Enforcement in Argumentation is a kind of Update*

Pierre Bisquert Claudette Cayrol Florence Dupin de Saint-Cyr Marie-Christine Lagasquie-Schiex
IRIT – UPS, Toulouse, France
{bisquert,ccayrol,dupin,lagasq}@irit.fr

June 2013

Abstract

In the literature, enforcement consists in changing an argumentation system in order to force it to accept a given set of arguments. In this paper, we extend this notion by allowing incomplete information about the initial argumentation system. Generalized enforcement is an operation that maps a propositional formula describing a system and a propositional formula that describes a goal, to a new formula describing the possible resulting systems. This is done under some constraints about the allowed changes. We give a set of postulates restraining the class of enforcement operators and provide a representation theorem linking them to a family of proximity relations on argumentation systems.

Keywords: dynamics in argumentation, belief change.

1 Introduction

During a trial, a lawyer makes her final address to the judge; the lawyer of the opposite party, say O, is able to build the argumentation system (a graph containing arguments and attacks relation between them) corresponding to this pleading. O is also able to compute all the arguments that are accepted according to the pleading, *i.e.*, the set of consensual arguments. Suppose now that O wants to force the audience to accept another set of arguments. She has to make a change to the argumentation system, either by adding an argument or by making an objection about an argument (to remove it) in order to achieve this goal. In the literature, the operation to perform on an argumentation system in order to ensure that a given set of arguments is accepted given a set of authorized changes is called "enforcement" [3].

This enforcement may be done more or less easily, since it may involve more or less changes (costs to add/remove arguments may be introduced). The aim of the speaker will be to find the least expensive changes to make to the argumentation system.

The previous example is a particular case of a more general enforcement operator. Since we could consider cases where Agent O does not know exactly the argumentation system on which she must make a change but knows only some information about it (*e.g.* some arguments that are accepted or that are present in the system). In this more general case, the idea is to ensure that the argumentation system after change satisfies a given goal whatever the initial system is. The result of enforcement will give a characterization of the set of argumentation systems that could be obtained (taking into account a set of authorized changes).

[*] This is a draft version, the article will appear In : 7th International Conference on Scalable Uncertainty Management (SUM'2013), Washington DC, USA, September 16 to 18, 2013

The key idea developed in this paper is the parallel between belief update theory [19, 16] and enforcement in argumentation. Enforcement consists in searching for the argumentation systems that are closest to a given starting argumentation system, in a set of argumentation systems in which some target arguments are accepted. This gives us the parallel with preorders on worlds in belief update. Hence worlds correspond to argumentation systems while formulas should represent knowledge about these argumentation systems. In classical enforcement this knowledge is expressed in terms of a description of an initial argumentation system and a set of arguments that one wants to see accepted. This is why we propose to introduce a propositional language in which this kind of information may be expressed. This language enables us to generalize enforcement with a broader expressiveness.

Our paper is situated in the growing domain of dynamics of argumentation systems [8, 7, 9, 3, 18, 17] which covers both addition and removal of arguments or interactions. It is organized as follows. We first restate abstract argumentation theory. Then we present a framework that illustrates a particular case of change in argumentation, it concerns an agent that wants to act on a given target system, this agent has a given goal and her possible actions are limited. We then recall classical enforcement. In the third section we propose a generalization of classical enforcement. Finally, we do a parallel with belief update. As classical update postulates do not allow to deal with restrictions about the authorized changes, we had to introduce a new set of postulates that characterizes generalized enforcement. All the proofs can be found in [4].

2 Framework

2.1 Abstract argumentation

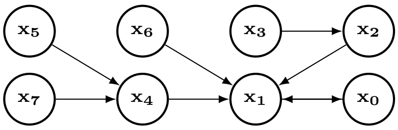
Let us consider a set $\mathtt{Arg}$ of symbols (denoted by lower case letters) representing a set of arguments and a relation $\mathtt{Rel}$ on $\mathtt{Arg} \times \mathtt{Arg}$. The pair $\langle \mathtt{Arg}, \mathtt{Rel} \rangle$, called *universe*, allows us to represent the set of possible arguments together with their interactions. More precisely, $\mathtt{Arg}$ represents a maybe infinite set of arguments usable in a given domain (*e.g.* if the domain is a knowledge base then $\mathtt{Arg}$ and $\mathtt{Rel}$ are the set of all arguments and interactions that may be built from the formulas of the base). We can also, as in the following example borrowed from [5], assume that $\mathtt{Arg}$ and $\mathtt{Rel}$ are explicitly provided.

Example 1 *During a trial*[1] *concerning a defendant (Mr. X), several arguments can be involved to determine his guilt. This set of arguments i.e., the set* $\mathtt{Arg}$ *and the relation* $\mathtt{Rel}$ *are given below.*

x_0	*Mr. X is not guilty of premeditated murder of Mrs. X, his wife.*
x_1	*Mr. X is guilty of premeditated murder of Mrs. X.*
x_2	*The defendant has an alibi, his business associate has solemnly sworn that he met him at the time of the murder.*

[1] In real life, lawyers may be confronted to tougher problems than the one presented here. Namely objection should often be done before an argument is fully laid out in order to stop the jury forming an impression. Unfortunately, this side of real life argumentation is not yet handled in our proposal.

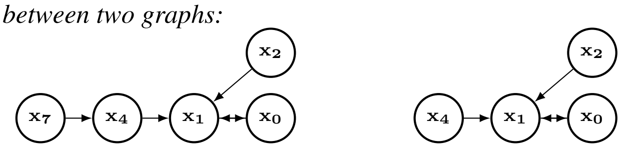
x_3	*The close working business relationships between Mr. X and his associate induce suspicions about his testimony.*
x_4	*Mr. X loves his wife so deeply that he asked her to marry him twice. A man who loves his wife cannot be her killer.*
x_5	*Mr. X has a reputation for being promiscuous.*
x_6	*The defendant had no interest to kill his wife, since he was not the beneficiary of the huge life insurance she contracted.*
x_7	*The defendant is a man known to be venal and his "love" for a very rich woman could be only lure of profit.*



A new definition of argumentation system derives directly from a universe $\langle \texttt{Arg}, \texttt{Rel} \rangle$. It differs slightly from the definition of [13] by the fact that arguments and interactions are taken in the universe. In the following, we will use indifferently "argumentation system" or "argumentation graph".

Definition 1 *An* argumentation graph *G is a pair (A, R) where $A \subseteq \texttt{Arg}$ is the* finite *set of vertices of G called "arguments" and $R \subseteq R_A = \texttt{Rel} \cap (A \times A)$ (R_A is the restriction of $\texttt{Rel}$ on A) is its set of edges, called "attacks". The set of argumentation graphs that may be built on the universe $\langle \texttt{Arg}, \texttt{Rel} \rangle$ is denoted by Γ. In the following, $x \in G$ when x is an argument, is a shortcut for $x \in A$.*

Example 2 *In Example 1, we consider that all the arguments of the universe are known by Agent O, but she is not sure about the content of the jury's argumentation system. She hesitates between two graphs:*



In argumentation theory, see [13], given such graphs, there are several ways to compute a set of "accepted" arguments. This computation depends on the way to select admissible groups of arguments, called "extensions"; several definitions can be considered for the "extensions", they are called "semantics". It depends also on the attribution of a status to arguments, for instance an argument can be "accepted skeptically" or respectively "credulously", if it belongs to all, respectively, to at least one extension. For sake of generality, we are not interested in a particular semantics nor on the mechanism used to instate the status of the arguments. We only consider a function $acc : \Gamma \rightarrow 2^{\texttt{Arg}}$ which associates with any argumentation graph G the set of arguments that have an accepted status in G according to a given semantics and a given status computation[2].

We will define a propositional language $\mathscr{L}$ in order to be able to describe an argumentation system and its set of accepted arguments. Its semantics will be defined with respect to Γ. $\forall \varphi \in \mathscr{L}$, we denote by $[\varphi]$ the set of argumentation graphs such that φ is

[2] This function could be parameterized by the precise semantics used.

true in these graphs, namely $[\varphi] = \{G \in \Gamma \text{ s.t. } \varphi \text{ is true in } G\}$. As usual, we denote $G \models \varphi$ iff $G \in [\varphi]$ and $\varphi \models \psi$ iff $[\varphi] \subseteq [\psi]$.

For sake of simplicity in all the examples, we are going to use a restricted propositional language $\mathscr{L}_{\texttt{Arg}}$, only able to express conditions about the presence or the accepted status of an argument in a graph. With this language, we can only handle examples about argument addition or removal. Hence, changes about interactions won't be considered, which allows us to assume that R is always equal to R_A in all our examples.

Definition 2 *Let $\Gamma_{\texttt{Arg}}$ be the set of argumentation graphs (A, R_A) that may be built on* $\texttt{Arg}$. *Let $\mathscr{L}_{\texttt{Arg}}$ be the propositional language associated with the vocabulary $\{a(x), on(x) \mid x \in \texttt{Arg}\}$*[3]*, with the usual connectives $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ and constants $\bot$ and $\top$. Its semantics is defined with respect to $\Gamma_{\texttt{Arg}}$ as follows: let $G \in \Gamma_{\texttt{Arg}}$*

- *the formula $\bot$ is always false in G*
- *the formula $\top$ is always true in G*
- *if $x \in \texttt{Arg}$ then*
  - *the formula $a(x)$ is true in G iff $x \in acc(G)$,*
  - *the formula $on(x)$ is true in G iff $x \in G$*
- *the non atomic formulas are interpreted as usual, $\neg\varphi$ is true in G if φ is not true in G, $\varphi_1 \vee \varphi_2$ is true in G if φ_1 or φ_2 is true in G, etc.*

Note that every accepted argument in a graph should belong to the graph, hence in $\mathscr{L}_{\texttt{Arg}}, \forall G \in \Gamma_{\texttt{Arg}}, \forall x \in \texttt{Arg}, G \models a(x) \rightarrow on(x)$.

Definition 3 *The* characteristic function $f_{\texttt{Arg}}$ *associated with $\mathscr{L}_{\texttt{Arg}}$, $f_{\texttt{Arg}} : \Gamma_{\texttt{Arg}} \rightarrow \mathscr{L}_{\texttt{Arg}}$, is defined by:*
$\forall G \in \Gamma_{\texttt{Arg}}, f_{\texttt{Arg}}(G) = \bigwedge_{x \in G} on(x) \wedge \bigwedge_{x \in \texttt{Arg} \setminus G} \neg on(x)$.

Note that, in Definition 2, the attack relation being fixed, if the set of arguments belonging to G is known then G is perfectly known. More formally, $f_{\texttt{Arg}}(G)$ characterizes G in a unique way:

Property 1 $\forall G \in \Gamma_{\texttt{Arg}}$, $[f_{\texttt{Arg}}(G)] = \{G\}$

Example 3 *The jury's system is not completely known by Agent O. It is represented in $\mathscr{L}_{\texttt{Arg}}$ by the formula $\varphi_{Jury} = on(x_0) \wedge on(x_1) \wedge on(x_2) \wedge on(x_4) \wedge \neg on(x_3) \wedge \neg on(x_5) \wedge \neg on(x_6) \wedge (on(x_7) \vee \neg on(x_7))$ which covers the two graphs drawn in Example 2; the disjunction between $on(x_7)$ and $\neg on(x_7)$ expresses the fact that Agent O hesitates. Moreover, x_0, x_2 and (x_4 or x_7) are the only members of the "grounded extension" [13]. Hence, $\varphi_{Jury} \models a(x_0) \wedge a(x_2) \wedge (a(x_4) \vee a(x_7))$.*

Note that the idea to write propositional formulas for expressing acceptability of arguments was first proposed in [11]. This was done with a completely different aim, namely to generalize Dung's argumentation framework by taking into account additional constraints (expressed in logic) on the admissible sets of arguments.

2.2 Change in argumentation

In this section we propose a definition of change in argumentation based on the work of [9, 6] and adapted to the encoding of generalized enforcement operators. [9] have distinguished four change operations. An elementary change is either adding/removing an

[3] "a" stands for "accepted in G" while "on" stands for "belongs to G".

argument with a set of attacks involving it, or adding/removing an attack. According to the restriction explained in Section 2.1, we only present in Definition 4 the operations of addition and removal on arguments. Moreover operations are only defined for specific argumentation systems of the form (A, R_A) where $R_A = \texttt{Rel} \cap (A \times A)$ i.e. R_A contains all the attacks concerning arguments of A that are present in the universe $(\texttt{Arg}, \texttt{Rel})$. Note that this definition gives only a particular example of change operations when the attack relation is fixed.

The purpose of the following definitions is the introduction of a particular framework, that will be used to illustrate enforcement. In this framework, we consider an agent that may act on a target argumentation system. This agent has a goal and should follow some constraints about the actions she has the right to do. For instance, an agent can only advance arguments that she knows. Hence some restrictions are added on the possible changes that may take place on the system. These constraints are represented by the notion of *executable operation*.

We first refine the notion of *elementary operation* within the meaning of [9] in four points: first a precise syntax is given; then we define an *allowed operation* w.r.t. a given agent's knowledge; we restrict this notion w.r.t. its feasibility on the target system (it is not possible to add an already present argument or to remove an argument which was not in the graph), it leads to the notion of *executable operation*; and finally, we study the impact of an operation on an argumentation system. Note that considering only elementary operations does not result in a loss of generality since any change can be translated into a sequence of elementary operations, called program in Definition 5.

Definition 4 *Let k be an agent and $G_k = \langle A_k, R_{A_k} \rangle$ be her argumentation system and let $G = \langle A, R_A \rangle$ be an argumentation system.*

- *An* elementary operation *is a pair $o = \langle op, x \rangle$ where $op \in \{\oplus, \ominus\}$ and $x \in \texttt{Arg}$.*
- *An elementary operation $\langle op, x \rangle$ is* allowed *for k iff $x \in A_k$.*[4]
- *An operation* executable *by k on G is an operation $\langle op, x \rangle$ allowed for k such that:*
  - *if $op = \oplus$ then $x \notin A$*
  - *if $op = \ominus$ then $x \in A$.*
- *An operation $o = \langle op, x \rangle$ executable by k on G provides a new argumentation system $G' = o(G) = \langle A', R_{A'} \rangle$ such that:*
  - *if $op = \oplus$ then $G' = \langle A \cup \{x\}, R_{A \cup \{x\}} \rangle$*
  - *if $op = \ominus$ then $G' = \langle A \setminus \{x\}, R_{A \setminus \{x\}} \rangle$*

Example 4 *From* `Arg` *and* `Rel` *given in Example 1, several elementary operations are syntactically correct, e.g., $\langle \oplus, \{x_2\} \rangle$ and $\langle \ominus, \{x_4\} \rangle$. Among the elementary operations, Agent O is only allowed to use those concerning arguments she knows. Since O learnt all about this trial, all the elementary operations are allowed for O. $\langle \oplus, \{x_5\} \rangle$, $\langle \ominus, \{x_4\} \rangle$, $\langle \ominus, \{x_2\} \rangle$ are some executable operations for O on the systems described by φ_{Jury}.*

Finally, we consider sequences of operations executed by an agent on an argumentation system, called *programs*, which are providing the possibility for an agent to perform several elementary operations one after the other.

Definition 5 *Let $G = \langle A, R_A \rangle$ be an argumentation system. A* program *p executable by an agent k on G is a finite ordered sequence of n operations $(o_1, \cdots, o_n)$ s.t.:*

[4] Note that in the case of an argument addition, if the attack relation had not been imposed then it would have been possible to add an argument with only a part of the known attacks and therefore to "lie by omission" or to add attacks unknown to the agent and therefore lie in an "active" way. This will be the subject of future work.

- $n = 1$ *:* o_1 *is executable by* k *on* G*. Hence* $p(G) = o_1(G)$*.*
- $n > 1$ *:* $(o_1, \cdots, o_{n-1})$ *is a program* p' *executable by* k *on* G *such that* $p'(G) = G'$ *and* o_n *is executable by* k *on* G'*. Hence* $p(G) = o_n(G')$*.*
- *By extension, an empty sequence is also a program. Hence, for* $p = ()$*,* $p(G) = G$*.*

2.3 Enforcement

The main references about enforcement are [3, 2] that address the following question : is it possible to change a given argumentation system, by applying change operations, so that a desired set of arguments becomes accepted? Baumann has specified necessary and sufficient conditions under which enforcements are possible, in the case where change operations are restricted to the addition of new arguments and new attacks. More precisely, [2] introduces three types of changes called expansions: the *normal expansion* adds new arguments and new attacks concerning at least one of the new arguments, the *weak expansion* refines the normal expansion by the addition of new arguments not attacking any old argument and the *strong expansion* refines the normal expansion by the addition of new arguments not being attacked by any old argument.

It is not the case in general that any desired set of arguments is enforceable using a particular expansion. Moreover, in some cases, several enforcements are possible, some of them requiring more effort than others. In order to capture this idea, [2] introduces the notion of characteristic which depends on a semantics and on a set of possible expansions. The characteristic of a set of arguments is defined as the minimal number of modifications (defined by the differences between the attacks on the two graphs) that are needed in order to enforce this set of arguments. This number equals 0 when each argument of the desired set is already accepted. It equals infinity if no enforcement is possible. [2] provides means to compute the characteristic w.r.t. a given type of expansion and a given semantics.

3 Towards Generalized Enforcement

Let us formalize enforcement using the definitions presented in Section 2.1. Let $G \in \Gamma$ and $X \subseteq \texttt{Arg}$. An enforcement of X on G is a graph $G' \in \Gamma$ obtained from G by applying change operations and such that $X \subseteq acc(G')$. Different enforcements of X on G can be compared using a preorder $\preceq_G$. For instance, it seems natural to look for enforcements performing a minimal change on G. Minimality can be based on a distance for instance. In that case, given two enforcements G' and G'' of X on G, $G' \preceq_G G''$ may be defined as $distance(G, G') \leq distance(G, G'')$.

This preorder $\preceq_G$ suggests to draw a parallel between the enforcement problem and an update problem. Indeed, as we will see in Section 4.1, update is also related to the same kind of preorder on worlds w.r.t. a given world. More precisely an update operator maps a knowledge base and a piece of information to a new knowledge base, where knowledge bases are expressed in terms of propositional formulas. The semantic counterpart of this mapping is defined by operations on models of formulas, *i.e.*, worlds. This gives birth to the idea that graphs are to worlds what formulas characterizing sets of graphs are to formulas characterizing sets of worlds.

Definition 2 enables us to continue the parallel. Let $S \subseteq \texttt{Arg}$ and $\alpha = \bigwedge_{x \in S} a(x)$. $[\alpha]$ can be considered as the set of graphs in which the elements of S are accepted. In other words, $[\alpha]$ plays the role of the set of graphs that accept S.

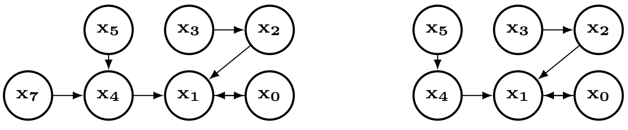
This leads to formalize an enforcement problem as an operator applying to propositional formulas, with a semantic counterpart working with argumentation graphs. So enforcing a propositional formula α on a propositional formula φ means enforcing α on the graphs that satisfy φ.

This setting allows us to have two generalizations of enforcement: first it is now possible to use enforcement not only to impose that a set of arguments is accepted, but also to make enforcement with any goal that can be expressed in a propositional language describing graphs. Second, the initial graph does not necessarily have to be completely known since a description in a propositional language allows for a richer expressivity. Hence, a set of graphs will be considered as representing the initial state of the argumentation system.

Let us explain more precisely the notion of *goal*: it reflects conditions that an agent would like to see satisfied in a particular argumentation system. We may consider two types of goals, namely "absolute" and "relative". An absolute goal only takes into account the resulting system after modifying the target system; it formally focuses on G'. A relative goal takes into account the target system and its resulting system; it formally focuses on (G, G'). An example of relative goal could be that the number of accepted arguments increases after enforcement. In the following, we only consider absolute goals, since relative goals are difficult to express in an update manner.

These goals could involve the arguments, the extensions, the set of extensions as well as its cardinality, the set of extensions containing a particular argument as well as its cardinality. Hence goals are represented by expressions involving these notions and that may contain classic comparison operators ($=$, $<$, $>$, etc.), quantifiers $\forall$ and $\exists$, membership ($\in$) and inclusion ($\subseteq$), union ($\cup$) and intersection ($\cap$) of sets, classical logic operators ($\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\neg$). If we associate a propositional formula with an absolute goal then a goal is satisfied in a graph if the associated formula holds in this graph.

Example 5 *We know that O wants to enforce the set $\{x_1\}$. This goal can be expressed in $\mathscr{L}_{\texttt{Arg}}$ by the formula $a(x_1)$. To enforce Argument x_1 on the Jury's graph, O can use the program $(\langle\oplus, x_5\rangle, \langle\oplus, x_3\rangle)$ which has the impact shown in the following graphs. A more complex goal could be e.g., $\neg a(x_4) \vee a(x_0)$.*



We are now able to define formally generalized enforcement.

Requirement: *Generalized enforcement is based on a propositional language $\mathscr{L}$ able to describe any argumentation system and its set of accepted arguments, and a characteristic function f associated with $\mathscr{L}$, such that $\forall G \in \Gamma$, $[f(G)] = \{G\}$.*

For instance, $\mathscr{L}_{\texttt{Arg}}$ of Definition 2 could be used as $\mathscr{L}$. However, $\mathscr{L}_{\texttt{Arg}}$ does not allow to express conditions about the cardinality of each extension after enforcement. $\mathscr{L}_{\texttt{Arg}}$ is only an example that has been introduced for illustrative purpose. The following results hold for any propositional language $\mathscr{L}$.

In order to capture classical enforcement we also need to be able to restrict the ways that graphs are allowed to change. This is done by introducing a set $T \subseteq \Gamma \times \Gamma$ of allowed transitions between graphs.

Here are three examples of sets of allowed transitions that could be used:

- If the allowed changes are executable elementary operations for an agent k then $T_e^k = \{(G, G') \in \Gamma \times \Gamma$, $\exists o$ s.t. o is an elementary executable operation by k on G s.t. $o(G) = G'\}$.
- If the allowed changes are executable programs by an agent k then $T_p^k = \{(G, G') \in \Gamma \times \Gamma$, $\exists p$ s.t. p is an executable program by k on G s.t. $p(G) = G'\}$
- Baumann's normal expansion can be translated in terms of allowed transitions as follows: $T_B = \{(G, G') \in \Gamma \times \Gamma$, with $G = (A, R_A)$ and $G' = (A', R_{A'})$ s.t. $A \subsetneq A'\}$. It means that the transitions admitted by Baumann's normal expansion are restricted to the addition of a new set of arguments.

Now, we are in position to define formally a generalized enforcement operator:

Definition 6 *A generalized enforcement operator is a mapping relative to a set of authorized transitions $T \subseteq \Gamma \times \Gamma$ from $\mathscr{L} \times \mathscr{L} \to \mathscr{L}$ which associates with any formula φ giving information about a target argumentation system, and any formula α encoding a goal, a formula, denoted $\varphi \Diamond_T \alpha$, characterizing the argumentation systems in which α holds, that can be obtained by a change belonging to T.*

Example 6 *In Example 5, if Agent O wants to enforce acceptation of x_1 when x_2 and x_4 are present (w.r.t. the grounded semantics) with an executable program then she can use the following result: $\varphi_{Jury} \Diamond_{T_p^O} \ (a(x_1) \wedge on(x_2) \wedge on(x_4)\ ) \models on(x_3) \wedge on(x_5)$*

Notation: $\forall \varphi, \psi \in \mathscr{L}$, *a transition in T is possible between a set of graphs satisfying φ to a set of graphs satisfying ψ, denoted $(\varphi, \psi) \models T$, iff ($[\varphi] \neq \varnothing$ and $\forall G \in [\varphi]$, $\exists G' \in [\psi]$, $(G, G') \in T$).* In other words, a transition from a given set of graphs towards another set is possible, iff there is a possible transition from *each graph* of the first set (which should not be empty) towards at least one graph of the second set.

4 Generalized Enforcement Postulates

4.1 Background on Belief Change Theory

In the field of belief change theory, the paper of AGM [1] has introduced the concept of "belief revision". Belief revision aims at defining how to integrate a new piece of information into a set of initial beliefs. Beliefs are represented by sentences of a formal language. Revision consists in adding information while preserving consistency.

A very important distinction between belief revision and belief update was first established in [19]. The difference is in the nature of the new piece of information: either it is completing the knowledge of the world or it informs that there is a change in the world. More precisely, update is a process which takes into account a physical evolution of the system while revision is a process taking into account an epistemic evolution, it is the knowledge about the world that is evolving. In this paper, we rather face an update problem, since in enforcement, the agent wants to change a graph in order to ensure that some arguments are now accepted (graphs play the role of worlds, as explained in Section 3)[5].

We need to recall some background on belief update. An update operator [19, 16] is a function mapping a knowledge base φ, expressed in a propositional logic $\mathscr{L}$, representing knowledge about a system in an initial state and a new piece of information

[5] A revision approach would apply to situations in which the agent learns some information about the initial argumentation system and wants to correct her knowledge about it. This would mean that the argumentation system has not changed but the awareness of the agent has evolved.

$\alpha \in \mathscr{L}$, to a new knowledge base $\varphi \diamond \alpha \in \mathscr{L}$ representing the system after this evolution. In belief update, the input α should be interpreted as the projection of the expected effects of some "explicit change", or more precisely, the expected effect of the action "make α true". The key property of belief update is Katsuno and Mendelzon's Postulate U8 which tells that models of φ are updated independently (contrarily to belief revision). We recall here the postulates of Katsuno and Mendelzon, where $\mathscr{L}$ denotes any propositional language and $[\varphi]$ denotes the set of models of the formula φ:[6] $\forall \varphi, \psi, \alpha, \beta \in \mathscr{L}$,

U1: $\varphi \diamond \alpha \models \alpha$
U2: $\varphi \models \alpha \implies [\varphi \diamond \alpha] = [\varphi]$
U3: $[\varphi] \neq \varnothing$ and $[\alpha] \neq \varnothing \implies [\varphi \diamond \alpha] \neq \varnothing$
U4: $[\varphi] = [\psi]$ and $[\alpha] = [\beta] \implies [\varphi \diamond \alpha] = [\psi \diamond \beta]$
U5: $(\varphi \diamond \alpha) \wedge \beta \models \varphi \diamond (\alpha \wedge \beta)$
U8: $[(\varphi \vee \psi) \diamond \alpha] = [(\varphi \diamond \alpha) \vee (\psi \diamond \alpha)]$
U9: if $card([\varphi]) = 1$ then $[(\varphi \diamond \alpha) \wedge \beta] \neq \varnothing \implies \varphi \diamond (\alpha \wedge \beta) \models (\varphi \diamond \alpha) \wedge \beta$ (where $card(E)$ denotes the cardinality of the set E)

These postulates allow Katsuno and Mendelzon to write the following representation theorem concerning update, namely, an operator satisfying these postulates can be defined by means of a ternary preference relation on worlds (the set of all worlds is denoted by Ω).

Theorem 1 ([16]) *There is an operator $\diamond : \mathscr{L} \times \mathscr{L} \to \mathscr{L}$ satisfying U1, U2, U3, U4, U5, U8, U9 iff there is a* faithful *assignment that associates with each $\omega \in \Omega$ a complete preorder, denoted $\preceq_\omega$ s.t. $\forall \varphi, \alpha \in \mathscr{L}$, $[\varphi \diamond \alpha] = \bigcup_{\omega \in [\varphi]} \{\omega' \in [\alpha]$ s.t. $\forall \omega'' \in [\alpha], \omega' \preceq_\omega \omega''\}$*
where an assignment of a preorder[7] *$\preceq_\omega$ to each $\omega \in \Omega$ is* faithful *iff $\forall \omega, \omega' \in \Omega$, $\omega \prec_\omega \omega'$.*

This set of postulates has already been broadly discussed in the literature (see *e.g.*, [15, 14, 12]). U2 for instance imposes inertia which is not always suitable. Herzig [14] proposes to restrict possible updates by taking into account integrity constraints, *i.e.*, formulas that should hold before and after update. Dubois et al. [12] proposes to not impose inertia and to allow for update failure even if the formulas are consistent. This is done by introducing an unreachable world called z in order to dispose of an upper bound of the proximity from a current world to an unreachable world. In the following, as seen in Section 3, we want to restrain the possible changes. Hence we have to allow for enforcement failure. As we have seen, we choose to introduce a set of allowed transitions T which restricts possible enforcements. The idea to define an update operator based on a set of authorized transitions was first introduced by Cordier and Siegel [10]. Their proposal goes beyond our idea since they allow for a greater expressivity by using prioritized transition constraints. However, this proposal is only defined at a semantical level (in terms of preorders between worlds), hence they do not provide postulates nor representation theorem associated with their update operator. Moreover our idea to define postulates related to a set T of authorized transitions generalizes [14] since integrity constraints can be encoded with T (the converse is not possible). Consequently, we have now to adapt update postulates in order to restrict possible transitions.

[6] Postulates U6 and U7 are not considered here since the set U1-U8 is only related to a family of partial preorders while replacing U6-U7 by U9 ensures a family of complete preorders.

[7] In the following, $\prec_\omega$ is defined from $\preceq_\omega$ as usual by: $a \prec_\omega b$ iff $a \preceq_\omega b$ and not $b \preceq_\omega a$.

4.2 Postulates characterizing enforcement on graphs with transition constraints

We are going to define a set of rational postulates for $\diamondsuit_T$. These postulates are constraints that aim at translating the idea of enforcement. Some postulates coming from update are suitable, namely U1, since it ensures that after enforcement the constraints imposed by α are true. U2 postulate is optional, it imposes that if α already holds in a graph then enforcing α means no change. This postulate imposes inertia as a preferred change, this may not be desirable in all situations. U3 transposed in terms of graphs imposes that if a formula holds for some graphs and if the update piece of information also holds for some graphs then the result of enforcement should give a non empty set of graphs. Here, we do not want to impose that any enforcement is always possible since some graphs may be unreachable from others. So we propose to replace U3 by a postulate called E3 based on the set of authorized transitions T: $\forall \varphi, \psi, \alpha, \beta \in \mathscr{L}$

E3: $[\varphi \diamondsuit_T \alpha] \neq \varnothing$ iff $(\varphi, \alpha) \models T$

Due to the definition of $(\varphi, \alpha) \models T$, E3 handles two cases of enforcement impossibility: no possible transition and no world (*i.e.* no graph satisfying φ or α, as it will be shown in Proposition 3).

U4 is suitable in our setting since enforcement operators are defined semantically. U5 is also suitable for enforcement since it says that graphs enforced by α in which β already holds are graphs in which the constraints α and β are enforced. Due to the fact that we want to allow for enforcement failure, this postulate had been restricted to "complete" formulas[8].

E5: if $card([\varphi]) = 1$ then $(\varphi \diamondsuit_T \alpha) \wedge \beta \models \varphi \diamondsuit_T (\alpha \wedge \beta)$if $card([\varphi]) = 1$ then $(\varphi \diamondsuit_T \alpha) \wedge \beta \models \varphi \diamondsuit_T (\alpha \wedge \beta)$

U8 captures the decomposability of enforcement with respect to a set of possible input attack graphs. We slightly change this postulate in order to take into account the possibility of failure, namely if enforcing something is impossible then enforcing it on a larger set of graphs is also impossible, else the enforcement can be decomposable:

E8 if $([\varphi] \neq \varnothing$ and $[\varphi \diamondsuit_T \alpha] = \varnothing)$ or $([\psi] \neq \varnothing$ and $[\psi \diamondsuit_T \alpha] = \varnothing)$
then $[(\varphi \vee \psi) \diamondsuit_T \alpha] = \varnothing$
else $[(\varphi \vee \psi) \diamondsuit_T \alpha] = [(\varphi \diamondsuit_T \alpha) \vee (\psi \diamondsuit_T \alpha)]$

Postulate U9 is a kind of converse of U5 but restricted to a "complete" formula φ *i.e.* such that, $card([\varphi]) = 1$, this restriction is required in the proof of KM theorem as well as in Theorem 2.

Note that the presence of U1 in the set of postulates characterizing an enforcement operator is not necessary since U1 can be derived from E3, E5 and E8.

Proposition 1 *U1 is implied by E3, E5 and E8.*

These postulates allow us to write the following representation theorem concerning enforcement, namely, an enforcement operator satisfying these postulates can be defined by means of the definition of a family of preorders on graphs.

Definition 7 *Given a set $T \subseteq \Gamma \times \Gamma$ of accepted transitions, an* assignment respecting T *is a function that associates with each $G \in \Gamma$ a complete preorder $\preceq_G$ such that $\forall G_1, G_2 \in \Gamma$, if $(G, G_1) \in T$ and $(G, G_2) \notin T$ then $G_2 \not\preceq_G G_1$.*

[8] Note that $card[\varphi] = 1$ iff $\exists G \in \Gamma$ s.t. $[\varphi] = [f(G)]$.

Theorem 2 *Given a set $T \subseteq \Gamma \times \Gamma$ of accepted transitions, there is an operator $\Diamond_T : \mathscr{L} \times \mathscr{L} \to \mathscr{L}$ satisfying E3, U4, E5, E8, U9 iff there is an assignment respecting T s.t. $\forall G \in \Gamma$, $\forall \varphi, \alpha \in \mathscr{L}$,*

- $[f(G)\Diamond_T\alpha] = \{G_1 \in [\alpha]$ *s.t.* $(G, G_1) \in T$ *and* $\forall G_2 \in [\alpha]$ *s.t.*$(G, G_2) \in T,$ $G_1 \preceq_G G_2\}$
- $[\varphi\Diamond_T\alpha] = \left| \begin{array}{ll} \varnothing & \textit{if } \exists G \in [\varphi] \textit{ s.t. } [f(G)\Diamond_T\alpha] = \varnothing \\ \bigcup_{G\in[\varphi]}[f(G)\Diamond_T\alpha] & \textit{otherwise} \end{array} \right.$

This result is a significant headway, but as usual for a representation theorem, it gives only a link between the existence of an assignment of preorders and the fact that an enforcement operator satisfies the postulates. It does not give any clue about how to assign these preorders i.e., how to design precisely an enforcement operator.

The following proposition establishes the fact that 5 postulates are necessary and sufficient to define an enforcement operator, namely E3, U4, E5, E8 and U9. Indeed, U1 can be derived from them (as seen in Proposition 1).

Proposition 2 *E3, U4, E5, E8, U9 constitute a minimal set: no postulate can be derived from the others.*

From Theorem 2 we can deduce two simple cases of impossibility: if the initial situation or the goal is impossible then enforcement is impossible (this result is a kind of converse of U3).

Proposition 3 *If $\Diamond_T$ satisfies E3, U4, E5, E8 and U9 then $([\varphi] = \varnothing$ or $[\alpha] = \varnothing \implies [\varphi\Diamond_T\alpha] = \varnothing)$.*

The following property ensures that if an enforcement is possible then a more general enforcement is also possible.

Proposition 4 *If $\Diamond_T$ satisfies E3 then $([\varphi] \neq \varnothing$ and $[\varphi\Diamond_T\alpha] \neq \varnothing \implies [\varphi\Diamond_T(\alpha \vee \beta)] \neq \varnothing)$.*

Note that there are some cases where U2 does not hold together with E3, U4, E5, E8 and U9. If U2 is imposed then the enforcement operator is associated with a preorder in which a given graph is always closer to itself than to any other graph. This is why it imposes to have a faithful assignment. In that case, the relation represented by T should be reflexive.

Definition 8 *A* faithful assignment *is a function that associates with each $G \in \Gamma$ a complete preorder*[9] *$\preceq_G$ such that $\forall G_1 \in \Gamma$, $G \prec_G G_1$.*

Proposition 5 *Given a* reflexive *relation $T \subseteq \Gamma \times \Gamma$ of accepted transitions, there is an operator $\Diamond_T : \mathscr{L} \times \mathscr{L} \to \mathscr{L}$ satisfying E3, U4, E5, E8, U9 that satisfies U2 iff there is a faithful assignment respecting T defined as in Theorem 2.*

If we remove the constraint about authorized transitions then we recover Katsuno and Mendelzon theorem, namely:

Proposition 6 *If $T = \Gamma \times \Gamma$ then $\Diamond_T$ satisfies U2, E3, U4, E5, E8, U9 iff $\Diamond$ satisfies U1, U2, U3, U4, U5, U8 and U9.*

[9] In the following, $\prec_G$ is defined from $\preceq_G$ as usual by: $a \prec_G b$ iff $a \preceq_G b$ and not $b \preceq_G a$.

Among the different kinds of changes proposed by Baumann, the normal expansion, *i.e.*, adding an argument with the attacks that concern it, can be encoded in our framework as follows.

Remark 1 *Baumann's enforcement by normal expansion is a particular enforcement operator $\diamondsuit_T : \mathscr{L} \times \mathscr{L} \to \mathscr{L}$ such that $T = T_B$. Moreover, the language used is restricted as follows: the formulas that describe the initial system are restricted to $\{\varphi \in \mathscr{L}_{on}, card([\varphi]) = 1\}$ and the formulas that describe the facts that should be enforced are only conjunctions of positive literals of $\mathscr{L}_a$, where $\mathscr{L}_a$ and $\mathscr{L}_{on}$ are respectively the propositional languages based only on $a(x)$ and on $on(x)$ variables.*

In Baumann's framework, the formula concerning the initial graph should be complete, *i.e.*, should correspond to only one graph. The formula concerning the goal of enforcement should describe a set of arguments that should be accepted (under a given semantics) after the change. Due to Theorem 2, there exists a family of preorders that could be defined. Baumann proposes to use the following: $G' \preceq_G G''$ iff $dist(G, G') \leq dist(G, G'')$ where $dist(G, G')$ is the number of attacks that differs in G and G'.[10]

5 Conclusion

The work of [2] gives the basics about enforcement, our approach investigates several new issues:

- we propose to take into account the ability to remove an argument, which could help to enforce a set of arguments with less effort. We also generalize what can be enforced, not only sets of arguments can be enforced but any goal that can be expressed in propositional logic is allowed.
- we enable the possibility to restrict the authorized changes. In generalized enforcement, authorized changes may be described by a set of possible transitions. Hence, the structure of the changes can be restricted (for instance to additions only or to elementary operations) as well as the arguments that are allowed to be added/removed.

Finally, our main contribution is to state that enforcement is a kind of update, which allows for an axiomatic approach. This kind of update is more general than classical update since it allows to take into account transition constraints.

In this paper, for sake of shortness, we use a simplified logical language for describing argumentation systems in our examples, this makes us focus only on changes about arguments hence allow us to consider a fixed attack relation. However our results hold on any given propositional logic, hence choosing a logic in which attacks are encoded would enable us to deal with changes on attacks. This deserves more investigation.

Another issue is to find postulates that are more specific for argumentation dynamics. Indeed, we have defined a set of postulates that may characterize changes in any kind of graphs that can be defined in propositional logic, provided that a transition function is given. Further research should take into account the particularities of the graphs representing argumentation systems (semantics notions should be introduced in the postulates). Moreover, in this paper we have focused on a representation theorem based on complete preorders between pairs of argumentation graphs ; another study

[10] Note that since Baumann's enforcement is defined on one graph and not on a set of graphs, then it is also a kind of belief revision since revision and update collapse when the initial world is completely known (this kind of belief revision won't be a pure AGM revision but rather a revision under transition constraints).

would be required for partial preorders. Finally, it would be worthwhile to study what could be the counterpart of enforcement for revision instead of update.

References

[1] Alchourrón, C., Gärdenfors, P., Makinson, D.: On the logic of theory change : partial meet contraction and revision functions. Journal of Symbolic Logic 50, 510–530 (1985)

[2] Baumann, R.: What does it take to enforce an argument? minimal change in abstract argumentation. In: ECAI. pp. 127–132 (2012)

[3] Baumann, R., Brewka, G.: Expanding argumentation frameworks: Enforcing and monotonicity results. In: Proc. of COMMA. pp. 75–86. IOS Press (2010)

[4] Bisquert, P., Cayrol, C., Dupin de Saint Cyr, F., Lagasquie-Schiex, M.C.: Axiomatic approach of enforcement in argumentation. Tech. rep., IRIT, Toulouse, France (2013), ftp://ftp.irit.fr/pub/IRIT/ADRIA/rap-IRIT-2013-24.pdf

[5] Bisquert, P., Cayrol, C., Dupin de Saint-Cyr, F., Lagasquie-Schiex, M.: Change in argumentation systems: exploring the interest of removing an argument. In: Proc. of SUM. pp. 275–288. Springer-Verlag (2011)

[6] Bisquert, P., Cayrol, C., de Saint Cyr, F.D., Lagasquie-Schiex, M.: Duality between Addition and Removal. In: Proc. of IPMU. pp. 219–229. Springer (2012)

[7] Boella, G., Kaci, S., van der Torre, L.: Dynamics in argumentation with single extensions: Abstraction principles and the grounded extension. In: Proc. of ECSQARU (LNAI 5590). pp. 107–118 (2009)

[8] Boella, G., Kaci, S., van der Torre, L.: Dynamics in argumentation with single extensions: Attack refinement and the grounded extension. In: Proc. of AAMAS. pp. 1213–1214 (2009)

[9] Cayrol, C., Dupin de Saint-Cyr, F., Lagasquie-Schiex, M.C.: Change in abstract argumentation frameworks: Adding an argument. Journal of Artificial Intelligence Research 38, 49–84 (2010)

[10] Cordier, M., Siegel, P.: Prioritized transitions for updates. In: Lecture Notes in Artificial Intelligence 946 (Proc. of ECSQARU-95). pp. 142–150. Fribourg, Switzerland (1995)

[11] Coste-Marquis, S., Devred, C., Marquis, P.: Constrained argumentation frameworks. In: Proc. of KR. pp. 112–122 (2006)

[12] Dubois, D., Dupin de Saint-Cyr, F., Prade, H.: Update postulates without inertia. In: Proc. of ECSQARU. pp. 162–170. No. 946 in LNAI, Springer-Verlag (1995)

[13] Dung, P.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence 77(2), 321–358 (1995)

[14] Herzig, A.: On updates with integrity constraints. In: Belief Change in Rational Agents (2005)

[15] Herzig, A., Rifi, O.: Propositional belief base update and minimal change. Artificial Intelligence 115, 107–138 (1999)

[16] Katsuno, H., Mendelzon, A.: On the difference between updating a knowledge base and revising it. In: Proc. of KR. pp. 387–394 (1991)

[17] Liao, B., Jin, L., Koons, R.: Dynamics of argumentation systems: A division-based method. Artificial Intelligence 175(11), 1790 – 1814 (2011)

[18] Moguillansky, M.O., Rotstein, N.D., Falappa, M.A., García, A.J., Simari, G.R.: Argument theory change through defeater activation. In: Proc. of COMMA. pp. 359–366. IOS Press (2010)

[19] Winslett, M.: Reasoning about action using a possible models approach. In: Proc. of AAAI. pp. 89–93 (1988)