Further Applications of the Gabbay-Rodrigues Iteration Schema in Argumentation and Revision Theories

D. M. Gabbay[1] and O. Rodrigues[2]

[1] Department of Informatics, King's College London,
Bar Ilan University, Ramat Gan, Israel
and University of Luxembourg, Luxembourg.
dov.gabbay@kcl.ac.uk
WWW home page: http:///www.inf.kcl.ac.uk/staff/dg
[2] Department of Informatics, King's College London,
The Strand, London, WC2R 2LS, UK
odinaldo.rodrigues@kcl.ac.uk,
WWW home page: http://www.inf.kcl.ac.uk/staff/odinaldo

Abstract. In [8], we proposed an iteration schema which operated on an extended argumentation framework whose nodes were assigned initial values in $[0, 1]$, coming from some application area, e.g., revision theory. We showed that the schema generated a new set of node values at each iteration and that after a finite number of steps no new values in the open interval $(0, 1)$ were generated. Any remaining nodes with values in the set $\{0, 1\}$ retain those values during all future iterations. The sequence eventually converges by turning as few values in $(0, 1)$ into $\{0, 1\}$ as necessary in order to yield a complete extension in the traditional sense (interpreting the value 1 as "in", the value 0 as "out", and any other value as "undecided". This traditional extension is the best at accommodating the $\{0, 1\}$-part of the initial set of values. Although the iteration schema operates on values in $[0, 1]$, in this work, we show a simplified form operating on the set $\left\{0, \frac{1}{2}, 1\right\}$ which is more suitable for use in a practical implementation.

Keywords: argumentation theory, revision theory, numerical methods

1 Introduction

An *abstract argumentation framework* is a tuple $\langle S, R\rangle$, where S is a set of arguments and R is an attack relation on $S \times S$. Semantics of argumentation frameworks can be defined in terms of *extensions*, which are subsets of the set of arguments S with special properties.

In [8], we proposed the Gabbay-Rodrigues Iteration Schema. The schema is an iterative method that can be used for calculating extensions in the traditional Dung sense mentioned above. The schema takes an assignment of initial values $V_0 : S \longmapsto [0, 1]$ and produces a new assignment V_{i+1} for each iteration $i \geq 0$. The values in the schema eventually converge and in the limit of the sequence we can construct a complete extension by taking the nodes with value 1.

One disadvantage of the schema is the need to calculate the limit values of the sequence. This can be approximated by iterating sufficiently many times until the difference between the values of the nodes of two consecutive iterations falls below a certain threshold ε. In GRIS [9], Rodrigues set the threshold ε as the upper bound of the relative error introduced due to the rounding in the calculations of the target machine.[3] In experimental tests, this required around 100 iterations. Since in GRIS the schema is used to "ground" the strongly connected components of an argumentation framework and hence invoked often, if we were able to obtain the same results without the need to do any approximation, this would increase the overall performance of the computation of the semantics significantly. This paper proposes and discusses a discrete version of the method called the *Discrete Gabbay-Rodrigues Iteration Schema* and its application in argumentation and revision theories. The computational steps of this method are developed in the main body of this paper and are summarised in Section 6. The reader might wish to refer to the summary to get a more complete overview of the paper.

The following example motivates how an argumentation framework can have requested values coming from revision theory. The interested reader may wish to refer to [6, 7] for a more in-depth analysis of the relationship between revision and argumentation theories.

Example 1 (Numerical Revision Under Constraints). John and Mary are young students and they are getting married. As part of the planning of the wedding, John and Mary, as well as both sets of parents, put forward names of people they want to invite to the wedding.

This can be modelled as a database Δ of names, where $x \in \Delta$ means "invite x to the wedding". There are however constraints on Δ. It may be the case that x and y are not in good terms and if x is invited, then y should not be invited. So let $\mathcal{C}$ be a set of contraints of the form

$$x \to \neg y.$$

The system $\Delta \cup \mathcal{C}$ should not be regarded as a classical theory with negation, in the sense that if it is inconsistent it proves everything. We do not mean that if both x and y are invited then we should invite everybody. We would rather regard $\Delta \cup \mathcal{C}$ as a set violating the constraints and that Δ should be revised to a smaller $\Delta' \subsetneq \Delta$ which satisfies the constraints.

The organisers decided to ask everyone who put a name forward to rank them by a number, indicating how they feel about x being invited. These number were aggregated into a single number $h(x)$ from $\{0, \frac{1}{2}, 1\}$, for every x, where $h(x) = 0$ means "do not invite x" and $h(x) = 1$ means "absolutely must invite x" and $h(x) = \frac{1}{2}$ means "indifferent with respect to x's invitation".

[3] When the threshold is reached we can no longer reliably tell if the difference in value is a legitimate approximation of the limit value or was introduced due to a rounding error in the arithmetic unit of the target machine.

In order now to revise the theory Δ, we present the constraints in a relational form, where xRy means "if you invite x, do not invite y". This turns $\langle \Delta, R \rangle$ into an abstract argumentation framework and the function h into a numerical assignment to the elements of Δ. Our task is to seek a function h', as near h as possible, such that

$$\Delta' = \{x \mid h'(x) = 1\}$$

If the wishes of everyone can be accomodated without conflicts, then we want $\Delta' = \Delta$. Otherwise, we want to invite as many people as possible subject to the constraint that "everyone whose attendance is not objected by anyone invited is also invited."

To solve this problem and other problems like it, we turn to the *discrete* version of the Gabbay-Rodrigues Iteration Schema [8].

The rest of the paper is structured as follows. In Section 2 we provide some background material from argumentation theory. In Section 3, we re-introduce the full-fledged version of the Gabbay-Rodrigues Iteration Schema. This is followed by the presentation of its discrete version in Section 4. We show some examples in Section 5 and conclude with a discussion in Section 6.

2 Background

This section provides a very brief overview of the key concepts from argumentation theory that will be needed in the remainder of this paper. For a more comprehensive introduction to these concepts the reader is referred to [5, 2, 3, 1].

We mentioned in the introduction that the semantics of an argumentation framework can be given in terms of extensions. In [2] Caminada has shown that the semantics can be alternatively presented in terms of labelling functions. Essentially, a labelling function λ is an assignment $\lambda : S \longmapsto \{\mathbf{in}, \mathbf{und}, \mathbf{out}\}$. There is a direct correspondence between the labelling semantics and the notion of extensions, by taking an extension to be the set of arguments with label **in** in labelling functions with special properties (defined below).

As numerical assignments will also give values to all arguments, it will be easier for us to think in terms of labellings instead of extensions. The correspondence between the numerical values and the values in $\{\mathbf{in}, \mathbf{und}, \mathbf{out}\}$ is given later in Definition 5. We start by defining the status of the label of an argument with respect to the labels of its attackers.

Definition 1 (Illegal labelling of an argument [4]). *Let $\langle S, R \rangle$ be an argumentation framework and λ a labelling function for S.*

1. *An argument $X \in S$ is illegally labelled* **in** *by λ if $\lambda(X) = \mathbf{in}$ and there exists $Y \in Att(X)$ such that $\lambda(Y) \neq \mathbf{out}$.*
2. *An argument $X \in S$ is illegally labelled* **out** *by λ if $\lambda(X) = \mathbf{out}$ and there is no $Y \in Att(X)$ such that $\lambda(Y) = \mathbf{in}$.*
3. *An argument $X \in S$ is illegally labelled* **und** *by λ if $\lambda(X) = \mathbf{und}$ and either for all $Y \in Att(X)$, $\lambda(Y) = out$ or there exists $Y \in Att(X)$, such that $\lambda(Y) = \mathbf{in}$.*

The conditions above are used to define labelling functions associated with admissible sets, grounded and complete extensions. The sets are defined in terms of the arguments labelled **in** by the corresponding labelling functions.

Definition 2 (Admissible labelling function [4, Definition 8]). *An* admissible labelling function *is a labelling function without arguments that are illegally labelled* **in** *and without arguments that are illegally labelled* **out**.

It is easy to see that any illegal labelling function λ_0 can be turned into an admissible labelling function λ_{da} by successively turning each node illegally labelled **in** or **out** into **und**.[4] This process was called a *contraction sequence* in [4]. The labelling function λ_{da} obtained in this way is the largest element of the set of all admissible labellings that are smaller or equal to λ.[5]

Definition 3 (Complete labelling function [4, Definition 9]). *A* complete labelling function *is a labelling function without any illegally labelled arguments.*

After a contraction sequence, by successively assigning the correct legal label to each illegally labelled undecided node in λ_{da} we eventually end up with a labelling function λ_{CP} without any illegally labelled nodes. By Definition 3, λ_{CP} is a complete labelling function. This corrective process was called an *expansion sequence* in [4]. Caminada and Pigozzi have further shown that λ_{CP} is the smallest element of the set of all complete labellings that are larger or equal to λ_{da} (in the sense of Footnote 5).

If we start with $\lambda_{da}(X) = \mathbf{und}$, for all $X \in S$, then obviously λ_{CP} will correspond to the smallest complete labelling function of all, which is the grounded labelling function.

Definition 4 (Grounded labelling function [4, Definition 10]). *Let λ be a complete labelling function. λ is a grounded labelling function if and only if the set $\{X \in S \mid \lambda(X) = \mathbf{in}\}$ is minimal with respect to set inclusion among all complete labelling functions.*

These results can be understood in our numerical setting through the following two-way translation mechanism.

Definition 5 (Caminada-Pigozzi/Gabbay-Rodrigues Translation). *A labelling function λ and a valuation function V can be inter-defined according to the table below.*

$\lambda(X) \to V_\lambda(X)$			$V(X) \to \lambda_V(X)$		
in	$\to$	1	1	$\to$	**in**
out	$\to$	0	0	$\to$	**out**
und	$\to$	$\frac{1}{2}$	$(0, 1)$	$\to$	**und**

[4] Here "da" reminds us that λ_{da} is the "down-admissible" labelling function resulting from λ_0.

[5] λ_1 is smaller or equal to λ_2 if $in(\lambda_1) \subseteq in(\lambda_2)$ and $out(\lambda_2) \subseteq out(\lambda_2)$. Conversely, λ_1 is larger or equal to λ_2 if $in(\lambda_1) \supseteq in(\lambda_2)$ and $out(\lambda_2) \supseteq out(\lambda_2)$.

What this gives us is that it is possible to turn labellings that are not complete (i.e., are not associated with a complete extension in Dung's sense) via a contraction sequence followed by an expansion sequence. The Gabbay-Rodrigues Iteration Schema can be used in a numerical context and in the limit of the sequence, the computed values will correspond to the same complete labelling. This is explained in more detail in the next section.

3 The Gabbay-Rodrigues Iteration Schema

In [8], we proposed the Gabbay-Rodrigues Iteration Schema. The schema is an iterative method that can be used for calculating extensions in the traditional Dung sense. The schema takes an assignment of initial values $V_0 : S \longmapsto [0, 1]$ and produces a new assignment V_{i+1} for each iteration $i \geq 0$. We will use U to denote the unit interval $[0, 1]$. The schema is defined as follows.

Definition 6. *Let $\mathcal{N} = \langle S, R\rangle$ be an argumentation framework and V_0 be an assignment of values to the nodes in S. The* Gabbay-Rodrigues Iteration Schema *is defined by the following system of equations $\boldsymbol{T}$, where for each node $X \in S$, the value $V_{i+1}(X)$ is defined in terms of the values of the nodes in $\{X\} \cup Att(X)$ in iteration V_i as follows:*

$$\begin{aligned} V_{i+1}(X) = (1 - V_i(X)) \cdot \min\left\{\tfrac{1}{2}, 1 - \max\nolimits_{Y \in Att(X)} V_i(Y)\right\} \; + \\ V_i(X) \cdot \max\left\{\tfrac{1}{2}, 1 - \max\nolimits_{Y \in Att(X)} V_i(Y)\right\} \end{aligned} \tag{T}$$

The schema guarantees that all node values generated remain in the unit interval U:

Proposition 1 ([8]). *Let $\mathcal{N} = \langle S, R\rangle$ be an argumentation framework and $V_0 : S \longmapsto U$ an assignment of initial values to the nodes in S. Let each assignment V_i, $i > 0$, be calculated by the Gabbay-Rodrigues Iteration Schema for $\mathcal{N}$. It follows that $V_i(X) \in U$, for all $i \geq 0$ and all $X \in S$.*

In order to understand what the schema computes based on the initial assignment V_0, it will prove useful to introduce some terminology first.

Definition 7. *For any assignment of values $V : S \longmapsto U$ define the sets $in(V) = \{X \in dom\ V \mid V(X) = 1\}$ and $out(V) = \{X \in dom\ V \mid V(X) = 0\}$.*

Definition 8. *Let $V : S \longmapsto U$ be an assignment of values to the nodes in S. The set of crisp values with respect to V, C_V is the defined as the set $in(V) \cup out(V)$*

By abuse of notation, we will also use $in(\lambda)$ (resp. $out(\lambda)$) to designate the nodes labelled **in** (resp. **out**) by λ and C_λ to denote $in(\lambda) \cup out(\lambda)$.

As in a contraction followed by an expansion sequence, the schema also operates in two phases. In the first phase, all nodes X whose initial values are crisp and whose equation

$$V(X) = 1 - \max_{Y \in Att(X)}\{V(Y)\} \tag{1}$$

is not satisfied have their values re-assigned to a value in the open interval $(0, 1)$. Intuitively, what this does is to turn illegal crisp values into the undecided range, but unlike in a contraction sequence, more than one illegal crisp value can be turned into undecided in the same iteration. The schema does not turn values in $(0, 1)$ into $\{0, 1\}$, as can be seen by the theorem below, although as many values as required will approximate their correct "legal" $\{0, 1\}$ values so as to yield a complete extension.[6]

Theorem 1 ([8]). *Let $\mathcal{N} = \langle S, R \rangle$ be an argumentation framework, $\boldsymbol{T}$ a system of equations for $\mathcal{N}$ using the Gabbay-Rodrigues Iteration Schema, and $V_0 : S \longmapsto U$ an assignment of initial values to the nodes in S. Let V_0, V_1, V_2, ... be a sequence of value assignments where each V_i, $i > 0$, is generated by $\boldsymbol{T}$. Then the following properties hold for all $X \in S$ and for all $k \geq 0$*

1. *If $V_k(X) = 0$, then $V_{k+1}(X) \neq 1$.*
2. *If $V_k(X) = 1$, then $V_{k+1}(X) \neq 0$.*
3. *If $0 < V_k(X) < 1$, then $0 < V_{k+1}(X) < 1$.*

We say that a sequence becomes *stable* if no new undecided nodes are generated between iterations.[7]

Definition 9. *Let $\mathcal{N} = \langle S, R \rangle$ be an argumentation framework and $V_0 : S \longmapsto U$ an assignment of initial values to the nodes in S. A sequence of assignments $V_i : S \longmapsto U$ where each $i > 0$ is said to be* stable at iteration k, *if for all nodes $X \in S$ we have that*

1. *If $V_k(X) \in \{0, 1\}$, then $V_{k+1}(X) = V_k(X)$; and*
2. *k is the smallest value for which the condition above holds.*

The theorem below shows that once all crisp values stabilise between two iterations, they remain unchanged throughout the rest of the sequence.

Theorem 2 ([8]). *Let $\mathcal{N} = \langle S, R \rangle$ be an argumentation framework, $\boldsymbol{T}$ its GR system of equations, and V_0 an initial assignment of values to the nodes in S. Let V_0, V_1, V_2, ... be a sequence of value assignments where each V_i, $i > 0$, is generated by $\boldsymbol{T}$. Assume that for some iteration i and all nodes $X \in S$ such that $V_i(X) \in \{0, 1\}$, we have that $V_{i+1}(X) = V_i(X)$, then for all $j \geq 1$, $V_{i+j}(X) = V_i(X)$.*

Since no new crisp values are generated and S is finite, then at some iteration k the last set of illegal crisp values is changed into the undecided range.

Corollary 1 ([8]). *Let $\mathcal{N} = \langle S, R \rangle$ be an argumentation framework, $V_0 : S \longmapsto U$ an assignment of initial values to the nodes in S and $\boldsymbol{T}$ its GR system of equations. The following hold:*

[6] The correct "legal" values are obtained in the *limit* of the sequence $\lim_{i \to \infty} V_i$.

[7] Stability here for us just means that the schema has corrected all incorrectly assigned crisp values and hence no new values that were not already in the $(0, 1)$ interval will be generated.

1. *If the sequence of value assignments is not stable at iteration k, then there exists $X \in S$, such that $V_k(X) \in \{0, 1\}$ and $V_{k+1}(X) \in (0, 1)$.*
2. *Let $|S| = n$. Then, the sequence is stable for some $k \leq n$.*

Corollary 1 shows that for some value $0 \leq k \leq |S|$, the sequence of value assignments $V_0(X), V_1(X), V_2(X), \ldots$ eventually becomes stable. That is, there exists $k \geq 0$, such that for all $j \geq 0$ and all nodes X

- if $V_k(X) = 0$, then $V_{k+j}(X) = 0$;
- if $V_k(X) = 1$, then $V_{k+j}(X) = 1$; and
- if $V_k(X) \in (0, 1)$, then $V_{k+j}(X) \in (0, 1)$.

Obviously, if the initial assignment already gives a value in $(0, 1)$ to all nodes, then the schema is already stable at the outset:

Proposition 2. *Consider the assignment V_0 such that $V_0(X) = \frac{1}{2}$ for all $X \in S$. Then the Gabbay-Rodrigues Iteration Schema is stable at iteration 0.*

Proof. *The initial assignment has no nodes in $\{0, 1\}$, so condition 1. for stability of Definition 9 is vacuously satisfied and obviously 0 is the smallest iteration value for which this holds.*

We mean "stable" in the sense that all crisp values at the stable point remain unchanged. Stability in the sequence does not guarantee that all values are legal, and hence it does not guarantee that the corresponding labelling is complete. What remains to be done is to "correct" as many illegal undecided nodes so as to yield an extension. We have shown in [8], that this is achieved in the limit of the sequence:

Theorem 3 ([8]). *Let $\langle S, R \rangle$ be an argumentation framework; V_0 be an initial assignment of values to the nodes in S; λ_0 an initial labelling of these nodes; and V_0 and λ_0 faithful to each other according to Definition 5. Let λ_{da} be the labelling at the end of a contraction sequence from λ_0 and λ_{CP} the labelling at the end of an expansion sequence after λ_{da}. Let k be the point at which the sequence V_0, V_1,... becomes stable and $V_e(X)$ the value of the node X in the limit of the sequence of values calculated through the Gabbay-Rodrigues Iteration Schema. Then λ_{CP} and V_e agree with each other according to Definition 5.*

The above theorem therefore establishes a correspondence between the results obtained in the limit of the sequence of the Gabbay-Rodrigues Iteration Schema and those obtained after a contraction and expansion sequence (in Caminada-Pigozzi's sense). However, by using the Gabbay-Rodrigues Iteration Schema in a numerical context, we can also use the values at the stable point as the admissible values closest to the initial assignment of values (in Dung's sense). The values at the stable point are not used in this paper.

We now present a simplified version of the schema which only operates on the set of values $\{0, \frac{1}{2}, 1\}$.

4 The Discrete Gabbay-Rodrigues Iteration Schema

We would like to simplify the calculations of the Gabbay-Rodrigues Iteration Schema by avoiding having to approximate the limit values of the sequence and yet keeping the same final results. This is the objective of the discrete schema we present next.

Definition 10. *Let $\mathcal{N} = \langle S, R \rangle$ be an argumentation framework and V_0 be an assignment of values from $\{0, \frac{1}{2}, 1\}$ to the nodes in S. The* Discrete Gabbay-Rodrigues Iteration Schema *is defined by the following system of equations* ($\mathrm{T}_{\mathbf{d}}$)*, where the value $V_{i+1}(X)$ of each node in iteration $i+1$ is defined in terms of the values of the nodes in iteration i as follows:*

$$V_{i+1}(X) = 1 - \max_{Y \in Att(X)} \{V_i(Y)\} \qquad (\mathrm{T}_{\mathbf{d}})$$

The theorem below shows that, as for the full-fledged Gabbay-Rodrigues Iteration Schema, initial assignments corresponding to complete labelling functions are preserved.

Theorem 4. *Let $\langle S, R \rangle$ be an argumentation framework and V_i an assignment of values from $\{0, \frac{1}{2}, 1\}$ to the nodes in S. λ_{V_i} is a complete labelling function if and only if $V_{i+1}(X) = V_i(X)$, for all $X \in S$.*

Proof. *(⇒) Take any $X \in S$ and assume λ_{V_i} is a complete labelling function.*

1. *If $V_i(X) = 1$, then $\lambda_{V_i}(X) =$ **in**. Since λ_{V_i} is a complete labelling function, then for all $Y \in Att(X)$, $\lambda_{V_i}(Y) =$ **out**. Therefore, $\max_{Y \in Att(X)}\{V_i(Y)\} = 0$, and hence $V_{i+1}(X) = 1 - 0 = 1$.*
2. *If $V_i(X) = 0$, then $\lambda_{V_i}(X) =$ **out**. Since λ_{V_i} is a complete labelling function, then there exists $Y \in Att(X)$, such that $\lambda_{V_i}(Y) =$ **in**. Therefore, $\max_{Y \in Att(X)}\{V_i(Y)\} = 1$, and hence $V_{i+1}(X) = 1 - 1 = 0$.*
3. *If $V_i(X) = \frac{1}{2}$, then $\lambda_{V_i}(X) =$ **und**. Since λ_{V_i} is a complete labelling function, then there exists $Y \in Att(X)$, such that $\lambda_{V_i}(Y) =$ **und** and for no $Y \in Att(X)$ do we have that $\lambda_{V_i}(Y) =$ **in**. Therefore, $\max_{Y \in Att(X)}\{V_i(Y)\} = \frac{1}{2}$, and hence $V_{i+1}(X) = 1 - \frac{1}{2} = \frac{1}{2}$.*

Therefore, for all nodes $X \in S$, if λ_{V_i} is a complete labelling function, then $V_{i+1}(X) = V_i(X)$.
(⇐) Take any $X \in S$, and assume that $V_{i+1}(X) = V_i(X)$.

1. *If $V_{i+1}(X) = 1$, then $\max_{Y \in Att(X)}\{V_i(Y)\} = 0$, and hence for all $Y \in Att(X)$, $V_i(Y) = 0$. It follows that λ_{V_i} labels all attackers Y of X **out**. Since $V_i(X) = V_{i+1}(X)$, then $\lambda_{V_i}(X) =$ **in** and therefore λ_{V_i} legally labels X.*
2. *If $V_{i+1}(X) = 0$, then there exists $Y \in Att(X)$, such that $V_i(Y) = 1$. Therefore, one of the attackers of X is labelled **in** by λ_{V_i}. Since $V_i(X) = V_{i+1}(X)$, $\lambda_{V_i}(X) =$ **out** and hence λ_{V_i} legally labels X.*

3. If $V_{i+1}(X) = \frac{1}{2}$, then $\max_{Y \in Att(X)}\{V_i(Y)\} = \frac{1}{2}$. Therefore, there exists $Y \in Att(X)$, such that $V_i(Y) = \frac{1}{2}$ and for no $Y \in Att(X)$ do we have that $V_i(Y) = 1$. This means one attacker of X is labelled **und** *by λ_{V_i}, but no attacker of X is labelled* **in** *by it. Since $V_i(X) = V_{i+1}(X)$, $\lambda_{V_i}(X) =$* **und**. *Therefore, X is legally labelled by λ_{V_i}.*

It follows that if for all nodes $X \in S$, $V_{i+1}(X) = V_i(X)$, then all nodes $X \in S$ are legally labelled by λ_{V_i} and hence λ_{V_i} is a complete labelling function.

Corollary 2. *Let $\langle S, R\rangle$ be an argumentation framework and take the assignment of values $V_i : S \longmapsto \{0, \frac{1}{2}, 1\}$. If λ_{V_i} is a complete labelling function, then $V_{i+j}(X) = V_i(X)$, for all $X \in S$ and all $j \geq 0$.*

Proof. Theorem 5 shows that if λ_{V_i} is a complete labelling function, then $V_{i+1}(X) = V_i(X)$ for all $X \in S$. Since each iteration only depends on the values of the nodes of the previous iteration and $V_{i+1}(X) = V_i(X)$ for all $X \in S$, then all subsequent iterations will produce the exact same values.

The above corollary shows that the sequence of values generated by the Discrete Gabbay-Rodrigues Iteration Schema does not change when the values of an iteration correspond to a complete labelling function. In practice what this means is that the discrete schema can be used to check whether an initial labelling function is complete. If the initial assignment corresponding to the labelling function is complete, then the values calculated at the second iteration will remain the same for all nodes.

It is easy to see that the complexity of this check is not higher than that of using labelling functions on $\{\mathbf{out}, \mathbf{in}, \mathbf{und}\}$ and that the check can be implemented in a single loop such as the one in Algorithm 1.

Algorithm 1 Checking whether a labelling is complete

procedure isComplete($\langle S, R, V\rangle$)
  for all nodes $X \in S$ **do**
    if $V(X) \neq 1 - \max_{Y \in Att(X)}\{V(Y)\}$ **then**
      return false ▷ **The value of the node X is illegal**
  return true ▷ **The values of all nodes are legal**

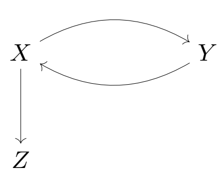

Fig. 1. An argumentation framework with multiple complete extensions.

Example 2. Consider the argumentation framework of Fig. 1 and the complete labellings $\lambda_0 = \{X = \textbf{und}, Y = \textbf{und}, Z = \textbf{und}\}$; $\lambda_1 = \{X = \textbf{in}, Y = \textbf{out}, Z = \textbf{out}\}$ and $\lambda_2 = \{X = \textbf{out}, Y = \textbf{in}, Z = \textbf{in}\}$. The associated translations are $V_{\lambda_0} = \{X = \frac{1}{2}, Y = \frac{1}{2}, Z = \frac{1}{2}\}$; $V_{\lambda_1} = \{X = 1, Y = 0, Z = 0\}$; and $V_{\lambda_2} = \{X = 0, Y = 1, Z = 1\}$. The table below shows how the Discrete Gabbay-Rodrigues Iteration Schema evolves using V_{λ_i} as the initial values for V_0^i.

$$
\begin{array}{ll}
V_0^0(X) = \frac{1}{2} & V_1^0(X) = 1 - \max\{\frac{1}{2}\} = \frac{1}{2} \\
V_0^0(Y) = \frac{1}{2} & V_1^0(Y) = 1 - \max\{\frac{1}{2}\} = \frac{1}{2} \\
V_0^0(Z) = \frac{1}{2} & V_1^0(Z) = 1 - \max\{\frac{1}{2}\} = \frac{1}{2} \\
V_0^1(X) = 1 & V_1^1(X) = 1 - \max\{0\} = 1 \\
V_0^1(Y) = 0 & V_1^1(Y) = 1 - \max\{1\} = 0 \\
V_0^1(Z) = 0 & V_1^1(Z) = 1 - \max\{1\} = 0 \\
V_0^2(X) = 0 & V_1^2(X) = 1 - \max\{1\} = 0 \\
V_0^2(Y) = 1 & V_1^2(Y) = 1 - \max\{0\} = 1 \\
V_0^2(Z) = 1 & V_1^2(Z) = 1 - \max\{0\} = 1
\end{array}
$$

Unlike the full-fledged Gabbay-Rodrigues Iteration Schema, its discrete version *does not* correct all possible illegal initial assignments. For instance, if the assignment $V_0 =$ all-**in** is given to the Discrete Gabbay-Rodrigues Iteration Schema, then the sequence of values will not converge, as shown in Example 3. The sequence in the full-fledged Gabbay-Rodrigues Iteration Schema will however converge to the values $X = \frac{1}{2}$, $Y = \frac{1}{2}$, $Z = \frac{1}{2}$, which do correspond to the empty complete extension.

Example 3. Consider the argumentation framework of Fig. 1 and the initial assignment $V_0 = \{X = 1, Y = 1, Z = 1\}$. The table below shows how the Discrete Gabbay-Rodrigues Iteration Schema evolves using V_0 as initial values.

$$
\begin{array}{lll}
V_0(X) = 1 & V_1(X) = 1 - \max\{1\} = 0 & V_2(X) = 1 - \max\{0\} = 1 \ldots \\
V_0(Y) = 1 & V_1(Y) = 1 - \max\{1\} = 0 & V_2(Y) = 1 - \max\{0\} = 1 \ldots \\
V_0(Z) = 1 & V_1(Z) = 1 - \max\{1\} = 0 & V_2(Z) = 1 - \max\{0\} = 1 \ldots
\end{array}
$$

We will see that under the particular initial assignment all-**und**, the discrete version of the schema *will* always converge to values whose corresponding labelling function is complete.

This means that given the all-**und** initial assignment for an argumentation framework $\langle S, R \rangle$, the Discrete Gabbay-Rodrigues Iteration Schema will compute its grounded extension.

Theorem 5. *Let $V_k(X)$ be the values of the Gabbay-Rodrigues Iteration Schema at the stable point. Then the labelling λ_{V_k} is admissible.*

Proof. *We only need to show that if* $V_k(X) = 1$ *then* X *is legally labelled* **in** *by* λ_{V_k} *and that if* $V_k(X) = 0$ *then* X *is legally labelled* **out** *by* λ_{V_k}.

So suppose $V_k(X) = 1$. *Since the sequence is stable at* k, $V_{k+1}(X) = V_k(X) = 1$. *Hence,*

$$\begin{aligned} 1 &= 0 \cdot \min\{\frac{1}{2}, 1 - \max_{Y \in Att(X)}\{V_i(Y)\} + 1 \cdot \max\{\frac{1}{2}, 1 - \max_{Y \in Att(X)}\{V_i(Y)\} \\ &= \max\{\frac{1}{2}, 1 - \max_{Y \in Att(X)}\{V_i(Y)\} \end{aligned}$$

Therefore, $\max_{Y \in Att(X)}\{V_i(Y)\} = 0$, *and hence for all* $Y \in Att(X)$, $V_k(Y) = 0$. *Therefore,* λ_{V_k} *labels all attackers of* X **out**, *and hence* X *is legally labelled* **in** *by* λ_{V_k}.

On the other hand, if $V_k(X) = 0$, *since the sequence is stable at* k, $V_{k+1}(X) = V_k(X) = 0$. *Hence,*

$$\begin{aligned} 0 &= 1 \cdot \min\{\frac{1}{2}, 1 - \max_{Y \in Att(X)}\{V_i(Y)\} + 0 \cdot \max\{\frac{1}{2}, 1 - \max_{Y \in Att(X)}\{V_i(Y)\} \\ &= \min\{\frac{1}{2}, 1 - \max_{Y \in Att(X)}\{V_i(Y)\} \end{aligned}$$

Therefore, $\max_{Y \in Att(X)}\{V_i(Y)\} = 1$, *and hence there exists* $Y \in Att(X)$ *such that* $V_k(Y) = 1$. *Therefore,* λ_{V_k} *labels some attacker of* X **in**, *and hence* X *is legally labelled* **out** *by* λ_{V_k}.

Proposition 3. *Let* λ *be an admissible labelling function, then no expansion sequence can change the labels of the nodes in* C_λ.

Proof.
1. *Take* $\lambda(X) =$ **in**. *Since* λ *is admissible, then all attackers of* X *must be labelled* **out** *by* λ, *and hence none of them is labelled* **und**.
   The significance of this is that no attacker of X *can have its label changed to* **in** *in any expansion sequence, and hence* X *cannot itself be changed from* **in** *to* **out** *or to* **und**. *So the label of* X *remains* **in** *in all expansion sequences.*
2. *Take* $\lambda(X) =$ **out**. *Since* V *is admissible, then for at least one attacker* Y *of* X *do we have that* $\lambda(Y) =$ **in**, *and therefore, by 1. above,* $\lambda(Y)$ *cannot change from* **in**. *Furthermore, any change in the values of any other attacker of* X *cannot affect* X*'s label, since it is already* **out**, *and hence it remains* **out**.

The counterpart for the above proposition using the schema ($\mathrm{T}_{\mathbf{d}}$) is shown below.

Proposition 4. *Let* λ *be an admissible labelling and let* V_0 *be* V_λ *according to Definition 5. Now consider the schema* ($\mathrm{T}_{\mathbf{d}}$). *If* $V_0(X) \in \{0, 1\}$, *then* $V_1(X) = V_0(X)$.

Proof.
1. *Suppose* $V_0(X) = 1$. *Since* λ *is admissible, then all attackers* Y *of* X *are labelled* **out**, *and hence for all such attackers* $V_0(Y) = 0$. *Therefore,*

$$V_1(X) = 1 - \max_{Y \in att(X)}\{V_0(Y)\} = 1 - 0 = 1 = V_0(X).$$

2. *Suppose $V_0(X) = 0$. Since λ is admissible, then there exists one attacker Y of X which is labelled* **in***, and hence $V_0(Y) = 1$. Therefore,*

$$V_1(X) = 1 - \max_{Y \in att(X)} \{V_0(Y)\} = 1 - 0 = 1 = V_0(X).$$

So it is easy to see that given an admissible assignment of values V_0, $(\mathrm{T}_\mathbf{d})$ will only change the values of the nodes X such that $V_0(X) \in (0, 1)$. But what nodes can change? Suppose $V_0(X) = \frac{1}{2}$. We have three cases

1. Either $max_{Y \in Att(X)}\{V_0(Y)\} = 1$, and then $V_1(X) = 0$. Notice that this change cannot alter the value of any node Z attacked by X with $V_0(Z) \in \{0, 1\}$. If V_0 is admissible and X attacks Z, then certainly $V_0(Z) \neq 1$. If $V_0(Z) = 0$, then again since V_0 is admissible, Z would have been attacked by another node W such that $V_0(W) = 1$, so the change of X to 1 is irrelevant to Z.
2. Or $max_{Y \in Att(X)}\{V_0(Y)\} = 0$, and then $V_1(X) = 1$. Notice that this change cannot again alter the value of any node Z attacked by X with $V_0(Z) \in \{0, 1\}$. If V_0 is admissible and X attacks Z, then if $V_0(Z) = 0$, it would have been attacked by some node W with $V_0(W) = 1$, and so $V_1(Z)$ will remain 0. We cannot have that $V_0(Z) = 1$, since V_0 is admissible and $V_0(X) = \frac{1}{2}$, so the change in the value of X to 1 is irrelevant to all nodes attacked by it.
3. Or $max_{Y \in Att(X)}\{V_0(Y)\} = \frac{1}{2}$, and then $V_1(X) = \frac{1}{2}$. Therefore all nodes attacked by X will remain unaffected.

Conjecture 1. Let $V_e(X)$ be the equilibrium value of the node X calculated according to the Gabbay-Rodrigues Iteration Schema and $V_e^d(X)$ its value calculated according to the discrete version of the schema, where

$$V_0^d(X) = \begin{cases} V_k(X), \text{ if } V_k(X) \in \{0, 1\} \\ \frac{1}{2}, \text{ otherwise} \end{cases} \tag{2}$$

Then for all nodes X, $V_e(X) = V_e^d(X)$.

Sketch of proof.

1. Use the full-fledged Gabbay-Rodrigues Iteration Schema. By Theorem 5, the values in V_k correspond to an admissible labelling.
2. Proposition 1 shows that the sequence of values becomes stable at some iteration k. Definition 9 says that crisp values do not change after the stable point. Turn the remaining values in $(0, 1)$ into $\frac{1}{2}$, generating V_0^d. Notice that V_0^d is still admissible, since no nodes with values in $\{0, 1\}$ were changed.
3. Run the discrete version of the Schema using V_0^d as initial values. Proposition 4 guarantees that old crisp values remain the same throughout.
4. All nodes for which the sum of the value of the attackers is 0 will turn into 1. This may then change the value of some nodes attacked by them and so forth. Any change of values will not affect the original crisp values in V_k, because of Proposition 4.
   Proceeding in this way will generate the minimal complete labelling including the initial admissible labelling.

5 Worked examples

This section presents some examples and discusses both the differences between the two versions of the schema and how they can be combined.

1. We have seen in Fig. 1, that the Discrete Gabbay-Rodrigues Iteration Schema is not guaranteed to converge if given an arbitrary illegal initial assignment. So for the argumentation framework of Fig. 1, and initial assignment $V_0^d(X) = V_0^d(Y) = V_0^d(Z) = 1$, we get the values of the nodes in odd steps of the iteration as being 0 and in the even steps as being 1, without convergence. The full-fledged Gabbay-Rodrigues Iteration Schema does not suffer from this. The values calculated will be as follows:

   Discrete

	V_0^d	V_1^d	V_2^d
X	1	0	1
Y	1	0	1
Z	1	0	1

   Full-fledged

	V_0	V_k	V_e
X	1	$\frac{1}{2}$	$\frac{1}{2}$
Y	1	$\frac{1}{2}$	$\frac{1}{2}$
Z	1	$\frac{1}{2}$	$\frac{1}{2}$

   Our suggestion is to let the discrete version take over from iteration k, giving (in the discrete version):

	V_0^d	$V_i^d = V_0^d$
X	$\frac{1}{2}$	$\frac{1}{2}$
Y	$\frac{1}{2}$	$\frac{1}{2}$
Z	$\frac{1}{2}$	$\frac{1}{2}$

   Which does converge in a finite number of steps (without the need to calculate the limit of the sequence).
2. Consider the argumentation framework of Fig. 2. It has the grounded labelling $\lambda_1 = \{X = \textbf{in}, Y = \textbf{out}, W = Z = \textbf{und}\}$ corresponding to the grounded extension $E_1 = \{X\}$.
   Given the all-$\frac{1}{2}$ initial assignment V_0 below, the Discrete Gabbay-Rodrigues Iteration Schema will compute the subsequent values in the sequence as follows.

	V_0^d	V_1^d	V_2^d	V_3^d	
X	$\frac{1}{2}$	1	1	$\cdots$	λ_1
Y	$\frac{1}{2}$	$\frac{1}{2}$	0	$\cdots$	
W	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\cdots$	
Z	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\cdots$	

The values converge at iteration 2, and the corresponding labelling function λ_{V_2} is the same as λ_1, so $in(V_2) = E_1$.
Notice that the complete labellings $\lambda_2 = \{X = \mathbf{in}, Y = \mathbf{out}, W = \mathbf{in}, Z = \mathbf{out}\}$ and $\lambda_3 = \{X = \mathbf{in}, Y = \mathbf{out}, W = \mathbf{out}, Z = \mathbf{in}\}$ cannot be obtained using the Discrete Gabbay-Rodrigues Iteration Schema directly using $\frac{1}{2}$ as initial values. This is because λ_1 and λ_2 are complete but not grounded. However, if the initial assignments $V_0^{d_1}(X) = 1$, $V_0^{d_1}(Y) = 0$, $V_0^{d_1}(W) = 1$, $V_0^{d_1}(Z) = 0$ and $V_0^{d_2}(X) = 1$, $V_0^{d_2}(Y) = 0$, $V_0^{d_2}(W) = 0$, $V_0^{d_2}(Z) = 1$ are given to the discrete schema, the values will immediately stabilise as they correspond to complete extensions:

	$V_0^{d_1}$	$V_1^{d_1}$
X	1	1
Y	0	0
W	1	1
Z	0	0

	$V_0^{d_2}$	$V_1^{d_2}$
X	1	1
Y	0	0
W	0	0
Z	1	1

$$X \longrightarrow Y \longrightarrow W \longleftrightarrow Z$$

Fig. 2. A sample argumentation framework.

6 Conclusions and Discussion

In [8] we put forward the Gabbay-Rodrigues Iteration Schema, which given any initial assignment of values to the nodes of an argumentation framework, will successively turn each node with an illegal value 1 or 0 into the undecided range (i.e., in the open interval $(0, 1)$). The schema will then, *in the limit* of the sequence of values, turn each illegal undecided values into 0 or 1 so as to yield a complete extension.

The disadvantage of this is that we need some means of computing the values in the limit of the sequence. Every computer has an upper bound of the relative error introduced due to the rounding in the arithmetic calculations. When the maximum variation in node values between two successive iterations becomes smaller than or equal to this value, we can no longer be certain if the variation is genuine or due to rounding errors. In [9], Rodrigues used this as the halting point of the approximation.

In this paper we proposed a simplified version of the schema, which we called the *Discrete Gabbay-Rodrigues Iteration Schema*.

Using the simplified schema, we can only give two guarantees: 1) if the initial assignment corresponds to a complete labelling (i.e., yields a complete extension), then the values of the schema will remain the same; 2) If the initial values are all $\frac{1}{2}$, then the schema will converge to values corresponding to the grounded labelling of the argumentation framework. However, there is no guarantee of correction of any other initial illegal values other than $\frac{1}{2}$.

Our suggestion is to combine the two schema as follows.

1. Start with the Gabbay-Rodrigues Iteration Schema and iterate until no new nodes with undecided values are generated (say, iteration k). This is the stable point for the schema.
2. At the stable point k, Corollary 1 and Theorem 1 guarantee that all crisp values are stable (you can read this as "they are legal"). This is also the largest possible set of such values and nodes cannot swap values within $\{0, 1\}$.
   The limit theorem of the Gabbay-Rodrigues Iteration Schema gives us that all values will eventually converge to one of $\{0, \frac{1}{2}, 1\}$. This means that the remaining values in $(0, 1)$ will all converge to one of $\{0, \frac{1}{2}, 1\}$ **and** this convergence will not affect the previously calculated crisp values.
   Instead of approximating the limit, let us take the discrete version of the iteration schema and use the following assignment V_0^d as the initial assignment for the discrete schema:
$$V_0^d(X) = \begin{cases} V_k(X), \text{ if } V_k(X) \in \{0, 1\} \\ \frac{1}{2}, \text{ otherwise} \end{cases} \tag{3}$$
3. Now apply the discrete schema with the initial values V_0^d.
$$V_{i+1}^d(X) = 1 - \max_{Y \in Att(X)} \{V_i^d(Y)\} \tag{T$_\mathbf{d}$}$$
   The initial crisp values in V_0^d are all legal, since $V_0^d = V_k$ is admissible, so by Proposition 4, they will not change.
   The illegal $\frac{1}{2}$ values will change, but only so as to yield a complete extension. We conjecture that this extension has to be the same as the one calculated by the full-fledged Gabbay-Rodrigues Iteration Schema, since that extension is the minimal extension including the crisp values calculated at the stable point.

References

1. P. Baroni, M. Caminada, and M. Giacomin. An introduction to argumentation semantics. *The Knowledge Engineering Review*, 26:365–410, 12 2011.
2. M. Caminada. On the issue of reinstatement in argumentation. In *Proceedings of the 10th European Conference on Logics in Artificial Intelligence*, JELIA'06, pages 111–123, Berlin, Heidelberg, 2006. Springer-Verlag.
3. M. Caminada and D. M. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2-3):109–145, 2009.
4. M. Caminada and G. Pigozzi. On judgment aggregation in abstract argumentation. *Autonomous Agents and Multi-Agent Systems*, 22(1):64–102, 2011.
5. P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

6. M. A. Falappa, A. J. García, G. Kern-Isberner, and G. R. Simari. On the evolving relation between belief revision and argumentation. *The Knowledge Engineering Review*, 26:35–43, 2 2011.
7. M. A. Falappa, A. J. García, G. Kern-Isberner, and G. R. Simari. Stratified belief bases revision with argumentative inference. *Journal of Philosophical Logic*, 42(1):161–193, 2013.
8. D. M. Gabbay and O. Rodrigues. Equilibrium states in numerical argumentation networks. *Logica Universalis*, pages 1–63, 2015.
9. O. Rodrigues. Gris system description. In Matthias Thimm and Serena Villata, editors, *System Descriptions of the First International Competition on Computational Models of Argumentation (ICCMA'15)*, pages 37–40. Cornell University Library, 2015. http://arxiv.org/abs/1510.05373.