Gradual Semantics for Logic-Based Bipolar Graphs Using T-(Co)norms
Martin Jedwabny, Madalina Croitoru, Pierre Bisquert

To cite this version:

HAL Id: lirmm-03037284
https://hal-lirmm.ccsd.cnrs.fr/lirmm-03037284
Submitted on 3 Dec 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License
Gradual Semantics for Logic-Based Bipolar Graphs Using T-(Co)nons

Martin Jedwabny¹ and Madalina Croitoru² and Pierre Bisquert³

Abstract. In this paper we consider a bipolar graph structure encoding conflicting knowledge with logic formulas. Gradual semantics provide a way to assign strength values in the unit interval to nodes (i.e. logical inference steps) in the bipolar graph. Here, we introduce a new class of semantics based on the notions of T-norms and T-conorms and show that they handle circular reasoning and satisfy desirable properties such as provability and rewriting.

1 INTRODUCTION

Techniques for query answering over knowledge bases with existential rules mostly rely on the two classical ways of processing rules, namely forward chaining and backward chaining.

In forward chaining (i.e. chase [11]) the rules are applied to enrich the fact base and query answering is performed against the resulting saturation.

The backward chaining process is divided into two steps: first, the query is rewritten using the rules into a first-order query, then, second, the rewritten query is evaluated against the fact base. Depending on the considered class of existential rules the chase and/or query rewriting may terminate or not [6].

When considering knowledge bases where conflict arises, in order to avoid the explosion principle, one has to deploy reasoning mechanisms that do not follow the classical logical inference. In this paper we consider Statement Graphs, a bipolar (i.e. two-colored edges) graph structure that is able to capture argumentation semantics, defeasible reasoning semantics [7] and maxi consistent reasoning semantics [17] via node labeling functions. The nodes of this graph (statements) contain logical formulae while the edges represent support and attack (respectively) between them.

Following on the steps of several semantics in the literature defined for abstract argumentation frameworks [1, 9] and bipolar argumentation frameworks [3], we consider how to define gradual semantics for statement graphs.

To this end we adapt existing principles of gradual semantics to the context of statement graphs, taking into account the fact that the nodes of our graphs are composed of logical formulae. We define a scheme for generating gradual semantics based on the notions of T-norms and T-conorms in order to satisfy those principles.

The significance of our work is two fold:

1. We extend Statement Graphs with T-norm and T-conorm inspired operators allowing to reason in a non boolean manner.
2. We propose a semantics for bipolar graphs that can handle circular reasoning.

The paper is structured as follows. The Preliminaries section introduces the basic notions of statement graphs. The Existing semantics for statement graphs section places our semantics within the state of the art, and two main guiding principles they should satisfy based on its specificity. The New semantics based on T-(co)norms section presents the newly introduced gradual semantics while the Properties of the proposed semantics section discusses its properties. Finally, we conclude with the Discussion section.

2 PRELIMINARIES

We consider a first order language $\mathcal{L}$ composed of (possibly infinite) constants $C = \{a, b, \ldots\}$, variables $V = \{X, Y, \ldots\}$, null variables $N = \{\text{Null}_1, \text{Null}_2, \ldots\}$, no additional function symbols (other than constants), a (finite) set of predicates $\mathcal{P}$, and formulas built with the quantifiers ($\exists, \forall$) and the connectors for implication ($\rightarrow$) and conjunction ($\land$). An atom (or atomic formula) is $\top, \bot$ (i.e. top, bottom) or of the form $t = p(t_1, \ldots, t_k)$, where $p$ is a predicate and $t_i$ are the terms i.e. $t_i \in C \cup V \cup N$. Given a formula $\phi$ of $\mathcal{L}$, terms($\phi$) and vars($\phi$) are the set of terms and variables of $\phi$.

A fact on $\mathcal{L}$ is a formula of the form $\exists \vec{X} p(\vec{a}, \vec{X})$, where $\vec{a}$ and $\vec{X}$ are tuples of constants and variables respectively.

Rules are used as an inference mechanism in order to generate new knowledge. We consider here defeasible rules $r$: formulae of the form $\forall \vec{X}, \vec{Y}, Body(\vec{X}, \vec{Y}) \rightarrow \exists \vec{Z} Head(\vec{X}, \vec{Z})$ such that $\vec{X}, \vec{Y}$ are tuples of variables, $\vec{Z}$ is a tuple of (existential) variables, $Body(\vec{X}, \vec{Y})$, $Head(\vec{X}, \vec{Z})$ are finite non-empty conjunctions of atoms, also denoted $Body(r)$ and $Head(r)$. Defeasible rules express a weak implication i.e. if $Body(r)$ then generally $Head(r)$ also holds. We call them fact rules when $Body(r) = \top$.

A knowledge base is a tuple $KB = (\mathcal{T}, R, N)$, where $\mathcal{T}$ is a set of fact rules, i.e. $\mathcal{T} \rightarrow \exists \vec{X} p(\vec{a}, \vec{X}), R$ is a set of non-fact rules, i.e. $Body(r) \neq \top, N$ is the negative constraints $\forall \vec{X} Body(\vec{X}) \rightarrow \bot$ expressing incompatibility.

Example 1. Let us consider an example about whether we should or not keep a dog named oscar. The knowledge we formalize below is the following: Oscar has a collar, a GPS tracker and is found alone. If an animal is found alone then he is usually put for adoption; if he has a collar then he usually has an owner; if he has a GPS tracker then he usually has an owner. An animal with an owner is usually kept. It is impossible to be kept and put for adoption at the same time.

$$\mathcal{T} = \{ \top \rightarrow \text{alone(oscar)}, \top \rightarrow \text{hasCollar(oscar)},$$

$$\top \rightarrow \text{hasGPS(oscar)} \}$$
\[ R = \{ \forall X \, \text{alone}(X) \rightarrow \text{adoption}(X) \} \]
\[
\forall X \, \text{hasCollar}(X) \rightarrow \exists Y \, \text{hasOwner}(X, Y) \quad \forall X \, \text{hasGPS}(X) \rightarrow \exists Y \, \text{hasOwner}(X, Y) \\
\forall X, Y \, \text{hasOwner}(X, Y) \rightarrow \text{keep}(X) \}
\]
\[ N = \{ \forall X \, \text{adoption}(X) \wedge \text{keep}(X) \rightarrow \bot \} \]

In order to characterize and reason with knowledge bases, [17] proposed statement graphs, a general graph based representation containing statements that support and attack each other. A statement \( s = (\Phi \rightarrow \psi) \) expressed in \( \mathcal{L} \) contains a logic formula in \( \mathcal{L} \) where \( \Phi \) is either \( \top \) or a non-empty conjunction of grounded atoms and \( \psi \) is a grounded atom. We denote \( \text{Premise}(s) = \Phi \) and \( \text{Conc}(s) = \psi \). We call \( s \) a query statement iff \( \psi = \emptyset \) and \( \Phi \neq \top \).

We can retrieve the statements produced by a knowledge base \( KB = (T, R, \neg, N) \) expressed in \( \mathcal{L} \) through forward or backwards chaining. A statement graph expressed in \( \mathcal{L} \), is a directed graph \( SG = (V, E_S, E_A) \):

- \( V \) is a set of statements.
- \( E_S \subseteq V \times V \) is the set of support edges. There is a support edge \( e = (s_1, s_2) \in E_S \) iff \( \text{Conc}(s_1) \subseteq \text{Premise}(s_2) \).
- \( E_A \subseteq V \times V \) is the set of attack edges. There is an attack edge \( e = (s_1, s_2) \in E_A \) iff the statement \( s_1 \) undercuts \( s_2 \) on a premise \( f \), i.e. \( \exists f \in \text{Premise}(s_2) \) s.t. \( f \) and \( \text{Conc}(s_1) \) are in conflict w.r.t. a set of negative constraints \( N \).

A comprehensive description on retrieving statements using chase can be found in [16]. However, such procedures are out of the scope of this paper. Here, for a given knowledge base and a query we consider the statement graph already built.

Example 2. Let us depict in Figure 1 the statement graph for Example 1 and query \( Q = (\text{keep}(\text{oscar}) \rightarrow \emptyset) \). In this graph, the statement \( s_1 = (\top \rightarrow \text{alone}(\text{oscar})) \) supports \( s_2 = (\text{alone}(\text{oscar}) \rightarrow \text{adoption}(\text{oscar})) \) as the conclusion of the first is included in the premise of the second. Also, \( s_2 \) attacks \( Q \) as \( \text{oscar} \) cannot be given for adoption and kept as we model the negative constraint \( \forall X \, \text{adoption}(X) \wedge \text{keep}(X) \rightarrow \bot \) as an attack edge.

![Figure 1: Statement graph of Example 1.](image-url)

Let us now introduce a key notion for the remainder of the paper. The notion of complete support describes the situation where a statement has a supporting statement for each one of its premises.

A complete support for a statement \( s \) is a set of statements \( C \) such that:

- \( \forall f \in \text{Premise}(s) \), \( \exists s' \in C \) such that \( s' \) supports \( s \) on \( f \).
- \( \not\exists C' \subseteq C \) such that \( C' \) is a complete support for \( s \) (minimality w.r.t. set inclusion).

We also say that the empty set \( \emptyset \) is a complete support for any fact statement \( s = (\top \rightarrow \psi) \) i.e. \( \text{Premise}(s) = \emptyset \). Then, we can extend the notion of complete support to trees as follows. A complete support tree for a statement \( s \) is a set of statements \( C \) such that:

- \( s \in C \).
- \( \forall s', s'' \in C, \exists f \in \text{Premise}(s') \), \( \exists s''' \in C \) such that \( s''' \) supports \( s' \) on \( f \).
- \( \not\exists s', s'' \in C \) such that \( s' \) attacks \( s'' \).
- \( \not\exists s', s'' \in C \) such that there is a directed cycle of support edges between \( s' \) and \( s''' \).
- \( \not\exists C' \subseteq C \) such that \( C' \) is a complete support tree for \( s \) (minimality w.r.t. set inclusion).

We denote \( \text{CST}(s) \) the set of complete support trees of \( s \). We say that a complete support tree \( C \in \text{CST}(s) \) attacks \( C' \in \text{CST}(s') \) with respect to a statement graph \( SG \) iff \( \exists s''' \in C' \) such that \( (s, s''') \in E_A \).

Example 3. In the statement graph of Figure 1, statement \( (\top \rightarrow \text{hasCollar}(\text{oscar})) \) is a complete support for \((\text{hasCollar}(\text{oscar}) \rightarrow \text{hasOwner}(\text{oscar}, \text{Null}))\).

Moreover, the set of statements \( (\top \rightarrow \text{hasCollar}(\text{oscar})), (\text{hasCollar}(\text{oscar}) \rightarrow \text{hasOwner}(\text{oscar}, \text{Null})), (\text{hasOwner}(\text{oscar}, \text{Null}) \rightarrow \text{keep}(\text{oscar})), (\text{keep}(\text{oscar}) \rightarrow \emptyset)) \) is a complete support tree for the query statement \( (\text{keep}(\text{oscar}) \rightarrow \emptyset)) \).

3 EXISTING SEMANTICS FOR STATEMENT GRAPHS

In order to evaluate statements, [17] proposed the use of labeling functions, which assign a truth value to each of the statements.

In this paper, we propose the use of gradual semantics to finer grain characterize the set of statements from the most accepted one(s) to the weakest one(s).

To this end we will need two notions, both expressed as values between 0 and 1. On one hand we will need to characterize the confidence in each statement. On the other we need to characterize the confidence that complete support trees leading to a statement are correct.

Definition 1. (Gradual semantics) Given a statement graph \( SG = (V, E_S, E_A) \), a gradual semantics \( W \) takes the statement graph and an inner strength function \( I : V \rightarrow [0, 1] \) and produces an outer strength value for each statement, i.e.:

\[ W(SG, I) : O : V \rightarrow [0, 1]. \]

We also denote the outer strength of \( s \in V \) as \( W_{SG, I}(s) = O(s) \).

Unlike abstract argumentation, statement graph nodes contain formulas. Their content plays an important role in the evaluation of other nodes, so our semantics should take it into account. Consider the following examples in Figure 2:

In case (a) the statement \( A(a) \wedge B(a) \rightarrow C(a) \) has a support for each of its premises, which is not the case in (b).
Gradual semantics have been studied in the literature in the context of argumentation frameworks.

Let us explain how the previous properties render such semantics not usable for Statement Graphs. For instance, consider the euler-based semantics as defined by [3]. In this work, the authors present a semantics for bipolar abstract argumentation.

In our terms, the outer strength of a statement in the graph would be determined as:

\[ O(s) = 1 - \frac{1 - I(s)^2}{1 + I(s)^2}, \]

where

\[ E = \sum_{s' \text{ supports } s} O(s') - \sum_{s' \text{ attacks } s} O(s'). \]

**Figure 2**: Support in statement graphs.

In the second case that same inference step would be unjustified under any presumption for the given graph.

To account for such a property we define the notion of provability.

**Definition 2. (Provability)** Let \( SG = (\mathcal{V}, E_S, E_A) \) be a statement graph, \( I : \mathcal{V} \rightarrow [0, 1] \) and \( W \) a gradual semantics such that \( W(SG, I) = O \). We say that \( W \) satisfies provability iff \( \forall s \in \mathcal{V} \) such that \( s \) is unsupported (i.e. there is a premise without any supporting statement) then \( O(s) = 0 \).

Moreover, we desire our semantics to be resistant to some forms of rule rewriting. Consider the following examples in Figure 3:

Due to the fact that arguments are considered in an abstract manner, these semantics cannot satisfy the properties defined above. Consider Figure 4, statement \((A(a) \land B(a) \land C(a) \rightarrow D(a))\) does not have a support for each of its premises, yet its outer strength is equal to 1 (thus violating provability).

**Figure 4**: Euler-based semantics.

**Figure 3**: Rule rewriting

In case (a) we have three statements containing fact rules supporting \( s_1 = (A(a) \land B(a) \land C(a) \rightarrow D(a)) \). In case (b) we rewrite \( s_1 \) by factoring out \( A(a) \) and \( B(a) \) through the intermediate statement \( s_2 = (A(a) \land B(a) \rightarrow D'(a)) \) and \( s_3 = (D'(a) \land C(a) \rightarrow D(a)) \).

We would like our semantics to consider such rewriting, as the length of a proof in our logical language must not necessarily affect our confidence in it.

**Definition 3. (Rewriting)** Given \( SG = (S \cup \{s_1, s_2, s_3\}, E_S, E_A) \), and \( I : S \cup \{s_1, s_2, s_3\} \rightarrow [0, 1] \), where:

- \( s_1 = (L_1 \land \ldots \land L_n \land L_{n+1} \rightarrow L) \),
- \( s_2 = (L_1 \land \ldots \land L_n \rightarrow L') \),
- \( s_3 = (L' \land L_{n+1} \rightarrow L) \),
- \( L' \) is a fresh atom (i.e. not in other statements of \( S \)),
- \( I(s_1) = I(s_3) = x \) and \( I(s_2) = 1 \).

We say that a gradual semantics \( W \) satisfies rewriting iff \( W_{SG, I}(s_1) = W_{SG, I}(s_3) \).

**4 NEW SEMANTICS BASED ON T-(CO)NORMS**

In this section, we present a new class of gradual semantics for Statement Graphs based on the concepts of T-norms and T-conorms (also called S-norms). These are inspired from and generalize the concepts of conjunction and disjunction (respectively) in the context of fuzzy logics [19, 8].

A **T-norm** (i.e. triangular norm) is a binary operation \( \otimes : [0, 1] \times [0, 1] \rightarrow [0, 1] \) satisfying the following conditions:

- \( \otimes(x, y) = \otimes(y, x) \) (i.e. commutativity)
- \( \otimes(x, \otimes(y, z)) = \otimes(\otimes(x, y), z) \) (i.e. associativity)
- \( y \leq z \Rightarrow \otimes(x, y) \leq \otimes(x, z) \) (i.e. monotonicity)
- \( \otimes(x, 1) = x \) (i.e. neutral element 1)

Some examples are the minimum \( \otimes(x, y) = \min(x, y) \), the product \( \otimes(x, y) = x \cdot y \) and Lukasiewicz t-norm \( \otimes(x, y) = \max(x + y - 1, 0) \).

A **T-conorm** (also called S-norm) is a binary operation \( \oplus : [0, 1] \times [0, 1] \rightarrow [0, 1] \) satisfying the following conditions:

- \( \oplus(x, y) = \oplus(y, x) \) (i.e. commutativity)
- \( \oplus(x, \oplus(y, z)) = \oplus(\oplus(x, y), z) \) (i.e. associativity)
- \( y \leq z \Rightarrow \oplus(x, y) \leq \oplus(x, z) \) (i.e. monotonicity)
- \( \oplus(x, 0) = x \) (i.e. neutral element 0)

The maximum function \( \oplus(x, y) = \max(x, y) \), probabilistic sum \( \oplus(x, y) = x + y - x \cdot y \) and Lukasiewicz t-conorm \( \oplus(x, y) = \min(x + y, 1) \) are all examples of T-conorms.

A **negation** is a function \( \neg : [0, 1] \rightarrow [0, 1] \) such that \( \neg(0) = 1 \) and \( \neg(1) = 0 \). It is **strict** if \( \neg \) is continuous and strictly decreasing.

A negation is **strong** if it is strict and satisfies \( \neg(\neg(x)) = x \).
The most widely used strong negation is the standard negation \( \neg(x) = 1 - x \).

The standard negation introduced t-norms as duals of t-norms. Given a t-norm \( \otimes \) and a strict negation \( \neg \), one obtains a t-conorm \( \oplus \), which is \( \neg \)-dual to \( \otimes \) as:

\[
\oplus(x, y) = \neg^{-1}(\otimes(\neg(x), \neg(y)))
\]

A triple \((\otimes, \oplus, \neg)\), where \( \otimes \) is a t-norm, \( \oplus \) is a t-conorm and \( \neg \) is a negation is called a De Morgan triple iff \( \forall x, y \in [0, 1]: \)

- \( \otimes(x, y) = \neg(\neg(x), \neg(y)) \)
- \( \oplus(x, y) = \neg(\otimes(\neg(x), \neg(y))) \)

Given a t-norm \( \otimes \), \((\otimes, \oplus, \neg)\) is a De Morgan triple if and only if \( \neg \) is a strong negation and \( \oplus \) is the \( \neg \)-dual of \( \otimes \).

We generalize T-norms and T-conorms for \( n \) parameters inductively i.e.

- \( \otimes(\emptyset) = 1 \), and \( \otimes(x_i) = \otimes(x_0, x_i) \), \( i = 1, \ldots, n \).
- \( \oplus(\emptyset) = 0 \), and \( \oplus(x_i) = \oplus(x_0, x_i) \), \( i = 1, \ldots, n \).

To quantify the strength of a statement, we need to define a way to aggregate the strength of all the statements supporting and attacking each of its premises.

Given a statement \( s = (p_1 \land \ldots \land p_n \rightarrow c) \), we use T-norms (representing conjunction) to aggregate the strength of each complete support tree that leads to \( s \) separately. Then, we use T-conorms (representing disjunction) to aggregate the strength of all the mentioned trees together. We do the same process for every statement which attacks \( s \) as well. Finally, we use the negation \( \neg \) to capture how the strength of complete support trees attacking another affects the final strength of a statement.

**Definition 4. (T-norm semantics)** Let \( SG = (V, E_S, E_A) \) be a statement graph, \( I : V \mapsto [0, 1] \) an inner strength function, and \( DT = (\otimes, \oplus, \neg) \) a De Morgan triple, we define the T-norm semantics as \( W_{DT}(SG, I) = O \) such that:

\[
O(s) = \bigoplus_{c \in CST(s)} O(C)
\]

\[
O(C) = I(C) \otimes \neg \bigoplus_{c' \in CST(s', E \in V) \text{Where C attacks C'}} I(C')
\]

\[
I(C) = \bigotimes_{s \in C} I(s)
\]

Recall that \( CST(s) \) denotes the complete support trees of \( s \).

The intuition of this definition is that the outer strength of a statement \( O(s) \) is given by the disjunction (e.g. max) of outer strength value of all the complete support trees justifying it. The outer-strength of a CST \( O(C) \) is given by the conjunction (e.g. min) between the inner strength of all the statements contained by it \( I(C) \) and the inner-strength of all the CSTs attacking it \( I(C') \).

Please note that this definition makes our semantics not to be impacted by cycles. Indeed, the inner strength of statements are part of the input, the outer strength of CSTs only depend on those inner strengths, and the outer strengths of single statements only depend on outer strengths of CSTs. Therefore they only depend on the inner strength as given by the input.

Also, note that both the time and space complexity of an algorithm that computes the outer strength of every statement would depend on the amount of CSTs. For every CST, one has to check if it contains a cycle, and iterate over all the CSTs that attack it. Each CST is a directed tree in the graph, and an algorithm will find \((n - 1)^2\) subtrees in a general directed graph with \( n \) nodes in the worst case. However, such case will not happen in practice because of the induced structure caused by the logical formulae.

**Example 4.** Consider the statement graph \( SG = (V, E_S, E_A) \) and inner strength function \( I : V \mapsto [0, 1] \) as depicted in Figure 5, alongside the triple \( DT = (\otimes, \oplus, \neg) \) given by:

- \( \otimes(x, y) = x \cdot y \) (multiplication),
- \( \oplus(x, y) = x + y - x \cdot y \) (probabilistic sum), and
- \( \neg(x) = 1 - x \) (standard negation).

Statements containing fact rules, such as \( s_1 = (\top \rightarrow A(a)) \), receive an outer strength equivalent to their inner strength i.e. \( O(s_1) = I(s_1) = 0.6 \).

The outer strength of rules only supported by a fact rule which are not attacked, such as \( s_5 = (B(a) \rightarrow A(a)) \), depend on the conjunction of their own inner strength and that of their support, which in this case is implemented as the multiplication function i.e. \( O(s_5) = I(s_5) \cdot I(s_5) = 0.5 \).

To calculate the strength of a statement that is both supported and attacked, such as \( s_7 \), we take the conjunctive strength of its complete support trees \( \otimes(I(s_1), I(s_5), I(s_7)) = 0.3 \) and \( \otimes(I(s_2), I(s_5), I(s_4), I(s_7)) = 0.25 \).

Then, we repeat the process for the statements attacking any complete support of \( s_7 \) (in this case the one leading to \( s_8 \)): \( \otimes(\otimes(I(s_3, s_6, s_8), s_7)) = 0.2 \).

Finally, we merge them into a single outer strength value as \( O(s_7) = (0.3 \otimes -0.2) \oplus (0.25 \otimes -0.2) = 0.392 \).

Let us exemplify how our semantics handles circular reasoning and contradiction with the defined T-(co)norm operators.

**Example 5.** Now, consider the statement graph \( SG = (V, E_S, E_A) \) and inner strength function \( I : V \mapsto [0, 1] \) as depicted in Figure 6, alongside the triple \( DT = (\otimes, \oplus, \neg) \) as defined previously:
As before, statements containing fact rules have the same inner and outer strength values, i.e. $O(s_1) = T(s_1)$, $O(s_3) = T(s_3)$, $O(s_5) = T(s_5)$, and $O(s_9) = T(s_9)$.

Then, to get the outer strength of statement $s_2$, we observe that the statement graph contains a cycle leading to it. In particular, the inference chain $\{s_3, s_4, s_6, s_8, s_2\}$ is not a CST of $s_2$ because it contains a cycle of support edges. Therefore, it does not improve the outer strength of $s_2$. Concretely, $CST(s_2) = \{\{s_1, s_2\}, \{s_3, s_4, s_5, s_6, s_7\}, \{s_8, s_9, s_2\}\}$. Thus, $O(s_2) = O(s_1) \otimes O(s_2) \otimes O(s_3) \otimes O(s_4) \otimes O(s_5) \otimes O(s_6) \otimes O(s_7) \otimes O(s_8) \otimes O(s_9) \otimes O(s_2) = 0.7 \otimes 0.8 = 0.56$.

Next, statement $s_4$ has complete support trees $CST(s_4) = \{\{s_8, s_4\}, \{s_1, s_2, s_4\}, \{s_3, s_4, s_5, s_6\}\}$ with $CST(s_3) = \{\{s_3\}\}$ attacking all of them. Thus, $O(s_4) = O(s_1) \otimes O(s_3) \otimes O(s_4) \otimes (1 - O(s_5)) \otimes (1 - O(s_6)) \otimes (1 - O(s_7)) \otimes (1 - O(s_8)) \otimes (1 - O(s_9)) = 0.6 \otimes 1.0 \otimes 1.0 \otimes 0.6 \otimes 0.7 \otimes 0.8 \otimes 1.0 \otimes 1.0 \otimes 1.0 \otimes 1.0 \otimes 0.6 = 0.63$.

Likewise, we repeat the procedure for the rest of the statements.

### 5 PROPERTIES OF THE PROPOSED SEMANTICS

In this section, we study properties that characterize our newly introduced semantics.

**Proposition 1.** Given $SG = (\mathcal{V}, \mathcal{E}_S, \mathcal{E}_A)$ a statement graph, $I : \mathcal{V} \rightarrow [0, 1]$ an inner strength function and $DT = (\otimes, \oplus, \neg)$ a De Morgan triple, $W_{DT}(SG, I) = O$ always exists and is unique.

*Proof.* It follows from the fact that $O$ only computes its values using $(\otimes, \oplus, \neg)$, which themselves only depend on combinations, according to the complete support trees, of the inner strengths $I$ of statements as given by the input. The complete support trees of each statement are easily determined by the graph structure.

**Proposition 2.** Given $SG = (\mathcal{V}, \mathcal{E}_S, \mathcal{E}_A)$ a statement graph, $I : \mathcal{V} \rightarrow [0, 1]$ an inner strength function and $DT = (\otimes, \oplus, \neg)$ a De Morgan triple, $W_{DT}(SG, I) = O$ satisfies provability.

*Proof.* Let $s = (p_1 \wedge \ldots \wedge p_n \rightarrow c) \in \mathcal{V}$ be a statement of $SG = (\mathcal{V}, \mathcal{E}_S, \mathcal{E}_A)$, if there is a premise $f \in \text{Premise}(s)$ with no support, then $CST(s) = \emptyset$ (there are no complete support trees for $s$). Thus, $O(s) = \bigoplus(\emptyset) = 0$.

**Proposition 3.** Given $SG = (\mathcal{V}, \mathcal{E}_S, \mathcal{E}_A)$ a statement graph, $I : \mathcal{V} \rightarrow [0, 1]$ an inner strength function and $DT = (\otimes, \oplus, \neg)$ a De Morgan triple, $W_{DT}(SG, I) = O$ satisfies rewriting.

*Proof.* Given $SG$, and $I$ as defined in the property, let $W_{DT}(SG, I) = O$, then:

- $\forall C \subseteq S \setminus \{s_1, s_2, s_3\}, C \cup \{s_1\} \in CST(s_1)$ iff $C \cup \{s_2, s_3\} \in CST(s_3)$. Let $C \cup \{s_1\} \in CST(s_1)$:
  \[
  \otimes I(s) = \otimes (I(s_1), I(s)) = \otimes (I(s_2), I(s_3)) = \otimes I(s).
  \]
  \[
  \forall s \in S \cup \{s_1, s_2, s_3\}, s \text{ attacks } s_1 \text{ if } s \text{ attacks } s_2 \text{ or } s_3.
  \]

Thus, $\forall C, C' \subseteq S, C \cup \{s_1\} \in CST(s_1)$ and $C'$ attacks $C$ iff $C'$ attacks $C \cup \{s_2, s_3\} \in CST(s_3)$.

Therefore, $O(s_1) = O(s_3)$.

In the following we also adopt some of the principles described by [3] in the context of statement graphs and show how the proposed semantics satisfy them.

**Definition 5.** Let $SG = (\mathcal{V}, \mathcal{E}_S, \mathcal{E}_A)$ be a statement graph, $I$ an inner strength function, and $W$ a gradual semantics, we say that it satisfies:

- **Directionality:** if the strength of a statement only depends on the statements that are connected to it. Given statement graph $SG' = (\mathcal{V} \cup \{s_1\}, \mathcal{E}_S, \mathcal{E}_A)$, where $v \in \mathcal{E}_S \cup \mathcal{E}_A$ or $s_1$ is the left or right part of $v$ such that there is no directed path (either with attack and/or support edges) from $s_1$ to $s_2$ in $SG'$, it holds that $W_{SG', I}(s_2) = W_{SG, I}(s_2)$.

- **Support Reinforcement:** if making a support stronger, makes a statement weaker. Given statement graph $SG = (\mathcal{V}, \mathcal{E}_S, \mathcal{E}_A)$, inner strength functions $I_1, I_2$, and statements $s_1, s_2 \in \mathcal{V}$ where $s_2$ is part of a CST of $s_1$ but not of a CST attacking a CST of $s_1$, $I_1(s_2) \leq I_2(s_2)$ and $\forall s \neq s_2, I_1(s) = I_2(s)$, then it holds that $W_{SG, I_1}(s_1) \leq W_{SG, I_2}(s_1)$.

- **Attack Reinforcement:** if making an attack stronger, makes a statement weaker. Given statement graph $SG = (\mathcal{V}, \mathcal{E}_S, \mathcal{E}_A)$, inner strength functions $I_1, I_2$, and statements $s_1, s_2 \in \mathcal{V}$ where $s_2$ is part of a CST attacking a CST of $s_1$ but not of a CST that supports $s_1$, $I_1(s_2) \leq I_2(s_2)$ and $\forall s \neq s_2, I_1(s) = I_2(s)$, then it holds that $W_{SG, I_1}(s_1) \geq W_{SG, I_2}(s_1)$.

- **Support Monotonicity:** if adding a support makes a statement stronger. Given statement graphs $SG = (\mathcal{V}, \mathcal{E}_S, \mathcal{E}_A)$, $SG' = (\mathcal{V}', \mathcal{E}_S', \mathcal{E}_A')$, where:
  - $\mathcal{V}' = \mathcal{V} \cup \{s_1\}$, $\mathcal{V}' = \mathcal{V} \cup \{s_1\}$.
  - $s_2$ is part of a CST attacking a CST of $s_1$ but not of a CST that supports $s_1$, $I_1(s_2) \leq I_2(s_2)$ and $\forall s \neq s_2, I_1(s) = I_2(s)$, then it holds that $W_{SG, I_1}(s_1) \geq W_{SG, I_2}(s_1)$.

- **Attack Monotonicity:** if adding an attack makes a statement weaker. Given statement graphs $SG = (\mathcal{V}, \mathcal{E}_S, \mathcal{E}_A)$, $SG' = (\mathcal{V}', \mathcal{E}_S', \mathcal{E}_A')$, where:
  - $\mathcal{V}' = \mathcal{V} \cup \{s_1\}$, $\mathcal{V}' = \mathcal{V} \cup \{s_1\}$.
  - $s_2$ is part of a CST attacking a CST of $s_1$ but not of a CST that supports $s_1$ in $SG'$. Then, it holds that $W_{SG, I_1}(s_1) \leq W_{SG', I_2}(s_1)$. 

Figure 6: Handling circular reasoning via CSTs.
- $\forall e \in \mathcal{E}_g \cup \mathcal{E}_a, e \in \mathcal{E}_g \cup \mathcal{E}_a$ or $s_2$ is the left or right part of $e$.

Then, it holds that $W_{SG,I}(s_1) \geq W_{SG',I}(s_1)$.

**Proposition 4.** Given $SG = (V, \mathcal{E}_g, \mathcal{E}_a)$ a statement graph, $I$ an inner strength function and $W$ a T-norm semantics defined by the triple $(\otimes, \oplus, \neg)$, $W$ satisfies directionality.

**Proof.** Given a statement $s_2$, if there is no directed path from $s_1$ to $s_2$, then the CSTs that attack and support $s_2$ do not change, thus the outer strength does not change either. \hfill $\Box$

**Proposition 5.** Given $SG = (V, \mathcal{E}_g, \mathcal{E}_a)$ a statement graph, $I$ an inner strength function and $W$ a T-norm semantics defined by the triple $(\otimes, \oplus, \neg)$, $W$ satisfies support reinforcement.

**Proof.** Every CST $C \in CST(s_1)$ that supports $s_1$ has an equal or higher strength value using $I_2$ than $I_1$: $\forall e \in C I_1(s) \leq \otimes \forall e \in C I_2(s)$ because $\otimes$ is a T-norm and thus, is monotonically increasing. Also, every CST $C' \in CST(s_3)$ attacking a CST of $s_1$ has an equal or higher strength value using $I_2$ than $I_1$ i.e. $\forall C' \in CST(s_3)$ such that $C$ attacks $C' \in CST(s_1)$ it holds that $I_1(C) \geq I_2(C)$. Thus, $W_{SG,I}(s_1) \leq W_{SG,I}(s_1)$ because $\otimes$ is monotonically increasing. \hfill $\Box$

**Proposition 6.** Given $SG = (V, \mathcal{E}_g, \mathcal{E}_a)$ a statement graph, $I$ an inner strength function and $W$ a T-norm semantics defined by the triple $(\otimes, \oplus, \neg)$, $W$ satisfies attack reinforcement.

**Proof.** Every CST $C \in CST(s_1)$ that supports $s_1$ has an equal or higher strength value using $I_2$ than $I_1$, so $\forall e \in C I_1(s) \geq \otimes \forall e \in C I_2(s)$. Then, every CST $C \in CST(s_3)$ attacking a CST of $s_1$ has an equal or higher strength value using $I_2$ than $I_1$ i.e. $\forall C \in CST(s_3)$ such that $C$ attacks $C' \in CST(s_1)$ it holds that $I_1(C) \leq I_2(C)$. Thus, $W_{SG,I}(s_1) \geq W_{SG,I}(s_1)$ because $\otimes$ is monotonically increasing and $\neg$ is strictly decreasing. \hfill $\Box$

**Proposition 7.** Given $SG = (V, \mathcal{E}_g, \mathcal{E}_a)$ a statement graph, $I$ an inner strength function and $W$ a T-norm semantics defined by the triple $(\otimes, \oplus, \neg)$, $W$ satisfies support monotonicity.

**Proof.** Given statement $s_1$, $CST_{SG}(s_1) \subseteq CST_{SG'}(s_1)$. Also, a set $C$ of statements is a CST attacking $s_1$ in $SG$ (i.e. $C \in CST_{SG}(s_1)$ and $s_3$ attacks a CST of $s_1$ if and only if it is also present in $SG'$). Thus, it follows that $W_{SG,I}(s_1) \leq W_{SG,I}(s_1)$. \hfill $\Box$

**Proposition 8.** Given $SG = (V, \mathcal{E}_g, \mathcal{E}_a)$ a statement graph, $I$ an inner strength function and $W$ a T-norm semantics defined by the triple $(\otimes, \oplus, \neg)$, $W$ satisfies attack monotonicity.

**Proof.** Given statement $s_1$, $CST_{SG}(s_1) = CST_{SG'}(s_1)$. Also, if a set $C$ of statements is a CST attacking $s_1$ in $SG$ (i.e. $C \in CST_{SG}(s_1)$ and $s_3$ attacks a CST of $s_1$), then it will be also present in $SG'$. Thus, it follows that $W_{SG,I}(s_1) \geq W_{SG,I}(s_1)$. \hfill $\Box$

6 DISCUSSION

In this work, we presented the first gradual semantics for Statement Graphs and analysed their properties. Before laying our future work directions let us place this paper in the context of existing literature. The most prominent knowledge representation and reasoning domain related to this work is that of bipolar argumentation frameworks.

Bipolar argumentation frameworks are built upon argumentation graphs by considering as underlying representation a bi-colored graph: the nodes represent arguments and the bi-colored edges represent, respectively, support and attack relations between the arguments [26, 22, 13, 25, 29]. From this perspective Statement Graphs can, indeed, be seen as a bipolar argumentation graph. The arguments are the statements (i.e. logical implications), the support is represented by the support relation and the attack by the attack relation.

Bipolar argumentation systems enjoy the study of several kinds of support relations: deductive support [31], evidential support [24], necessity support [23, 22] etc. If we view statement graphs as a bipolar argumentation framework it is not straightforward to place the Statement Graph support as any of the supports in the argumentation literature (even for more general frameworks such as ADFs [10] where arguments are not instantiated). This is due to the particular structure of Statement Graphs. To illustrate this problem let us consider a simple example in which we can have a node $n_1$ concluding to $A$, a node $n_2$ concluding to $A$ as well, a node $t$ concluding to not $A$ and, finally, a node $n_3$ using $A$ as a premise. The support of Statement Graphs is not a necessary support because we could have $n_1$ IN, $n_2$ OUT, $t$ is OUT and therefore $n_3$ IN (i.e if $n_3$ is IN then $n_2$ should be IN according to the definition of necessity support). The support is not a deductive support because if we can consider an example where $n_1$ is IN, $t$ is IN and $n_3$ is AMBIG. A similar construction can be made for evidential supports.

On bipolar graphs, existing ranking based semantics provide a partial order between the arguments [18, 20, 14, 1, 9, 4, 5]. Existing ranking semantics for bipolar argumentation graphs that rely on the intuition of a recursive aggregation of supports and attacks [12, 30, 5], were proven not to be able to converge for the general case of graphs without a particular structure other than the existence of cycles [21] if they are calculated iteratively. Moreover, bipolar weighted argumentation frameworks containing cycles can be made to converge under certain conditions as in [28] by solving a linear equation model as proposed in [15], or by continuing the discrete recursive algorithms as discussed in [27]. To this end, if viewing statement graphs as a particular case of instantiated bipolar argumentation frameworks, we provide a semantics considering all supports and attacks for bipolar graph that avoids convergence problems in the general case by not defining the nodes strengths recursively. Of course, this is possible because our approach is based on the logical instantiation of statement graphs, where recursive definition is less crucial than in abstract argumentation.

Last, let us mention that when evaluating the semantics provided in this paper we relied on the work of [9] regarding the postulates to follow for the semantics. We mention here that, for bipolar argumentation graphs, several works have been concerned with defining an automatization of ranking semantics (see [2, 4]).

The actual practical motivation of this work comes from the explanation needs of Statement Graphs. When we tried to apply Statement Graphs to real world knowledge bases we noticed the important size of the graphs. This hindered the visual capabilities of the formalism. It was difficult to explain the status of a node because the set of the nodes supporting it was very large (and therefore scattered on the screen). To this end we proposed the semantics described in this paper that could be used as a filtering mechanism on the graph visualization software. For instance, it could be used to select the strongest CST as a full justification for the state of a query. Such explanation techniques would require, of course, empirical evaluation from the users and will probably vary according to the needs of the application at hand.
However, please note that the definition of our semantics implies considering all possible CST. This is a difficult algorithmic problem to address in the context of large graphs. While the practical impact of this problem is not as important (such computation can be performed offline) it would be useful to be able to have incremental algorithms allowing for online interaction with the Statement Graph. Therefore, immediate future work is concerned with algorithmic aspects for the semantics and how the chase variants impact on the computation of Statement Graphs and, subsequently, on their gradual semantics.

7 ACKNOWLEDGEMENTS

We would like to thank the reviewers for their comments, which helped improve this paper considerably. We also acknowledge the support of the H2020 NoAW project (project ID 688338).

REFERENCES