Functional ASP with Intensional Sets: Application to Gelfond-Zhang Aggregates

Pedro Calabar, Jorge Fandinno, Luis Fariñas del Cerro, David Pearce

doi:10.1017/S1471068418000169

Functional ASP with Intensional Sets: Application to Gelfond-Zhang Aggregates∗

PEDRO CABALAR
Department of Computer Science, University of Corunna, Corunna, Spain
(*e-mail:* cabalar@udc.es)

JORGE FANDINNO and LUIS FARIÑAS DEL CERRO
IRIT, Université de Toulouse, CNRS, Toulouse, France
(*e-mails:* {jorge.fandinno, luis}@irit.fr)

DAVID PEARCE
Universidad Politécnica de Madrid, Madrid, Spain
(*e-mail:* david.pearce@upm.es)

Abstract

In this paper, we propose a variant of Answer Set Programming (ASP) with evaluable functions that extends their application to sets of objects, something that allows a fully logical treatment of aggregates. Formally, we start from the syntax of First Order Logic with equality and the semantics of Quantified Equilibrium Logic with evaluable functions ($\mathrm{QEL}^{=}_{\mathcal{F}}$). Then, we proceed to incorporate a new kind of logical term, *intensional set* (a construct commonly used to denote the set of objects characterised by a given formula), and to extend $\mathrm{QEL}^{=}_{\mathcal{F}}$ semantics for this new type of expression. In our extended approach, intensional sets can be arbitrarily used as predicate or function arguments or even nested inside other intensional sets, just as regular first-order logical terms. As a result, aggregates can be naturally formed by the application of some evaluable function (`count`, `sum`, `maximum`, etc) to a set of objects expressed as an intensional set. This approach has several advantages. First, while other semantics for aggregates depend on some syntactic transformation (either via a reduct or a formula translation), the $\mathrm{QEL}^{=}_{\mathcal{F}}$ interpretation treats them as regular evaluable functions, providing a compositional semantics and avoiding any kind of syntactic restriction. Second, aggregates can be explicitly defined now within the logical language by the simple addition of formulas that fix their meaning in terms of multiple applications of some (commutative and associative) binary operation. For instance, we can use recursive rules to define `sum` in terms of integer addition. Last, but not least, we prove that the semantics we obtain for aggregates coincides with the one defined by Gelfond and Zhang for the $\mathcal{A}log$ language, when we restrict to that syntactic fragment.

KEYWORDS: Answer Set Programming, Equilibrium Logic, Partial Functions, Aggregates

∗ Partially supported by MINECO, Spain, grant TIC2017-84453-P, Xunta de Galicia, Spain (GPC ED431B 2016/035 and 2016-2019 ED431G/01, CITIC). The second author is funded by the Centre International de Mathématiques et d'Informatique de Toulouse (CIMI) through contract ANR-11-LABEX-0040-CIMI within the program ANR-11-IDEX-0002-02. The fourth author is supported by UPM project RP151046021.

1 Introduction

Due to its extensive use for practical Knowledge Representation and Reasoning (KRR), the paradigm of *Answer Set Programming* (ASP; Baral 2003) has been continuously subject to multiple extensions of its input language and, frequently, its formal semantics. One of those possible extensions is the addition of *evaluable functions* (see Cabalar 2013 for a survey). This extension allows us, for instance, to replace the conjunction $mother(cain, X) \wedge mother(abel, X)$ by the equality $mother(cain) = mother(abel)$ so that (i) *mother* can be better captured as a function (a person has a unique mother) and (ii) it is not treated as a Herbrand function, since syntactically different terms may refer to the same object. Although several prototypes for functional ASP have been developed (Lin and Wang 2008; Cabalar 2011; Balduccini 2013; Bartholomew and Lee 2014), the use of evaluable functions has not been commonly adopted in the mainstream ASP solvers yet. Still, their logical definition can also be useful for other common ASP extensions, as happened with their application to constraint ASP (Cabalar *et al.* 2016). Another ASP extension that can be examined under the functional viewpoint is the use of aggregates. An *aggregate* is the result of an operation on a set of values, such as their cardinality, their sum, their maximum/minimum value, etc. ASP introduces this feature via so-called *aggregate atoms*, that allow comparing the result of an aggregate with some fixed value. To put an example, suppose $p(X)$ means that Agatha Christie wrote book X. Then, adding the aggregate atom $\texttt{count}\{X : p(X)\} \geq n$ in a rule body checks that she wrote at least n books. Defining the semantics for these atoms may become tricky, since it is easy to build self-referential rules like:

$$p(a) \leftarrow \texttt{count}\{X : p(X)\} \geq n. \tag{1}$$

to express that Mrs. Christie also writes an autobiography a if she writes at least n books. Different alternative semantics have been proposed for ASP aggregates (Simons *et al.* 2002; Pelov *et al.* 2007; Son and Pontelli 2007; Ferraris 2011; Faber *et al.* 2011; Gelfond and Zhang 2014) but all of them have treated each aggregate atom as a whole, without providing a semantics for its individual components. A different, and perhaps more natural possibility, is to consider inequality as a standard predicate and interpret $\texttt{count}(S)$ as an *evaluable function*, whose argument S happens to be a set.

In this paper, we propose an extension of ASP with evaluable functions that allows their application to sets of objects and the treatment of aggregates as functions. To this aim, we start from the first-order logic characterisation of ASP, *Quantified Equilibrium Logic* (QEL; Pearce and Valverde 2004) plus its extension to evaluable functions ($\text{QEL}^{=}_{\mathcal{F}}$; Cabalar 2011). Then, we proceed to include a new type of logical term, *intensional set*, with the form $\{\vec{\tau}(\vec{x}) : \varphi(\vec{x})\}$ and the expected meaning, that is, the set of tuples $\vec{\tau}(\vec{c})$ for which the formula $\varphi(\vec{c})$ holds (having $\vec{c}$ and $\vec{x}$ the same arity). Intensional sets constitute a quite common mathematical notation and, in fact, have been already studied in the context of Prolog (Dovier *et al.* 1991) and Constraint Logic Programming (Dovier *et al.* 2003). In our case, we will treat them as regular, first-order logical terms, without syntactic restrictions, so they can be arbitrarily nested in other expressions. One interesting feature inherited from $\text{QEL}^{=}_{\mathcal{F}}$ is that functions can be partial, so we can use them to represent that, say, $mother(adam)$, $mother(eve)$ or $division(3, 0)$ are undefined, but also that the maximum value of an empty set $\texttt{max}(\emptyset)$ is undefined too. The new

extension allows now to define new aggregates within the logical language. It suffices to add formulas to fix their meaning in terms of multiple applications of some (commutative and associative) binary operation. For instance, we show how to define the `sum` aggregate using recursive rules in terms of integer addition. Finally, we are also able to prove that, when restricted to the the *Alog* language (Gelfond and Zhang 2014), there is a semantic, one-to-one correspondence.

The rest of the paper is organised as follows. In Section 2 we recall the basic definitions of Functional ASP under the $\text{QEL}^{=}_{\mathcal{F}}$ interpretation. Section 3 introduces intensional sets while Section 4 studies their use for aggregates. Section 5 focuses on the correspondence to (Gelfond and Zhang 2014) aggregates. Finally, Section 6 concludes the paper.

2 Background: Quantified Equilibrium Logic with Evaluable Functions

The definition of propositional Equilibrium Logic (Pearce 1996) relied on a selection criterion on models of the intermediate logic of *Here-and-There* (HT; Heyting 1930). The first order case (Pearce and Valverde 2004) followed similar steps, introducing a quantified version of HT, called $\text{SQHT}^{=}$ that stands for *Quantified HT with static domains*[1] *and equality*. In this section we describe the syntax and semantics of a variant of the latter, called $\text{SQHT}^{=}_{\mathcal{F}}$ (Cabalar 2011), for dealing with evaluable functions.

We begin by defining a first-order language by its *signature*, a tuple $\Sigma = \langle \mathcal{C}, \mathcal{F}, \mathcal{P} \rangle$ of disjoint sets where $\mathcal{C}$ and $\mathcal{F}$ are sets of *function names* and $\mathcal{P}$ a set of *predicate names*. We assume that each function (resp. predicate) name has the form f/n where f is the function (resp. predicate) symbol, and $n \geq 0$ is an integer denoting the number of arguments (or *arity*). Elements in $\mathcal{C}$ will be called *Herbrand functions* (or *constructors*), whereas elements in $\mathcal{F}$ will receive the name of *evaluable functions* (or *operations*). The sets $\mathcal{C}_0$ (Herbrand constants) and $\mathcal{F}_0$ (evaluable constants) respectively represent the elements of $\mathcal{C}$ and $\mathcal{F}$ with arity 0. We assume $\mathcal{C}_0$ contains at least one element.

First-order formulas follow the syntax of classical predicate calculus with equality "=". We assume that default negation $\neg\varphi$ is defined as $\varphi \rightarrow \bot$. We use letters x, y, z and their capital versions to denote variables, τ to denote terms, and letters c, d, e to denote ground terms. Tuples of variables, terms and ground terms are respectively represented by $\vec{x}, \vec{\tau}, \vec{c}$. By abuse of notation, when a tuple contains a single element, we write just τ instead of $\langle\tau\rangle$. When writing formulas, we assume that all free variables are implicitly universally quantified. An atom like $\tau = \tau'$ is called an *equality atom*, whereas an atom like $p(\tau_1, \ldots, \tau_n)$ for any predicate p/n different from equality receives the name of *predicate atom*. Given any set of functions S we write $\mathit{Terms}_{\mathcal{F}}(S)$ to stand for the set of ground terms built from functions (and constants) in S. In particular, the set of all possible ground terms for the signature $\Sigma = \langle \mathcal{C}, \mathcal{F}, \mathcal{P} \rangle$ would be $\mathit{Terms}_{\mathcal{F}}(\mathcal{C} \cup \mathcal{F})$ whereas the subset $\mathit{Terms}_{\mathcal{F}}(\mathcal{C})$ will be called the *Herbrand Universe* of the language $\mathcal{L}_{\mathcal{F}}(\Sigma)$. The *Herbrand Base* $\mathcal{HB}_{\mathcal{F}}(\mathcal{C}, \mathcal{P})$ is the set containing all atoms that can be formed with predicates in $\mathcal{P}$ and terms in the Herbrand Universe, $\mathit{Terms}_{\mathcal{F}}(\mathcal{C})$.

Definition 1 ($\text{SQHT}^{=}_{\mathcal{F}}$-assignment)**.** *An* $\text{SQHT}^{=}_{\mathcal{F}}$*-assignment* σ *for a signature* $\langle \mathcal{C}, \mathcal{F}, \mathcal{P} \rangle$ *is a function* $\sigma : \mathit{Terms}_{\mathcal{F}}(\mathcal{C} \cup \mathcal{F}) \rightarrow \mathit{Terms}_{\mathcal{F}}(\mathcal{C}) \cup \{\mathbf{u}\}$ *that maps any ground term in the*

[1] The term *static domain* means that the universe is shared among all worlds in the Kripke frame.

language to some ground term in the Herbrand Universe or the special value $\mathbf{u} \notin \mathit{Terms}_{\mathcal{F}}(\mathcal{C} \cup \mathcal{F})$ *(standing for* undefined*). The function* σ *must satisfy:*

(i) $\sigma(c) \stackrel{\text{def}}{=} c$ *for all* $c \in \mathit{Terms}_{\mathcal{F}}(\mathcal{C})$.

(ii) $\sigma(f(\tau_1, \ldots, \tau_n)) \stackrel{\text{def}}{=} \begin{cases} \mathbf{u} & \text{if } \sigma(\tau_i) = \mathbf{u} \text{ for some } i = 1 \ldots n \\ \sigma(f(\sigma(\tau_1), \ldots, \sigma(\tau_n))) & \text{otherwise} \end{cases}$ □

As we can see, the value of any functional term is an element from the Herbrand Universe $\mathit{Terms}_{\mathcal{F}}(\mathcal{C})$, excluding the cases in which operations are left undefined (i.e., they are *partial* functions) – if so, they are assigned the special element $\mathbf{u}$ (outside the universe) instead. Condition (i) asserts, as expected, that any term c from the Herbrand Universe has the fixed valuation $\sigma(c) = c$. Condition (ii) asserts that a functional term with an undefined argument becomes undefined in its turn (functions like these are called *strict*). Otherwise, if all arguments are defined, then functions preserve their interpretation through subterms – for instance, if we have $\sigma(f(a)) = c$ we expect that $\sigma(g(f(a)))$ and $\sigma(g(c))$ coincide. It is easy to see that (ii) implies that σ is completely determined by the mappings $f(\vec{c}) = d$ where f is any operation, $\vec{c}$ a tuple of elements from $\mathit{Terms}_{\mathcal{F}}(\mathcal{C})$, and d an element in the latter. We call these expressions *ground functional facts.*

Definition 2 (Ordering $\preceq$ among assignments)**.** *Given two assignments* σ, σ' *we define* $\sigma \preceq \sigma'$ *as the condition:* $\sigma(\tau) = \sigma'(\tau)$ *or* $\sigma(\tau) = \mathbf{u}$*, for all terms* $\tau \in \mathit{Terms}_{\mathcal{F}}(\mathcal{C} \cup \mathcal{F})$. □

As usual, we write $\sigma \prec \sigma'$ when $\sigma \preceq \sigma'$ and $\sigma \neq \sigma'$. The intuitive meaning of $\sigma \preceq \sigma'$ is that both contain *compatible information*, but the former contains *less information* than the latter: any defined function in σ must preserve the same value in σ'.

Definition 3 (SQHT$^{=}_{\mathcal{F}}$-interpretation)**.** *An* SQHT$^{=}_{\mathcal{F}}$-interpretation $\mathcal{I}$ *for a signature* $\Sigma = \langle \mathcal{C}, \mathcal{F}, \mathcal{P} \rangle$ *is a quadruple* $\langle \sigma^h, \sigma^t, I^h, I^t \rangle$ *where* $I^h \subseteq I^t \subseteq \mathcal{HB}_{\mathcal{F}}$ *are two sets of ground atoms and* σ^h *and* σ^t *are two assigments satisfying* $\sigma^h \preceq \sigma^t$. □

The superindices h, t represent two intuitionistic Kripke worlds (respectively standing for *here* and *there*) with a reflexive ordering relation satisfying $h \leq t$. Accordingly, world h contains less information than t, as we can see in the conditions $I^h \subseteq I^t$ and $\sigma^h \preceq \sigma^t$. We say that the interpretation $\mathcal{I}$ is *total*[2] when both worlds contain the same information, that is, $I^h = I^t$ and $\sigma^h = \sigma^t$, and we abbreviate it as the pair $\langle \sigma^t, I^t \rangle$. Moreover, given $\mathcal{I} = \langle \sigma^h, \sigma^t, I^h, I^t \rangle$ we define its corresponding total interpretation $\mathcal{I}^t$ as $\langle \sigma^t, I^t \rangle$ that is, the one in which all the uncertainty in world h is "filled" with the information in t.

An interpretation $\mathcal{I} = \langle \sigma^h, \sigma^t, I^h, I^t \rangle$ *satisfies* a formula φ at some world $w \in \{h, t\}$, written $\mathcal{I}, w \models_{\mathcal{F}} \varphi$, when any of the following conditions are satisfied:

i) $\mathcal{I}, w \models_{\mathcal{F}} p(\tau_1, \ldots, \tau_n)$ if $p(\sigma^w(\tau_1), \ldots, \sigma^w(\tau_n)) \in I^w$ for any predicate $p/n \in \mathcal{P}$;
ii) $\mathcal{I}, w \models_{\mathcal{F}} \tau_1 = \tau_2$ if $\sigma^w(\tau_1) = \sigma^w(\tau_2) \neq \mathbf{u}$;
iii) $\mathcal{I}, w \not\models_{\mathcal{F}} \bot$; $\mathcal{I}, w \models_{\mathcal{F}} \top$;
iv) $\mathcal{I}, w \models_{\mathcal{F}} \alpha \wedge \beta$ if $\mathcal{I}, w \models_{\mathcal{F}} \alpha$ and $\mathcal{I}, w \models_{\mathcal{F}} \beta$;
v) $\mathcal{I}, w \models_{\mathcal{F}} \alpha \vee \beta$ if $\mathcal{I}, w \models_{\mathcal{F}} \alpha$ or $\mathcal{I}, w \models_{\mathcal{F}} \beta$;
vi) $\mathcal{I}, w \models_{\mathcal{F}} \alpha \to \beta$ if for all $w' \geq w$: $\mathcal{I}, w' \not\models_{\mathcal{F}} \alpha$ or $\mathcal{I}, w' \models_{\mathcal{F}} \beta$;

[2] Note that by *total* we do not mean that functions cannot be left undefined. We may still have some term d for which $\sigma^h(d) = \sigma^t(d) = \mathbf{u}$.

vii) $\mathcal{I}, w \models_{\mathcal{F}} \forall x\ \alpha(x)$ if for each $c \in \mathit{Terms}_{\mathcal{F}}(\mathcal{C})$: $\mathcal{I}, w \models_{\mathcal{F}} \alpha(c)$;
viii) $\mathcal{I}, w \models_{\mathcal{F}} \exists x\ \alpha(x)$ if for some $c \in \mathit{Terms}_{\mathcal{F}}(\mathcal{C})$: $\mathcal{I}, w \models_{\mathcal{F}} \alpha(c)$.

The first condition above implies that an atom with an undefined argument will always be evaluated as false since, by definition, **u** never occurs in ground atoms of I^h or I^t. Something similar happens with equality: $\tau_1 = \tau_2$ will be false if any of the two operands, or even both, are undefined. As usual, I is called a *model* of a theory Γ, written $I \models_{\mathcal{F}} \Gamma$, when $I, h \models_{\mathcal{F}} \varphi$ for all $\varphi \in \Gamma$.

Proposition 1 (From Cabalar 2011)**.** $\mathcal{I}, h \models_{\mathcal{F}} \neg\varphi \ \Leftrightarrow\ \mathcal{I}, t \models_{\mathcal{F}} \neg\varphi \Leftrightarrow\ \mathcal{I}, t \not\models_{\mathcal{F}} \varphi$. □

We define next a particular ordering among SQHT$^{=}_{\mathcal{F}}$-interpretations. We say that $\mathcal{I}_1 = \langle \sigma_1^h, \sigma_1^t, I_1^h, I_1^t \rangle$ is smaller than $\mathcal{I}_2 = \langle \sigma_2^h, \sigma_2^t, I_2^h, I_2^t \rangle$, also written $\mathcal{I}_1 \preceq \mathcal{I}_2$ by abuse of notation, when $I_1^t = I_2^t$, $\sigma_1^t = \sigma_2^t$, $I_1^h \subseteq I_2^h$ and $\sigma_1^h \preceq \sigma_2^h$. That is, they have the same information at world t, but $\mathcal{I}_1$ can have less information than $\mathcal{I}_2$ at world h. Again, we write $\mathcal{I}_1 \prec \mathcal{I}_2$ when $\mathcal{I}_1 \preceq \mathcal{I}_2$ and $\mathcal{I}_1 \neq \mathcal{I}_2$. Nonmonotonicity is obtained by the next definition, which introduces the idea of equilibrium models for SQHT$^{=}_{\mathcal{F}}$.

Definition 4 (Equilibrium model)**.** *A total model $\mathcal{I} = \langle \sigma, I \rangle$ of a theory Γ is an* equilibrium model *if there is no strictly smaller interpretation $\mathcal{I}' \prec \mathcal{I}$ that is also a model of Γ. A set of atoms I is a* stable model[3] *of Γ iff $\langle \sigma, I \rangle$ is an equilibrium model for some σ.* □

3 QEL with Evaluable Functions and Intensional Sets

In this section, we define SQHT$^{=}_{\mathcal{S}}$: a logic that extends SQHT$^{=}_{\mathcal{F}}$ with *intensional sets*. Intensional sets are terms of the form $\{\vec{x} : \vec{\tau}(\vec{x}) : \varphi(\vec{x})\}$ where $\vec{x}$ is a tuple of variables and $\vec{\tau}(\vec{x})$ and $\varphi(\vec{x})$ are respectively a tuple of terms and a formula with free variables $\vec{x}$.

Note that, as opposed to terms in SQHT$^{=}_{\mathcal{F}}$ (and also in first order logic), intensional sets are terms whose structure not only depends on other terms, but also on formulas. Hence, we define terms and formulas recursively such that the definition of i-terms will depend on the definition of $(i-1)$-formulas while the definition of i-formulas will depend on the definition of i-terms. We depart from a similar first-order signature $\Sigma = \langle \mathcal{C}, \mathcal{F}, \mathcal{P} \rangle$ as in SQHT$^{=}_{\mathcal{F}}$, but we build the terms as follows. For any $i \geq 0$, we define an *i-term* as any of the following cases:

i) every constant $c \in \mathcal{C}_0 \cup \mathcal{F}_0$.
ii) every variable x.
iii) $f(\tau_1, \ldots, \tau_n)$, where $f/n \in (\mathcal{C} \cup \mathcal{F})$ and $\tau_1, \ldots, \tau_n$ are i-terms, in their turn.
iv) the construct $\{\vec{\tau}_1, \ldots, \vec{\tau}_m\}$ (called *extensional set*), where $m \geq 0$ and $\vec{\tau}_1, \ldots, \vec{\tau}_m$ are n-tuples (of the same arity $n \geq 1$) of i-terms . If $m = 0$ we write $\emptyset$ instead of $\{\}$.
v) the construct $\{\vec{\tau} : \varphi\}$ (called *intensional set*) if $i > 0$, φ is an $(i-1)$-formula and $\vec{\tau}$ is a tuple of i-terms.

Now, for any $i \geq 0$, *i-atoms* and *i-formulas* are defined over i-terms as follows:

vi) if $\tau_1, \ldots, \tau_n$ are i-terms and $p/n \in \mathcal{P}$, then $p(\tau_1, \ldots, \tau_n)$ is an i-atom,

[3] Apart from atoms, we could additionally include ground functional facts $f(\vec{c}) = d$. However, we only consider atoms here, for better comparison to other (non-functional) semantics of aggregates.

vii) if τ_1 and τ_2 are i-terms, then $\tau_1 = \tau_2$ is an i-atom
viii) every i-atom is an i-formula,
ix) $\bot$ and $\top$ are 0-formulas,
x) if φ_1 and φ_2 are i-formulas, then $\varphi_1 \otimes \varphi_2$ is an i-formula with $\otimes \in \{\wedge, \vee, \rightarrow\}$.
xi) if φ is an i-formula and x is a variable, then $\forall x\, \varphi$ and $\exists x\, \varphi$ are i-formulas.

A *formula* (resp. *term*) is any i-formula (resp. i-term) for any $i \geq 0$. Note that v) is the only case in the term definition that refers to a formula, but this formula has less rank than the term, so the definition is well-founded. $\mathit{Terms}^i(\mathcal{C} \cup \mathcal{F})$ denotes all the ground i-terms while $\mathit{Terms}^i(\mathcal{C})$ denotes all ground i-terms without evaluable functions. $\mathit{Terms}(\mathcal{C} \cup \mathcal{F}) = \bigcup_{0 \leq i} \mathit{Terms}^i(\mathcal{C} \cup \mathcal{F})$ and $\mathit{Terms}(\mathcal{C}) = \bigcup_{0 \leq i} \mathit{Terms}^i(\mathcal{C})$ denote the set of all ground terms and ground terms without evaluable functions. In particular, $\mathit{Terms}^0(\mathcal{C})$ corresponds to the *Herbrand Universe* that includes not only $\mathit{Terms}_{\mathcal{F}}(\mathcal{C})$ we had before, but also all possible formations of extensional sets, that act as a new constructor. For instance, if we have the singleton $\mathcal{C} = \{c\}$, then $\mathit{Terms}_{\mathcal{F}}(\mathcal{C}) = \{c\}$ is finite but $\mathcal{D}$ additionally contains an infinite number of (finite) extensional sets including, among others, the sets of tuples $\emptyset, \{c\}$[4], $\{\langle c, c\rangle\}$, $\{\langle c, c, c\rangle\}$, ..., or combinations of nested sets such as $\{\{c\}\}$ or $\{\{c\}, \{\langle c, c\rangle\}\}$, etc. We also define $\mathcal{SB} \stackrel{\text{def}}{=} \mathit{Terms}(\mathcal{C}) \setminus \mathit{Terms}_{\mathcal{F}}(\mathcal{C})$, that is, the subset of the Herbrand universe consisting of extensional sets. In the previous example $\mathcal{SB} = \mathit{Terms}(\mathcal{C}) \setminus \{c\}$. By $\mathcal{HB}$ we denote the *Herbrand Base*, that is, the set of all ground atoms of the form $p(c_1, \ldots, c_n)$ with $p/n \in \mathcal{P}$ and $\{c_1, \ldots, c_n\} \subseteq \mathit{Terms}(\mathcal{C})$.

If we consider terms also formed with evaluable functions, we have that $\mathit{Terms}_{\mathcal{F}}(\mathcal{C} \cup \mathcal{F}) \subseteq \mathit{Terms}^0(\mathcal{C} \cup \mathcal{F})$ again, and so, every $\mathrm{SQHT}^{=}_{\mathcal{F}}$ formula is also a $\mathrm{SQHT}^{=}_{\mathcal{S}}$ formula – obviously the converse does not hold, as the latter may contain set constructors such as $p(\{c\})$. Still, we could take each possible extensional set in $\mathcal{SB}$ as a kind of Herbrand constant like those in $\mathcal{C}$. Doing so, $\mathit{Terms}_{\mathcal{F}}(\mathcal{C} \cup \mathcal{SB} \cup \mathcal{F}) = \mathcal{D}$ and, thus, every 0-formula over a signature $\Sigma = \langle \mathcal{C}, \mathcal{F}, \mathcal{P}\rangle$ is also a $\mathrm{SQHT}^{=}_{\mathcal{F}}$ formula over the signature $\Sigma' = \langle \mathcal{C} \cup \mathcal{SB}, \mathcal{F}, \mathcal{P}\rangle$. Note, however, that intensional sets are syntactically different from any $\mathrm{SQHT}^{=}_{\mathcal{F}}$ term, so there are $\mathrm{SQHT}^{=}_{\mathcal{S}}$ terms (resp. formulas) that are not $\mathrm{SQHT}^{=}_{\mathcal{F}}$ terms (resp. formulas) over any signature.

For any expression α (term or formula), we define next when an occurrence of a variable is either *free* or *bound* to some quantifier/intensional set:

i) all free occurrences of x in ψ are bound in $\exists x\, \psi$ to its prefix quantifier $\exists x$
ii) all free occurrences of x in ψ are bound in $\forall x\, \psi$ to its prefix quantifier $\forall x$
iii) if x occurs in $\vec{x}$, then all free occurrences of x in ψ and $\vec{\tau}$ are bound in $\{\vec{x} : \vec{\tau} : \psi\}$ to the outermost intensional set.
iv) In the remaining cases, an occurrence of x is bound (to some connective) in a formula iff it is so in some subformula; otherwise, it is free.

As in the case of $\mathrm{SQHT}^{=}_{\mathcal{F}}$, when we write standalone formulas, we assume that all free variables are implicitly universally quantified. Similarly, we write $\{\vec{\tau} : \psi\}$ instead of $\{\vec{x} : \vec{\tau} : \psi\}$ when $\vec{x}$ contains exactly all free variables in $\vec{\tau}$. Note that intensional sets play a role similar to quantifiers. As an example, suppose we want to obtain the maximum number of times that the character *Poirot* is mentioned in an Agatha Christie book b, and assume that predicate $\mathit{word}(b, i, w)$ tells us that the i-th word of book b is w. We

[4] Recall that, here, c stands for the unary tuple $\langle c\rangle$.

assume by now that we have functions `count` and `max` on sets: their meaning will be fixed later on. The set $\{i : word(b, i, Poirot)\}$ collects all occurrences of word *Poirot* in book b. Since i is the only free variable to the left of ':', the intensional set is an abbreviation of $\{i : i : word(b, i, Poirot)\}$ revealing that i is being varied. On the contrary, variable b is left free. Now, take the expression

$$\texttt{max}\{\texttt{count}\{i : word(b, i, Poirot)\} : author(Agatha, b)\} \tag{2}$$

The left term $\texttt{count}\{i : word(b, i, Poirot)\}$ contains a free occurrence of b while i is bound to the inner intensional set. Therefore, (2) actually stands for:

$$\texttt{max}\{b : \texttt{count}\{i : i : word(b, i, Poirot)\} : author(Agatha, b)\}$$

that is obviously less readable than (2). However, in the general case, we may need to make use of the explicit list of quantified variables. For instance, if we want to parameterise the expression above for some author x and character y whose values are determined outside the term (as part of a formula), then we would necessarily write:

$$\texttt{max}\{b : \texttt{count}\{i : word(b, i, y)\} : author(x, b)\} \tag{3}$$

because the free occurrence of y in $\texttt{count}\{i : word(b, i, y)\}$ could make us incorrectly assume that it is being varied in the set, as happened with b.

3.1 Semantics

First, we need to define the domain in which terms are going to be interpreted. Given a set S, let us define the set of all possible n-tuples of elements from S, for any $n \geq 1$, as

$$Tup(S) \stackrel{\text{def}}{=} \bigcup_{n \geq 1} \{\ \vec{e} \mid \vec{e} \in S^n\ \}$$

Our domain $\mathcal{D}$ is inductively constructed as follows:

$$\mathcal{D}^0 \stackrel{\text{def}}{=} Terms_{\mathcal{F}}(\mathcal{C}) \qquad\qquad \mathcal{D}^{i+1} \stackrel{\text{def}}{=} \mathcal{D}^i \cup 2^{Tup(\mathcal{D}^i)}$$

so that $\mathcal{D} \stackrel{\text{def}}{=} \bigcup_{0 \leq i} \mathcal{D}^i$. We also define the subset of $\mathcal{D}$ consisting of sets as $\mathcal{S} \stackrel{\text{def}}{=} \mathcal{D} \backslash \mathcal{D}^0$.

Definitions of assignments and interpretations are then straightforward: we just extend the domain of σ from $Terms_{\mathcal{F}}(\mathcal{C} \cup \mathcal{F})$ to $Terms(\mathcal{C} \cup \mathcal{F})$ and replace the Herbrand Universe $Terms_{\mathcal{F}}(\mathcal{C})$ by its corresponding $\mathcal{D}$.

Definition 5 (Assignment)**.** *An* assignment σ *for a signature* $\Sigma = \langle \mathcal{C}, \mathcal{F}, \mathcal{P} \rangle$ *is a function* $\sigma : Terms(\mathcal{C} \cup \mathcal{F}) \to \mathcal{D} \cup \{\mathbf{u}\}$ *that maps some ground term in the Herbrand Universe or the special value* $\mathbf{u} \notin Terms(\mathcal{C} \cup \mathcal{F})$ *(standing for* undefined*) to any ground term in the language. Function* σ *must satisfy:*

(i) $\sigma(c) \stackrel{\text{def}}{=} c$ *for all* $c \in \mathcal{D}$.

(ii) $$\sigma(f(\tau_1, \ldots, \tau_n)) \stackrel{\text{def}}{=} \begin{cases} \mathbf{u} & \text{if } \sigma(\tau_i) = \mathbf{u} \text{ for some } i = 1 \ldots n \\ \sigma(f(\sigma(\tau_1), \ldots, \sigma(\tau_n))) & \text{otherwise} \end{cases}$$

(iii) $$\sigma(\{\vec{\tau}_1, \ldots, \vec{\tau}_n\}) \stackrel{\text{def}}{=} \begin{cases} \mathbf{u} & \text{if } \sigma(\vec{\tau}_i) = \mathbf{u} \text{ for some } i = 1 \ldots n \\ \{\, \sigma(\vec{\tau}_1), \ldots, \sigma(\vec{\tau}_n)\, \} & \text{otherwise} \end{cases}$$

where

$$\sigma(\langle \tau_1, \ldots, \tau_m \rangle) \stackrel{\text{def}}{=} \begin{cases} \mathbf{u} & \text{if } \sigma(\tau_j) = \mathbf{u} \text{ for some } j = 1 \ldots m \\ \langle\, \sigma(\tau_1), \ldots, \sigma(\tau_m)\, \rangle & \text{otherwise} \end{cases}$$

□

Note that we added a third case (iii) for extensional sets, but there is no restriction on the values of intensional sets: their meaning will be fixed later, once we describe the satisfaction of formulas.

Definition 6 (Ordering $\preceq$ among assignments)**.** *Given two assignments σ, σ' we define $\sigma \preceq \sigma'$, as the condition: $\sigma(\tau) = \sigma'(\tau)$ or $\sigma(\tau) = \mathbf{u}$ for all terms $\tau \in Terms(\mathcal{C} \cup \mathcal{F})$.* □

Interpretations $\mathcal{I} = \langle \sigma^h, \sigma^t, I^h, I^t \rangle$ have the same form as before, with $I^h \subseteq I^t \subseteq \mathcal{HB}$ and $\sigma^h \preceq \sigma^t$, but under the extended definition of assignment and Herbrand Base.

Definition 7 ($\mathcal{S}$-satisfaction)**.** *Given an interpretation $\mathcal{I} = \langle \sigma^h, \sigma^t, I^h, I^t \rangle$, we define when $\mathcal{I}$* $\mathcal{S}$-satisfies *a formula φ at some world $w \in \{h, t\}$, written $\mathcal{I}, w \models_{\mathcal{S}} \varphi$ as follows:*

i) $\mathcal{I}, w \models_{\mathcal{S}} p(\tau_1, \ldots, \tau_n)$ *if* $p(\sigma^w(\tau_1), \ldots, \sigma^w(\tau_n)) \in I^w$ *for any predicate* $p/n \in \mathcal{P}$*;*
ii) $\mathcal{I}, w \models_{\mathcal{S}} \tau_1 = \tau_2$ *if* $\sigma^w(\tau_1) = \sigma^w(\tau_2) \neq \mathbf{u}$*;*
iii) $\mathcal{I}, w \not\models_{\mathcal{S}} \bot$*;* $\mathcal{I}, w \models_{\mathcal{S}} \top$*;*
iv) $\mathcal{I}, w \models_{\mathcal{S}} \alpha \wedge \beta$ *if* $\mathcal{I}, w \models_{\mathcal{S}} \alpha$ *and* $\mathcal{I}, w \models_{\mathcal{S}} \beta$*;*
v) $\mathcal{I}, w \models_{\mathcal{S}} \alpha \vee \beta$ *if* $\mathcal{I}, w \models_{\mathcal{S}} \alpha$ *or* $\mathcal{I}, w \models_{\mathcal{S}} \beta$*;*
vi) $\mathcal{I}, w \models_{\mathcal{S}} \alpha \rightarrow \beta$ *if for all* $w' \geq w$*:* $\mathcal{I}, w' \not\models_{\mathcal{S}} \alpha$ *or* $\mathcal{I}, w' \models_{\mathcal{S}} \beta$*;*
vii) $\mathcal{I}, w \models_{\mathcal{S}} \forall x\ \alpha(x)$ *if for each* $c \in \mathcal{D}$*:* $\mathcal{I}, w \models_{\mathcal{S}} \alpha(c)$*;*
viii) $\mathcal{I}, w \models_{\mathcal{S}} \exists x\ \alpha(x)$ *if for some* $c \in \mathcal{D}$*:* $I, w \models_{\mathcal{S}} \alpha(c)$*.*

As usual, we write $I \models_{\mathcal{S}} \varphi$, when $I, h \models_{\mathcal{S}} \varphi$. □

It is easy to see that rules i)-vi) for $\models_{\mathcal{S}}$ are the exactly the same as for $\models_{\mathcal{F}}$. Rules vii) and viii) are just the result of replacing $Terms_{\mathcal{F}}(\mathcal{C})$ by $\mathcal{D}$. From this observation, we can immediately establish a correspondence between $\models_{\mathcal{S}}$ and $\models_{\mathcal{F}}$ as follows. Given any interpretation $\mathcal{I} = \langle \sigma^h, \sigma^t, I^h, I^t \rangle$ for a signature $\Sigma = \langle \mathcal{C}, \mathcal{F}, \mathcal{P} \rangle$, by $\hat{\mathcal{I}} = \langle \hat{\sigma}^h, \hat{\sigma}^t, I^h, I^t \rangle$ we denote a $\mathrm{SQHT}^{=}_{\mathcal{F}}$-interpretation for the signature $\Sigma = \langle \mathcal{C} \cup \mathcal{S}, \mathcal{F}, \mathcal{P} \rangle$, for each $w \in \{h, t\}$, where the assignment $\hat{\sigma}^w$ is the restriction of σ^w to $Terms^0(\mathcal{C} \cup \mathcal{F}) = Terms_{\mathcal{F}}(\mathcal{C} \cup \mathcal{S} \cup \mathcal{F})$.

Proposition 2. *Any interpretation $\mathcal{I}$ and 0-formula φ satisfy: $\mathcal{I}$,$w \models_{\mathcal{S}} \varphi$ iff $\hat{\mathcal{I}}$,$w \models_{\mathcal{F}} \varphi$.* □

As, in general, not all $\mathrm{SQHT}^{=}_{\mathcal{S}}$ formulas are $\mathrm{SQHT}^{=}_{\mathcal{F}}$ formulas, we cannot directly extend this result beyond 0-formulas. However, we may still expect that what can be proved to be a tautology using the $\mathrm{SQHT}^{=}_{\mathcal{F}}$ rules, still holds for $\mathrm{SQHT}^{=}_{\mathcal{S}}$ formulas. For instance, the $\mathrm{SQHT}^{=}_{\mathcal{S}}$ formula $f = \{\vec{\tau} : \varphi\} \rightarrow f = \{\vec{\tau} : \varphi\}$ is an obvious tautology which is not a $\mathrm{SQHT}^{=}_{\mathcal{F}}$ formula. However, we may replace every occurrence of the intensional set $\{\vec{\tau} : \varphi\}$ by a fresh evaluable constant c and observe that $f = c \rightarrow f = c$ is an $\mathrm{SQHT}^{=}_{\mathcal{F}}$ tautology, using this to conclude that it is an $\mathrm{SQHT}^{=}_{\mathcal{S}}$ tautology too. To formalise this intuition, let $\mathcal{F}_\varphi$ be a set disjoint from $\mathcal{C} \cup \mathcal{F} \cup \mathcal{P}$ containing a fresh constant per each different intensional set occurring in φ and let κ be a bijection mapping each intensional set in φ to its corresponding constant in $\mathcal{F}_\varphi$. Let us also denote by $\kappa(\varphi)$ the result of replacing in φ each intensional set by its κ image. By $\tilde{\mathcal{I}} = \langle \tilde{\sigma}^h, \tilde{\sigma}^t, I^h, I^t \rangle$, we denote the $\mathrm{SQHT}^{=}_{\mathcal{F}}$-interpretation for the signature $\Sigma' = \langle \mathcal{C} \cup \mathcal{S}, \mathcal{F} \cup \mathcal{F}_\varphi, \mathcal{P} \rangle$ where, for each $w \in \{h, t\}$, we have $\tilde{\sigma}^w(\tau) = \sigma^w(\kappa^{-1}(\tau))$ if $\tau \in \mathcal{F}_\varphi$ and $\tilde{\sigma}^w(\tau) = \sigma^w(\tau)$ otherwise.

Proposition 3. *For any interpretation $\mathcal{I}$ and formula φ: $\mathcal{I}$,$w \models_{\mathcal{S}} \varphi$ iff $\tilde{\mathcal{I}}$,$w \models_{\mathcal{F}} \kappa(\varphi)$.* □

Note that, as illustrated by Proposition 3, intensional sets act just as new fresh evaluable constants with respect to $\mathcal{S}$-satisfaction. To fix the expected meaning of each

intensional set $\{\vec{\tau}(\vec{x}) : \varphi(\vec{x})\}$ we still have to relate its value in σ^w to the satisfaction of formula $\varphi(\vec{x})$ in $\mathcal{I}$. Given $\tau = \{\vec{x} : \vec{\tau}(\vec{x}) : \varphi(\vec{x})\}$, let us define its *extension* at $\mathcal{I}, w$ as:

$$ext^0(\mathcal{I}, w, \tau) \stackrel{\text{def}}{=} \{\ \sigma^w(\vec{\tau}(\vec{c})) \mid \mathcal{I}, w \models_s \varphi(\vec{c}) \text{ for some } \vec{c} \in \mathcal{D}^{|\vec{x}|}\ \}$$

$$ext(\mathcal{I}, w, \tau) \stackrel{\text{def}}{=} \begin{cases} \mathbf{u} & \text{if } \mathbf{u} \in ext^0(\mathcal{I}, w, \tau) \\ ext^0(\mathcal{I}, w, \tau) & \text{otherwise} \end{cases}$$

To put an example, if $\tau_1 = \{X * n/X : p(X)\}$ and $\mathcal{I}_1$ contains $I_1^t = \{p(0), p(1), p(2)\}$ then the set of tuples would be $\{0*n/0,\ 1*n/1,\ 2*n/2\}$. Since the obtained set is finite, its evaluation coincides with the case of extensional sets in Def. 5, item (iii).

In the example, if we have, for instance, $\sigma^t(n) = 10$, then $ext(\mathcal{I}_1, t, \tau_1) = \sigma^t(\ \{0 * 10/0,\ 1 * 10/1,\ 2 * 10/2\}\ ) = \{\mathbf{u}, 10, 10\} = \mathbf{u}$. On the other hand, if I_1^t consists of the sequence $p(0), p(s(0)), p(s(s(0))), \ldots$, we similarly obtain the set $\{0 * n/0,\ s(0) * n/s(0),\ s(s(0)) * n/s(s(0)), \ldots\}$ which, being infinite, is not covered by Def. 5. With the definition of extension, for $\sigma^t(n) = 10$, we get $ext(\mathcal{I}_1, t, \tau_1) = \{\sigma^t(0 * 10/0),\ \sigma^t(s(0) * 10/s(0)),\ \sigma^t(s(s(0)) * 10/s(s(0))), \ldots\}\ ) = \{\mathbf{u}, 10, 10, \ldots\} = \mathbf{u}$.

Now, given any interpretation $\mathcal{I} = \langle \sigma^h, \sigma^t, I^h, I^t \rangle$, we define the assignments $\sigma^w_{\mathcal{I}}$ for $w \in \{h, t\}$ as follows. If $\tau \in \mathit{Terms}^0(\mathcal{C} \cup \mathcal{F})$ (i.e. not an intensional set) we can drop the $\mathcal{I}$ subindex, that is $\sigma^w_{\mathcal{I}}(\tau) \stackrel{\text{def}}{=} \sigma^w(\tau)$. If τ is an intensional set

$$\sigma^t_{\mathcal{I}}(\tau) \stackrel{\text{def}}{=} ext(\mathcal{I}, t, \tau)$$

$$\sigma^h_{\mathcal{I}}(\tau) \stackrel{\text{def}}{=} \begin{cases} ext(\mathcal{I}, h, \tau) \text{ if } ext(\mathcal{I}, h, \tau) = ext(\mathcal{I}, t, \tau) \\ \quad \boldsymbol{u} \qquad \text{otherwise} \end{cases}$$

As we can see, we have two potential sources of undefinedness. One may appear because some element in $ext(\mathcal{I}_1, t, \tau)$ cannot be evaluated, as we had before with $0 * n/0$. But a second one may occur if the extension at h is different from the one at t. For instance, for the same example, the extension of $\tau_2 = \{X : p(X)\}$ at t is $\sigma^t(ext(\mathcal{I}, t, \tau_2)) = \{0, 1, 2\}$. If we had $I_1^h = \{p(0), p(2)\}$, then the extension at h would be $ext(\mathcal{I}, h, \tau_2) = \{0, 2\} \neq ext(\mathcal{I}, t, \tau_2) = \{0, 1, 2\}$ and so $\sigma^h_{\mathcal{I}}(\tau_2) = \mathbf{u}$. We also define the interpretation $\mathit{Coh}(\mathcal{I}) \stackrel{\text{def}}{=} \langle \sigma^h_{\mathcal{I}}, \sigma^t_{\mathcal{I}}, I^h, I^t \rangle$. Note that $\mathit{Coh}(\mathcal{I})$ is determined by the interpretation of predicates and terms in $\mathit{Terms}^0(\mathcal{C} \cup \mathcal{F})$. In this sense, given a SQHT$^=_{\mathcal{F}}$-interpretation $\mathcal{I}$ for the signature $\Sigma' = \langle \mathcal{C} \cup \mathcal{S}, \mathcal{F}, \mathcal{P} \rangle$, by $\mathit{Coh}(\mathcal{I})$, we also denote the interpretation $\mathit{Coh}(\mathcal{J})$ for any interpretation $\mathcal{J}$ over the signature $\Sigma' = \langle \mathcal{C}, \mathcal{F}, \mathcal{P} \rangle$ such that $\mathcal{J} = \hat{\mathcal{I}}$.

Definition 8 (Coherent interpretation)**.** *An interpretation $\mathcal{I} = \langle \sigma^h, \sigma^t, I^h, I^t \rangle$ is said to be* coherent (w.r.t. intensional sets) *iff $\mathcal{I} = \mathit{Coh}(\mathcal{I})$.* □

Definition 9 (Satisfaction)**.** *We say that an interpretation $\mathcal{I}$* satisfies (w.r.t. intensional sets) *a formula φ at $w \in \{h, t\}$, in symbols $\mathcal{I}, w \models \varphi$, if both $\mathcal{I}$ is coherent and $\mathcal{I}, w \models_s \varphi$. We also write $\mathcal{I} \models \varphi$ when $\mathcal{I}, h \models \varphi$. Given a theory Γ, we write $\mathcal{I} \models \Gamma$ if $\mathcal{I}$ is coherent and $\mathcal{I} \models \varphi$ for all formulas $\varphi \in \Gamma$. We say that a formula φ is a* tautology *if every coherent interpretation $\mathcal{I}$ satisfies $\mathcal{I} \models \varphi$.* □

Proposition 4. *For any* SQHT$^=_{\mathcal{F}}$*-interp. $\mathcal{I}$ and 0-formula φ: $\mathcal{I} \models_{\mathcal{F}} \varphi$ iff $\mathit{Coh}(\mathcal{I}) \models \varphi$.* □

Proposition 5. *Any coherent interpretation $\mathcal{I}$ satisfies:*
i) $\mathcal{I}, w \models \varphi$ implies $\mathcal{I}, t \models \varphi$,
ii) $\mathcal{I}, w \models \neg\varphi$ iff $\mathcal{I}, t \not\models \varphi$, □

Proposition 6. *Given a formula φ, the following statements hold:*

i) if $\kappa(\varphi)$ is an $\mathrm{SQHT}^{=}_{\mathcal{F}}$ *tautology, then φ is an* $\mathrm{SQHT}^{=}_{\mathcal{S}}$ *tautology,*
ii) if φ is a 0-formula, then φ is an $\mathrm{SQHT}^{=}_{\mathcal{S}}$ *tautology iff it is a* $\mathrm{SQHT}^{=}_{\mathcal{F}}$ *tautology.* □

Nonmonotonicity is obtained with by the definition of equilibrium models for $\mathrm{SQHT}^{=}_{\mathcal{S}}$.

Definition 10 (Equilibrium model)**.** *A total (coherent) model $\mathcal{I} = \langle \sigma, I \rangle$ of a theory Γ is an* equilibrium model *if there is no interpretation $\mathcal{I}' \prec \mathcal{I}$ which is also model of Γ. A set of atoms I is a* stable model *of Γ iff $\langle \sigma, I \rangle$ is an equilibrium model of Γ for some σ.* □

Proposition 7. *Let Γ be a theory just containing 0-formulas over a signature $\Sigma = \langle \mathcal{C}, \mathcal{F}, \mathcal{P} \rangle$ and let I be a set of ground atoms. Then, I is a stable model of Γ according to Definition 10 iff I is a stable model of Γ according to Definition 4 with signature $\Sigma' = \langle \mathcal{C} \cup \mathcal{S}, \mathcal{F}, \mathcal{P} \rangle$.* □

Proposition 7 shows that our semantics is a conservative extension of (Cabalar 2011). Furthermore, as we will see later, our approach coincides with (Gelfond and Zhang 2014) and rejects *vicious circles* as shown by the following example from (Dovier *et al.* 2003).

Example 1. *Consider the following logic program P_1:*

$$r(1). \qquad q(1). \qquad p(Y) \leftarrow Y = \{X : q(X)\}$$
$$r(2). \qquad q(2) \leftarrow Z = \{X : r(X)\} \wedge p(Z).$$

□

P_1 has a unique equilibrium model $\langle \sigma_1, I_1 \rangle$ with $I_1 = \{q(1), p(\{1\}), r(1), r(2)\}$ and

$$\sigma_1(\{X : r(X)\}) \;=\; \{1, 2\} \qquad\qquad \sigma_1(\{X : q(X)\}) \;=\; \{1\}$$

Under (Dovier *et al.* 2003) semantics, there exists a second stable model $I_2 = \{q(1), q(2), p(\{1, 2\}), r(1), r(2)\}$ not corresponding to any equilibrium model. To see why, consider the coherent interpretation $\mathcal{I}'_2 = \langle \sigma^h_2, \sigma_2, I^h_2, I_2 \rangle$ with $I^h_2 = I_2 \backslash \{q(2), p(\{1, 2\})\}$ and:

$$\sigma_2(\{X : r(X)\}) = \{1, 2\} \qquad\qquad \sigma_2(\{X : q(X)\}) = \{1, 2\}$$
$$\sigma^h_2(\{X : r(X)\}) = \{1, 2\} \qquad\qquad \sigma^h_2(\{X : q(X)\}) = \boldsymbol{u}$$

Note also that any total, coherent interpretation that agrees with I_2 on the true atoms must be of the form $\mathcal{I}_2 = \langle \sigma_2, I_2 \rangle$ and that $\mathcal{I}'_2 \prec \mathcal{I}_2$. Hence, $\mathcal{I}_2$ is not an equilibrium model. Finally, note that I_2 violates the *Vicious-Circle Principle* (Gelfond and Zhang 2014) because the truth of $q(2)$ depends on $p(\{1, 2\})$ which, in its turn, depends on the fact that $\{X : q(X)\}$ contains element 2. This last fact only holds if $q(2)$ holds. □

4 Aggregates based on evaluable functions and intensional sets

From now on, we assume that $\mathcal{F}$ contains a subset $\mathcal{A}$ of function names of arity 1 used to denote aggregate names and that each aggregate name $f/1 \in \mathcal{A}$ has an associated predefined function $\hat{f} : \mathcal{S} \longrightarrow \mathcal{C} \cup \{\boldsymbol{u}\}$ that computes its value as expected (maximum, count, sum, etc). Now we further restrict Definition 8 to fix the meaning of aggregates:

Definition 11. *An interpretation $\mathcal{I} = \langle \sigma^h, \sigma^t, I^h, I^t \rangle$ is said to be* coherent (w.r.t. aggregates) *if it is coherent w.r.t. intensional sets and, in addition, it satisfies:*

i) for $f/n \in \mathcal{A}$, $\sigma^w(f(\tau)) = \hat{f}(\sigma^w(\tau))$ *if* $\sigma^w(\tau) \in \mathcal{S}$; $\sigma^t(f(\tau)) = \boldsymbol{u}$ *otherwise.*

We say that an interpretation $\mathcal{I}$ *is a* model (w.r.t. aggregates) *of a formula* φ*, in symbols* $\mathcal{I} \models \varphi$*, if both* $\mathcal{I}$ *is coherent and* $\mathcal{I} \models_s \varphi$*. Given a theory* Γ*, we write* $\mathcal{I} \models \Gamma$ *if* $\mathcal{I}$ *is coherent and* $\mathcal{I} \models_s \varphi$ *for all formulas* $\varphi \in \Gamma$. □

For the rest of the paper, we assume that the terms 'coherent' and 'model' are understood w.r.t. aggregates (Definition 11). In particular, we assume that $\mathcal{A}$ contains at least the aggregate names `count` and `sum` with the following semantics

1. $\widehat{\texttt{count}}(S) = d$ if d the number of elements in S,
2. $\widehat{\texttt{sum}}(S) = d$ if S is a set of tuples, each of which has as an integer number as first component, and d is the sum of all first components of all tuples in S,

$\hat{f}(S) = \boldsymbol{u}$ with $f \in \{\texttt{count}, \texttt{sum}\}$ otherwise. We also assume that $\mathcal{P}$ and $\mathcal{F}$ respectively contain predicates $\leq, \geq, <, >, \neq$ and evaluable functions $+, -, \times, /$ for the arithmetic relations and functions with the standard meanings. Similarly, $\mathcal{P}$ and $\mathcal{F}$ also contain the predicate $\in$ and the evaluable functions $\cup, \cap$ and $\setminus$ with the standard meanings in set theory. As usual, we use infix notation for arithmetic and set predicates and functions. We also omit the parentheses around intensional sets, so we write $\texttt{count}\{X : p(X)\}$ instead of $\texttt{count}(\{X : p(X)\})$.

Example 2. *Let* P_2 *be a theory over a signature with set of constants* $\mathcal{C} = \{a, b\}$ *that contains the rule* (1) *with* $n = 1$ *plus the fact* $p(b)$. □

The theory in Example 2 has no stable model. On the one hand, it is clear that every stable model I must satisfy $p(b) \in I$. Furthermore, $\{p(b)\}$ is not a stable model because, every coherent total interpretation $\langle \sigma, \{p(b)\}\rangle$ must satisfy $\sigma(\{X : p(X)\}) = \{b\}$ and $\sigma(\texttt{count}\{X : p(X)\}) = 1$. Hence, $\langle \sigma, \emptyset\rangle$ does not satisfy (1). On the other hand, the only other alternative is $\{p(a), p(b)\}$ and we have that $\mathcal{I} = \langle \sigma, \{p(a), p(b)\}\rangle$, with $\sigma(\{X : p(X)\}) = \{a, b\}$ and $\sigma(\texttt{count}\{X : p(X)\}) = 2$, satisfies (1). To show that $\mathcal{I}$ is not an equilibrium model, let us define $\mathcal{I}' = \langle \sigma', \sigma, I', I\rangle$ with $I' = \{p(b)\}$. It is easy to see that $\mathcal{I}' \models p(b)$. Furthermore, we have that:

$$ext(\mathcal{I}', t, \{X : p(X)\}) \;=\; \{a, b\} \neq \{b\} \;=\; ext(\mathcal{I}', h, \{X : p(X)\}) \tag{4}$$

Since these two extensions are different, it follows that $\sigma'(\{X : p(X)\}) = \mathbf{u}$ and, consequently, that $\mathcal{I}' \not\models \texttt{count}\{X : p(X)\} \geq 1$. In its turn, this implies that $\mathcal{I}'$ is also a model of rule (1) with $n = 1$ and a model of P_2. Finally, it easy to check that $\mathcal{I}' < \mathcal{I}$ and, therefore, $\mathcal{I}$ is not an equilibrium model. As we will see in Section 5, this behaviour agrees with *Alog* (Gelfond and Zhang 2014), but differs from other approaches like (Son and Pontelli 2007) and (Ferraris 2011) in which $\{p(a), p(b)\}$ is a stable model of P_2.

Interestingly, the use of evaluable functions allows defining aggregates within the logical language. First, let us recall the notion of *(directional) assignment* from (Cabalar 2011). By $f(\vec{\tau}) := \tau'$ we denote the implication[5] $(\tau' = \tau') \rightarrow f(\vec{\tau}) = \tau'$. Then, rather than providing predefined $\widehat{\texttt{max}}$ and $\widehat{\texttt{min}}$ functions, we can specify their meaning as aggregates

[5] Note that $\tau' = \tau'$ can be read as "τ' is defined."

`max` and `min` by including, instead, the formulas:

$$\texttt{max}(S) := X \quad \leftarrow \quad X \in S \wedge \neg\exists Y(Y \in S \wedge Y > X)$$
$$\texttt{min}(S) := X \quad \leftarrow \quad X \in S \wedge \neg\exists Y(Y \in S \wedge Y < X)$$

Clearly, $\texttt{max}(\emptyset)$ and $\texttt{min}(\emptyset)$ are always left undefined, because no rule body can satisfy $X \in \emptyset$. Similarly, `count` can be inductively defined in terms of addition as follows:

$$\texttt{count}(\emptyset) := 0 \tag{5}$$
$$\texttt{count}(S) := 1 + \texttt{count}(S \backslash \{Y\}) \quad \leftarrow \quad Y \in S \tag{6}$$

That is, the cardinality of the empty set is 0, and the cardinality of any other set is 1 plus the cardinality of any set obtained by removing one of its elements.[6] In general, we can easily define aggregate functions based on any associative and commutative binary function. For instance, to define the `sum` aggregate in terms of addition, we can just use:

$$\texttt{sum}(\emptyset) := 0 \tag{7}$$
$$\texttt{sum}(S) := \texttt{sum}(S \backslash \{Y\}) + Y \quad \leftarrow \quad Y \in S \tag{8}$$

If we now consider a program P_3 containing these two rules, facts $p(2)$ and $p(3)$ plus

$$q(Y) \leftarrow \texttt{sum}\{X : p(X)\} = Y \tag{9}$$

we can check that it has a unique stable model $I = \{p(2), p(3), q(5)\}$. The stable model I corresponds to the equilibrium model $\langle \sigma, I \rangle$ which satisfies the assignments: $\sigma(\texttt{sum}(\{2\})) = 2$, $\sigma(\texttt{sum}(\{3\})) = 3$ and $\sigma(\texttt{sum}(\{2, 3\})) = 5$.

5 Relation to GZ-aggregates for propositional formulas and *Alog*

A term τ is said to be *arithmetic* if it only contains variables, numbers and arithmetic functions $+$, $-$, $\times$, etc. A *(GZ-)set name* is an intensional set of the form $\{\vec{x} : \varphi\}$ where $\vec{x}$ is a tuple of variables and φ is a 0-formula. A *(GZ-)set atom* is an expression of the form $f(\tau) \preceq \tau'$ with $f \in \mathcal{A}$ an aggregate function, τ a set name, τ' an arithmetic term and $\preceq \in \{=, \leq, \geq, <, >, \neq\}$ an arithmetic relation. A *GZ-predicate atom* is an expression of the form $p(\tau_1, \ldots, \tau_n)$ with $p/n \in \mathcal{P}$ a predicate name and $\tau_1, \ldots, \tau_n \in \mathcal{D}$. A GZ-atom is either a GZ-predicate atom or a set atom. GZ-formulas are the universal closure of formulas built over GZ-atoms using the connectives $\vee$, $\wedge$ and $\rightarrow$ as usual. A GZ-theory is a set of GZ-formulas. We say that a GZ-formula φ is *ground* when there are no quantifiers and all arithmetic terms have been evaluated, that is, the only variables occurring in φ are bound to some set name and the only arithmetic terms are numbers. A GZ-theory is said to be *ground* when all its formulas are ground. The following definitions extend the semantics of *Alog* (Gelfond and Zhang 2014) to arbitrary propositional formulas (Cabalar *et al.* 2017):

Definition 12. *A set of atoms T* satisfies *a ground GZ-formula φ, denoted by $T \models_{cl} \varphi$, iff*
i) $T \not\models_{cl} \bot$

[6] For `count` and `sum`, we are assuming that set S is finite. Otherwise, we would need additional formalisation to deal with infinite sets and cardinalities.

ii) $T \models_{cl} p(\vec{c})$ *iff* $p(\vec{c}) \in T$ *for any ground atom* $p(\vec{c})$
iii) $T \models_{cl} f\{\vec{x}{:}\varphi(\vec{x})\} \trianglelefteq n$ *if* $\hat{f}\big(\{\ \vec{c} \in \mathcal{D}^{|\vec{x}|} \mid T \models_{cl} \varphi(\vec{c})\ \}\big)$ *has some value* $k \in \mathbb{Z}$ *and* $k \trianglelefteq n$ *holds for the usual meaning of arithmetic relation* $\trianglelefteq$
iv) $T \models_{cl} \varphi \wedge \psi$ *iff* $T \models_{cl} \varphi$ *and* $T \models_{cl} \psi$
v) $T \models_{cl} \varphi \vee \psi$ *iff* $T \models_{cl} \varphi$ *or* $T \models_{cl} \psi$
vi) $T \models_{cl} \varphi \to \psi$ *iff* $T \not\models_{cl} \varphi$ *or* $T \models_{cl} \psi$.
We say that T *is a* model *of a ground GZ-theory* Γ *if* $T \models_{cl} \varphi$ *for all* $\varphi \in \Gamma$. □

Given a formula $\varphi(\vec{x})$ with free variables $\vec{x}$, by $\mathtt{Gr}(\varphi(\vec{x})) \stackrel{\text{def}}{=} \{\ \varphi(\vec{c}) \mid \vec{c} \in \mathcal{D}^{|\vec{x}|}\ \}$ we denote the set of its ground instances. By $\mathtt{Gr}(\Gamma) \stackrel{\text{def}}{=} \bigcup\{\ \mathtt{Gr}(\varphi(\vec{x})) \mid \forall\vec{x}\varphi(\vec{x}) \in \Gamma\ \}$ we also denote the grounding of a theory Γ. Furthermore, given some set of atoms T, we divide any theory Γ into the two disjoint subsets: $\Gamma^+_T \stackrel{\text{def}}{=} \{\varphi \in \Gamma \mid T \models_{cl} \varphi\}$ and $\Gamma^-_T \stackrel{\text{def}}{=} \Gamma \setminus \Gamma^+_T$, that is, the formulas in Γ satisfied by T and not satisfied by T, respectively. When set Γ is parametrized, say $\Gamma(z)$, we write $\Gamma^+_T(z)$ and $\Gamma^-_T(z)$ instead of $\Gamma(z)^+_T$ and $\Gamma(z)^-_T$. We also omit the set brackets when the theory is a singleton. For instance, $\mathtt{Gr}^+_T(\varphi)$ collects the formulas from $\mathtt{Gr}(\{\varphi\})$ satisfied by T.

Definition 13. *The reduct of a ground GZ-formula* φ *w.r.t. a set of atoms* T *written* φ^T*, is defined as* $\bot$ *if* $T \not\models_{cl} \varphi$*, otherwise:*

$$\varphi^T \stackrel{\text{def}}{=} \begin{cases} a & \text{if } \varphi \text{ is some atom } a \in I \\ \big(\bigwedge \mathtt{Gr}^+_T(\psi(\vec{x}))\big)^T & \text{if } T \models_{cl} \varphi \text{ with } \varphi = f\{\vec{x}{:}\psi(\vec{x})\} \trianglelefteq n \\ \varphi_1^T \otimes \varphi_2^T & \text{if } T \models_{cl} \varphi \text{ and } \varphi = (\varphi_1 \otimes \varphi_2) \text{ for some } \otimes \in \{\wedge, \vee, \to\} \end{cases}$$

T *is a* stable model *of a GZ-theory* Γ *iff* T *is the* $\subseteq$*-minimal model of* $\mathtt{Gr}(\Gamma)^T$. □

Theorem 1. *For any GZ-theory* Γ*, a set of atoms* T *is a stable model of* Γ *according to Definition 13 iff* T *is a stable model of* Γ *according to Definition 10.* □

Let us return to Example 2 and recall we have shown that program P_2 has no stable model according to Definition 10. To illustrate the behaviour of this program with respect to *Alog* (Definition 13), note first that the only possible candidate for being a stable model is the set $I = \{p(a), p(b)\}$; otherwise, the rules would not be satisfied according to Definition 12. Then, we have that P_2^I corresponds to the normal program

$$p(a) \leftarrow p(a) \wedge p(b)$$
$$p(b)$$

whose unique $\subseteq$-minimal model is $\{p(b)\}$. Hence, I is not a stable model of P_2. Intuitively, this is explained by the fact that our belief in $p(a)$ depends on the extension of intensional set $\{X : p(X)\}$ which, in its turn, depends on our belief in $p(a)$, forming what Gelfond and Zhang (2014) call a "vicious circle." According to the *vicious circle principle*, set I should be rejected as a stable model. In our approach, the "vicious circle" can be easily spotted by observing that evaluation of the set in world h is left undefined because its extension is different from the one at world t, as shown in (4).

As an example of non-vicious definition, consider the following variation.

Example 3. *Let* P_4 *be the following program:*

$$p(a) \leftarrow \mathtt{count}\{X : p(X) \wedge X \neq a\} \geq 1 \tag{10}$$
$$p(b)$$

□

Again, the only candidate interpretation that satisfies all rules is $I = \{p(a), p(b)\}$, but the reduct corresponds now to:

$$p(a) \leftarrow p(b)$$
$$p(b)$$

whose unique minimal model is I, becoming a stable model under Definition 13. Therefore, the same will happen under Definition 10. Let us put $\tau = \{X : p(X) \wedge X \neq a\}$. It is easy to see that $\mathcal{I} = \langle \sigma, I \rangle$, with $\sigma(\tau) = \{b\}$ and $\sigma(\texttt{count}(\tau)) = 1$, satisfies (10) and obviously $p(b)$ as well. Now take the smaller interpretation $\mathcal{I}' = \langle \sigma', \sigma, I', I \rangle$ with $I' = \{p(b)\}$. Then, we have:

$$ext(\mathcal{I}', t, \tau) \ = \{b\} = \ ext(\mathcal{I}', h, \tau)$$

so $\sigma'(\tau) = \{b\}$ and, consequently, $\mathcal{I}' \models \texttt{count}(\tau) \geq 1$. In its turn, this implies that $\mathcal{I}'$ does not satisfy (10) and so is not a model of P_4. It is easy to see that there is no other smaller interpretation $\mathcal{I}'' < \mathcal{I}$ that satisfies $p(b)$, and so $\mathcal{I}$ is an equilibrium model.

An interesting property of $\mathcal{A}log$ is that

it is always possible to introduce auxiliary variables for the aggregate value. For instance, we can always replace (1) by:

$$p(a) \leftarrow \texttt{count}\{X : p(X)\} = Y \wedge Y \geq n \tag{11}$$

This transformation is not safe in other semantics such as (Son and Pontelli 2007; Faber *et al.* 2011; Ferraris 2011). In particular, under these semantics, if we take $n = 0$, a program consisting of (1) has a unique stable model $\{p(a)\}$ while a program consisting of (11) has no stable model. These two programs are equivalent in $\mathcal{A}log$ and have no stable model. We can generalize the safety of this transformation to any context, using $\mathrm{SQHT}^{=}_{\mathcal{S}}$:

Proposition 8 (Existential variable introduction)**.** *Let $p(\tau_1, \ldots, \tau_i, \ldots, \tau_n)$ be an atom. Then, $p(\tau_1, \ldots, \tau_i, \ldots, \tau_n)$ is equivalent in $\mathrm{SQHT}^{=}_{\mathcal{S}}$ to $\exists x[x = \tau_i \wedge p(\tau_1, \ldots, x, \ldots, \tau_n)]$.* □

From Proposition 8, it immediately follows that rule (1) is equivalent to (11) with an existential quantifier $\exists Y$ in the body that can be trivially removed.

6 Conclusions and Related Work

We have extended Quantified Equilibrium Logic with evaluable, partial functions by introducing *intensional sets*, that is, terms that allow defining elements in a set that satisfy some function or condition. By providing a logical interpretation, we define the semantics of these new expressions without any kind of syntactic restriction, so they can be arbitrarily nested within standard logical terms and formulas. The new extension yields a natural interpretation of an aggregate as an evaluable function applied to a set term. As a result, the semantics of an aggregate can be fixed within the logical language, by the addition of formulas fixing its meaning, rather than relying on an external, predefined function (although we assume that some elementary set predicates are available). This may become a powerful theoretical tool to analyse the fundamental properties of aggregate functions. We have also proved that, when restricted to the

syntactic fragment of language $\mathcal{A}log$, our semantics coincides with that of (Gelfond and Zhang 2014).

Extensions in Logic Programming with sets can be traced back to (Kuper 1990) and (Beeri *et al.* 1991). The approach of (Dovier *et al.* 2003) is based on the stable model semantics with a reduct definition, but does not include evaluable functions or allow complex terms (beyond simple variables) to appear as the definition of intensional sets. Still, an important difference for the common syntactic fragment is that (Dovier *et al.* 2003) does not satisfy the *vicious-circle free principle* as defined in (Gelfond and Zhang 2014) "no object or property can be introduced by the definition referring to the totality of objects satisfying this property" (see Example 1). The approaches (Lee and Meng 2009; Ferraris and Lifschitz 2010) or (Harrison *et al.* 2017) do not satisfy this principle either, but still share our orientation of defining the semantics of individual components of an aggregate. An important difference, however, is that these formalisations do not deal with general *evaluable* (i.e. non-Herbrand) functions, while we use them as a starting point and then understand aggregates just as one more case whose arguments happen to be sets. This allows us a completely arbitrary use of aggregates as terms and of terms inside aggregates, leading to expressive constructions such as (5)-(6).

Our future work will include the study of non-strict functions and their relation to (Son and Pontelli 2007; Ferraris 2011). We will also study the possible formalisation under Free Logics as in (Cabalar *et al.* 2014) or the kind of properties that allow functions to be recursively defined as in (5)-(6), and their application to (Gelfond and Zhang 2014).

Supplementary Material

To view supplementary material for this article, please visit http://dx.doi.org/10.1017/S1471068418000169.

References

Balduccini, M. 2013. ASP with non-herbrand partial functions: a language and system for practical use. *TPLP 13,* 4-5, 547–561.

Baral, C. 2003. *Knowledge representation, reasoning and declarative problem solving.* Cambridge university press.

Bartholomew, M. and Lee, J. 2014. Stable models of multi-valued formulas: Partial versus total functions. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014*, C. Baral, G. D. Giacomo, and T. Eiter, Eds. AAAI Press.

Beeri, C., Naqvi, S. A., Shmueli, O., and Tsur, S. 1991. Set constructors in a logic database language. *J. Log. Program. 10,* 3&4, 181–232.

Cabalar, P. 2011. Functional answer set programming. *Theory and Practice of Logic Programming 11,* 2-3, 203–233.

Cabalar, P. 2013. Setting the stage for ASP functions. *ALP Newsletter*.

Cabalar, P., Fandinno, J., Schaub, T., and Schellhorn, S. 2017. Gelfond-Zhang aggregates as propositional formulas. In *Logic Programming and Nonmonotonic Reasoning - 14th International Conference, LPNMR 2017, Espoo, Finland, July 3-6, 2017, Proceedings*, M. Balduccini and T. Janhunen, Eds. LNCS, vol. 10377. Springer, 117–131.

Cabalar, P., Fariñas del Cerro, L., Pearce, D., and Valverde, A. 2014. A free logic for stable models with partial intensional functions. In *Proc. of the 14th European Conference on*

Logics in Artificial Intelligence (JELIA'14). Lecture Notes in Artificial Intelligence, vol. 8761. 340–354.

Cabalar, P., Kaminski, R., Ostrowski, M., and Schaub, T. 2016. An ASP semantics for default reasoning with constraints. In *Proc. of the 25th Intl. Joint Conference on Artificial Intelligence (IJCAI'16), New York, USA, July 9-15th, 2016*, S. Kambhampati, Ed. 1015–1021.

Dovier, A., Omodeo, E. G., Pontelli, E., and Rossi, G. 1991. {log}: A logic programming language with finite sets. In *Logic Programming, Proceedings of the Eigth International Conference, Paris, France, June 24-28, 1991*, K. Furukawa, Ed. MIT Press, 111–124.

Dovier, A., Pontelli, E., and Rossi, G. 2003. Intensional sets in CLP. In *Logic Programming, 19th International Conference, ICLP 2003, Mumbai, India, December 9-13, 2003, Proceedings*, C. Palamidessi, Ed. Lecture Notes in Computer Science, vol. 2916. Springer, 284–299.

Faber, W., Pfeifer, G., and Leone, N. 2011. Semantics and complexity of recursive aggregates in answer set programming. *Artificial Intelligence 175,* 1, 278–298.

Ferraris, P. 2011. Logic programs with propositional connectives and aggregates. *ACM Transactions on Computational Logic 12,* 4, 25.

Ferraris, P. and Lifschitz, V. 2010. On the stable model semantics of first-order formulas with aggregates. In *Proceedings of the 2010 Workshop on Nonmonotonic Reasoning*.

Gelfond, M. and Zhang, Y. 2014. Vicious circle principle and logic programs with aggregates. *Theory and Practice of Logic Programming 14,* 4-5, 587–601.

Harrison, A., Lifschitz, V., and Raju, D. 2017. Program completion in the input language of GRINGO. *TPLP 17,* 5-6, 855–871.

Heyting, A. 1930. Die formalen Regeln der intuitionistischen Logik. *Sitzungsberichte der Preussischen Akademie der Wissenschaften, Physikalisch-mathematische Klasse*, 42–56.

Kuper, G. M. 1990. Logic programming with sets. *J. Comput. Syst. Sci. 41,* 1, 44–64.

Lee, J. and Meng, Y. 2009. On reductive semantics of aggregates in answer set programming. In *Logic Programming and Nonmonotonic Reasoning*, E. Erdem, F. Lin, and T. Schaub, Eds. Springer Berlin Heidelberg, Berlin, Heidelberg, 182–195.

Lin, F. and Wang, Y. 2008. Answer set programming with functions. In *Proc. of the 11th Intl. Conf. on Principles of Knowledge Representation and Reasoning (KR'08)*.

Pearce, D. 1996. A new logical characterisation of stable models and answer sets. In *Non monotonic extensions of logic programming. Proc. NMELP'96. (LNAI 1216)*. Springer-Verlag.

Pearce, D. and Valverde, A. 2004. Towards a first order equilibrium logic for nonmonotonic reasoning. In *Logics in Artificial Intelligence, 9th European Conference, JELIA 2004, Lisbon, Portugal, September 27-30, 2004, Proceedings*, J. J. Alferes and J. A. Leite, Eds. Lecture Notes in Computer Science, vol. 3229. Springer, 147–160.

Pelov, N., Denecker, M., and Bruynooghe, M. 2007. Well-founded and stable semantics of logic programs with aggregates. *Theory and Practice of Logic Programming 7,* 3, 301–353.

Simons, P., Niemelä, I., and Soininen, T. 2002. Extending and implementing the stable model semantics. *Artificial Intelligence 138,* 1-2, 181–234.

Son, T. C. and Pontelli, E. 2007. A constructive semantic characterization of aggregates in answer set programming. *Theory and Practice of Logic Programming 7,* 3, 355–375.