abrantes 2013), or immediate consequences of results there contained.

The remainder of the paper proceeds as follows. In section 2 we define the language of normal logic programs and the terminology to be used in the sequel. In section 3 the families $ASM$, $ASM^h$ and $ASM^m$ are defined, as well as some semantics pertaining to them. In section 4 we characterize the property of cumulativity for the families $ASM^h$ and $ASM^m$, while in section 5 the properties of defectivity, excessiveness and irregularity are defined, and some relations among existence, relevance and cumulativity, which are revealed by means of these properties, are stated. Section 6 is dedicated to final remarks.

2 Language and Terminology of Logic Programs

A normal logic program defined over a language $L$ is a set of normal rules, each of the form

$$b_0 ← b_1, \cdots, b_m, \text{not } c_1, \cdots, \text{not } c_n$$

(1)

where $m, n$ are non-negative integer numbers and $b_i, c_k$ are atoms of $L$; $b_i$ and not $c_k$ are generically designated literals, not $c_k$ being specifically designated default literal. The operator ‘,$\text{not }$’ stands for the conjunctive connective, the operator ‘,$\text{not }$’ stands for negation by default and the operator ‘,$\text{not }$’ stands for a dependency operator that establishes a dependence of $b_0$ on the conjunction on the right side of ‘,$\text{not }$’. $b_0$ is the head of the rule and $b_1, \cdots, b_m, \text{not } c_1, \cdots, \text{not } c_n$ is the body of the rule. A rule is a fact if $m = n = 0$. A literal (or a program) is ground if it does not contain variables. The set of all ground atoms of a normal logic program is called Herbrand base of $P$, $\mathcal{H}_P$. A program is finite if it has a finite number of rules.

Given a program $P$, program $Q$ is a subprogram of $P$ if $Q \subseteq P$, where $Q, P$ are envisaged as sets of rules.

For ease of exposition we henceforth use the following abbreviations: $Atoms(E)$, is the set of all atoms that appear in the ground structure $E$, where $E$ can be a rule, a set of rules or a set of logic expressions; $Body(r)$, is the set of literals in the body of a ground rule $r$; $Facts(E)$, is the set of all facts that appear in the set of rules $E$; $Heads(E)$, is the set of all atoms that appear in the heads of the set of rules $E$; if $E$ is unitary, we may use ‘$Head$’ instead of ‘$Heads$’. We may compound some of these abbreviations, as for instance $Atoms(Body(r))$ whose meaning is straightforward. Each of the abbreviations may also be taken as the conjunction of the elements contained in the respective sets.

The following concepts concern the structure of programs. Let $P$ be a logic program and $r, s$ any two rules of $P$.

1. Complete rule graph. (adapted from (Pinto and Pereira 2011)) The complete rule graph of $P$, denoted by $CRG(P)$, is the directed graph whose vertices are the rules of $P$. Two vertices representing rules $r$ and $s$ define an arc from $r$ to $s$ iff $Head(r) \subseteq Atoms(Body(s))$.

2. Rule depending on a rule. (adapted from (Pinto and Pereira 2011)) We say that $s$ depends on $r$ iff there is a directed path in $CRG(P)$ from $r$ to $s$.

3. Subprogram relevant to an atom. (adapted from (Dix 1995b)) We say that a rule $r \in P$ is relevant to an atom $a \in \mathcal{H}_P$ iff there is a rule $s$ such that $Head(s) = \{a\}$ and $s$ depends on $r$. The set of all rules of $P$ relevant to $a$ is represented by $Rel_P(a)$, and is named subprogram (of $P$) relevant to $a$.

4. Loop. (adapted from (Costantini 1995)) A set of rules $R$ forms a loop (or the rules of set $R$ are in loop) iff, for any two rules $r, s \in R$, $r$ depends on $s$ and $s$ depends on $r$. We say that rule $r \in R$ is in loop through literal $L \in Body(r)$ iff there is a rule $s \in R$ such that $Head(s) = Atoms(L)$.

5. Rule layering. (adapted from (Pinto and Pereira 2011)) The rule layering (or just layering, for simplicity) of $P$ is the labeling of each rule $r \in P$ with the smallest possible natural number, layer($r$), in the following way: for any two rules $r$ and $s$, (1) if rules $r, s$ are in loop, then layer($r$) = layer($s$); (2) if rule $r$ depends on rule $s$ but rule $s$ does not depend on rule $r$, then layer($r$) > layer($s$). Every integer number $T$ in the image of the layer function defines a layer of $P$, meaning the set of rules of $P$ labeled with number $T$ – we use the expression ‘layer’ to refer both to a set of all rules with the same label, and to the label itself. We represent by $P^{\leq T}$ ($P^{> T}$) the set of all rules of $P$ whose layer is less or equal to (greater than) $T$.

1In this work, if nothing to the contrary is said, by ‘logic program’, or simply by ‘program’, we mean a finite set of normal ground rules.
6. **T-segment of a program.** We say that \( P^{\leq T} \) is the \( T \)-segment of \( P \) iff \( \text{Atoms}(P^{\leq T}) \cap \text{Heads}(P^{> T}) = \emptyset \). We may also say “segment \( T \)” to mean the set of rules corresponding to segment \( P^{\leq T} \).

Let \( \mathit{SEM} \) be a 2-valued semantics and \( \mathit{SEM}(P) \) the set of \( \mathit{SEM} \) models of a logic program \( P \). Let also the set of atoms \( \ker_{\mathit{SEM}}(P) = \bigcap_{M \in \mathit{SEM}(P)} M^+ \) be dubbed semantic kernel of \( P \) with respect to \( \mathit{SEM} \). The following properties concern semantics of logic programs. We say that \( \mathit{SEM} \) is:\(^2\)

1. **Existential** iff every normal logic program has at least one \( \mathit{SEM} \) model;
2. **Cautious monotonic** iff for every normal logic program \( P \), and for every set \( S \subseteq \ker_{\mathit{SEM}}(P) \), we have \( \ker_{\mathit{SEM}}(P) \subseteq \ker_{\mathit{SEM}}(P \cup S) \);
3. **Cut** iff for every normal logic program \( P \), and for every set \( S \subseteq \ker_{\mathit{SEM}}(P) \), we have \( \ker_{\mathit{SEM}}(P \cup S) \subseteq \ker_{\mathit{SEM}}(P) \);
4. **Cumulative** iff it is cautious monotonic and cut;
5. **Relevant** iff for every normal logic program \( P \) we have
\[
\forall a \in \mathcal{H}(P) (a \in \ker_{\mathit{SEM}}(P) \iff a \in \ker_{\mathit{SEM}}(\mathit{Rel}_P(a)))
\] (2)
with \( \mathit{Rel}_P(a) \) is the subprogram of \( P \) relevant to atom \( a \);
6. **Global to local relevant** iff the logical entailment \( \Rightarrow \) stands in formula (2);
7. **Local to global relevant** iff the logical entailment \( \Leftarrow \) stands in formula (2).

3 **Conservative Extensions of the SM Semantics**

In this section we define a family of abductive 2-valued semantics,\(^3\) the affix stable model family, \( \mathit{ASM} \), whose members are conservative extensions of the \( \mathit{SM} \) semantics. For this purpose we need to introduce two concepts that can be found in the literature: the concept of reduction system (Brass et al. 2001) and the concept of minimal hypotheses semantics (Pinto and Pereira 2011).

**Reduction System**

In (Brass et al. 2001) the authors propose a set of five operations to reduce a program (i.e., eliminate rules or literals) – positive reduction, \( \mathit{PR} \), negative reduction, \( \mathit{NR} \), success, \( \mathit{S} \), failure, \( \mathit{F} \) and loop detection, \( \mathit{L} \) – whose definitions are as follows (\( P_1 \) and \( P_2 \) being two ground logic programs):

1. **Positive reduction, \( \mathit{PR} \).** Program \( P_2 \) results from \( P_1 \) by positive reduction iff there is a rule \( r \in P_1 \) and a default literal \( \neg b \in \text{Body}(r) \) such that \( b \notin \text{Heads}(P_1) \), and \( P_2 = (P_1 \setminus \{r\}) \cup \{\text{Head}(r) \leftarrow (\text{Body}(r) \setminus \{\neg b\})\} \).

2. **Negative reduction, \( \mathit{NR} \).** Program \( P_2 \) results from \( P_1 \) by negative reduction iff there is a rule \( r \in P_1 \) and a default literal \( \neg b \in \text{Body}(r) \) such that \( b \notin \text{Facts}(P_1) \), and \( P_2 = P_1 \setminus \{r\} \).

3. **Success, \( \mathit{S} \).** Program \( P_2 \) results from \( P_1 \) by success iff there is a rule \( r \in P_1 \) and a fact \( b \in \text{Facts}(P_1) \) such that \( b \in \text{Body}(r) \), and \( P_2 = (P_1 \setminus \{r\}) \cup \{\text{Head}(r) \leftarrow (\text{Body}(r) \setminus \{b\})\} \).

4. **Failure, \( \mathit{F} \).** Program \( P_2 \) results from \( P_1 \) by failure iff there is a rule \( r \in P_1 \) and a positive literal \( b \in \text{Body}(r) \) such that \( b \notin \text{Heads}(P_1) \), and \( P_2 = P_1 \setminus \{r\} \).

5. **Loop Detection, \( \mathit{L} \).** Program \( P_2 \) results from \( P_1 \) by loop detection iff there is a set \( A \) of ground atoms such that:

(a) For each rule \( r \in P_1 \), if \( \text{Head}(r) \in A \), then \( \text{Body}(r) \cap A = \emptyset \);
(b) \( P_2 \) := \( \{r \in P_1 \mid \text{Body}(r) \cap A = \emptyset\} \).

We represent this set of operations as \( \mapsto_{\mathit{WFS}} := \{\mathit{PR}, \mathit{NR}, \mathit{S}, \mathit{F}, \mathit{L}\} \). By applying non-deterministically this set of operations on a program \( P \), we obtain the program \( P \), named remainder of \( P \), which is invariant under a further application of any of the five operations. This transformation is terminating and confluent, meaning that for any finite ground program \( P \) the number of operations needed to reach \( P \) is finite, and the order in which the operations are performed is irrelevant. We denote the transformation of \( P \) into \( P \) as \( P \mapsto_{\mathit{WFS}} P \). We also write \( P = \text{remainder}_{\mathit{WFS}}(P) \). It is shown in (Brass et al. 2001) that \( WFM(P) = WFM(P) \), where \( WFM \) stands for the well-founded model (Gelder 1993). The next example shows this transformation at work.

**Example 1 Computing the remainder of a program.** Let \( P \) be the set of all rules below. The remainder \( P \) is the non shaded part of the program (the labels (i)–(v) indicate the operations used in the corresponding reductions, as pinpointed in the legend).

\[
\begin{align*}
a & \leftarrow \neg f \quad (i) \\
\mathbf{a} & \leftarrow \neg b \quad (i) \\
\mathbf{b} & \leftarrow \neg a \quad (ii) \\
c & \leftarrow \mathbf{a} \quad (iii) \\
d & \leftarrow f \quad (iv)
\end{align*}
\]

Legend: (i) \( \mathit{PR} \), (ii) \( \mathit{NR} \), (iii) \( \mathit{S} \), (iv) \( \mathit{F} \), (v) \( \mathit{L} \).

One way to obtain conservative extensions of the \( \mathit{SM} \) semantics, is to relax some operations of the reduction system \( \mapsto_{\mathit{WFS}} \), which yields weaker reduction systems, that is, systems that erase less rules or literals than \( \mapsto_{\mathit{WFS}} \).

Every semantics \( \mathit{SEM} \) of the family \( \mathit{ASM} \) is associated with such a reduction system, \( \mapsto_{\mathit{SEM}} \), which allows to write \( \mathit{SEM}(P) := \mathit{SEM}(P^*) \), where \( P \mapsto_{\mathit{SEM}} P^* \). We also write \( P^* = \text{remainder}_{\mathit{SEM}}(P) \). It is implicit that the reduction systems are both terminating and confluent. As an example, the \( \mathit{SM} \) semantics is associated with the reduction system \( \mapsto_{\mathit{WFS}} \), i.e., \( \mapsto_{\mathit{SM}} \Rightarrow \mapsto_{\mathit{WFS}} \), meaning that \( \mathit{SM}(P) = \mathit{SEM}(P) \).
Minimal Hypotheses Semantics

The minimal hypotheses semantics, MH (Pinto and Pereira 2011), is a semantics whose reduction system $\rightarrow_{WFS}$ is obtained from $\rightarrow_{WFS}$ by replacing the negative reduction operation, $NR$, by the layered negative reduction operation, $LNR$, i.e., $\rightarrow_{MH} = \{ PR, LNR, S, F, L \}$. LNR is a weaker version of $NR$ that instead of eliminating any rule $r$ containing say not $b$ in the body, in the presence of the fact $b$, as $NR$ does, only eliminates rule $r$ if this rule is not in loop through literal not $b$. We write $P \rightarrow_{MH} P'$, where $P'$ is the layered remainder of $P$. We also write $P = \text{remainder}_{MH}(P)$. In the case of program $P$ in the example above, the layered remainder $P'$ is the non shaded part of the program below.

$$
\begin{align*}
  a &\leftarrow \text{not } f \\
  a &\leftarrow \text{not } b \\
  b &\leftarrow \text{not } a \\
  c &\leftarrow a \\
  d &\leftarrow f
\end{align*}
$$

Notice that rule $r = (b \leftarrow \text{not } a)$ is no longer eliminated by the fact $a$, since rule $r$ and rule $a \leftarrow \text{not } b$ are in loop, and in the case of rule $r$ the loop is through the literal not $b$. The MH models of a program $P$ are then computed as follows: (1) Take as assumable hypotheses set, $Hyps(P)$, the set of all atoms that appear default negated in $P'$; in the case of the previous program we have $Hyps(P) = \{ a, b \}$; (2) Form all programs $P \cup H$, for all possible subsets $H \subseteq Hyps(P)$, $H \neq \emptyset$ (if $Hyps(P) = \emptyset$, then $H = \emptyset$ is the only set to consider); take all the interpretations for which $WF\text{M}(P \cup H)$ is a total model (meaning a model that has no undefined literals); $H$ is the hypotheses set of the interpretation $WF\text{M}(P \cup H)$; (3) Take all the interpretations obtained in the previous point, and chose as MH models the ones that have minimal $H$ sets with respect to set inclusion. The MH models of program $P$ in the example above, and the corresponding hypotheses sets, are

$$
\begin{align*}
  M_1 &= \{ a, \text{not } b, c, \text{not } d, \text{not } e, \text{not } f \} \\
  M_2 &= \{ a, b, c, \text{not } d, \text{not } e, \text{not } f \}
\end{align*}
$$

Notice that $M_1$ is the only SM model of $P$. The MH reduction system keeps some loops intact, which are used as choice devices for generating MH models, allowing us to have $MH(P) \supset SM(P)$. The sets $H$ considered may be taken as abductive explanations (Denecker and Kakas 2002) for the corresponding examples. It can be shown (Abrantes 2013) that the MH semantics solves problems that have no solutions by means of stable models.

**Definition 1 Affix Stable Interpretation.** Let $P$ be a normal logic program, $SEM$ a 2-valued semantics with a corresponding reduction system $\rightarrow_{SEM}$, and $X \subseteq Atoms(remainder_{SEM}(P))$. We say that $I$ is an affix stable interpretation of $P$ with respect to set $X$ and semantics $SEM$ (or simply a SEM stable interpretation with affix $X$) iff $I = WF\text{M}(P \cup X)$ and $WF\text{M}^\ast(P \cup X) = \emptyset$, that is, $I$ is the only stable model of the program $P \cup X$. We name $X$ an affix (or hypotheses set) of interpretation $I$. We also name assumable hypotheses set of program $P$, $Hyps(P)$, the union of all possible affixes that may be considered to define the stable interpretations ($Hyps(P) \subseteq Atoms(remainder_{SEM}(P))$).

The purpose of computing the remainder of a program is to obtain the assumable hypotheses set of the program.

**Definition 2 Affix Stable Model Semantics Family, ASM.** A 2-valued semantics $SEM$, with a corresponding reduction system $\rightarrow_{SEM}$, belongs to the affix stable model semantics family, $ASM$, iff, given any normal logic program $P$, $SEM(P)$ contains all the SM models of $P$, in case they exist, plus a subset (possibly empty) of the affix stable interpretations of $P$ chosen by resorting to specifically enounced criteria. An atom $b$ of the language of $P$ is true (false) under the semantics $SEM$ iff $b$ pertains to every (neither) model in $SEM(P)$; in other cases $b$ is undefined under the semantics $SEM$.

Both semantics $SM$ and $MH$ belong to the $ASM$ family. The two non-disjoint subfamilies of $ASM$ next defined, $ASM^b$ and $ASM^m$, will be the classes whose formal properties we study in the sequel.

**Definition 3 $ASM^b$ and $ASM^m$ Families.** A semantics $SEM \in ASM$ belongs to the $ASM^b$ or $ASM^m$ families iff, for any normal logic program $P$:

1. The set of assumable hypotheses, $Hyps(P)$, is contained in the set of atoms that appear default negated in $remainder_{SEM}(P)$;
2. In case $SEM \in ASM^b$, the affixes of the models of $P$ are either those non empty minimal with respect to set inclusion, if $Hyps(P) \neq \emptyset$, or the empty set if $Hyps(P) = \emptyset$; in case $SEM \in ASM^m$, the models in $SEM(P)$ are minimal in the classical sense.

The reasons to study this two particular classes of semantics are two fold: on the one hand they encompass a number of $MH$ semantics variations, and $MH$, as already pointed out, solves a larger number of problems than $SM$; on the other hand both classes contain the $SM$ semantics, whose characterization on the properties of existence, relevance and cumulativity will be clarified as the result of the study of these properties for the $ASM^b \cup ASM^m$ class. We now set forth some examples of $ASM^b$ and $ASM^m$ members.

Besides $SM$, $MH$ and others, the following are $ASM^b$ family members, referred to subsequently: 4

---

4The first three semantics were suggested by Alexandre Pinto.
MH\textsuperscript{LS}: the reduction system is obtained by replacing the success operation in $\rightarrow MH$ by the layered success operation;\textsuperscript{5} $MH\textsuperscript{LS}$ models are computed as in the $MH$ case.

MH\textsuperscript{Loop}: the reduction system is $\rightarrow MH$; the assumable hypotheses set of a program $P$, Hyps($P$), is formed by the atoms that appear default negated in literals involved in loops in the layered remainder $P$; $MH\textsuperscript{Loop}$ models are computed as in the $MH$ case.

MH\textsuperscript{Sustainable}: the reduction system is $\rightarrow MH$; $MH\textsuperscript{Sustainable}$ models are computed as in the $MH$ case with the following additional condition: if $H$ is a set of hypotheses of a $MH\textsuperscript{Sustainable}$ model $M$ of $P$, then
\[ \forall h \in H \left[ (H \setminus \{h\}) \neq \emptyset \Rightarrow h \in WFM^u(P \cup (H \setminus \{h\})) \right] , \]
that is, no single hypothesis may be defined in the well-founded model if we join to $P$ all the other remaining hypotheses.

MH\textsuperscript{Sustainable}\textsubscript{min}: the reduction system is $\rightarrow MH$; $MH\textsuperscript{Sustainable}\textsubscript{min}$ retrieves the minimal models contained in $MH\textsuperscript{Sustainable}$ for any normal logic program $P$. $MH\textsuperscript{Sustainable}\textsubscript{min}$ also belongs to the $ASM^m$ family, since all models it retrieves are minimal.

MH\textsuperscript{Regular}: the reduction system is $\rightarrow MH$; retrieves the same models as $MH$, except for the irregular ones (cf. Definition 6 below).

Besides $SM$ and others, the following are $ASM^m$ family members, referred to subsequently:

Navy: the reduction system is $\rightarrow WFS$. Given a normal logic program $P$, Navy($P$) contains all the minimal models of $P$.

Blue: the reduction system is $\rightarrow WFS$. Given a normal logic program $P$, Blue($P$) contains all the models in Navy($P \cup K$) where $K$ is obtained after terminating the following algorithm:\textsuperscript{6}
(a) Compute $K = \ker_{Navy}(\hat{P})$;
(b) Compute $K' = \ker_{Navy}(P \cup K)$;
(c) If $K \neq K'$, then let $P$ be the new designation of program $P \cup K'$; go to step (a). Repeat steps (a) – (c) until $K \neq K'$ comes false in (c).

Cyan: the reduction system is $\rightarrow WFS$. Given a normal logic program $P$, Cyan($P$) computes Cyan($P$) through the steps of Blue computation, but taking only the regular models (cf. Definition 6 below) to compute the semantic kernel at steps (a) and (b).

Green: the reduction system is $\rightarrow WFS$. Given a normal logic program $P$, Green($P$) contains all the minimal models of $\hat{P}$ that have the smallest (with respect to set inclusion) subsets of classically unsupported atoms.\textsuperscript{7}

\textsuperscript{5}Layered success is an operation proposed by Alexandre Pinto. It weakens the operation of success by allowing it to be performed only in the cases where the rule $r$, whose body contains the positive literal $b$ to be erased, is not involved in a loop through literal $b$.

\textsuperscript{6}This algorithm is presented in (Dix 1995a).

\textsuperscript{7}Given a logic program $P$, a model $M$ of $P$ and an atom $b \in M$, we say that $b$ is classically unsupported by $M$ if there is no rule $r \in P$ such that $Hcad(r) = \{b\}$ and all literals in $Body(r)$ are true with respect to $M$.  

4 Characterization of Cumulativity for the $ASM^h \cup ASM^m$ Class

In this section we lay down a characterization of cumulativity for semantics $SEM$ of the $ASM^h \cup ASM^m$ class.

Cautious monotony is characterized by proposition 1 and corollary 2 below.

**Proposition 1** Let $SEM$ be a semantics of the $ASM^h \cup ASM^m$ class. $SEM$ is not cautious monotonic iff, for some program $P$ and for some subset $S \subseteq ker_{SEM}(P)$, there is a 2-valued interpretation $M$ such that $M \in SEM(P \cup S)$ and $M \notin SEM(P)$.

The following corollary is an immediate consequence of proposition 1.

**Corollary 2** A semantics $SEM$ of the $ASM^h \cup ASM^m$ class is cautious monotonic iff, for every program $P$ and for every subset $S \subseteq ker_{SEM}(P)$, it is the case that $SEM(P \cup S) \subseteq SEM(P)$.

The definition of cautious monotony (see section 2) presupposes $SEM(P) \neq \emptyset$. Meanwhile, proposition 7, in section 5, states the failure of cautious monotony if existence fails for any semantics of the $ASM^h \cup ASM^m$ class. This means, considering the terminology used in the above results, cautious monotony fails if $SEM(P) = \emptyset$ or $SEM(P \cup S) = \emptyset$.

Cut is characterized by proposition 3 and corollary 4 below.

**Proposition 3** Let $SEM$ be a semantics of the $ASM^h \cup ASM^m$ class. $SEM$ is not cut iff, for some program $P$ and for some subset $S \subseteq ker_{SEM}(P)$, there is a 2-valued interpretation $M$ such that $M \in SEM(P)$ and $M \notin SEM(P \cup S)$.

The definition of cut (see section 2) presupposes $SEM(P \cup S) \neq \emptyset$. Meanwhile, it can be proved that cut fails for $SEM \in ASM^h \cup ASM^m$ if $SEM(P \cup S) = \emptyset$ and $SEM(P) \neq \emptyset$, for some program $P$ and some subset $S \subseteq ker_{SEM}(P)$. Notwithstanding, knowing that existence fails for some semantics $SEM$ (with no additional information on how this failure occurs), for a semantics of the $ASM^h \cup ASM^m$ class, is not enough to establish conclusions about cut – e.g., $MH$ is existential and is not cut, $Blue$ is existential and is cut, $SM$ is not existential and is cut, $MH\textsuperscript{Sustainable}$ is not existential and is not cut (these properties are proved in (Abrantes 2013)).

The following theorem stems directly from the above corollaries.
**Theorem 5** Let $SEM$ be a semantics of the $ASM^h \cup ASM^m$ class. Then $SEM$ is cumulative iff $SEM(P) = SEM(P \cup S)$, for every finite ground normal logic program $P$ and every subset $S \subseteq ker_{SEM}(P)$.

The general procedure to spot the failure of cumulativity by resorting to counter examples is the following (e.g. (Dix 1995a; 1995b)): compute all the $SEM$ models of a program $P$; add to $P$ subsets $S \subseteq ker_{SEM}(P)$, and compute all the models of the resulting programs $P \cup S$, drawing a conclusion about cumulativity failure only in cases where $ker_{SEM}(P) \neq ker_{SEM}(P \cup S)$. Meanwhile, corollaries 2, 4 and theorem 5 allow us to conclude that a semantics of $ASM^h$ or $ASM^m$ families is not cumulative by resorting to counter examples, even in some cases where the programs used to set counter examples do not show explicitly a failure of this property (i.e., cases where the use of the general procedure is inconclusive). To make this point clear, consider the following examples.

**Example 2** Let $P$ be the 1-layer program
\[
\begin{align*}
a & \leftarrow \text{not } b, \text{ not } s & d & \leftarrow b & d & \leftarrow a \\
b & \leftarrow \text{not } a, \text{ not } c & d & \leftarrow \text{not } d & c & \leftarrow k \\
c & \leftarrow \text{not } b, \text{not } k & k & \leftarrow a, d & s & \leftarrow \text{not } a, d
\end{align*}
\]
whose $SM$ models are \{$a, d, c, k$\} and \{$b, d, s$\}, and thus $ker_{SM}(P) = \{d\}$. Now $P \cup \{d\}$ has the stable models \{$a, d, c, k$\}, \{$b, d, s$\} and \{$c, d, s$\}, and thus $ker_{SM}(P) = ker_{SM}(P \cup \{d\}) = \{d\}$. Hence no negative conclusion can be afforded about cumulativity, by means of the usual general procedures. But using the result of theorem 5 it is straightforward to conclude that the $SM$ does not enjoy the property of cumulativity, because $SEM(P) \neq SEM(P \cup \{d\})$.

More specifically, corollary 2 tells us, via this example, that the $SM$ is not cautious monotonic.

**Example 3** The following 1-layer program $P = \hat{P}$ shows that none of the semantics $MH$, $MH^{LS}$, $MH^{Loop}$, $MH^{Sustainable}$ and $MH^{Regular}$ is either cautious monotonic or cut.
\[
\begin{align*}
u & \leftarrow b & a & \leftarrow \text{not } b \\
u & \leftarrow c & b & \leftarrow \text{not } c \\
t & \leftarrow a & c & \leftarrow h, u \\
t & \leftarrow h & h & \leftarrow \text{not } h, \text{not } t
\end{align*}
\]
Let $SEM$ represent any of these semantics. The minimal hypotheses models are the same with respect to any of the four semantics (models are represented considering only positive literals): \{c, u, a, t\} with affix \{c\}; \{b, h, u, c, t\} with affix \{b, h\}; \{t, b, u\} with affix \{t\}. Thus $ker_{SEM}(P) = \{t, u\}$. Now it is the case that the remainder of $P \cup \{u\}$ is the same for any of these semantics:
\[
\begin{align*}
u & \leftarrow b & a & \leftarrow \text{not } b \\
u & \leftarrow c & b & \leftarrow \text{not } c \\
t & \leftarrow a & c & \leftarrow h \\
t & \leftarrow h & h & \leftarrow \text{not } h, \text{not } t & u & \leftarrow
\end{align*}
\]
(as a matter of fact, the remainder for the $MH^{LS}$ has the rule $c \leftarrow h, u$ instead of $c \leftarrow h$; but this does not change the sequel of this reasoning). The minimal hypotheses models of $P \cup \{u\}$ are the same with respect to any of the four semantics (models are represented considering only positive literals): \{c, u, a, t\} with affix \{c\}; \{h, u, c, t, a\} with affix \{h\}; \{t, b, u\} with affix \{t\}. Thus $ker_{SEM}(P \cup \{u\}) = \{t, u\} = ker_{SEM}(P)$, and no conclusions about cumulativity can be drawn by means of the usual general procedures. Meanwhile $M = \{h, u, c, t\}$, with affix \{h\}, is a minimal affix model of $P \cup \{u\}$, but is not a minimal affix model of $P$, which by corollary 2 renders any of these semantics not cautious monotonic. Also $N = \{b, h, u, c, t\}$, with affix \{b, h\}, is a minimal affix model of $P$, but not a minimal affix model of $P \cup \{u\}$, which by corollary 4 renders any of these semantics as not cut.

It should be stressed that there are 2-valued cumulative semantics to which $SEM(P) \neq SEM(P \cup S)$ for some normal logic program $P$ and some $S \subseteq ker_{SEM}(P)$ – theorem 5 states this is not the case if $SEM$ belongs to the $ASM^h \cup ASM^m$ class. Consider, for instance, the semantics Picky defined as follows: for any normal logic program $P$, $SEM(P) \neq \emptyset$, Picky($P$) = $SEM(P)$ iff $ker_{SM}(P) = ker_{SM}(P \cup S)$, for every $S \subseteq ker_{SM}(P)$; otherwise Picky($P$) = \emptyset. This semantics is cumulative but it is not always the case that $SEM(P) = SEM(P \cup S)$: for program $P$ of example 2, we have Picky($P$) = \{\{a, d, c, k\}, \{b, d, s\}\} and Picky($P \cup \{d\}$) = \{\{a, d, c, k\}, \{b, d, s\}, \{c, d, s\}\}. Picky is not a $ASM$ semantics, because it does not conservatively extend the $SM$ semantics: for program $P$ in example 4 below, we have $SM(P) \neq \emptyset$ and Picky($P$) = \emptyset.

Theorem 5 application for dismissing the cumulativity property by means of counter examples, demands computing the set of models $SEM(P)$ of a program $P$, the set $ker_{SEM}(P)$, and after this it needs the computation of the set of models $SEM(P \cup S)$, $S \subseteq ker_{SEM}(P)$, to look for a case that eventually makes $SEM(P) = SEM(P \cup S)$ false. In the next section three structural properties are defined, defectivity, excessiveness and irregularity, that will turn the dismissal of existence, relevance or cumulativity spottable by means of one model only.

5 **Defectivity, Excessiveness and Irregularity**

In this section we define three new structural properties of 2-valued semantics, defectivity, excessiveness and irregularity, and show that for semantics of the $ASM^h \cup ASM^m$ class, defectivity is equivalent to the failure of existence and to the failure of global to local relevance, and also entails the failure of cautious monotony, whilst excessiveness entails the failure of cut, and irregularity is equivalent to the failure of local to global relevance. One of the virtues of these structural properties is that in some cases we do not even need to compute all the models of a program to spot their validity, hence providing a shortcut to detect the failure of existence, relevance or cumulativity, as we shall see in the sequel.
Defectivity

The rationale for the concept of defective semantics is the following: if a segment $P^{≤T}$ has a SEM model $M$ that is not contained in any whole model of $P$, then we say the semantics is defective, in the sense that it “does not use” all the models of segment $T$ in order to get whole models of $P$.

**Definition 4** Defective semantics. A 2-valued semantics SEM is called defective if there is a normal logic program $P$, $SEM(P) ≠ ∅$, a segment $P^{≤T}$ of $P$, and a SEM model $M$ of the segment $P^{≤T}$, such that $SEM(P^{>T}∪M^+) = ∅$. We also say that $SEM$ is defective with respect to segment $T$ of program $P$, and that $M$ is a defective model of $P$ with respect to segment $T$ and semantics $SEM$.

**Example 4** Program $P = \{a ← not b, b ← not a, c ← a, c ← not c\}$ may be used to show that the SEM semantics is defective. In fact, the only $SM$ model of $P$ is $N = \{not b, c\}$ with affix $\{a\}$, $P^{≤1} = \{a ← not b, b ← not a\}$ is a segment that has the stable model $M = \{not a, b\}$, and we have $SM(P^{>1}∪\{b\}) = ∅$.

It is clear from the above definition that defectivity spots the non existence property of a semantics. The next result shows a tight relation between existence and defectivity, for semantics of the $ASM^h ∪ ASM^m$ class.

**Proposition 6** Defectivity $⇔ ¬$ Existence. A semantics $SEM$ of the $ASM^h ∪ ASM^m$ class is defective iff it is non-existential.

This result allows to detect the failure of the existence property for semantics of the $ASM^h ∪ ASM^m$ class, by resorting to counter examples, even in some cases where the programs used as counter examples have a semantics. E.g., program $P$ in example 4 can be used to detect the failure of existence for $SM$ semantics, in spite of the existence of stable models for program $P$, since it reveals the defectivity of $SM$. Notice that there are 2-valued semantics for which this property fails, e.g., $M_P^{Supp}$ (Apt, Blair, and Walker 1988) which is not defective in spite of failing the existence property — it is the case that $M_P^{Supp}$ is a $ASM$ semantics, since it does not conservatively extend the $SM$ semantics.

The next result shows that for semantics of the $ASM^h ∪ ASM^m$ class, lack of existence implies the failure of cautious monotony.

**Proposition 7** $¬$ Existence $⇒ ¬$ Cautious monotony. If a semantics $SEM$ of the $ASM^h ∪ ASM^m$ class is not existential, then it is not cautious monotonic.

The above referred semantics, $M_P^{Supp} ∉ ASM$, fails this property.

The converse of proposition’s 7 statement, $¬$ Cautious Monotony $⇒ ¬$ Existence, is not valid for the $ASM^h ∪ ASM^m$ class. To see this is the case, notice that on the side of $ASM^h$ family the semantics $MH$ is existential (Pinto and Pereira 2011), although in example 3 it is shown that this semantics is not cautious monotonic; on the side of $ASM^m$ family, Green is not cautious monotonic, since it mimics the $SM$ behavior with respect to program in example 4, but is existential (Abrantes 2013).

In (Dix 1995b), section 5.6, the author says, about a program alike the one in example 4, that the $SM$ is not cumulative and that this fact does not depend on the non existence of stable models. Although this claim is true when considering the program in example 4, proposition 7 above shows that the $SM$ semantics could not be cautious monotonic since it is non-existential. Thus a relation between these properties in what concerns the $SM$, seems to be attainable only by considering the universe of the normal logic programs, in the vein of the approach taken in this paper. It appears not to come out on a basis of individual programs analysis.

Moreover, with respect to the $SM$ semantics a stronger result than the one in proposition 7 is stated below.

**Proposition 8** There is a program $P$ such that $SM(P) = ∅$ iff $SM$ cautious monotony fails for some program $P^*$.

To the best of our knowledge, this result had not yet been stated. The following corollary is immediate after propositions 6 and 7.

**Corollary 9** Defectivity $⇒ ¬$ Cautious monotony. If a semantics $SEM$ of the $ASM^h ∪ ASM^m$ class is defective, then it is not cautious monotonic.

The next two results show that defectivity is equivalent to the failure of global to local relevance, for any semantics of the $ASM^h ∪ ASM^m$ class. Hence, due to proposition 6, lack of existence immediately implies lack of relevance for semantics of this class.

**Proposition 10** Defectivity $⇒ ¬$ Global to local relevance. If a semantics $SEM$ of the $ASM^h ∪ ASM^m$ class is defective, then it fails the property of global to local relevance.

**Proposition 11** ¬ Global to local relevance $⇒$ Defectivity. If a semantics $SEM$ of the $ASM^h ∪ ASM^m$ class is not global to local relevant, then it is defective.

Proposition 10, together with example 4, show that $SM$ semantics is not global to local relevant and hence not relevant. The theorem below stems directly from the results presented above in this subsection.

**Theorem 12** The following relations are valid for any semantics of the $ASM^h ∪ ASM^m$ class:

$¬$Existence $⇔$ Defectivity $⇔ ¬$Global to Local Relevance

As a corollary of this theorem, we have the following implicit relation between relevance and cumulativity.

**Corollary 13** ¬ Global to local relevance $⇒ ¬$ Cautious monotony. If a semantics $SEM$ of the $ASM^h ∪ ASM^m$ class fails global to local relevance, then it also fails cautious monotony.
**Excessiveness and Irregularity**

The rationale of the concept of *excessive semantics* is the following: if a normal logic program $P$ has a model $N$ and a layer $P^{\leq T}$ such that for every model $M_* \in \text{SEM}(P^{\leq T})$ it is the case that $N \notin \text{SEM}(P^{\geq T} \cup M_*^+)$, then we say that model $N$ (and thus the semantics) is excessive, in the sense that it "goes beyond" the semantics of the segment $P^{\leq T}$ by not being a "consequence" of it. The concept of *irregularity* resembles excessiveness (see definition in the sequel), but they exhibit a kind of independence, meaning that they can both occur, both fail, or only one of them fail in semantics of the $ASM^h \cup ASM^m$ class.

**Definition 5 Excessive semantics.** A 2-valued semantics $SEM$ is called *excessive* iff there is a logic program $P$, a segment $T$ of $P$, a model $M \in \text{SEM}(P^{\leq T})$ and a model $N \in \text{SEM}(P)$ such that:

1. $M^+ = N^+_{\leq T}$, where $N^+_{\leq T} = N^+ \cap \text{Heads}(P^{\leq T})$;
2. For every model $M_* \in \text{SEM}(P^{\leq T})$ it is the case that $N \notin \text{SEM}(P^{\geq T} \cup M_*^+)$;
3. There is at last a $SEM$ model $N^*$ of $P$, such that $N^* \in \text{SEM}(P^{\geq T} \cup M^+)$.

We also say that $SEM$ is *excessive* with respect to segment $T$ of program $P$, and that $N$ is an *excessive model* of $P$ with respect to segment $T$ and semantics $SEM$.

**Example 5** The following program $P$ shows that semantics $MH$, $MH^{LS}$, $MH^{Loop}$, $Navy$ and $Green$ are excessive (the dashed lines divide the program into layers; top layer is layer 1, bottom layer is layer 4),

\[
\begin{align*}
a & \leftarrow \text{not } b \\
b & \leftarrow \text{not } a \\
& \quad -- -- -- -1 \\
u & \leftarrow a \\
u & \leftarrow b \\
p & \leftarrow \text{not } p, \text{not } u \\
& \quad -- -- -- -3 \\
q & \leftarrow \text{not } q, \text{not } p \\
& \quad -- -- -- -4.
\end{align*}
\]

In fact $N = \{a, u, p, \text{not } b, \text{not } q\}$, with affix $\{a, p\}$, is a model of $P$ under any of these semantics, and for no $SEM$ model $M_* \in \text{SEM}(P^{\leq 2}) = \{\{a, u, p, \text{not } b, \text{not } q\}\} \cup \{\{a, \text{not } b, u\}, \{\text{not } a, \text{not } b, u\}\}$, do we have $N \in \text{SEM}(P^{\geq 2} \cup M_*^+)$, because atom $u \in M_*^+$ eliminates the rule in layer 3 via layered negative reduction operation (which has here the same effect as negative reduction operation), and thus $p$ belongs to no model in $\text{SEM}(P^{\geq 2} \cup M_*^+)$. The rational of the concept of irregularity is the following: if a model $N \in \text{SEM}(P)$ contains no model of a segment $P^{\leq T}$, then we say that $SEM$ is irregular, since $N$ "is not a consequence" of the semantics of segment $T$.

**Definition 6 Irregular semantics.** A 2-valued semantics $SEM$ is called *irregular* iff there is a logic program $P$, a segment $P^{\leq T}$ of $P$ and a $SEM$ model $N$ of $P$, such that for no model $M$ of $P^{\leq T}$ do we have $N^*_+ = M^*$, where $N^+\leq T = N^+ \cap \text{Heads}(P^{\leq T})$. We also say that $SEM$ is *irregular* with respect to segment $T$ of program $P$, and that $N$ is an *irregular model* of $P$ with respect to segment $T$ and semantics $SEM$. A model that is not irregular is called *regular*, and a semantics that produces only regular models is called *regular*.

The concepts of excessiveness and irregularity exhibit independence for semantics of the $ASM^h \cup ASM^m$ class, meaning the existence of semantics in this class for any of the four possible cases of validity or failure of excessiveness and irregularity. As a matter of fact, it can be shown (Abrantes 2013) that *Blue* is irregular whilst not excessive (that is, *irregularity $\Rightarrow$ excessiveness*); it is also the case that $MH^\text{Regular}$ is excessive but not irregular (that is, *excessiveness $\nRightarrow$ irregularity*). Also $MH$ is excessive and irregular, and $Cyan$ is not excessive and is not irregular. The following result tells us that a stable model is neither excessive nor irregular, and hence $SM$ is a regular semantics.

**Proposition 14** Let $P$ be a normal logic program and $M \in SM(P)$. Then $M$ is neither excessive nor irregular.

The ensuing result shows that for any semantics of the $ASM^h \cup ASM^m$ class, excessiveness implies the failure of cut.

**Proposition 15 Excessiveness $\Rightarrow \neg$ Cut.** If a semantics $SEM$ of the $ASM^h \cup ASM^m$ class is excessive, then it is not cut.

This result, together with example 5, shows that semantics $MH$, $MH^{LS}$, $MH^{Loop}$, $Navy$ and $Green$ are not cut.

The converse of proposition’s 15 statement, $\neg$ Cut $\Rightarrow$ Excessiveness, is not valid for semantics of the $ASM^h \cup ASM^m$ class. For example, $MH^{\text{Sustainable}} \in ASM^h$ is not cut and is not excessive (Abrantes 2013). The same for the semantics $MH^{\text{Sustainable}} \in ASM^m$ (Abrantes 2013).

The result that follows states that irregularity is equivalent to the failure of local to global relevance, for semantics of the $ASM^h \cup ASM^m$ class.

**Proposition 16 Irregularity $\Leftrightarrow \neg$ Local to global relevance.** A semantics $SEM$ of the $ASM^h \cup ASM^m$ class is irregular if it is not local to global relevant.

The following result is a corollary of this proposition and of proposition 14.

**Corollary 17** $SM$ is (vacuously) local to global relevant.

Notice that this corollary, together with example 4 and propositions 10, 11, let clear the cause for $SM$ relevance failure: $SM$ fails relevance because it fails global to local relevance. This is a more precise characterization than just saying that $SM$ is not relevant, as usually stated in literature (e.g., (Dix 1995b)). It can be shown (Abrantes 2013) that semantics $MH$, $MH^{LS}$ and $MH^{Loop}$, $MH^{\text{Sustainable}}$, $Green$, $Navy$ and $Blue$ fail local to global relevance, and
are thus not relevant. The following theorem stems directly from the statements of propositions 15 and 16.

**Theorem 18** The following relations stand for any semantics of the $ASM^h \cup ASM^m$ class:

- Excessiveness $\Rightarrow$ ~Cut
- Irregularity $\Leftrightarrow$ ~Local to Global Relevance.

It should be noticed that for semantics of the $ASM^h \cup ASM^m$ class:

- Defectivity and cut are unrelated properties: $MH_{sustainable}$ is defective and is not cut; $MH$ is not defective and is not cut; Blue is not defective and is cut; $SM$ is defective and is cut.
- Excessiveness and cautious monotony are unrelated properties: $SM$ is not excessive and is not cautious monotonic; $MH$ is excessive and is not cautious monotonic; $Navy$ is excessive and is cautious monotonic; Blue is not excessive and is cautious monotonic.
- Irregularity and cautious monotony are unrelated properties: $MH$ is irregular and is not cautious monotonic; $SM$ is not irregular and is not cautious monotonic; $Navy$ is irregular and is cautious monotonic; $Cyan$ is not irregular and is cautious monotonic.
- Irregularity and cut are unrelated properties: $MH$ is irregular and is not cut; Blue is irregular and is cut; $SM$ is not irregular and is cut; $MH_{Regular}$ is not irregular and is not cut.

6 Final Remarks

We have presented in this paper a study on formal properties of 2-valued conservative extensions of the $SM$ semantics, under a structural point of view. For that purpose we focused on two subsets of $ASM$, the non-disjoint classes $ASM^h$ and $ASM^m$. Our intention is to characterize semantics of these two classes on the properties of existence, relevance and cumulativity. This was fulfilled by refining the usual definition of cumulativity, and also by analyzing the decomposition of models with respect to the layers of the programs. This approach reveals itself advantageous on various aspects, when compared to common processes for characterizing semantics on these properties, as we have shown in this paper: (i) It furnishes a larger universe of normal logic programs that may be used as counter examples to spot the failure of one or more of these properties – e.g., corollaries 2, 4 and theorems 5, 12 and 18; (ii) It reveals relations among those three properties that allow to draw conclusions about some of them on basis of held knowledge about others – e.g., theorems 12 and 18. This last point builds on top of the new structural properties of defectivity, excessiveness and irregularity, that in certain cases require the computation of only one model of a program to draw conclusions about the above referred formal properties over the universe of normal logic programs, providing an alternative to the sometimes harsh path of direct proofs.

If we consider the five formal properties of existence ($\exists$), global to local relevance ($gl$), local to global relevance ($lg$), cautious monotony ($cm$) and cut ($cut$), the validity or failure of each of these properties allow, in the general case, the existence of $2^5 = 32$ types of semantics. Meanwhile, the study we present in this work shows that only 12 such types of semantics may exist in the $ASM^h \cup ASM^m$ class. They are represented in table 1, where ‘0’ flags the failure of a property, and ‘1’ means the property is verified. The 20 missing types of semantics correspond to cases where $(\exists = 0$ and $gl = 1)$, or $(\exists = 1$ and $gl = 0)$, or $(\exists = 0$ and $cm = 1)$. Each of these cases going against the statement of theorem 12. The correspondence of the $ASM^h \cup ASM^m$ class semantics presented in this text and the entries in table 1 is as follows: 1. $MH_{sustainable}$, $MH_{sustainable}$ 2. $MH_{min}$, $MH_{max}$ 3. $MH_{Reg}$, $MH_{Loops}$, $Green$ 4. $Blue$ 5. $MH_{Loops}$, $MH_{Reg}$, $Green$ 6. $Cyan$ 7. $Blue$. Whether semantics of the $ASM^h \cup ASM^m$ class exist for the types marked with ‘~’ may be envisaged as an open issue.

### Table 1: The 12 possible types of $ASM^h$ and $ASM^m$ semantics

<table>
<thead>
<tr>
<th>$\exists$</th>
<th>$gl$</th>
<th>$lg$</th>
<th>$cm$</th>
<th>$cut$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Our research reveals a ‘syntactic’ nature of the studied formal properties, and clearly opens the possibility to tackle from a syntactic perspective other formal properties of semantics of logic programs. It also proffers a path of investigation to extend/redefine the properties of defectivity, excessiveness and irregularity to $n$-valued logic program semantics, $n \neq 2$.

### Acknowledgments

We thank Alexandre Pinto for some important debates on conservative extensions to $SM$ semantics. The work on this paper has been partially supported by Fundação para a Ciência e Tecnologia and Instituto Politécnico de Bragança grant PROTEC : SFHR/49747/2009.

### References


deductive databases and logic programming, 89–142. Los Altos, CA: Morgan Kaufmann.


