

Properties of Stable Model Semantics Extensions

Mário Abrantes

*Departamento de Matemática, Escola Superior de Tecnologia e de Gestão,
Instituto Politécnico de Bragança, 5300-253 Bragança, Portugal*

Luís Moniz Pereira

*Centro de Inteligência Artificial (CENTRIA), Departamento de Informática,
Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal*

submitted 1 January 2003; revised 1 January 2003; accepted 1 January 2003

Abstract

The stable model (SM) semantics lacks the properties of existence, relevance and cumulativity. If we prospectively consider the class of conservative extensions of the SM semantics (i.e., semantics that for each normal logic program P retrieve a superset of the set of stable models of P), one may wonder how do the semantics of this class behave in what concerns the aforementioned properties. That is the type of issue dealt with in this paper. We define a large class of conservative extensions of the SM semantics, dubbed affix stable model semantics (ASM), and study the above referred properties into two non-disjoint subfamilies of the class ASM, here dubbed ASM^h and ASM^m . From this study a number of results stem which facilitate the assessment of semantics in the class $ASM^h \cup ASM^m$ with respect to the properties of existence, relevance and cumulativity, whilst unveiling relations among these properties. As a result of the approach taken in our work, light is shed on the characterization of the SM semantics, as we show that 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. We also characterize the relevance failure of SM semantics in a more clear way than usually stated 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 (Gelfond and Lifschitz 1988) 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). Model *existence* guarantees that every normal logic program has a semantics. This is important to allow arbitrary updates and/or merges involving Knowledge Bases, possibly from different authors or sources (Pinto and Pereira 2011). *Relevance* allows for top-down query solving without the need to always compute complete models, but just the sub-models that sustain the answer to a query, though guaranteed extendable to whole ones (Pinto and Pereira 2011). As for *cumulativity*, it allows the programmer to take advantage of tabling techniques (Swift 1999) for speeding up computations (Pinto and Pereira 2011). Independently of the motivations that underlay the design of a semantics for logic programs, one may ask if it is easy to guarantee some or all of

the above properties, or even if it is easy to assess the profile of the resulting semantics in what concerns these properties. In this work we define a family of 2-valued conservative extensions of the *SM* semantics, the *affix stable model* semantics family, *ASM*. We then take two subclasses, $ASM^h \subset ASM$ and $ASM^m \subset ASM$, and present a number of results that simplify the task of assessing the semantics in $ASM^h \cup ASM^m$ on the properties of existence, relevance and cumulativity. The semantics in these two classes bear resemblance with the already known *SM* and *MH* semantics (see section 3), and this stands for the motivation to consider them. The following results, obtained in this work, should be emphasized: (1) We present a refined definition of cumulativity for semantics in the class $ASM^h \cup ASM^m$, which 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 $ASM^h \cup ASM^m$ class, and at the same time facilitate the assessment of semantics in this class with respect to those properties; (3) As a result of the approach in our work light is shed on the characterization of *SM* semantics, as we show that 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; we also characterize the relevance failure of *SM* semantics in a more clear way than usually stated in the literature. It should be stressed that this study is on the properties of a class of 2-valued semantics, under a prospection motivation. The weighing of such semantics rationales under an ‘intuitive’ point of view (or any other equivalently non-objective concept) is beyond the reach of our study. 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. In section 4 we characterize the property of cumulativity for the families ASM^h and ASM^m , whilst in section 5 the properties of defectivity, excessiveness and irregularity are defined. Some relations among existence, relevance and cumulativity, which are revealed by means of those 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 \mathcal{L} is a set of *normal rules*, each of the form

$$b_0 \leftarrow b_1, \dots, b_m, \text{not } c_1, \dots, \text{not } c_n \quad (1)$$

where m, n are non-negative integers and b_j, c_k are *atoms* of \mathcal{L} ; b_i and *not* c_k are generically designated *literals*, *not* c_k being specifically designated *default literal*. The operator ‘,’ stands for the conjunctive connective, the operator ‘*not*’ stands for negation by default and the operator ‘ \leftarrow ’ stands for a *dependency* operator that establishes a dependence of b_0 on the conjunction on the right side of ‘ \leftarrow ’. b_0 is the *head* of the rule and $b_1, \dots, b_m, \text{not } c_1, \dots, \text{not } c_n$ 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 ,

H_P . A program is *finite* if it has a finite number of rules¹. 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.

Given a 2-valued interpretation I of a logic program P , we represent by I^+ (resp. I^-) the set of its positive literals (resp. atoms whose default negations are true with respect to I). If I is 3-valued, we additionally represent by I^u the set of undefined atoms with respect to I .

The following concepts concern the structure of programs. Let P be a logic program and r, s any two rules of P . **Complete rule graph, $CRG(P)$** ²: 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 $Head(r) \subseteq Atoms(Body(s))$. **Rule depending on a rule**²: rule s *depends* on rule r iff there is a directed path in $CRG(P)$ from r to s . **Subprogram relevant to an atom**³: a rule $r \in P$ is *relevant* to an atom $a \in H_P$ 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* . **Loop**⁴: 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 $Head(s) = Atoms(L)$. **Rule layering**³: 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 that label, and to the label itself. We represent by $P^{\leq T}$ (resp. $P^{> T}$) the set of all rules of P whose layer is less than or equal to (resp. greater than) T . **T-segment of a program**: we say that $P^{\leq T}$ is the *T-segment* of P iff $Atoms(P^{\leq T}) \cap Heads(P^{> T}) = \emptyset$. We may also say ‘*segment T*’ to mean the set of rules corresponding to segment $P^{\leq T}$.

Let SEM be a 2-valued semantics and $SEM(P)$ the set of SEM models of a logic program P . Let also the set of atoms $ker_{SEM}(P) = \bigcap_{M \in SEM(P)} M^+$ be dubbed *semantic kernel* of P with respect to SEM (the semantic kernel is not defined if $SEM(P) = \emptyset$). The following properties concern semantics of logic programs. We say that a semantics SEM is: **Existential** iff every normal logic

¹ In 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.

² Adapted from (Pinto and Pereira 2011).

³ Adapted from (Dix 1995b)

⁴ Adapted from (Costantini 1995)

program has at least one *SEM* model; **Cautious monotonic**⁵ iff for every normal logic program P , and for every set $S \subseteq \text{ker}_{SEM}(P)$, we have $\text{ker}_{SEM}(P) \subseteq \text{ker}_{SEM}(P \cup S)$; **Cut** iff for every normal logic program P , and for every set $S \subseteq \text{ker}_{SEM}(P)$, we have $\text{ker}_{SEM}(P \cup S) \subseteq \text{ker}_{SEM}(P)$; **Cumulative** iff it is cautious monotonic and cut; **Relevant** iff for every normal logic program P we have

$$\forall_{a \in H_P} (a \in \text{ker}_{SEM}(P) \Leftrightarrow a \in \text{ker}_{SEM}(\text{Rel}_P(a))) \quad (2)$$

where $\text{Rel}_P(a)$ is the subprogram of P relevant to atom a ; **Global to local relevant** iff the logical entailment ' \Rightarrow ' stands in formula (2); **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 semantics⁶, the *affix stable model* family, *ASM*, whose members are conservative extensions of the *SM* semantics. For that purpose we need the concepts of *reduction system* and *MH semantics*.

3.1 Reduction System and MH Semantics

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, PR, negative reduction, NR, success, S, failure, F and loop detection, L (see Appendix A for the definitions of these operations). We represent this set of operations as $\mapsto_{WFS} := \{PR, NR, S, F, L\}$. By non-deterministically applying this set of operations on a program P , we obtain the program \hat{P} , the *remainder* of P , which is invariant under a further application of any of the five operations. This transformation is *terminating* and *confluent* (Brass et al. 2001). We denote the transformation of P into \hat{P} as $P \mapsto_{WFS} \hat{P}$. We also write $\hat{P} = \text{remainder}_{WFS}(P)$. It is shown in (Brass et al. 2001) that $WFM(P) = WFM(\hat{P})$, where *WFM* stands for the well-founded model (Gelder 1993). See Appendix B for an example of the computation of the remainder of a program.

One way to obtain conservative extensions of the *SM* semantics, is to relax some operations of the reduction system \mapsto_{WFS} , which yields weaker reduction systems, that is, systems that erase less rules or literals than \mapsto_{WFS} . An example of such a semantics is the *minimal hypotheses semantics*, *MH* (Pinto and Pereira 2011), whose reduction system \mapsto_{MH} is obtained from \mapsto_{WFS} by replacing the negative reduction operation, *NR*, by the *layered negative reduction* operation, *LNR*, i.e., $\mapsto_{MH} := \{PR, LNR, S, F, L\}$. *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 $P \mapsto_{MH} \hat{P}$, where \hat{P} is the *layered remainder* of P . We also write $\hat{P} = \text{remainder}_{MH}(P)$. See Appendix C for an example of the computation of the layered remainder of a program.

3.2 ASM, ASM^h and ASM^m Families

We define *affix stable interpretation* and then use this concept to put forward the definition of *ASM* family.

⁵ Adapted from (Dix 1995a; Dix 1995b).

⁶ See (Denecker and Kakas 2002) for *abductive semantics*.

Definition 1

Affix Stable Interpretation. Let P be a normal logic program, SEM a 2-valued semantics with a corresponding reduction system \mapsto_{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 $I = WFM(P \cup X)$ and $WFM^u(P \cup X) = \emptyset$,⁷ 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 (we have $Hyps(P) \subseteq Atoms(remainder_{SEM}(P))$).

Definition 2

Affix Stable Model Semantics Family, ASM. A 2-valued semantics SEM , with a corresponding reduction system \mapsto_{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.

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

Definition 3

ASM^h and ASM^m Families. A semantics $SEM \in ASM$ belongs to the ASM^h or ASM^m families iff, for any normal logic program P , the models are computed as follows:

1. For both ASM^h and ASM^m the set of *assumable hypotheses*, $Hyps(P)$, is contained in the set of atoms that appear default negated in $remainder_{SEM}(P)$ ⁸;
2. For semantics in the class ASM^h , the affixes of the models of P are either those non empty minimal with respect to set inclusion, if $Hyps(P) \neq \emptyset$, or else the empty set if $Hyps(P) = \emptyset$. For semantics in the class ASM^m , the models in $SEM(P)$ are always minimal models.

We now refer some examples of ASM^h and ASM^m members, whose definitions can be found in Appendix D. Besides *SM*, *MH* and others, the following are ASM^h family members, referred to subsequently:⁹ MH^{LS} , MH^{Loop} , $MH^{Sustainable}$, $MH_{min}^{Sustainable}$, $MH^{Regular}$. Besides *SM* and others, the following are ASM^m family members, referred to subsequently: **Navy**, **Blue**, **Cyan**, **Green**.

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, via the following theorem.

⁷ Notice that $WFM^u(P \cup X)$ is the set of undefined atoms in the model $WFM(P \cup X)$.

⁸ The purpose of computing the remainder of a program, is to obtain the assumable hypotheses set of the program.

⁹ The first three semantics were suggested by Alexandre Pinto.

Theorem 1

Let SEM be a semantics of the $ASM^h \cup ASM^m$ class. For every program P and for every subset $S \subseteq ker_{SEM}(P)$, the following results stand: (1) SEM is cautious monotonic iff $SEM(P \cup S) \subseteq SEM(P)$; (2) SEM is cut iff $SEM(P) \subseteq SEM(P \cup S)$; (3) SEM is cumulative iff $SEM(P) = SEM(P \cup S)$ – this is a consequence of statements (1) and (2).¹⁰

The three items of this theorem correspond to refinements of the classical definitions of cautious monotony, cut and cumulativity (see section 2). The new definitions establish the properties by means of relations among sets of models, as opposed to the relations among sets of atoms that characterize the classical definitions.

The results stated in this theorem are advantageous to spot cumulativity failure in semantics of the $ASM^h \cup ASM^m$ class by means of counter-examples (logic programs), when compared with common procedures (e.g., (Dix 1995a; Dix 1995b)). The reason is that common procedures always need the counter-examples to fail cumulativity¹¹, whilst the results of theorem 1 allow us to spot failure of cumulativity even in some cases where the counter-examples used do not show any failure of this property. To make this point clear see the examples in Appendix E and Appendix F.

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 $S \subseteq ker_{SEM}(P)$ (for an example, see the definition of the 2-valued semantics *Picky* in Appendix G). Theorem 1 states this is not the case if $SEM \in ASM^h \cup ASM^m$.

5 Defectivity, Excessiveness and Irregularity

Theorem 1 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 sets of models $SEM(P \cup S)$, $S \in ker_{SEM}(P)$, to look for a case that eventually makes $SEM(P) = SEM(P \cup S)$ false. In this 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. It will be shown 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.

5.1 Defectivity

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

¹⁰ Notice that $SEM(P)$ represents the set of all SEM models of P .

¹¹ The general procedure to spot the failure of cumulativity by resorting to counter-examples is as follows: 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 $ker_{SEM}(P) \neq ker_{SEM}(P \cup S)$.

Definition 4

Defective semantics. A 2-valued semantics SEM is called *defective* iff there is a normal logic program P , $SEM(P) \neq \emptyset$, a segment $P^{\leq T}$ of P , and a SEM model M of the segment $P^{\leq T}$, such that $SEM(P^{>T} \cup M^+) = \emptyset$. 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 1

Program $P = \{a \leftarrow \text{not } b, b \leftarrow \text{not } a, c \leftarrow a, c \leftarrow \text{not } c\}$ may be used to show that the SM semantics is defective. In fact, the only SM model of P is $N = \{a, \text{not } b, c\}$ with affix $\{a\}$. Meanwhile, $P^{\leq 1} = \{a \leftarrow \text{not } b, b \leftarrow \text{not } a\}$ is a segment that has the stable model $M = \{\text{not } a, b\}$, and we have $SM(P^{>1} \cup \{b\}) = \emptyset$.

The next theorem shows how conclusions about existence, relevance and cumulativity may be immediately taken in the case of a defective semantics.

Theorem 2

The following relations are valid for any semantics of the $ASM^h \cup ASM^m$ class:

1. Defectivity $\Leftrightarrow \neg$ Existence $\Leftrightarrow \neg$ Global to Local Relevance;
2. Defectivity $\Rightarrow \neg$ Cautious Monotony.

The reader should notice the importance of this theorem: not only defectivity is enough to dismiss existence, relevance and cumulativity, as also these properties appear strongly related for semantics of the class $ASM^h \cup ASM^m$: if existence fails then relevance also fails (through global to local relevance failure); if existence fails then cumulativity also fails (through cautious monotony failure); if relevance fails (through global to local relevance failure), then cumulativity also fails (through cautious monotony failure). Definition 4 above shows the structural nature of defectivity, which allows the verification of the property by wisely constructing a program that satisfies it. This may turn easier the assessment of existence, relevance and cumulativity, when compared to dealing with this issue on the basis of abstract proofs. Even more, the relation between existence and defectivity stated in theorem 2, allows the failure of the existence property to be detected by resorting to counter-examples, even in some cases where the program used as counter-example has models. E.g., program P in Appendix E 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 .¹²

The results stated in theorem 2 also shed some light on the characterization of SM semantics with respect to the properties of existence and cumulativity. In (Dix 1995b), section 5.6, the author says that the SM is not cumulative and that this fact does not depend on the non existence of stable models (i.e., the author states that lack of cumulativity is not a consequence of lack of existence). Meanwhile theorem 2 above shows that SM is non-existential due to being defective, which in turn makes it not cautious monotonic and thus not cumulative. Thus the failure of cumulativity for the SM semantics case is indeed a consequence of the failure of existence for this semantics. Moreover, with respect to the SM semantics a stronger result relating existence

¹² It should be pointed out that there are 2-valued semantics for which the equivalence *defectivity* $\Leftrightarrow \neg$ *existence* fails, e.g., M_P^{Supp} (Apt et al. 1988) which is not defective in spite of failing the existence property – it is the case that M_P^{Supp} is not a ASM semantics, since it does not conservatively extend the SM semantics.

and cautious monotony may be enounced: these two properties show up equivalence in the sense stated in proposition 3 below. To the best of our knowledge, this connection between these two properties had not yet been stated.

Proposition 3

For the *SM* semantics the following result stands: there is a program P that shows existence failure iff there is a program P^* that shows cautious monotony failure.

5.2 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 $M_* \in SEM(P^{\leq T})$ it is the case that $N \notin SEM(P^{>T} \cup M_*^+)$, 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.

Definition 5

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

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

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 .

In the excessiveness example in Appendix H it is shown that the semantics $MH, MH^{LS}, MH^{Loop}, Navy, Green$ are excessive.

The rationale of the concept of *irregularity* is as follows: given a certain whole model $N \in SEM(P)$, if the set $N^+ \cap Heads(P^{\leq T})$ is not a 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 $P^{\leq T}$ and a SEM model N of P , such that for no model M of $P^{\leq T}$ do we have $N_{\leq T}^+ = M^+$, where $N_{\leq T}^+ = N^+ \cap Heads(P^{\leq T})$. 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*.¹³

The concepts of excessiveness and irregularity exhibit independence for semantics of the $ASM^h \cup ASM^m$ class, meaning there is a 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 (i.e., *irregularity* $\not\Rightarrow$ *excessiveness*); it is also the case that $MH^{Regular}$ is excessive but not irregular (i.e., *excessiveness* $\not\Rightarrow$ *irregularity*).

¹³ In comparing excessiveness and irregularity, notice that a whole model can be excessive whilst containing models for all the segments of the program (i.e., be a regular model) - see the excessiveness example in Appendix H.

Also *MH* is excessive and irregular, and *Cyan* is not excessive and is not irregular.

The following result states relations between excessiveness and cut, and between irregularity and relevance.

Theorem 4

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

1. Excessiveness $\Rightarrow \neg$ Cut;
2. Irregularity $\Leftrightarrow \neg$ Local to Global Relevance.

As excessiveness and irregularity are structural properties, being thus detectable by construction of adequate programs, they facilitate, via this theorem, the dismissal of cut and relevance. For instance, this result together with the excessiveness example in Appendix H, shows that semantics *MH*, MH^{LS} , MH^{Loop} , *Navy* and *Green* are excessive, and thus not cut. Also, this result together with the irregularity example in Appendix H, shows that semantics *MH*, MH^{LS} and MH^{Loop} , *Green*, *Navy* and *Blue* are irregular, and thus not relevant. As was the case for the relation between the properties of existence and cumulativity for the *SM* semantics, our work sheds also some light on the *SM* semantics relevance failure, through the following results.

Proposition 5

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

Corollary 6

SM is (vacuously) local to global relevant.

Notice that this corollary, together with the example in Appendix E and theorem 2, let clear the cause for *SM* semantics 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)).

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 I 1 in Appendix I.

6 Final Remarks

In this paper we considered the characterization of 2-valued conservative extensions of the *SM* semantics on the properties of existence, relevance and cumulativity. This theoretical endeavor is reasonable under a point of view of prospectively assessing the behavior of such types of semantics with respect to a set of properties that are desirable, both under a computational (relevance and cumulativity) and a semantical (existence) standpoint. For that purpose we focused our study on two subsets of the here defined *ASM* class of 2-valued conservative extensions of the *SM* semantics, the non-disjoint classes ASM^h and ASM^m , whose elements maintain a degree of resemblance with already known 2-valued semantics, such as the *SM* and the *MH* semantics. As a result of this study, refined definitions of cautious monotony, cut and cumulativity were set.

This new definitions turn into an easier job the dismissal of the properties of existence, relevance and cumulativity, as shown in section 4. This study also reveals relations among these properties, unveiled by theorems 2 and 4, that allow to draw conclusions about some of them on basis of held knowledge about others. This last point builds on top of the new structural properties of defectivity, excessiveness and irregularity, which provide an analytical shortcut to assess existence, relevance and cumulativity. The approach taken in this work (characterizing families of semantics, instead of individual semantics), revealed itself advantageous also in clarifying the profile of the well known and studied *SM* semantics, via the results stated in proposition 3 and corollary 6. Our work also states a maximum of 12 types of semantics in the class $ASM^h \cup ASM^m$, with respect to the satisfaction/failure of the properties of existence (\exists), global to local relevance (*gl*), local to global relevance (*lg*), cautious monotony (*cm*) and cut (*cut*).

Finally, the structural approach put forward in this paper has the potential of being used with semantics other than 2-valued ones, and with other strong and weak properties besides existence, relevance or cumulativity.¹⁴

Acknowledgments

We thank Alexandre Pinto for some important debates on conservative extensions of the *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

- ABRANTES, M. 2013. Revision based total semantics for extended normal logic programs. Ph.D. thesis, Universidade Nova de Lisboa.
- APT, K., BLAIR, H. A., AND WALKER, A. 1988. Towards a theory of declarative knowledge. In *Foundations of deductive databases and logic programming*, J. Minker, Ed. Morgan Kaufmann, Los Altos, CA, 89–142.
- BRASS, S., DIX, J., FREITAG, B., AND ZUKOWSKI, U. 2001. Transformation-based bottom-up computation of the well-founded model. *TPLP*, 497–538.
- COSTANTINI, S. 1995. Contributions to the stable model semantics of logic programs with negation. *Theoretical Computer Science* 149, 2 (2 Oct.), 231–255.
- DENECKER, M. AND KAKAS, A. C. 2002. Abduction in logic programming. In *Computational Logic: Logic Programming and Beyond'02*. 402–436.
- DIX, J. 1995a. A classification theory of semantics of normal logic programs: I. strong properties. *Fundam. Inform* 22, 3, 227–255.
- DIX, J. 1995b. A classification theory of semantics of normal logic programs: II. weak properties. *Fundam. Inform* 22, 3, 257–288.
- GELDER, A. V. 1993. The alternating fixpoint of logic programs with negation. *J. of Comp. System Sciences* 47, 1, 185–221.
- GELFOND, M. AND LIFSCHITZ, V. 1988. The stable model semantics for logic programming. In *ICLP/SLP*. MIT Press, 1070–1080.
- PINTO, A. M. AND PEREIRA, L. M. 2011. Each normal logic program has a 2-valued minimal hypotheses semantics. *INAP 2011, CoRR abs/1108.5766*.
- SWIFT, T. 1999. Tabling for non-monotonic programming. *Ann. Math. Artif. Intell* 25, 3-4, 201–240.

¹⁴ The terms *strong* and *weak* applied to formal properties, are here adopted after (Dix 1995a; Dix 1995b).

Appendix A Reduction Operations

In the definitions below, P_1 and P_2 are two ground logic programs.

1. **Positive reduction, PR.** Program P_2 results from P_1 by *positive reduction* iff there is a rule $r \in P_1$ and a default literal $not\ b \in Body(r)$ such that $b \notin Heads(P_1)$, and $P_2 = (P_1 \setminus \{r\}) \cup \{Head(r) \leftarrow (Body(r) \setminus \{not\ b\})\}$.
2. **Negative reduction, NR.** Program P_2 results from P_1 by *negative reduction* iff there is a rule $r \in P_1$ and a default literal $not\ b \in Body(r)$ such that $b \in Facts(P_1)$, and $P_2 = P_1 \setminus \{r\}$.
3. **Success, S.** Program P_2 results from P_1 by *success* iff there is a rule $r \in P_1$ and a fact $b \in Facts(P_1)$ such that $b \in Body(r)$, and $P_2 = (P_1 \setminus \{r\}) \cup \{Head(r) \leftarrow (Body(r) \setminus \{b\})\}$.
4. **Failure, F.** Program P_2 results from P_1 by *failure* iff there is a rule $r \in P_1$ and a positive literal $b \in Body(r)$ such that $b \notin Heads(P_1)$, and $P_2 = P_1 \setminus \{r\}$.
5. **Loop Detection, L.** Program P_2 results from P_1 by *loop detection* iff there is a set \mathcal{A} of ground atoms such that:
 - (a) For each rule $r \in P_1$, if $Head(r) \in \mathcal{A}$, then $Body(r) \cap \mathcal{A} \neq \emptyset$;
 - (b) $P_2 := \{r \in P_1 \mid Body(r) \cap \mathcal{A} = \emptyset\}$.

Appendix B Remainder Computation Example

Let P be the set of all rules below. The remainder \hat{P} is the non shadowed part of the program. The labels (i)–(v) indicate the operations used in the corresponding reductions: (i) PR, (ii) NR, (iii) S, (iv) F, (v) L.

$\{a \leftarrow not\ f\ (i),\ e \leftarrow d\ (v),\ a \leftarrow not\ b\ (i),\ d \leftarrow e\ (v),\ b \leftarrow not\ a\ (ii),\ c \leftarrow a\ (iii),\ d \leftarrow f\ (iv)\}$

Appendix C Minimal Hypotheses Models Computation

Let P be the set of rules below, which is equal to the program in Appendix B. The layered remainder \hat{P} is the non-shadowed part of the program.

$$\begin{array}{ll} a \leftarrow not\ f & d \leftarrow f \\ a \leftarrow not\ b & e \leftarrow d \\ b \leftarrow not\ a & d \leftarrow e \\ c \leftarrow a & \end{array}$$

Notice that rule $b \leftarrow not\ a$ is no longer eliminated by the fact a , since this rule and rule $a \leftarrow not\ b$ are in loop, and in the case of rule $b \leftarrow not\ a$ the loop is through the literal $not\ a$.

The *MH* models of a program P are computed as follows: (1) Take as assumable hypotheses set, $Hyps(P)$, the set of all atoms that appear default negated in \hat{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 $H \subseteq Hyps$, $H \neq \emptyset$ (if $Hyps = \emptyset$, then $H = \emptyset$ is the only set to consider); take all the interpretations for which $WFM(P \cup H)$ is a total model (meaning a model that has no undefined literals); H is the *hypotheses set* of the interpretation $WFM(P \cup H)$; (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

$$\begin{aligned} M_1 &= \{a, \text{not } b, c, \text{not } d, \text{not } e, \text{not } f\} & H &= \{a\} \\ M_2 &= \{a, b, c, \text{not } d, \text{not } e, \text{not } f\} & H &= \{b\}. \end{aligned}$$

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 $MH(P) \supseteq SM(P)$. The sets H considered may be taken as abductive explanations (Denecker and Kakas 2002) for the corresponding models.

Appendix D Definitions of some Elements of ASM^h and ASM^m Families

Besides SM and others, the following are ASM^h family members.

MH^{LS}: the reduction system is obtained by replacing the success operation in \mapsto_{MH} by the *layered success* operation;¹⁵ MH^{LS} models are computed as in the MH case.

MH^{Loop}: the reduction system is \mapsto_{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 \hat{P} ; MH^{Loop} models are computed as in the MH case.

MH^{Sustainable}: the reduction system is \mapsto_{MH} ; $MH^{Sustainable}$ models are computed as in the MH case with the following additional condition: if H is a set of hypotheses of a $MH^{Sustainable}$ model M of P , then

$$\forall h \in H [(H \setminus \{h\}) \neq \emptyset \Rightarrow h \in WFM^u(P \cup (H \setminus \{h\}))],$$

that is, no single hypothesis may be defined in the well-founded model if we join to P all the other remaining hypotheses.

MH^{Sustainable}_{min}: the reduction system is \mapsto_{MH} ; $MH_{min}^{Sustainable}(P)$ retrieves the minimal models contained in $MH^{Sustainable}(P)$ for any normal logic program P . $MH_{min}^{Sustainable}$ also belongs to the ASM^m family, due to the minimality of its models.

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

Besides SM , $MH_{min}^{Sustainable}$ (defined above in this appendix) and others, the following are ASM^m family members.

Navy: the reduction system is \mapsto_{WFS} . Given a normal logic program P , $Navy(P)$ contains all the minimal models of \hat{P} .¹⁶

Blue: the reduction system is \mapsto_{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:¹⁷

- (a) Compute $K = kernel_{Navy}(\hat{P})$;
 - (b) Compute $K' = kernel_{Navy}(\widehat{P \cup K})$;
 - (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).

¹⁵ *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 .

¹⁶ See definition of \hat{P} in subsection 3.1.

¹⁷ This algorithm is presented in (Dix 1995a).

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

Green: the reduction system is \mapsto_{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.¹⁸

Appendix E Example of Cumulativity Failure Detection

The following 1-layer program P is a counter-example for showing, using theorem 1, that *SM* semantics is not cumulative, due to being not cautious monotonic (program P does not allow us to spot the failure of any of these properties by means of the usual definitions of cumulativity and cautious monotony presented in section 2).

$$\begin{array}{lll} a \leftarrow \text{not } b, \text{not } s & d \leftarrow b & d \leftarrow a \\ b \leftarrow \text{not } a, \text{not } c & d \leftarrow \text{not } d & c \leftarrow k \\ c \leftarrow \text{not } b, \text{not } k & k \leftarrow a, d & s \leftarrow \text{not } a, d \end{array}$$

In fact, the *SM* models of P 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 $ker_{SM}(P \cup \{d\}) = \{d\}$. Hence no negative conclusion can be afforded about cumulativity, by means of the usual definition of this property. Meanwhile, by using the statement (3) of theorem 1 it is straightforward to conclude that *SM* semantics does not enjoy the property of cumulativity, because $SM(P) \neq SM(P \cup \{d\})$. Moreover, statement (1) of the theorem tells us, via this example, that *SM* semantics is not cautious monotonic because $SM(P \cup \{d\}) \not\subseteq SM(P)$.

Appendix F Proof of Cautious Monotony and Cut Failure

The following 1-layer program $P = \hat{P}$ is a counter-example for showing, using theorem 1, that none of the semantics *MH*, *MH^{LS}*, *MH^{Loop}*, *MH^{Sustainable}* and *MH^{Regular}* is either cautious monotonic or cut (program P does not allow us to spot the failure of any of these properties by means of the usual definitions of cautious monotony and cut presented in section 2).

$$\begin{array}{ll} u \leftarrow b & a \leftarrow \text{not } b \\ u \leftarrow c & b \leftarrow \text{not } c \\ t \leftarrow a & c \leftarrow h, u \\ t \leftarrow h & h \leftarrow \text{not } h, \text{not } t \end{array}$$

Let *SEM* represent any of the above 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

¹⁸ 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 iff there is no rule $r \in P$ such that $Head(r) = \{b\}$ and all literals in $Body(r)$ are true with respect to M .

semantics:

$$\begin{array}{llll}
 u \leftarrow b & & a \leftarrow \text{not } b & \\
 u \leftarrow c & & b \leftarrow \text{not } c & \\
 t \leftarrow a & & c \leftarrow h & \\
 t \leftarrow h & & h \leftarrow \text{not } h, \text{not } t & \quad u \leftarrow
 \end{array}$$

(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, $M = \{h, u, c, t, a\}$, with affix $\{h\}$, is a minimal affix model of $P \cup \{u\}$ but is not a minimal affix model of P , which by point (1) of theorem 1 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 point (2) of theorem 1 renders any of these semantics as not cut.

Appendix G Picky, a Special 2-valued Cumulative Semantics

The semantics *Picky* is defined as follows: for any normal logic program P (1) if $SM(P) = \emptyset$, then $Picky(P) = \emptyset$; (2) if $SM(P) \neq \emptyset$, then (2a) $Picky(P) = SM(P)$ iff $\ker_{SM}(P) = \ker_{SM}(P \cup S)$, for every $S \subseteq \ker_{SM}(P)$; (2b) otherwise $Picky(P) = \emptyset$. This semantics is cumulative, by definition, but it is not always the case that $Picky(P) = Picky(P \cup S)$, $S \subseteq \ker_{SM}(P)$: for program P of the example in Appendix E, we have $Picky(P) = \{\{a, d, c, k\}, \{b, d, s\}\}$ and $Picky(P \cup \{d\}) = \{\{a, d, c, k\}, \{b, d, s\}, \{c, d, s\}\}$, which means, by theorem 1, that *Picky* is not cumulative. Notice that *Picky* is not a *ASM* semantics, because it does not conservatively extend the *SM* semantics: for program P in the referred example, we have $SM(P) \neq \emptyset$ and $Picky(P) = \emptyset$.

Appendix H Excessiveness and Irregularity

Excessiveness. 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),

$$\begin{array}{l}
 a \leftarrow \text{not } b \\
 b \leftarrow \text{not } a \\
 \text{-----}1 \\
 u \leftarrow a \\
 u \leftarrow b \\
 \text{-----}2 \\
 p \leftarrow \text{not } p, \text{not } u \\
 \text{-----}3 \\
 q \leftarrow \text{not } q, \text{not } p \\
 \text{-----}4.
 \end{array}$$

Let SEM represent any of these semantics. It is the case that $N = \{a, u, p, \text{not } b, \text{not } q\}$ with affix $\{a, p\}$, is a model of P under any of the referred semantics, and for no SEM model $M_* \in SEM(P^{\leq 2})$, where $SEM(P^{\leq 2}) = \{\{a, \text{not } b, u\}, \{\text{not } a, b, u\}\}$, do we have $N \in SEM(P^{>2} \cup M_*^+)$, 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 $SEM(P^{>2} \cup M_*^+)$.

Irregularity. Program P below shows that the semantics MH , MH^{LS} and MH^{Loop} , *Green*, *Navy* and *Blue* are all irregular.

$$\begin{aligned} a &\leftarrow \text{not } b \\ b &\leftarrow \text{not } a \\ \text{---} &\text{---} \text{---} \text{---} \text{---} 1 \\ p &\leftarrow \text{not } p, \text{not } a \\ q &\leftarrow \text{not } q, \text{not } b \end{aligned}$$

In fact, all these semantics admit the model $N = \{a, b, \text{not } p, \text{not } q\}$. The models of segment $P^{\leq 1}$ are $\{a, \text{not } b\}$ and $\{b, \text{not } a\}$, none of whose positive sets of atoms equals $N_{\leq T}^+ = \{a, b\}$. As *Blue* is not excessive, this example shows *irregularity* $\not\Rightarrow$ *excessiveness*.

Appendix I The 12 possible types of ASM^h and ASM^m semantics

In table I 1 below ‘0’ flags the failure of a property and ‘1’ means the property is verified.

Table I 1. *The 12 possible types of ASM^h and ASM^m semantics*

	\exists	gl	lg	cm	cut
1	0	0	0	0	0
2	0	0	0	0	1
3	0	0	1	0	0
4	0	0	1	0	1
5	1	1	0	0	0
6	1	1	0	0	1
7	1	1	0	1	0
8	1	1	0	1	1
9	1	1	1	0	0
10	1	1	1	0	1
11	1	1	1	1	0
12	1	1	1	1	1

The 20 missing types of semantics correspond to cases where $(\exists = 0 \text{ and } gl = 1)$, or $(\exists = 1 \text{ and } gl = 0)$, or $(\exists = 0 \text{ and } cm = 1)$, each of these cases going against the statement of theorem 2. The correspondence of the $ASM^h \cup ASM^m$ class semantics presented in this text and

the entries in table I1 is as follows: 1. $MH^{sustainable}$, $MH_{min}^{Sustainable}$ 2. — 3. — 4. SM 5. MH , MH^{LS} , MH^{Loop} , $Green$ 6. — 7. $Navy$ 8. $Blue$ 9. $MH^{Regular}$ 10. — 11. — 12. $Cyan$. Whether semantics of the $ASM^h \cup ASM^m$ class exist for the types marked with '—', may be envisaged as an open issue.