# Argumentation Semantics for Agents

Cristian Gratie and Adina Magda Florea

Computer Science Department,
University "Politehnica", Bucharest, Romania

**Abstract.** The paper introduces an argumentation semantics that can deal with several challenges that arise when using abstract argumentation within multi-agent systems. The extensions are computed with respect to initial constraints that specify the desired justification state of some arguments. The constraints can come from the agent's goals, its confidence in information from other agents or they may describe a decision context, where the agent must choose between several alternatives. The core idea behind the approach is the fact that, in order to find an extension that satisfies the constraints, an agent needs to find a suitable set of arguments to defeat.

We provide a full scenario where an auction for two items is modeled as a game where the participating agents take turns at updating an argumentation framework describing the possible states of the environment as well as the agents' intentions. The agents' goals and the consistency of the environment's state are described with constraints. Our argumentation semantics is shown to provide a very natural strategy for the agents playing this game. It can also be used at the end of the game for deciding its outcome, namely the final state of the environment and the actions of the agents.

**Keywords:** argumentation, semantics, multi-agent systems.

## 1 Introduction

Abstract argumentation was introduced by Dung [8] in 1995 and has been a hot research topic since. Several approaches were defined in the literature for using argumentation in artificial intelligence and several works deal with abstract argumentation itself.

The most common approach for using argumentation with multi-agent systems relies on extending the model with some additional features that make it more expressive for use with agents, such as preferences or values.

This paper aims to provide a different approach, by defining an argumentation semantics that can deal with the challenges of using argumentation frameworks in multi-agent systems. More precisely, our approach does not change the formal model proposed by Dung, it only defines a new semantics that has properties relevant for use in multi-agent systems.

Section 2 provides some argumentation background, together with a discussion of related work. Our approach is presented in Section 3. The details of an

M. Cossentino et al. (Eds.): EUMAS 2011, LNAI 7541, pp. 129–144, 2012.

argumentation-based multi-agent system corresponding to an auction scenario is discussed in Section 4. The paper ends in Section 5 with conclusions and ideas for future research.

## 2 Argumentation Basics and Related Work

This section is focused on argumentation research that is relevant to this paper, but is also aimed at providing the reader with basic argumentation background. We start with the definition of argumentation frameworks, as introduced by Dung in [8], and the basic terminology used when talking about arguments.

**Definition 1.** *An **argumentation framework** is a pair $F = (\mathcal{A}, \mathcal{R})$, where $\mathcal{A}$ is a set of arguments and $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is a binary attack relation on $\mathcal{A}$. We say that an argument $a$ **attacks** another argument $b$ and we write this as $a \rightarrow b$ iff $(a, b) \in \mathcal{R}$. Otherwise, $a$ does not attack $b$ and we write $a \not\rightarrow b$. Also, we say that a set of arguments $S$ attacks an argument $a$ iff $S$ contains an attacker of $a$. A set of arguments $S$ **defends** an argument $a$ iff $S$ attacks all the attackers of $a$. The **characteristic function** $\mathcal{F}_F$ returns, for every set of arguments $S$, the set of arguments defended by $S$ in $F$.*

We have split the presentation of argumentation research related to our work into several subsections with respect to different facets of our approach.

### 2.1 Extension-Based Semantics

Given a set of arguments and the attack relation between them, one must be able to identify the arguments that are acceptable. Several semantics were defined in the literature for finding the extensions of an argumentation framework, namely the sets of arguments that satisfy certain properties. Definition 2 lists the semantics introduced by Dung himself in [8].

**Definition 2.** *Let $F = (\mathcal{A}, \mathcal{R})$ be an argumentation framework and let $S$ be a set of arguments.*

- *$S$ is **conflict-free** ($\mathcal{CF}$) iff $S$ does not attack any of its arguments.*
- *$S$ is **admissible** ($\mathcal{AS}$) iff $S$ is conflict-free and $S$ defends all its arguments.*
- *$S$ is a **complete extension** ($\mathcal{CO}$) iff $S$ is admissible and it contains all the arguments it defends.*
- *$S$ is a **stable extension** ($\mathcal{ST}$) iff $S$ is conflict-free and it attacks all the arguments it does not contain.*
- *$S$ is a **preferred extension** ($\mathcal{PR}$) iff $S$ is a maximal (with respect to set inclusion) admissible set.*
- *$S$ is the **grounded extension** ($\mathcal{GR}$) of $F$ iff $S$ is the least fixed point of the characteristic function.*

*For an argumentation semantics $\mathcal{S}em$ we will use $\mathcal{E}_{\mathcal{S}em}$ to denote the set of all extensions prescribed by it, for example $\mathcal{E}_{\mathcal{CO}}(F)$ stands for all the complete extensions of $F$.*

Of the six types of sets introduced in Definition 2, only the last four correspond to actual argumentation semantics, whereas the first two describe properties satisfied by almost all semantics defined in the literature. The four semantics are not independent: stable extensions are also preferred, preferred extensions are also complete and the grounded extension is the minimal (with respect to set inclusion) complete extension [8].

Several additional semantics were defined in the literature, such as: semi-stable [3], ideal [9], eager [4], prudent [6], $CF2$ [2], resolution-based grounded [1], enhanced preferred [11]. We will provide more details about the last two, as this work combines ideas from both of them.

For the resolution-based grounded semantics we introduce the corresponding terminology in Definition 3.

**Definition 3.** *Two arguments $a$ and $b$ are* ***conflicting*** *iff $a \rightarrow b$ or $b \rightarrow a$. For an argumentation framework $F$, the set of all conflicting pairs of arguments is denoted by $\mathcal{CONF}(F)$. Given two argumentation frameworks $F_1 = (\mathcal{A}_1, \mathcal{R}_1)$ and $F_2 = (\mathcal{A}_2, \mathcal{R}_2)$, we say that $F_1$ is* ***more skeptical*** *than $F_2$ and we write $F_1 \preceq F_2$ iff $\mathcal{CONF}(F_1) = \mathcal{CONF}(F_2)$ and $\mathcal{R}_2 \subseteq \mathcal{R}_1$. Two frameworks $F_1$ and $F_2$ are* ***comparable*** *(with respect to skepticism) iff $F_1 \preceq F_2$ or $F_2 \preceq F_1$. The set of maximal (with respect to $\preceq$) frameworks comparable with a given framework $F$ is denoted with $\mathcal{RES}(F)$. The* ***resolution-based version*** *of a given argumentation semantics $\mathcal{S}em$ is defined as $\mathcal{E}_{\mathcal{S}em^*}(F) = \mathcal{MIN}(\bigcup_{F' \in \mathcal{RES}(F)} \mathcal{E}_{\mathcal{S}em}(F'))$, where $\mathcal{MIN}(X)$ denotes the minimal (with respect to set inclusion) elements of $X$.*

In other words, it is easy to see that an argumentation framework $F_1$ is less skeptical than a framework $F_2$ iff some of $F_2$'s mutual attacks are replaced with unidirectional attacks in $F_1$. With the terminology in [10], $F_1$ is a partial resolution of $F_2$. In a complete resolution, all mutual attacks are replaced with unidirectional ones. Thus, computing the resolution-based version of some argumentation semantics $\mathcal{S}em$ consists of taking all the complete resolutions of the argumentation framework, applying $\mathcal{S}em$ to each of them, then choosing the minimal (with respect to set inclusion) of all the resulting extensions.

In the partial resolution of an argumentation framework, some of the mutual attacks are converted to unidirectional ones, which is the same as discarding some attacks. In our approach we apply the same idea, but to arguments instead of attacks.

We now turn to the work of Zhang and Lin on enhanced preferred extensions [11]. We summarize their work in Definition 4, but using defense instead of acceptability with respect to a set.

**Definition 4.** *A pair of sets of arguments $(S, H)$ defends an argument $a$ iff $a \notin H$, $H \cap S = \varnothing$ and $S$ defends $a$ against all attacks that do not come from $H$. Given a framework $F = (\mathcal{A}, \mathcal{R})$ a conflict-free set of arguments $S$ and a set of arguments $H$, we say that $(S, H)$ is an* ***admissible pair*** *iff (1) $S \neq \varnothing$ or $H = \mathcal{A}$, and (2) $(S, H)$ defends all arguments in $S$. A pair $(S, H)$ is a* ***minimal admissible pair*** *if it is an admissible pair and its second element $H$ is minimal (with respect to cardinality) among all admissible pairs. A pair $(S, H)$ is an*

***enhanced preferred extension*** *iff its first element $S$ is maximal (with respect to set inclusion) among all minimal admissible pairs. The first element of an enhanced preferred extension is called* ***proper enhanced preferred extension.***

In other words, for computing the enhanced preferred extension, admissible extensions are computed with respect to subframeworks of $F$, then the maximal extensions are picked among those of all subframeworks of maximal cardinality.

As we have already mentioned while discussing resolution-based semantics, our approach relies on ignoring arguments as well and computing a given semantics on subframeworks. However, the sets of ignored arguments are minimized with respect to set inclusion instead of cardinality, additional constraints are imposed on the pair of sets and the extensions are not maximized, nor is the second element dropped. Instead we work with several sets (we actually use labelings).

### 2.2 Argument Labelings

In this section we will focus on argument labelings, as proposed in [3]. We do this because our proposal is most intuitively expressed in terms of labelings. At the same time, using labels allows us to compare our approach with existing semantics.

**Definition 5.** *Let $F = (\mathcal{A}, \mathcal{R})$ be an argumentation framework. A* ***labeling*** *is a total function $\mathcal{L} : \mathcal{A} \rightarrow \{\mathtt{in}, \mathtt{out}, \mathtt{undec}\}$. A labeling $\mathcal{L}$ is complete iff (1) an argument is labeled* `in` *iff all its attackers are labeled* `out`*, and (2) an argument is labeled* `out` *iff it has an attacker that is labeled* `in`*.*

Alternatively, a labeling $\mathcal{L}$ can be seen as a partition of the set of arguments into three sets $(\mathtt{in}(\mathcal{L}), \mathtt{out}(\mathcal{L}), \mathtt{undec}(\mathcal{L}))$. It is shown in [3] that any complete labeling $\mathcal{L}$ is uniquely defined by either $\mathtt{in}(\mathcal{L})$ or $\mathtt{out}(\mathcal{L})$. For the grounded labeling, $\mathtt{in}(\mathcal{L})$ and $\mathtt{out}(\mathcal{L})$ are minimal, whereas $\mathtt{undec}(\mathcal{L})$ is maximal. Preferred labelings have maximal $\mathtt{in}(\mathcal{L})$ and $\mathtt{out}(\mathcal{L})$, whereas the semi-stable extensions correspond to labelings that have a minimal $\mathtt{undec}(\mathcal{L})$.

We will provide a labeling-based definition for our approach and then compare it with complete labelings in Section 3.

### 2.3 Constrained Argumentation Frameworks

Our proposal is also related to constrained argumentation frameworks [7], but we prefer to relate the constraints to the labelings rather than to the framework itself, thus obtaining parameterized semantics.

The basic idea, roughly speaking, for constrained argumentation frameworks is that the extensions are computed for regular semantics, then only the extensions satisfying the constraints are kept. We do something similar, but with labelings and we apply constraints on a general enough set of labelings so as to be able to satisfy any reasonable constraint.

## 3 Constrained Strict Semantics

In this section we introduce the constrained strict semantics, describing it in terms of labelings. In order to do this, we first enrich the usual set of labels $\{\mathtt{in}, \mathtt{out}, \mathtt{undec}\}$ with an additional label, `ign`, which stands for arguments that are ignored.

The idea behind ignoring arguments is that a rational agent may choose to doubt some of the information it has in order to be able to take a decision or enforce one of its goals in the extensions of the corresponding argumentation framework.

**Definition 6.** *Let $F = (\mathcal{A}, \mathcal{R})$ be an argumentation framework. An* ***open labeling*** *is a mapping $\mathcal{L} : \mathcal{A} \to \{\mathtt{in}, \mathtt{out}, \mathtt{undec}, \mathtt{ign}\}$ such that:*

- *an argument is labeled* `in` *iff all its attackers are labeled either* `out` *or* `ign`
- *an argument is labeled* `out` *iff it has an attacker labeled* `in`

As one can see from Definition 6, the open labelings can be seen as complete labelings for subframeworks that ignore some arguments from the original framework. Note that arguments are only provisionally ignored, while computing extensions and searching for those satisfying certain constraints (as we shall see further on). In the end, however, the ignored arguments must be defeated (with new arguments, for example) so that the corresponding open labeling becomes a complete labeling of the framework. More about this aspect in Section 4.

We regard the ability to ignore arguments as a tool for being more decided about the status of the arguments that are not ignored. This leads to the definition of decided open labelings.

**Definition 7.** *An open labeling is said to be* ***decided*** *iff it has no* `undec`*-labeled argument.*

Let us first see that such labelings exist for any argumentation framework. Just as in [5], we will use $\mathtt{in}(\mathcal{L})$ to stand for the set of `in`-labeled arguments for an open labeling $\mathcal{L}$ and so on for the other labels. A complete labeling $\mathcal{L}_c$ can thus be viewed as a partition of the set of arguments into three sets $(\mathtt{in}(\mathcal{L}_c), \mathtt{out}(\mathcal{L}_c), \mathtt{undec}(\mathcal{L}_c))$, whereas an open labeling $\mathcal{L}$ corresponds to a partition into four sets $(\mathtt{in}(\mathcal{L}), \mathtt{out}(\mathcal{L}), \mathtt{undec}(\mathcal{L}), \mathtt{ign}(\mathcal{L}))$.

**Proposition 1.** *For any argumentation framework $F = (\mathcal{A}, \mathcal{R})$, the following open labelings are decided:*

*(i)* $\mathcal{L} = (S, \varnothing, \varnothing, \mathcal{A} \setminus \mathcal{S})$*, where $S$ is any conflict-free set of $F$.*
*(ii)* $\mathcal{L} = (\mathtt{in}(\mathcal{L}_c), \mathtt{out}(\mathcal{L}_c), \varnothing, \mathtt{undec}(\mathcal{L}_c))$*, where $\mathcal{L}_c$ is any complete labeling of $F$.*

Note that applying the condition from Definition 7 to complete labelings leads to stable labelings, which may not exist for certain argumentation frameworks. So ignoring some arguments does indeed enable us to enforce stronger restrictions on the arguments that are not ignored.

Since a decided open labeling only uses three labels (in , out and ign), similarly to the complete labelings, it is natural to ask ourselves whether our approach does indeed bring something new. It may seem that decided open labelings are only able to ignore parts of a complete labeling, thus only reducing the in and out parts. We show that this is not the case.

*Example 1.* Consider the argumentation framework $F = (\mathcal{A}, \mathcal{R})$, with $\mathcal{A} = \{a, b, c\}$ and $\mathcal{R} = \{(a, b), (b, c), (c, a)\}$. Its only complete labeling is $\mathcal{L}_c = (\varnothing, \varnothing, \{a, b, c\})$. On the other hand, the framework has 7 decided labelings: $\mathcal{L}_1 = (\{a\}, \{b\}, \varnothing, \{c\})$, $\mathcal{L}_2 = (\{b\}, \{c\}, \varnothing, \{a\})$, $\mathcal{L}_3 = (\{c\}, \{a\}, \varnothing, \{b\})$, $\mathcal{L}_4 = (\{a\}, \varnothing, \varnothing, \{b, c\})$, $\mathcal{L}_5 = (\{b\}, \varnothing, \varnothing, \{a, c\})$, $\mathcal{L}_6 = (\{c\}, \varnothing, \varnothing, \{a, b\})$, $\mathcal{L}_7 = (\varnothing, \varnothing, \varnothing, \{a, b, c\})$. All labelings except the trivial $\mathcal{L}_7$ are able to accept one argument, whereas the complete labeling was undecided.

It is known that complete labelings are uniquely identified by either their in or out parts. For open labelings this is generally not the case, as for each ign part there are several complete labelings for the resulting subframework. This means that the open labelings are uniquely identified by two of their sets, either ign and in or ign and out. These observations hold even if we focus on decided labelings only. Indeed, consider the following example:

*Example 2.* Let $F = (\mathcal{A}, \mathcal{R})$, with $\mathcal{A} = \{a, b, c, d\}$ and $\mathcal{R} = \{(a, b), (b, c), (c, d), (d, a)\}$. We consider the following decided labelings: $\mathcal{L}_1 = (\{a, c\}, \{b, d\}, \varnothing, \varnothing)$, $\mathcal{L}_2 = (\{a, c\}, \{b\}, \varnothing, \{d\})$, $\mathcal{L}_3 = (\{a\}, \{b\}, \varnothing, \{c, d\})$ and $\mathcal{L}_4 = (\{b, d\}, \{a, c\}, \varnothing, \varnothing)$. Note that we have $\texttt{in}(\mathcal{L}_1) = \texttt{in}(\mathcal{L}_2)$, $\texttt{out}(\mathcal{L}_2) = \texttt{out}(\mathcal{L}_3)$ and $\texttt{ign}(\mathcal{L}_1) = \texttt{ign}(\mathcal{L}_4)$. Thus, none of the labels can uniquely identify decided labelings on its own (the undec label is not part of this discussion, as $\texttt{undec}(\mathcal{L}) = \varnothing$ for any decided labeling $\mathcal{L}$). On the other hand, let us see that $\mathcal{L}_2$ and $\mathcal{L}_3$ are uniquely determined by their ign parts. We consider this property useful because, given a set of such labelings, choosing the desired one only depends on choosing the arguments to ignore (and later defeat).

**Definition 8.** *An open labeling of an argumentation framework $F$ is* ***unique*** *if no other open labeling of $F$ has the same set of* ign*-labeled arguments.*

Such labelings exist for all argumentation frameworks, as the labelings from Proposition 1 (i) are also unique. In the general case, however, not all unique labelings are also decided. Indeed, consider the framework from Example 1 and notice that $\mathcal{L} = (\varnothing, \varnothing, \{a, b, c\}, \varnothing)$ is a unique but not decided open labeling of $F$.

In fact, the decided labelings correspond to stable labelings of subframeworks, whereas the unique labelings correspond to grounded labelings of subframeworks that have a single complete labeling.

**Definition 9.** *An open labeling is said to be* ***strict*** *iff it is both decided and unique.*

Again, we rely on Proposition 1 (i) to see that strict labelings exist for any argumentation framework.

We are now ready to add constraints to our labelings. The approach is similar to that used for constrained argumentation frameworks in [7] We will use $PL_S$ to denote the propositional language defined in the usual inductive way from the set of propositional symbols $S$ and the logical connectives $\top, \bot, \neg, \wedge, \vee$.

**Definition 10.** *Let $F = (\mathcal{A}, \mathcal{R})$ be an argumentation framework, $\mathcal{L}$ one of its open labelings and $\varphi \in PL_{\mathcal{A}}$. We say that $\mathcal{L}$* ***satisfies*** *$\varphi$ and write $\mathcal{L} \models \varphi$, where satisfiability is recursively defined for each formula as follows:*

- $\mathcal{L} \models \top$
- $\mathcal{L} \not\models \bot$
- $\mathcal{L} \models a$ *iff* $a \in \texttt{in}(\mathcal{L})$, *for all* $a \in \mathcal{A}$
- $\mathcal{L} \models \neg a$ *iff* $a \in \texttt{out}(\mathcal{L})$ *or* $a \in \texttt{ign}(\mathcal{L})$, *for all* $a \in \mathcal{A}$
- $\mathcal{L} \models \phi \wedge \psi$ *iff* $\mathcal{L} \models \phi$ *and* $\mathcal{L} \models \psi$, *for all* $\phi \in PL_{\mathcal{A}}$ *and* $\psi \in PL_{\mathcal{A}}$
- $\mathcal{L} \models \phi \vee \psi$ *iff* $\mathcal{L} \models \phi$ *or* $\mathcal{L} \models \psi$, *for all* $\phi \in PL_{\mathcal{A}}$ *and* $\psi \in PL_{\mathcal{A}}$

Note that in Definition 10 there is no rule for arbitrary negations, but only for negated propositional symbols. This is because the negation of a propositional symbol does not only mean that the corresponding argument is not `in`, but also that it is not `undec`. We have chosen this approach because we consider that it makes little sense to actually want an argument to be undecided. Whenever writing constraints, we will make sure that all negations are applied to propositional symbols.

The interesting question is whether, given a formula $\varphi$, there is an open labeling that satisfies it. Clearly this is not possible for inconsistent formulas. Furthermore, consistency of a formula should also be related in some way to the attack relation, as it is clearly not possible to have both an argument and its attacker marked as `in`, for example.

**Definition 11.** *Let $F = (\mathcal{A}, \mathcal{R})$ be an argumentation framework and $\varphi \in PL_{\mathcal{A}}$ a satisfiable formula. We say that $\varphi$ is* ***consistent with*** *$F$ iff the set of formulas $\{\varphi\} \cup \{\neg a \vee \neg b \mid (a, b) \in \mathcal{R}\}$ is satisfiable.*

Satisfiability in Definition 11 refers to the usual satisfiability in propositional logic and is not connected to labelings. Whenever we talk about satisfiability with respect to labelings, we will explicitly say that the formula is satisfied by a labeling, to avoid any confusion.

**Proposition 2.** *A satisfiable formula $\varphi$ is consistent with an argumentation framework $F$ iff its disjunctive normal form contains at least one conjunction whose positive literals correspond to the elements of a conflict-free set of $F$.*

*Proof.* For the "$\Leftarrow$" part, suppose that the disjunctive normal form of $\varphi$ contains the conjunction $\psi = a_1 \wedge \ldots \wedge a_n \wedge \neg b_1 \wedge \ldots \wedge \neg b_k$ such that the set $S = \{a_1, \ldots, a_n\}$ is a conflict-free set of $F$. We assign $a_i = \top$ for all $i$'s and we assign $\bot$ to all the other arguments. This assignment is correct because the $a$'s and the $b$'s are distinct, as a result of the fact that $\varphi$ is satisfiable. Suppose that there is a formula $\neg a \vee \neg b$ that corresponds to an attack in $\mathcal{R}$ and is not satisfied. That would mean that both $a$ and $b$ are true and, thus, are elements of the conflict-free set $S$. But then they cannot attack one another, which contradicts our assumption. We can conclude that $\varphi$ is consistent with $F$.

For the "$\Rightarrow$" part, consider a truth assignment that satisfies $\varphi$. Then the disjunctive normal form of $\varphi$ contains at least one conjunction that is satisfied. The positive literals of that conjunction correspond to a conflict-free set, because otherwise there would be an attack whose corresponding formula is not satisfied, which would in turn violate the fact that $\varphi$ is consistent with $F$. This completes our proof. □

Note that, as a term in some conjunction of the disjunctive normal form of a formula, $\top$ corresponds to the empty set, which is a conflict-free set of every framework. This is in accordance with the expected fact that $\top$ is consistent with every framework.

We are now ready for the main theoretical result of this paper, namely the existence of strict labelings satisfying any reasonable constraint.

**Proposition 3.** *Any formula $\varphi$ that is consistent with an argumentation framework $F$ is satisfied by at least one strict labeling of $F$.*

*Proof.* From Proposition 2 we have that the disjunctive normal form of $\varphi$ contains a conjunction $\psi = a_1 \wedge \ldots \wedge a_n \wedge \neg b_1 \wedge \ldots \wedge \neg b_k$ such that the set $S = \{a_1, \ldots, a_n\}$ is a conflict-free set of $F$. We denote $B = \{b_1, \ldots, b_k\}$. Let $T = \{b \mid \exists a(a \in S \wedge (b, a) \in \mathcal{R})\}$, the set of arguments that attack elements of $S$. Let $\mathcal{L}_{gr}$ denote the grounded labeling of the restricted argumentation framework $F \downarrow_{\mathcal{A} \setminus (B \cup T)}$, where $F \downarrow_X = (X, \mathcal{R} \cap (X \times X))$. We consider the open labeling $\mathcal{L} = (\mathtt{in}(\mathcal{L}_{gr}), \mathtt{out}(\mathcal{L}_{gr}), \varnothing, \mathtt{undec}(\mathcal{L}_{gr}) \cup B \cup T)$.

First, let us see that $S \subseteq \mathtt{in}(\mathcal{L}_{gr})$. Indeed, since all attackers of arguments from $S$ were ignored, all elements of $S$ are unattacked in the restricted framework so they must be part of the grounded extension. Coupled with the fact that all arguments in $B$ are ignored, this leads to the fact that $\mathcal{L}$ satisfies $\psi$ and hence it satisfies $\varphi$ as well.

What is left to show is that $\mathcal{L}$ is indeed a strict labeling. Since the `in` and `out` parts come from $\mathcal{L}_{gr}$, the labeling satisfies the conditions for an open labeling. Also, the `in` and `out` arguments of the grounded labeling form a subframework that allows no other complete labeling, so $\mathcal{L}$ is unique. Since $\mathtt{undec}(\mathcal{L}) = \varnothing$, $\mathcal{L}$ is also decided and thus strict. □

The result of Proposition 3 is quite strong as it shows that, given any reasonable constraints, one can find strict labelings that satisfy them. This is the most important feature that distinguishes our work from the constrained argumentation frameworks in [7].

Since there may still be several labelings to choose from, we can refine the approach even more and finally define the constrained strict labelings.

**Definition 12.** *A* ***constrained strict labeling*** *of an argumentation framework $F$ with respect to a formula $\varphi$ that is consistent with $F$ is a strict open labeling*

*$\mathcal{L}$ that satisfies $\varphi$ and has* $\texttt{ign}(\mathcal{L})$ *minimal (with respect to set inclusion) among all strict open labelings that satisfy $\varphi$.*

The intuition behind Definition 12 is that one should decide the status of as many arguments as possible, while not violating the constraint $\varphi$ or the restrictions of strict labelings.

## 4 Constrained Argumentation Game

In this section we show that the constrained strict semantics is relevant for the multi-agent systems community by introducing an argumentation-based game featuring agents. We instantiate the approach by modeling a special auction scenario and we show how desired features of the scenario can be mapped into game elements.. At the end of the section we discuss the critical role that the constrained strict semantics plays for this game.

### 4.1 Auction Scenario

The example scenario that we will translate into a constrained argumentation game consists in a special kind of multiple items auction. To keep things simple, we will use an unspecified currency (just a positive integer). We will consider just three persons in our scenario: Anthony, Brian and Carol.

Anthony has two old pieces of furniture, a chair and a table, that he would like to sell. Since he has been a collector for quite some time now, he is rather good at appraising antiques so he knows that the chair values 200, while the table values 300. Anthony is determined to get at least these prices or keep the items. Anthony is familiar with most types of auctions, but he would like a bit more control over the outcome, so he organizes a special kind of auction.

The auction starts with Anthony announcing the rules, the items for sale and the minimum prices. To avoid any suspicions, the bidding process is public, spoken out loud. In a round-robin order, each participant can place, update or retract bids for any of the items that are put up for sale. Anthony can benefit from auctioning both items at the same time, as he is part of the auction himself and, on his turn, he reserves the right to impose additional restrictions or, on the contrary, relax some constraints. Each participant may also choose to pass, if satisfied with the current outcome (unless other restrictions are applicable, the highest bidder for each item wins that item). The auction ends when all participants pass.

We assume, for simplicity, that only two potential buyers show up for the auction: Brian and Carol. Brian is rather rich and knows very little about antiques, so he is willing to pay even twice their value: 400 for the chair and 600 for the table. However, he is determined to either get both items or none of them. That is why he likes Anthony's idea of an auction: he does not risk buying the first item only to find himself unable to acquire the second as well. Carol, on the other hand, has more limited resources so she cannot afford both items. However, she would really like acquiring one of them. She is willing to pay 300 for the chair and 500 for the table.

In the following subsections we will translate this auction scenario step by step into a turn-based game featuring agents that work with an abstract argumentation framework and constraints.

### 4.2 Arguments for the Environment

We start by discussing the environment of the multi-agent system and its representation using arguments. We will use a first order language containing predicates for various aspects of possible states of the environment and constants for the relevant objects. For our particular scenario, we have two items (*chair* and *table*) and three agents (*Anthony*, *Brian* and *Carol*).

The only relevant outcome of our auction is the final owner of each item. First, we consider the (rather naive) approach of assigning a first-order formula for each possible state: $s_1 = has(Anthony, chair) \wedge has(Anthony, table)$, $s_2 = has(Anthony, chair) \wedge has(Brian, table)$, and so on, for a total of $3^2 = 9$ states. We can read these states as arguments, deduce that no two of them can hold at the same time and decide to add attacks between all pairs of arguments. In order to ensure that any extension of the framework does select a state, we can add the constraint $\phi_{\mathcal{E}} = s_1 \vee \ldots \vee s_9$ and use strict constrained labelings. Since $\phi_{\mathcal{E}}$ can be satisfied, the corresponding strict constrained labelings ignore no argument so they are in effect complete labelings. Each of the 9 possible states forms a singleton complete extension. This representation, although semantically reasonable, is exponential in the number of items.

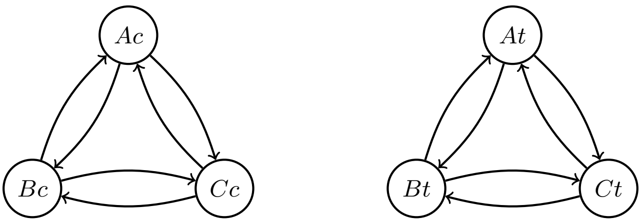


**Fig. 1.** State arguments for the auction example

We can come up, on the other hand, with an approach that is linear in the number of items. Indeed, it is rather easy to see that our states depend on two distinguishable parameters, the owners of each item. Thus, we can use just the following arguments: $Ac = has(Anthony, chair)$, $At = has(Anthony, table)$, $Bc = has(Brian, chair)$, $Bt = has(Brian, table)$, $Cc = has(Carol, chair)$ and $Ct = has(Carol, table)$. A suitable and intuitive framework for this case is the one in Figure 1, in conjunction with the following constraint: $\phi_{\mathcal{E}} = (Ac \vee Bc \vee Cc) \wedge (At \vee Bt \vee Ct)$. The constraint ensures that each item will have at least one owner, whereas the attacks between the states enforce at most one owner, thus leading to the desirable outcome that each item has exactly one owner. Let us note that the number of possible states is again given by the number of complete labelings, but this time each extension has exactly two elements.

### 4.3 Action and Reaction

We will use first order predicates to talk about actions as well. In fact, since our auction is more like a negotiation game, what we talk about are intentions of taking some action. We will see them as action arguments, to distinguish them from those describing the environment, which we shall call state arguments.

In our scenario, the only possible action consists in placing a bid for an item, for example $bids(table, 350)$. Such actions will be annotated with the name of the agent performing them, as in $Carol : bids(table, 350)$. The implicit attacks between such arguments come from the fact that a higher bid on the same item is preferred to a lower one.

Furthermore, actions may have an impact on the state of the environment. For example, the bid $Carol : bids(table, 350)$, if highest, should imply that Carol becomes the owner of the table. For this, we should have the following attacks: $Carol : bids(table, 350) \to At$ and $Carol : bids(table, 350) \to Bt$. We assume that such implicit attacks are common knowledge for all the participants at the auction.

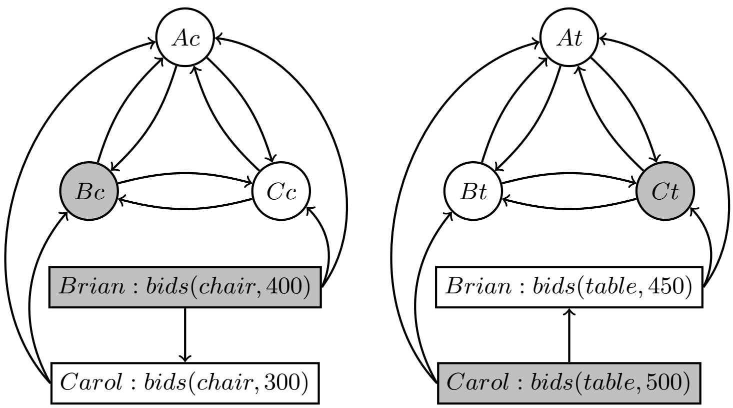


**Fig. 2.** Simple bid scenario on top of the initial environment representation. Grayed arguments form the only complete extension of the framework.

The framework in Figure 2 describes a possible moment from our auction. The grayed arguments form the only complete extension of the framework which, in addition, satisfies the environment constraint formula from the previous subsection. We will say that the extension is a valid outcome of the framework. Should this be the final state of the framework, it would mean that Brian is bound to pay 400 for the chair and Carol should pay 500 for the table. The resulting state after these actions would be the one also described by the extension, where the winning bidders actually get the items they pay for. But more about commitment and the outcome of a game later on.

### 4.4 Beliefs, Desires, Intentions

In this subsection we explore the role that beliefs, desires and intentions play in our model. While there is no one-to-one correspondence, we have several mechanisms that can help. First of all, let us see that the action arguments can be seen as intentions, or even plans, of the agents. They turn into actions only once an extension is chosen as the outcome of the framework.

Another mechanism consists in restricting the action arguments that are available to each agent. This can help us express the desires of the agent (the agent only considers the actions it would do) but it may also describe the abilities of the agent (some agents may have a smaller set of abilities with respect to others). In our case, we have that Brian's actions range from $bids(chair, 1)$ to $bids(chair, 400)$ and from $bids(table, 1)$ to $bids(table, 600)$, while Carol's actions range from $bids(chair, 1)$ to $bids(chair, 300)$ and from $bids(table, 1)$ to $bids(table, 500)$.

We also know that Carol can only afford one item. In order to say this, we will add attacks between Carol's bid for the chair and her bid for the table. Furthermore, Anthony wants some minimum price for each item. We can simulate this by having him bid as well, with the desired values. Thus, if he is the highest bidder for some item, that item will not be sold. The complete framework for this case is depicted in Figure 3.

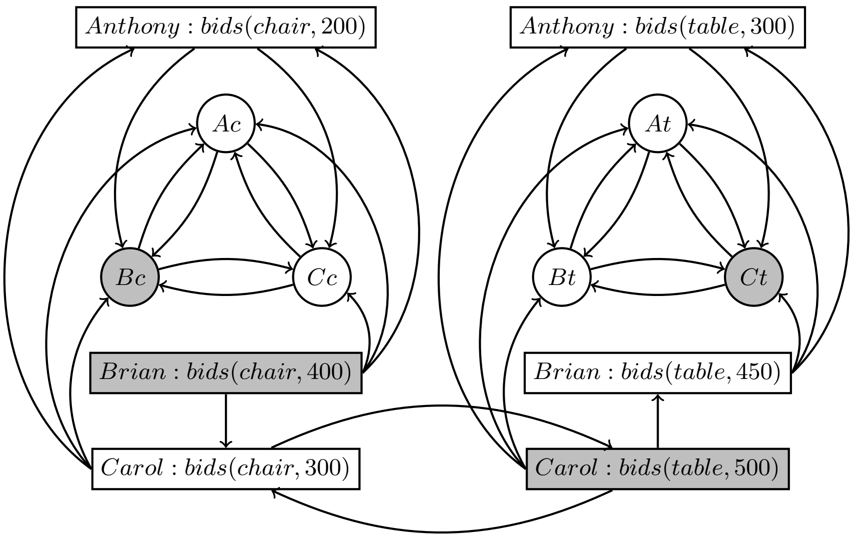


**Fig. 3.** Complete bid scenario extended from Figure 2. Grayed arguments form the only complete extension of the framework.

The goals of the agents can be expressed by means of constraints. Brian's desire to buy either both items or none can be expressed by $\phi_{Brian} = Bc \wedge Bt \vee \neg Bc \wedge \neg Bt$. Carol's wish to buy one of the items can be expressed by

$\phi_{Carol} = Cc \vee Ct$. Anthony's wish to sell both items can be written as $\phi_{Anthony} = \neg Ac \wedge \neg At$. We can see in Figure 3 that the only complete extension does satisfy the environment constraint, but does not satisfy Brian's.

Beliefs may also be encoded by means of additional state arguments that are not common knowledge, but are private to one or several agents. The agent may choose to disclose such information or not. For example, Carol may wish to debate the authenticity of the table. For this, she may provide the state arguments $auth(table)$ and $\neg auth(table)$, each attacking the other, then also add an action argument defeating the former: $Carol : newPaint(table) \rightarrow auth(table)$. It may also be implied that Anthony is no longer able to sell the table or that he must settle for a lower price. We will not detail this spin-off here, as it is not part of the initial scenario.

### 4.5 The Outcome of Changing the World

What we have seen so far is that the instantaneous state of the game can be described using an argumentation framework plus constraints. But we have said nothing about moving from one state to the other or about how the framework actually changes.

The auction will run as a turn-based game. During its turn, each agent can change the current argumentation framework so that it satisfies its goals (described with the corresponding constraint formula). However, not everything can be changed, as we shall see.

First of all, an agent is free to retract any of its own action arguments (intentions) or to add new ones to the framework Furthermore, each agent may have some influence on some of the state arguments. To maintain a unified approach, we shall assume that for each agent we can define the set of arguments it can control and that the set always includes its own actions. Now, given an argument it can control, an agent may add or remove attacks against it in order to express certain states of affairs. Let us explain this on our auction example.

All state arguments in our scenario are controlled by Anthony, since he is the one organizing the auction. This means that, if he so chooses, Anthony may change the rules of the game, for example by adding or removing attacks between state arguments. If Anthony prefers Carol to Brian, he may choose to favor her by removing the attacks $Bc \rightarrow Cc$ and $Bt \rightarrow Ct$. Thus, whenever both Carol and Brian might win, Carol will be preferred.

The bidding agents, each during its own turn, may change their bid on some of the items. They will generally do so by looking at the outcome of the current framework. From a strategical perspective, constrained strict labelings are fit for this task, because the agents may use their goals as constraints and find minimal sets of arguments that are to be attacked. From the arguments they have available, they may then choose which of them to put forward or in what way to change the framework.

The outcome of an instantaneous configuration, as well as the outcome of the game's final configuration, consists in the choice of a constrained extension of the framework, using an algorithm that is known to all participants of the game in advance, for example the constrained strict semantics. If all agents strive to maintain the environment constraint satisfiable at all times by a complete labeling, then at the end of the game this will also hold so none of the arguments will need to be ignored.

Let us consider the example in Figure 3 again and see how that configuration looks for each of the participants and what might be their next move, were it to be their turn. The only complete labeling is the one depicted in the figure and it also satisfies the environment constraint. Anthony gets to sell both items, so his goal is fulfilled. Thus, Anthony will pass. So will Carol, whose goal is to obtain one of the objects. However, this configuration is not good for Brian, who wishes either both objects or none.

What Brian must do in this case is recompute the open labelings using both the environment constraint and his own goal. In this case he will end up with sets of arguments to ignore. All such sets that contain arguments that Brian himself cannot control are to be discarded. He may then choose between remaining alternatives, if any.

In the particular case of Figure 3, Brian may notice that a simple solution to his problem consists in ignoring Carol's bid. Since actually ignoring it is not an option, Brian must defeat it. For this, it is enough to put forward a higher bid for the table.

### 4.6 End of the Game and Commitment

We have seen that the game proceeds in turn-based fashion, each agent changing the current configuration to better fulfill its goals. Whenever an agent is satisfied with the current configuration, or has no available action arguments to put forward for producing a favorable change, the agent will pass. The game ends after every agent has passed in a full round.

Once the game has reached its final configuration, all agents are committed to fulfill their intentions executing the corresponding actions. In doing so, the environment will also enter the state described by the chosen extension of the argumentation framework for the configuration.

An additional challenge for such a game is to arrange the initial configuration in such a way as to describe the actual initial state. In our auction scenario, instead of letting all states be acceptable, one might add Anthony's arguments so that the ownership of the items can only be his in acceptable extensions of the initial framework.

### 4.7 Discussion

We will end this section with a discussion of the relevance of constrained strict labelings for the presented argumentation-based multi-agent system. These extensions may play a part at the end of the game, where a suitable outcome is to be chosen. Since the environment constraint must be satisfied, in more liberal games, where it is not enforced at all points, it may be needed to ignore some of the arguments at the end of the game.

On the other hand, for the outcome algorithm it might be better to consider frameworks where there is a single extension satisfying the environment constraint. Of course, it may be declared in the beginning of such a game that some semantics that provides multiple extensions will be used and that the result will be undecided if several extensions are available. Thus, this may work as an incentive for participants that would not be satisfied with an undecided negotiation to work towards reaching a final state that has a single extension.

The most important part where our semantics can undoubtedly be put to good use is the agent's strategy for executing its move. At that point, the agent may use the same semantics as the one used for the game outcome, but update the constraint to include its own goals thus identifying arguments that are to be attacked or ignored. It is important that all the arguments found in the chosen ignored set can actually be controlled or attacked by the agent.

## 5 Conclusions

We have introduced a new argumentation semantics, based on labelings, which allows us to impose constraints on the labelings. We have proved that for any reasonable constraint we get at least one constrained strict labeling and we have shown how to use these labelings in an agent setting.

We have provided a detailed scenario that can be modeled as a multi-agent system based on argumentation where our semantics can play a significant part in determining the next move for each agent. The proposal is a significant contribution in itself, as it provides an alternative approach for using argumentation in multi-agent systems. Future work will compare our turn-based game with other approaches in the literature.

It would be interesting to analyze the proposed semantics with respect to its computational complexity, this being an important goal of future research, as the applicability of our approach in real multi-agent systems strongly depends on this.

**Acknowledgement.** This work has been funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labour, Family and Social Protection through the Financial Agreement POSDRU/88/1.5/S/61178 and by project ERRIC (Empowering Romanian Research on Intelligent Information Technologies), number 264207/FP7-REGPOT-2010-1. We would like to thank the reviewers for their useful comments that have helped us improve this paper.

## References

1. Baroni, P., Giacomin, M.: Resolution-based argumentation semantics. In: Proceedings of the 2nd Conference on Computational Models of Argument (COMMA 2008), Tolouse, France, pp. 25–36 (2008)
2. Baroni, P., Giacomin, M., Guida, G.: SCC-recursiveness: a general schema for argumentation semantics. Artificial Intelligence 168(1-2), 162–210 (2005)

3. Caminada, M.: Semi-stable semantics. In: Dunne, P.E., Bench-Capon, T.J.M. (eds.) Computational Models of Argument: Proceedings of COMMA 2006, September 11-12. Frontiers in Artificial Intelligence and Applications, vol. 144, pp. 121–130. IOS Press (2006)
4. Caminada, M.: Comparing two unique extension semantics for formal argumentation: Ideal and eager. In: Proceedings of the 19th Benelux Conference on Artificial Intelligence (BNAIC 2007), pp. 81–87 (2007)
5. Caminada, M., Gabbay, D.: A logical account of formal argumentation. Studia Logica 93(2), 109–145 (2009)
6. Coste-Marquis, S., Devred, C., Marquis, P.: Prudent semantics for argumentation frameworks. In: Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2005), Hong-Kong, China, pp. 568–572 (2005)
7. Coste-Marquis, S., Devred, C., Marquis, P.: Constrained argumentation frameworks. In: Proceedings of the 10th International Conference on Practices in Knowledge Representation and Reasoning, pp. 112–122 (2006)
8. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence 77(2), 321–357 (1995)
9. Dung, P.M., Mancarella, P., Toni, F.: Computing ideal sceptical argumentation. Artificial Intelligence 171(10-15), 642–674 (2007)
10. Modgil, S.: Hierarchical Argumentation. In: Fisher, M., van der Hoek, W., Konev, B., Lisitsa, A. (eds.) JELIA 2006. LNCS (LNAI), vol. 4160, pp. 319–332. Springer, Heidelberg (2006)
11. Zhang, Z., Lin, Z.: Enhancing Dung's Preferred Semantics. In: Link, S., Prade, H. (eds.) FoIKS 2010. LNCS, vol. 5956, pp. 58–75. Springer, Heidelberg (2010)