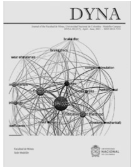
DYNA
http://dyna.medellin.unal.edu.co/



UNIVERSIDAD
NACIONAL
DE COLOMBIA

A rule-based argumentation framework for distributed contextual reasoning in dynamic environments

Helio Henrique Lopes Costa Monte-Alto [ab], Ayslan Trevizan Possebom [c], Miriam Mariela Mercedes Morveli-Espinoza [a] & Cesar Augusto Tacla [a]

[a] *Program in Electrical and Computer Engineering, Federal University of Technology - Paraná, Curitiba, Brazil. heliohenrique3@gmail.com, morveli.espinoza@gmail.com, tacla@utfpr.edu.br*
[b] *Federal University of Paraná, Brazil. heliohenrique@ufpr.br*
[c] *Federal Institute of Paraná, Campus Paranavaí, Paranavaí, Brazil. possebom@gmail.com*

Received: October 10th, 2020. Received in revised form: February 17th, 2021. Accepted: March 4th, 2021.

Abstract
In this study, we tackled the problem of distributed reasoning in environments in which agents may have incomplete and inconsistent knowledge. Conflicts between agents are resolved through defeasible argumentation-based semantics with a preference function. Support for dynamic environments, where agents constantly enter and leave the system, was achieved by means of rules whose premises can be held by arbitrary agents. Moreover, we presented a formalism that enables agents to share information about their current situation or focus when issuing queries to other agents. This is necessary in environments where agents have a partial view of the world and must be able to cooperate with one another to reach conclusions. Hence, we presented the formalization of a multi-agent system and the argument construction and semantics that define its reasoning approach. Using example scenarios, we demonstrated that our system enables the modeling of a broader range of scenarios than related work.

Keywords: multi-agent systems; distributed reasoning; formal argumentation; structured argumentation framework.

Un Framework de argumentación basado en reglas para razonamiento contextual distribuido en ambientes dinámicos

Resumen
En este estudio, abordamos el problema del razonamiento distribuido en entornos en los que los agentes pueden tener conocimiento incompleto e inconsistente. Los conflictos entre agentes se resuelven mediante una semántica basada en argumentación derrotable con una función de preferencia. El soporte a entornos dinámicos, donde los agentes entran y salen constantemente del sistema, se logra mediante reglas cuyas premisas pueden ser afirmadas por agentes arbitrarios. Además, presentamos un formalismo que permite a los agentes compartir información sobre su situación actual - o foco - al realizar consultas a otros agentes. Esto es necesario en entornos donde los agentes tienen una visión parcial del mundo y deben poder cooperar entre sí para llegar a conclusiones. Por tanto, presentamos la formalización de un sistema multiagente y la construcción de argumentos y la semántica que definen su abordaje de razonamiento. Mediante escenarios de ejemplo, demostramos que nuestro sistema permite modelar una gama de escenarios más amplia que los trabajos anteriores.

Palabras clave: sistemas multiagente; razonamiento distribuido; argumentación formal; *framework* de argumentación estructurado.

1. Introduction

Distributed reasoning is a central aspect required by many applications proposed in the last decades, including mobile and ubiquitous systems, smart spaces, ambient intelligence systems [1] and the semantic web [2]. In these types of systems, distributed entities are able to capture or receive information from their environment and are often able to derive new knowledge or make decisions based on the available knowledge. Given their ability to sense the

How to cite: Monte-Alto, H.H.L.C., Possebom, A.T., Morveli-Espinoza, M.M. and Tacla, C.A., A rule-based argumentation framework for distributed contextual reasoning in dynamic environments.. DYNA, 88(217), pp. 120-130, April - June, 2021.

Revista DYNA, 88(217), pp. 120-130, April - June, 2021, ISSN 0012-7353
DOI: https://doi.org/10.15446/dyna.v88n217.90858

environment, reason, act upon, and communicate with other entities, they are generically referred to as agents. Therefore, we consider distributed reasoning as the reasoning of agents in a multi-agent system (MAS) situated in an environment, whether physical or virtual.

We consider such an MAS in environments in which agents may have incomplete and inconsistent knowledge. *Incomplete knowledge* is the possibility that agents do not have a complete knowledge about the environment, thereby requiring knowledge imported from other agents to reach conclusions. For example, assume a robot agent a that can feel the wind but is blind, and has a knowledge expressed by the following rule: "I conclude that *it is going to rain* (conclusion) if *it is windy* (premise 1) and *there are clouds in the sky* (premise 2)". Agent a cannot reach the conclusion that is raining by itself because it cannot find the truth value of premise 2 by itself. However, suppose that agent b is known to agent a. Then, a can ask b whether there are clouds in the sky. If b answers positively, then a can conclude that it is going to rain. *Inconsistent knowledge* is the possibility of agents having an inaccurate or erroneous knowledge about the environment owing to imperfect sensing capabilities or badly intentioned or unreliable agents. For example, suppose that agent c has some defects in its visual sensor. If a asks c whether there are clouds in the sky, it is possible that c cannot see the sky perfectly, thus answering erroneously.

For such scenarios, the concept of *contextual reasoning* was proposed in the literature. Benerecetti et al. [3] defined contextual reasoning as that which operates on three fundamental dimensions along which a context-dependent knowledge representation may vary. The fundamental dimensions are *partiality* (the portion of the world that is considered), *approximation* (the level of detail at which a portion of the world is represented), and *perspective* (the point of view from which the world is observed). Contextual defeasible logic (CDL), introduced by Bikakis and Antoniou [1], was proposed as an argumentation-based contextual reasoning model based on the multi-context systems (MCS) paradigm, in which the local knowledge of agents is encoded in rule theories and information flow between agents is achieved through mapping rules that associate different knowledge bases. To resolve potential conflicts that may arise from the interaction of agents, a preference order for known agents combined with defeasible argumentation semantics was proposed.

However, the *approximation* dimension was not well studied in [1]. The idea of approximation is related to the idea of *focus*, in which the closer one is to an object of interest, the more it is aware of knowledge that is relevant to that object. For example, suppose agent d, which is located in another city distant to a, looks at the sky, sees the clouds, and is also able to feel the wind, but it does not know if this means that it is going to rain (it does not have the same rule as a). It then asks the other agents it knows; meanwhile, the only agent it can talk to at the moment is a. However, a, living in another city, is totally unaware of the weather conditions of d's location. In this case, it is necessary for d to send some knowledge to other agents to help them to reason properly. Thus, d sends two rules without premises (i.e., facts), stating that there are clouds in the sky and that it is windy. When a receives the query, it concludes that it is going to rain, and sends this answer to d.

In this study, we call *focus* the relevant knowledge at a given moment for a particular reasoning context. All the agents cooperating in the reasoning must be aware of this focus to reach reasonable conclusions in a fully decentralized manner. We are also concerned with the way agents can handle scenarios with multiple *foci* from different interactions.

Furthermore, this work allows mapping rules that reference the knowledge held by any known agent instead of only enabling references to specific agents. This enables the framework to be used in dynamic and open environments, in which the agents and their internal knowledge are not known *a priori* and where agents come in and out of the system at any time. We also demonstrate that the proposed approach can model a greater variety of scenarios than related work.

In this paper, we present a structured argumentation framework that includes argumentation semantics, which enables distributed contextual reasoning in dynamic environments, including the possibility of agents sharing focus knowledge when cooperating. This is an extension of a previous paper [4], furthering and developing the formalization details and adding new important features, namely: (1) a partial order relation between the agents using a preference function, which enables us to deal with more dynamic scenarios where not all agents are known by each agent and where agents can have equivalent trust in one another; (2) a similarity-based matching of schematic terms to concrete terms, which improves the range of supported scenarios involving approximate reasoning; and (3) an argument strength calculation that considers a chain of arguments to reach a conclusion when comparing arguments, and takes more advantage of the argument-based reasoning to reach more social-based conclusions. We also demonstrate the applicability of our approach using a more detailed example scenario that incrementally illustrates its modeling and behavior.

The paper is divided into five sections. In Section 2, we present a background on CDL. In Section 3, we present our contributions in the representational formalization of the framework, as well as the problem we are solving. In Section 4, we present a scenario that illustrates this problem. In Section 5, we present both argument construction and semantics, along with running examples. In Section 6, we present related studies. Finally, we discuss the conclusions and future work.

2. Preliminaries

In this section, we present some basic definitions and a fundamental background on CDL with slightly different nomenclature to facilitate the introduction of the contribution we make in this paper.

Definition 1: *A contextual defeasible logic system is composed of a set of* ***agents****. Each agent a_i is defined as a tuple of the form (KB_i, P_i) such that*

- *KB_i is the knowledge base (or belief base) of the agent;*
- *P_i is a preference relation as a total order relation over the other agents in the system which has the form $P_i =$*

$[a_2, a_j, a_3, \dots, a_1]$, *indicating that, for* a_i, a_2 *is preferable to* a_j, a_j *to* a_3, *and so on, and* a_1*is the least preferred agent.*

In this framework, knowledge is represented by sets of rules, referred to as knowledge bases, which comprise three different types of rules: *local strict rules*, *local defeasible rules,* and *mapping defeasible rules*. The rules are composed of terms of the form (a_i, x), where a_i is an agent called the *definer* of the term and x is a literal. A literal represents atomic information (x) or the negation of atomic information $(\neg x)$. For a literal x, its complementary literal is a literal corresponding to the strong negation of x, denoted as $\sim x$. More precisely, $\sim x \equiv \neg x$ and $\sim(\neg x) \equiv x$. For any term $p = (a_i, x)$, the complementary term is denoted as $\sim p = (a_i, \sim x)$.

The preference order P_i is useful for resolving conflicts that may arise from interaction among agents because we assume that the global knowledge base (the union of the knowledge bases of all the agents in the system) can be inconsistent.

Definitions 2-4 define the types of rules.

Definition 2: *A* ***local strict rule*** *is a rule of the form*

$$r: (a_i, x_n) \leftarrow (a_i, x_1), (a_i, x_2), \dots, (a_i, x_{n-1}) \qquad (1)$$

A strict rule with empty body denotes non-defeasible factual knowledge.

Note that the rule is local because the definer of each term in the body (premises) is the agent that owns the rule (a_i), as well as the term in the head (conclusion) of the rule. Local strict rules are interpreted by classical logic: whenever all the terms in the body of the rule are logical consequences of the set of local strict rules, then the term in the head of the rule is also a logical consequence.

Definition 3: *A* ***local defeasible rule*** *is a rule of the form*

$$r: (a_i, x_n) \Leftarrow (a_i, x_1), (a_i, x_2), \dots, (a_i, x_{n-1}) \qquad (2)$$

A defeasible rule with empty body denotes local factual knowledge that can be defeated.

A defeasible rule is similar to a strict rule in the sense that all terms in the body are defined by the same agent that defines the head. However, a defeasible rule cannot be applied to support its conclusion if there is adequate (not inferior) contrary evidence.

Definition 4: *A* ***mapping defeasible rule*** *is a rule of the form*

$$r: (a_i, x_n) \Leftarrow (a_i, x_1), (a_j, x_2), \dots, (a_l, x_{n-1}) \qquad (3)$$

A mapping defeasible rule has at least one term in its body that is a foreign term, which is a term whose definer is an agent different from the agent that defines the head of the rule - e.g. (a_j, x_2), *which is a term defined by* a_j.

A mapping rule denoted by $r: (a_i, x_n) \Leftarrow (a_j, x_1)$ has the following intuition: "If a_i knows that agent a_j concludes x_1, then a_i considers it a valid premise to conclude x_n if there is no adequate contrary evidence."

For convenience, given any rule r, $Head(r)$ denotes the head (conclusion) of the rule, which is a term, and $Body(r)$ denotes the body (premises) of the rule, which is a set of terms.

Based on the definitions of such agents with their respective rules, CDL proposes an argumentation framework that extends the argumentation semantics of defeasible logic presented in [5], which in turn is based on the grounded semantics of Dung's abstract argumentation framework [6], with arguments constructed in a distributed manner. The framework uses arguments of local range, that is, they are built based on the rules of a single agent only, which are then interrelated by means of an interdependent set of arguments constructed with respect to mapping rules. Further details concerning the argumentation semantics are presented in Section 5.

3. Architecture and problem formalization

In this section, we present some contributions to CDL's formalization with new elements to enable its use in a dynamic environment where agents are not known *a priori* and to enable contextual reasoning considering *focus*. The first difference lies in the definition of an agent.

Definition 5: *An* ***agent*** a_i *is defined as a tuple of the form* (KB_i, A_{K_i}, P_i) *such that*

- KB_i *is the knowledge base (or belief base) of the agent;*
- $A_{K_i} = \{a_1, a_2, \dots, a_j, \dots, a_k\}$ *is a set of agents known to* a_i*; and*
- $P_i: A_{K_i} \cup \{a_i\} \rightarrow [0,1]$ *is a preference function such that, given two agents* a_j *and* a_k *in* A_{K_i}, *if* $P_i(a_j) > P_i(a_k)$, a_j *is preferred to* a_k *by* a_i. *If* $P_i(a_j) = P_i(a_k)$, *then they are equally preferred by* a_i. *By default,* $P_i(a_i) = 1$.

The differences between Definition 1 and Definition 5 are the set of known agents and the preference relation, now represented as a function over the known agents, to support scenarios in which not every agent in the system is known by a given agent as well as when an agent has equivalent trust in two or more agents. Such a preference function, combined with the argument strength calculation and the strength comparison function (Definitions 17-22), enables a partial order relation over the known agents of an agent. The preference function can be derived from some degree of trust an agent has in another agent, which can result from various mechanisms, such as those presented in works on trust theory [7] and dynamic preferences [8].

To enable contextual defeasible reasoning in dynamic and open environments where agents are not known *a priori*, a new type of rule is necessary, which we call *schematic mapping defeasible rule* (called *schematic rule* in the rest of the paper for simplicity), as defined in Definition 6. These rules enable agents to import knowledge from arbitrary known agents without knowing *a priori* whether such agents have the required knowledge or not. This is a key feature that enables distributed reasoning in a dynamic and open environment where agents' knowledge about other agents' knowledge may be incomplete or non-existent. For example, suppose an agent a_1 with a rule whose premise x is not a logical consequence of a_1's local knowledge, and suppose that a few moments earlier, an agent a_2, which knows that x holds in the environment, entered the system. In this case, without schematic rules, a_1 would not ask a_2 about x.

The body of the schematic rules can be composed of a different type of term called *schematic term*, which has the form (X, x), where X is a variable that can be bound to any agent in the system and x is a literal.

We also consider the possibility of schematic terms being bound (or instantiated) to any term that is *sufficiently similar* to it. This means that terms whose literals have some similarity between them can be considered when matching schematic terms with *concrete terms* (not schematic). This enables us to support an even broader range of scenarios, especially those involving approximate reasoning. For example, suppose a scenario in which knowledge is represented through OWL (Ontology Web Language) ontologies [9]. In this case, a similarity function could be based on axioms that use the `owl:sameAs` property. Other examples include lexical similarity, such as WordNet [10], which could be used in scenarios involving speech or text recognition, and syntactic and semantic agent matchmaking processes, such as Larks [11].

Definition 6: *A* ***schematic rule*** *is a rule of the form*

$$r: (a_i, x_n) \Leftarrow (a_i, x_1), (X, x_2), \dots, (a_l, x_{n-1}) \qquad (4)$$

A schematic rule has at least one schematic term in its body.

A schematic rule whose body consists of a single schematic term, denoted $r: (a_i, x_n) \Leftarrow (X, x_1)$, has the following intuition: "If a_i knows that some agent concludes a literal sufficiently similar to x_1, then a_i considers it a valid premise to conclude x_n if there is no adequate contrary evidence."

The following presents concepts related to the similarity between terms and the concept of instantiated terms.

Definition 7: *A* ***similarity function*** *between terms is a function* $\Theta: T, T \rightarrow [0,1]$*, where* T *is the set of all possible terms, such that a greater similarity between the two terms* p *and* q *results in a greater* $\Theta(p, q)$ *value. A* ***similarity threshold*** $st \in [0,1]$ *is the minimum value required for the similarity between the two terms to be accepted. Two terms* p *and* q *are* ***sufficiently similar*** *if* $\Theta(p, q) \geq st$*. If* $p = (X, x)$ *is a schematic term,* $q = (a_i, y)$ *is a concrete term, and* p *and* q *are sufficiently similar, then there exists an* ***instantiated term*** *of the form* $p_{inst} = (a_i, y, \theta)$*, where* $\theta = \Theta(p, q)$ *is called the* ***similarity degree*** *between both terms.*

An agent a_i (which we call the *emitter agent*) is able to issue queries about the truth of a term to itself or to other agents. When an agent emits an *initial query,* which results in subsequent queries to other agents (which we call *cooperating agents*), it is necessary that every agent become aware of the specific knowledge related to the focus (or situation) of the initial query. This is a key feature that enables effective contextual reasoning because it nourishes the reasoning ability of the cooperating agents with the possibility of considering relevant - and possibly unknown *a priori* by these agents - information in the context of the emitter agent, which can be related to its current activity and/or location. We now define a *query context* as well as the concept of *focus knowledge base*.

Definition 8: *A* ***query context*** *for a term* p *is a tuple* $c_Q = (\alpha, p, a_i, KB_\alpha^F)$ *in which* α *is a unique identifier of the query context,* a_i *is the agent that created the query, and* KB_α^F *is the focus knowledge base of the query context* α*.*

Definition 9: *A* ***focus knowledge base*** KB_α^F *of query context* α *is a set of defeasible rules whose concrete terms can also be represented as focus terms. A focus term is a tuple* (F, x)*, where* F *indicates that the term must be interpreted temporarily as a local concrete term.*

Given these definitions, it is possible to define a MAS composed of agents with contextual defeasible reasoning capabilities, with their own knowledge bases comprising the types of rules presented. Furthermore, such agents have the capability of issuing queries and sharing focus knowledge bases.

Definition 10: *A* ***contextual defeasible multi-agent system (CDMAS)*** *is defined as a tuple* $M = (A, C_Q, \Theta, st)$ *in which* $A = \{a_1, a_2, \dots a_n\}$ *is a set of agents at a given moment situated in the environment,* C_Q *is a set of query contexts at a given moment,* Θ *is a similarity function, and* st *is a similarity threshold.*

For convenience, it is also useful to define additional definitions. The global knowledge base KB_M of a system M is the union of the knowledge bases of all the agents in the system, that is, $KB_M = \bigcup_{a_i}^{A} KB_i$. An *extended global knowledge base* $KB_{M\alpha} = KB_M \cup KB_\alpha^F$ includes the focus knowledge of a query context α. We also refer to the extended global knowledge base as **context**, given that it includes all the relevant knowledge for a given situation.

Based on these definitions, it is possible to formalize the problem at hand: "Given an agent a_i with a query context $c_q = (\alpha, p, a_i, KB_\alpha^F)$ for a term p in a multi-agent system M, a_i has to answer whether p is a logical consequence of the context $KB_{M\alpha}$ or not". This is a sub-problem of the following problem: "Given a multi-agent system M considering a query context $c_q = (\alpha, p, a, KB_\alpha^F)$, which terms in the context $KB_{M\alpha}$ are its logical consequences?". As explained in Section 5.3, it is possible that p in $KB_{M\alpha}$ is either a logical consequence of the context ($KB_{M\alpha} \vDash p$) or not ($KB_{M\alpha} \nvDash p$). However, it may not be possible to state whether p is a logical consequence or not when the accepted argument that concludes p is supported by an infinite argumentation tree, as explained in Section 5.2.

4. Example scenarios

This scenario takes place in an ambient intelligence environment of mushroom hunters in a natural park, who are able to communicate with one another through a wireless network, with the goal of collecting edible mushrooms. Suppose that, at a given moment, there are five agents in the system: Alice (a), Bob (b), Catherine (c), Dennis (d), and Eric (e). The knowledge bases for each agent are presented in Fig. 1. It is interesting to note here that this scenario cannot be modeled by the original CDL presented in [1], given the need to use schematic rules.

Alice's: a $KB_a = \left\{\begin{array}{l} r_{a1}: (a, \neg ed(Mus)) \Leftarrow (a,\ dc(Mus)) \\ r_{a2}: (a, col(Mus)) \Leftarrow (X, ed(Mus)) \\ r_{a3}: (a, \neg col(Mus)) \Leftarrow (X, \neg ed(Mus)) \end{array}\right\}$ $A_{Ka} = \{b, c, d, e\}$ $P_a(b) = 0.4; P_a(c) = 0.6;$ $P_a(d) = 0.2; P_a(e) = 0.8$	**Bob's:** b $KB_b = \left\{\begin{array}{l} r_{b1}: (b, \neg ed(Mus)) \Leftarrow (b, hv(Mus)) \\ r_{b2}: (b, col(Mus)) \Leftarrow (X, ed(Mus)) \\ r_{b3}: (b, \neg col(Mus)) \Leftarrow (X, \neg ed(Mus)) \end{array}\right\}$ $A_{Kb} = \{a, c, d, e\}$ $P_b(a) = P_b(c) = P_b(d) = P_b(e) = 0.5$
Catherine's: c $KB_c = \{ r_{c1}: (c, ed(Mus)) \Leftarrow (X, avl(Mus)) \}$ $A_{Kc} = \{a, b, d, e\}$ $P_c(a) = P_c(e) = 0.8$ $P_c(b) = P_c(d) = 0.4$	**Dennis':** d $KB_d = \{ r_{d1}: (d, \neg ed(Mus)) \Leftarrow (X, am(Mus)) \}$ $A_{Kd} = \{a, b, c, e\}$ $P_d(c) = P_d(e) = 0.8$ $P_d(a) = P_d(b) = 0.4$
Eric's: e $KB_e = \{ r_{e1}: (e, spa(Mus)) \Leftarrow (e, hv(Mus)), (e, pbc(Mus)) \}$ $A_{Ke} = \{a, b, c, d\}$ $P_e(a) = 0.4$ $P_e(b) = 0.6$ $P_e(c) = 0.2$ $P_e(d) = 0.8$	

Figure 1. Agents' mental states from the given scenario.
Source: The authors.

Alice has some knowledge about some species, such as the *death cap*, which she knows is not edible (rule r_{a1}). Suppose also that Alice and Bob share the same knowledge that, if a mushroom is edible, then they can collect it, and if it is not edible, they should not collect it. They are also willing to consider the opinion of any known agent about the edibleness of a mushroom, that is, if any agent states that a mushroom is edible, then they are willing to accept it to be true if there is no adequate contrary evidence. Such knowledge is thus presented in both a (rules r_{a2} and r_{a3}) and b (rules r_{b2} and r_{b3}). Bob also believes that an object is not edible if it has a *volva* (rule r_{b1}).

Catherine believes that a mushroom is edible if it is an *Amanita velosa* (rule r_{c1}). However, she is unable to describe a mushroom of this species; thus, the rule is a schematic rule, meaning that she is willing to accept that the mushroom is an *Amanita velosa* if any other agent states that a mushroom is sufficiently similar to *Amanita velosa*.

Dennis believes that an object is not edible if it is an *Amanita* (rule r_{d1}). However, he does not know anything about the characteristics of *Amanita*; thus, a schematic term is used in the body of the rule.

Finally, Eric believes that a mushroom is a *springtime amanita* if it has some properties, such as having a *volva* and a pale brownish cap (r_{e1}). *Springtime amanita* is the same type of mushroom as *Amanita velosa*, but we assume that the agents in the system are not so certain about it. Therefore, all the agents consider a similarity degree of 0.8 between them. Similarly, *springtime amanita* is a type of *Amanita*, and the agents consider a similarity degree of 0.5 between them. Hence, we consider the similarity function to have the value 1 for all lexically identical literals, and specifically the value 0.8 between *springtime amanita* and *Amanita velosa* and 0.5 between *Amanita* and any specific type of *Amanita*. Suppose also that the similarity threshold for the entire system is 0.4.

Each agent has different levels of confidence in the other agents, as shown in Fig. 1. For example, Alice trusts Eric more ($P_a(e) = 0.8$), followed by Catherine ($P_a(c) = 0.6$), Bob ($P_a(b) = 0.4$) and Dennis ($P_a(d) = 0.2$). In contrast, Bob trusts all the agents equally.

Based on this scenario, some possible queries are illustrated as follows.

Example 1: Suppose that, at some moment, Alice finds mushroom m_1 that has a *volva* and a pale brownish cap. Thus, she asks herself a query:

$$p: (a, col(m_1)) \quad c_q: (\alpha, p, a, KB^F_\alpha) \quad KB^F_\alpha = \{r^F_{\alpha 1}, r^F_{\alpha 2}\}$$
$$r^F_{\alpha 1}: (F, hv(m_1)) \Leftarrow \qquad r^F_{\alpha 2}: (F, pbc(m_1)) \Leftarrow \tag{5}$$

where c_q is a new query context whose focus knowledge base contains rules that represent the mushroom's characteristics.

When receiving her own query, Alice cannot reach a conclusion based only on locally available knowledge. Thus, she must send new queries to other agents, including, in each query, the same c_q created previously to enable the agents to reason effectively about the mushroom perceived.

Given this context, Alice concludes that she can collect mushroom m_1 because of the argument generated from Catherine with the help of Eric (r_{c1} and r_{e1}), although Bob alone (r_{b1}) and Dennis with Eric (r_{d1} and r_{e1}) state that it is not edible. The reason for this decision is based mainly on the preference function of Alice, and is formally presented in Section 5.

Example 2: Suppose that, simultaneously with the query context α of Example 1, Bob finds a mushroom (m_2) that he knows is a *death cap*, and he wants to know if he can collect it or not. He then issues the following query:

$$p: (b, col(m_2)) \quad c_q: (\beta, p, b, KB^F_\beta) \quad KB^F_\beta = \{r^F_{\beta 1}\}$$
$$r^F_{\beta 1}: (F, dc(m_2)) \Leftarrow \tag{6}$$

In this case, we have a new *query context, β,* originating from Bob. It is important to note that the context in this case does not include the focus knowledge bound to query context α. As presented in Section 5, Bob will decide not to collect m_2, given that Alice will answer the query stating that m_2 is not edible.

Example 3: Suppose that in a later moment, Eric leaves the system, and Alice finds another mushroom m_3 with the same characteristics as m_1. Suppose also that Alice did not learn the reasons for collecting m_1 in query α (such an ability will be studied in future works). In this case, when issuing a new query γ, which is identical to α, she only receives answers from Bob and Dennis. Bob still states that the mushroom is not edible, and Dennis cannot answer because his conclusion also depends on the knowledge held by Eric. This will result in Alice making the decision to not collect m_3. This example demonstrates how the framework supports reasoning in dynamic environments where agents leave and enter the environment continuously.

5. Argument construction and semantics

The CDMAS framework is based on the reasoning of agents according to a specific model of argument construction and defeasible argumentation semantics. In this section, we present the underlying argumentation theory in which this reasoning is based. The formalism for the construction of an interdependent set of arguments is presented in Section 5.1. In Section 5.2, we present the argumentation semantics, which, given an interdependent set of arguments, defines the arguments that are justified and those that are rejected, followed by examples in Section 5.3.

5.1 Argument construction

In this section, we present how arguments are constructed from the rules of each agent, together with focus knowledge from a given query context. The *framework* uses arguments of local range, in the sense that each argument is built upon the rules of a single agent only, which enables reasoning to be distributed among agents. The arguments of different agents are interrelated by means of an interdependent set of arguments $Args_{M\alpha}$, which is formed from an extended global knowledge base $KB_{M\alpha}$, given a CDMAS M with a query context α. $Args_{M\alpha}$ is a set of arguments that are either *base arguments*, that is, arguments based on rules with empty bodies, or arguments that depend on the existence of other arguments, according to Definition 11.

Definition 11: *Given **an interdependent set of arguments** $Args_{M\alpha}$ based on extended global knowledge base $KB_{M\alpha}$, an **argument** $arg_i \in Args_{M\alpha}$ for a term $p = (a_i, x)$, based on a rule $r_i \in KB_{M\alpha}$ such that $Head(r_i) = p$, is an n-ary tree with root p. Given $n = |Body(r_i)|$, if $n = 0$, then the tree has a single child τ, which makes the tree a **base argument**. If $n > 0$, then the tree has n children, each one corresponding to a term $q = (d, y)$ such that $q \in Body(r_i)$, and each child is either:*

1. *a local concrete term $q' = (a_i, y)$, if $d \in \{a_i, F\}$ (i.e., q is either a local or a focus term), and q' is the root of a sub-argument $arg_i' \in Args_{M\alpha}$ based on a rule $r_i' \in KB_{M\alpha}$ such that $Head(r_i') = q'$, or*
2. *an instantiated term $q_{inst} = (a_j, y, 1)$, with $a_j \neq a_i$, in the case $d = a_j$ (i.e. q is a foreign concrete term) and there exists an argument $arg_j \in Args_{M\alpha}$ for a term $q' = (a_j, y)$ based on a rule $r_j \in KB_{M\alpha}$ such that $Head(r_j) = q'$, or*
3. *an instantiated term $q_{inst} = (a_j, y', \theta)$, in the case $d = X$ (i.e. q is a schematic term) and there exists an argument $arg_j \in Args_{M\alpha}$ for a term $q' = (a_j, y')$ based on a rule $r_j \in KB_{M\alpha}$ such that $Head(r_j) = q'$ and $\theta = \Theta(q, q') \geq st$ (i.e., q and q' are sufficiently similar).*

*$ForeignLeaves(arg)$ is a function that returns all instantiated terms in the leaves of an argument. If $ForeignLeaves(arg) = \emptyset$, then arg is an **independent argument** because it does not depend on the terms concluded by other arguments. A base argument is an independent argument.*

*A **strict argument** is derived only through strict rules. An argument derived using at least one defeasible rule is a **defeasible argument**.*

An important definition for defining argumentation semantics is that of *sub-arguments*, which is defined as a relation over arguments. It is also important to define the concept of *conclusions* of arguments.

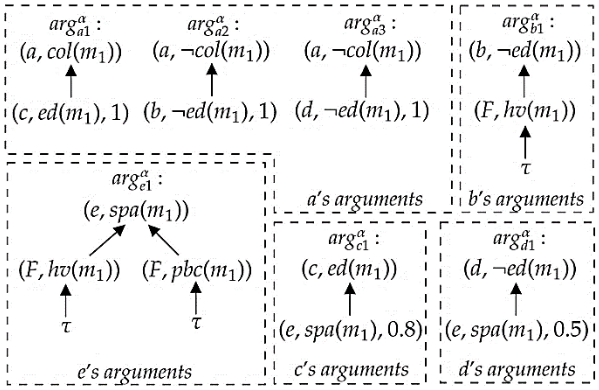


Figure 2. Interdependent set of arguments for query context α.
Source: The authors.

Definition 12: *A **sub-argument relation** $S \subset Args_{M\alpha} \times Args_{M\alpha}$ is such that $arg'\, S\, arg$ indicates that arg' is a sub-argument of arg, which occurs when $arg' = arg$, that is, an argument is always a sub-argument of itself, or when the root of arg' is a descendant of arg. A descendant of arg is either a child of the root of arg or a descendant of a child of the root of arg. Therefore, a sub-argument arg'' of argument arg' is also a sub-argument of arg. Thus, S is a reflexive and transitive relation.*

Definition 13: *Given an argument $arg \in Args_{M\alpha}$ with root p, p is called **the conclusion** of arg. The conclusion of any sub-argument arg' of arg is also called **a conclusion** of arg.*

Example 1 - Continuation: The $Args_{M\alpha}$ constructed for the query context α is shown in Fig. 2. To emphasize some properties, $ForeignLeaves(arg_{a1}^{\alpha}) = \{(c, ed(m_1), 1)\}$ and $ForeignLeaves(arg_{e1}^{\alpha}) = \emptyset$. This also means that arg_{e1}^{α} is an independent argument because it depends only on the knowledge locally available in agent *e*, given the query context α.

Note that from the schematic rule r_{a2}, the argument arg_{a1}^{α} is formed by instantiating the schematic term $(X, ed(m_1))$ with the conclusion of arg_{c1}^{α}. Similarly, arg_{a2}^{α} and arg_{a3}^{α} are formed by rule r_{a3}, having their schematic terms instantiated with the conclusions of arg_{b1}^{α} and arg_{d1}^{α}, respectively. Arguments arg_{b1}^{α} and arg_{e1}^{α} simply originate from their local rules. Note that the literals originating from the focus rules received in the query are considered to be incorporated by each agent - e.g., $(F, hv(m_1))$ in arg_{b1}^{α} and arg_{e1}^{α}, and $(F, pbc(m_1))$ in arg_{e1}^{α}. arg_{c1}^{α} has its schematic term instantiated with the conclusion of argument arg_{e1}^{α} with a similarity degree of 0.8, as well as arg_{d1}^{α} with a similarity degree of 0.5.

Example 2 - Continuation: From the *query context β*, a different interdependent set of arguments $Args_{M\beta}$ is created (Fig. 3).

It is interesting to note that $Args_{M\beta}$ is quite different from $Args_{M\alpha}$ and that both queries do not interfere with each other. This exemplifies how different *foci* or contexts lead to different interdependent sets of arguments.

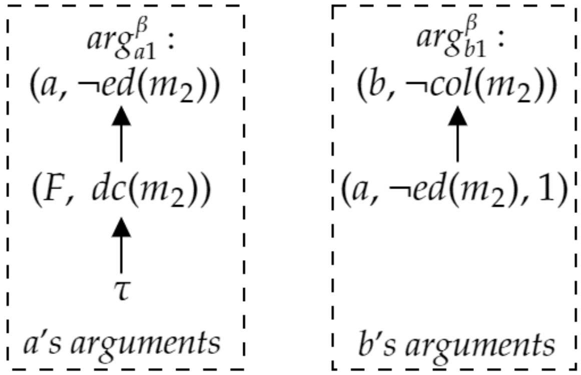


Figure 3. Interdependent set of arguments for query context β.
Source: The authors.

Example 3 - Continuation: The interdependent set of arguments for the query context γ is simply a subset of $Args_{M\alpha}$ containing only the arguments equivalent to arg^{α}_{a2} and arg^{α}_{b1}. The argument equivalent to arg^{α}_{a1} is not included because it depends on the existence of arg^{α}_{c1}, which in turn depends on the existence of arg^{α}_{e1}. Similarly, the argument equivalent to arg^{α}_{a3} is not included because it depends on the existence of arg^{α}_{d1}, which in turn depends on the existence of arg^{α}_{e1}.

5.2 Argumentation semantics

Once the interdependent set of arguments has been constructed, we need a way to determine which arguments (and consequently terms) will be accepted as logical consequences of the context. *Argumentation semantics* aim to define which arguments are *accepted*, which arguments are *rejected*, and which arguments are neither accepted nor rejected.

The definitions of *attack*, *defeat*, *argumentation tree*, *support*, *undercut*, *acceptable arguments*, *justified arguments* and *rejected arguments* represent the core of the argumentation semantics, which are all equivalent to the definitions given in [1]. These definitions are summarized in Section 5.2.1. Our contributions in this work deal more specifically with the definition of the strength comparison function ($Stronger$), which also depends on a new definition of *strength* of arguments, an evolution of the concept of *rank* given in [1]. Such specific contributions are developed in Section 5.2.2. For a more detailed description of the core definitions, we recommend the readers to [1] for the original semantic and to [12] for some variants of the semantic. Such works also present theorems and proofs regarding the consistency of the original CDL framework, which also applies to our work because we use the same core semantics.

5.2.1 Core semantics

Arguments with conflicting conclusions are said to attack each other. Therefore, there must be a way to resolve such conflicts to enable the choice of only one of the conflicting arguments. This is formalized by a defeat relation between the arguments, which depends on the definition of a strength comparison function $Stronger$, presented in Definition 22. The following defines the *attack and defeat between arguments*.

Definition 14: *An argument* arg_1 ***attacks*** *a defeasible argument* arg_2 *in* p *if* p *is a conclusion of* arg_2, $\sim p$ *is a conclusion of* arg_1, *and the sub-argument of* arg_2 *with conclusion* p *is not a strict argument.*

Definition 15: *Given a function* $Stronger:$ $Args_{M\alpha}, Args_{M\alpha}, A \rightarrow Args_{M\alpha}$, *where* A *is the set of agents in the system, an argument* arg_1 ***defeats*** *a defeasible argument* arg_2 *in* p *from the point of view of an agent* $a_i \in A$ *if* arg_1 *attacks* arg_2 *and, given the sub-argument of* arg_1 (arg_1'*) with conclusion* $\sim p$ *and the sub-argument of* arg_2 (arg_2'*) with conclusion* p, *then* $Stronger(arg'_1, arg'_2, a_i) \neq arg'_2$.

To connect different arguments, we introduce the concept of *argumentation tree*, which enables us to consider the chain of arguments used to support a conclusion.

Definition 16: *An argumentation tree* at *for a literal* p *is a tree, where each node of the tree is an argument, built using the following steps:*

1. *In the first step, add an argument* $arg_1 \in Args_{M\alpha}$ *for* p *as the root of* at*;*
2. *In the next steps, for each distinct term* $(a_j, x, \theta) \in ForeignLeaves(arg_1)$, *add an argument* arg_2 *with conclusion* (a_j, x) *as a child of* arg_1 *under the following condition:*
   - arg_2 *can be added in* at *only if there is no argument* arg_3 *for* (a_j, x) *which is already a child of* arg_1 *in* at*;*
3. *For each argument* arg_2 *added as a child in step 2, repeat step 2 for* arg_2.

The argument for p added in the first step is called the **head argument** of at, and its conclusion p is also the conclusion of at. Each node in at is the head of an argumentation tree, which is also a subtree of at. If the number of steps required to build at is finite, then at is a **finite argumentation tree**. For example, in Example 1, a possible argumentation tree with conclusion $(a, col(m_1))$ is given by $\left(arg^{\alpha}_{a1}\left(arg^{\alpha}_{c1}(arg^{\alpha}_{e1})\right)\right)$ in tree parentheses notation. An *infinite argumentation tree* arises when there are cycles in the global knowledge base, which occurs when, during the building of the argumentation tree, an argument is added that is equal to one of its ancestors, that is, its parent or an ancestor of its parent. In an actual computational implementation, as will be presented in future works, these cycles are detected at execution time using a history list passed on between agents, and the interrupted argumentation tree is then marked with a Boolean flag indicating that it is infinite.

The definition of **justified arguments** w.r.t. $KB_{M\alpha}$ is defined as $JArgs_{M\alpha} = \bigcup_{i=0}^{\infty} J_i^{M\alpha}$, where $J_i^{M\alpha}$ is a partial set of justified arguments, recursively defined as follows: (1) $J_0^{M\alpha} = \emptyset$; (2) $J_{i+1}^{M\alpha} = \{arg \in Args_{M\alpha} \mid arg$ is *acceptable* w.r.t. $J_i^{M\alpha}\}$. An **acceptable argument** arg_1 w.r.t. a set of arguments Acc is such that: (1) arg_1 is a strict argument, or (2) arg_1 is *supported* by Acc and every argument in $Args_{M\alpha}$ that defeats

arg_1 is *undercut* by Acc. An argument arg_1 is **supported** by a set of arguments Sup if: (1) every sub-argument of arg_1 is in Sup, and (2) there exists a finite argumentation tree at with head arg_1 such that every argument in at, except arg_1, is in Sup. A defeasible argument arg_1 is **undercut** by a set of arguments Und if there exists an argument arg_2 such that arg_2 is supported by Und, and arg_2 defeats a sub-argument of arg_1 or an argument in at for every argumentation tree at with head arg_1.

It is important to note that an argument arg_1 must be the head argument of a finite argumentation tree to be supported. Arguments that are the heads of infinite argumentation trees cannot be supported but they can be used to defeat other arguments. If the defeated argument is not supported by a set of arguments that undercuts the defeater, it prevents the defeated argument from being acceptable, leaving its conclusion undecided.

The set of **rejected arguments** w.r.t. $KB_{M\alpha}$ is denoted as $RArgs_{M\alpha} = \bigcup_{i=0}^{\infty} R_i^{M\alpha}$, where $R_i^{M\alpha}$ is a partial set of rejected arguments, recursively defined as follows: (1) $R_0^{M\alpha} = \emptyset$; (2) $R_{i+1}^{M\alpha} = \{arg \in Args_{M\alpha} \mid arg$ is rejected w.r.t. $R_i^{M\alpha}$, $JArgs_{M\alpha}\}$. An argument arg_1 is **rejected** w.r.t. the sets of arguments Rej (a set of already rejected arguments) and Acc (a set of acceptable arguments) when: (1) arg_1 is not a strict argument; and either (2.1) a sub-argument of arg_1 is in Rej; or (2.2) arg_1 is defeated by an argument supported by Acc; or (2.3) for every argumentation tree at with head arg_1, there exists an argument arg_2 in at, such that: either (2.3.1) a sub-argument of arg_2 is in Rej; or (2.3.2) arg_2 is defeated by an argument supported by Acc.

A justified argument implies that it resists any refutation. A term p (and its literal) is justified in $KB_{M\alpha}$ when there exists a justified argument for p, in which case p is a logical consequence of $KB_{M\alpha}$ ($KB_{M\alpha} \vDash p$). The intuition of a rejected argument is that it is not a strict argument and is either supported by the already rejected arguments or defeated by an argument supported by acceptable arguments. A term p is rejected in $KB_{M\alpha}$ if an argument with conclusion p does not exist in $Args_{M\alpha} - RArgs_{M\alpha}$, in which case p is not a logical consequence of $KB_{M\alpha}$ ($KB_{M\alpha} \nvDash p$).

5.2.2 Strength of arguments and strength comparison

In this work, we propose that strength comparison between arguments must be based not only on isolated arguments but also on an argumentation tree that supports this argument. Therefore, we denote AT_{arg} as the set of all *possible argumentation trees* with head arg. Specifically, we are interested in the most favorable argumentation tree in AT_{arg}, which we call the *chosen argumentation tree* (denoted at_{arg}^*) for an argument. To achieve this, we need a way to calculate the *strength of an argumentation tree*, which will also enable us to define the *strength of an argument*. However, these definitions first require the definitions of the *support set* of an argumentation tree and the *strength of an instantiated term*, as presented below.

Definition 17. *The* ***support set of an argumentation tree*** at*, denoted* SS_{at}*, is the set of all instantiated terms in the leaves of every argument in the argumentation tree* at*, that is,* $SS_{at} = \bigcup_{arg}^{at} ForeignLeaves(arg)$.

Definition 18. *The* ***strength of an instantiated term*** $p = (a_j, l, \theta)$ *from the point of view of agent* a_i *is* $St_p(p, a_i) = P_i(a_j) \times \theta$.

Definition 19. *The* ***strength of an argumentation tree*** *at with respect to its head argument* arg*, defined by agent* a_i*, is the sum of the strengths of all instantiated terms in the support set of the argumentation tree (*SS_{at}*), that is,* $St_t(at, a_i) = \sum_{p}^{SS_{at}} St_p(p, a_i)$.

An agent a_i assigns a strength to an argumentation tree based on the strengths of the instantiated terms that compose its support set. For each term, the agent gives it a strength based on its preference function P_i and the term's similarity degree θ. For example, in Example 1, agent a will assign a strength $P_a(e) \times 0.8 = 0.8 \times 0.8 = 0.64$ to the instantiated term $(e, spa(m_1), 0.8)$ and a strength $P_a(b) \times 1 = 0.4 \times 1 = 0.4$ to the instantiated term $(b, ed(m_1), 1)$.

The similarity degree θ is considered in the strength calculation based on the intuition that less similarity to the actual term that was used to create the instantiated term implies less certainty about the usefulness of the term to support an argument.

It is also important to note that this is the default definition of *strength*. It is possible to define different strength calculations, as well as different definitions for the function $Stronger$ (Definition 22), depending on the specificities of the application that will be implemented. For example, this default *strength* definition is based on the intuition that all arguments from all agents that contributed to reach a conclusion must be considered, including the total number of arguments and their respective strengths. In this way, a kind of social trust is implemented, similar to the skeptical agent proposed in [13]. Different formulas for strength calculation can also be used to implement more credulous or more skeptical agents. For example, a possible strength calculation could consider the "distance" of arguments in the argumentation tree to the head argument to demonstrate that an agent trusts less in arguments resulting from subsequent queries, e.g., as Alice asks Catherine, who then asks Eric, Alice could assign a lower value to Eric's argument.

Still with respect to the definition of strength, the consideration of the entire argumentation tree for strength calculation avoids strong bias toward a specific agent, as is the case with the default strategy proposed in the original work on CDL [1], which considers only the strengths of the foreign leaves in the head argument. Such a strategy has a problem that an agent will always choose responses from its most preferred agent.

As we are interested in the chosen argumentation tree (at_{arg}^*) with the most favorable strength among the argumentation trees with head arg considering the strength of each subtree from the point of view of the agent that defines arg, the following presents the definitions of the *chosen argumentation tree* and the *strength of an argument*.

Definition 20. *The **chosen argumentation tree** for an argument* arg*, denoted* $at^{*}_{arg} \in AT_{arg}$*, is the argumentation tree with head* arg *such that each child* arg' *is the head of the argumentation tree with the greatest strength among all argumentation trees in* $AT_{arg'}$*.*

Definition 21. *The **strength of an argument*** arg *is given by the strength of the chosen argumentation tree* at^{*}_{arg}*, that is,* $St_{arg}(arg, a_i) = St_t(at^{*}_{arg}, a_i)$*.*

Given the definition of the strength of an argument, it is possible to define the function *Stronger* that compares the strengths of two arguments.

Definition 22. *The **function*** $\boldsymbol{Stronger}$ *is such that, given two arguments* arg_1 *and* arg_2 *and an agent* a_i*,* $Stronger(arg_1, arg_2, a_i) = arg_1$ *if* $St_{arg}(arg_1, a_i) \geq St_{arg}(arg_2, a_i)$*. Otherwise,* $Stronger(arg_1, arg_2, a_i) = arg_2$*.*

It is important to note that there may be cases where the strengths of the two arguments are equal. In these cases, the first argument passed is chosen as the stronger argument.

5.3 Argumentation semantics example

Example 1 - Continuation: From the interdependent set of arguments $Args_{M\alpha}$ constructed in Section 5.1, the following are the strengths of the arguments arg^{α}_{a1}, arg^{α}_{a2} and arg^{α}_{a3} that attack each other:

$$\boldsymbol{St_{arg}(arg^{\alpha}_{a1}, a)} = St_p((c, ed(m_1), 1), a) + St_p((e, spa(m_1), 0.8), a) = P_a(c) \times 1 + P_a(e) \times 0.8 = 0.6 + 0.64 = \mathbf{1.24} \quad (7)$$

$$\boldsymbol{St_{arg}(arg^{\alpha}_{a2}, a)} = St_p((b, \neg ed(m_1), 1), a) = P_a(b) \times 1 = \mathbf{0.4} \quad (8)$$

$$\boldsymbol{St_{arg}(arg^{\alpha}_{a3}, a)} = St_p((d, \neg ed(m_1), 1), a) + St_p((e, spa(m_1), 0.5), a) = P_a(d) \times 1 + P_a(e) \times 0.5 = 0.2 + 0.4 = \mathbf{0.6} \quad (9)$$

Therefore, as illustrated in Fig. 4, arg^{α}_{a1} defeated both arg^{α}_{a2} and arg^{α}_{a3}. The following steps, also presented in Fig. 4, are used to create a set of justified arguments, starting from $J^{M\alpha}_0 = \emptyset$: (1) the arguments $arg^{\alpha}_{b1}{}'$, $arg^{\alpha}_{e1}{}'$ and $arg^{\alpha}_{e1}{}''$ are justified since they are base arguments and no other argument defeats them; thus $J^{M\alpha}_1 = J^{M\alpha}_0 \cup \{arg^{\alpha}_{b1}{}', arg^{\alpha}_{e1}{}', arg^{\alpha}_{e1}{}''\}$; (2) the arguments arg^{α}_{b1} and arg^{α}_{e1} are justified since their only respective sub-arguments are supported by $J^{M\alpha}_1$ and no other argument defeats them; thus $J^{M\alpha}_2 = J^{M\alpha}_1 \cup \{arg^{\alpha}_{b1}, arg^{\alpha}_{e1}\}$; (3) arg^{α}_{c1} and arg^{α}_{d1} are justified, since they are supported by $J^{M\alpha}_2$ (because all the other arguments in their respective argumentation trees are in $J^{M\alpha}_2$) and no other argument defeats them; although arg^{α}_{a2} is also supported by $J^{M\alpha}_2$, it is defeated by arg^{α}_{a1} and $J^{M\alpha}_2$ does not undercut it, thus it is not acceptable; therefore, $J^{M\,\alpha}_3 = J^{M\alpha}_2 \cup \{arg^{\alpha}_{c1}, arg^{\alpha}_{d1}\}$; (4) arg^{α}_{a1} is justified, because it is supported by $J^{M\alpha}_2$ and, although arg^{α}_{a1} is attacked by arg^{α}_{a2} and arg^{α}_{a3}, it is not defeated by them; arg^{α}_{a3} is not justified by the same reason as arg^{α}_{a2} in step 3; thus, $J^{M\alpha}_4 = J^{M\alpha}_3 \cup \{arg^{\alpha}_{a1}\} = JArgs_{M\alpha}$. As there are no more arguments to justify, $JArgs_{M\alpha} = J^{M\alpha}_4$.

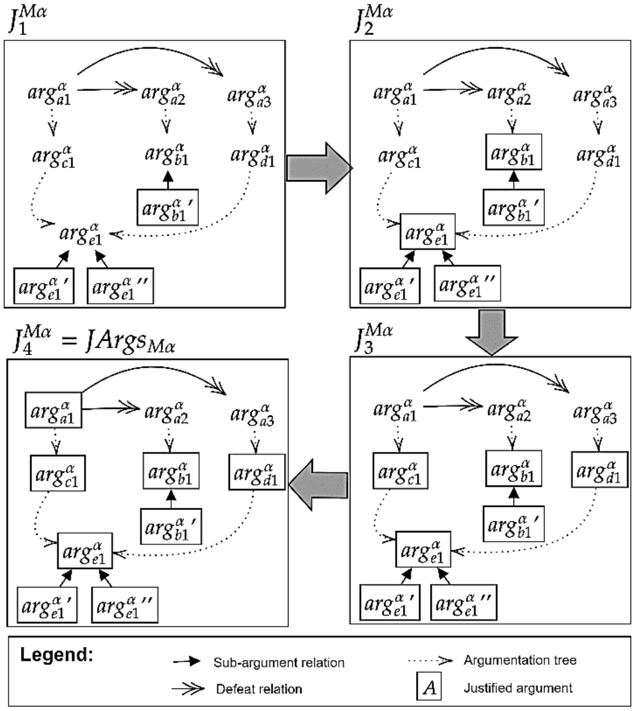


Figure 4. Illustration of the semantics of justified arguments for Example 1.
Source: The authors.

Example 2 - Continuation: From the interdependent set of arguments $Args_{M\beta}$ constructed in Section 5.2, the following steps are used to create a set of justified arguments, starting from $J^{M\beta}_0 = \emptyset$: (1) the argument $arg^{\beta}_{a1}{}'$ is justified since it is a base argument and no argument defeats it; thus, $J^{M\beta}_1 = J^{M\beta}_0 \cup \{arg^{\beta}_{a1}{}'\}$; (2) arg^{β}_{a1} is justified since its only sub-argument is supported by $J^{M\beta}_1$; thus, $J^{M\beta}_2 = J^{M\beta}_1 \cup \{arg^{\beta}_{a1}\}$; (3) arg^{β}_{b1} is justified since the argument arg^{β}_{a1}, which is the only other argument in the argumentation tree with head argument $B_{\beta 1}$, is supported by $J^{M\beta}_2$; thus, $J^{M\beta}_3 = J^{M\beta}_2 \cup \{arg^{\beta}_{b1}\}$. As there are no more arguments, then $JArgs_{M\beta} = J^{M\beta}_3$.

Example 3 - Continuation: It is easy to visualize this case when considering the steps of Example 1, as illustrated in Fig. 4, but ignoring the existence of the arguments equivalent to arg^{α}_{a1}, arg^{α}_{a3}, arg^{α}_{c1}, arg^{α}_{d1}, and arg^{α}_{e1} and their sub-arguments. The main difference is that arg^{γ}_{a2} (equivalent to arg^{α}_{a2} of Example 1) in this case is not attacked by any argument, and thus is accepted. Therefore, the set of justified arguments in this case is $\{arg^{\gamma}_{a2}, arg^{\gamma}_{b1}, arg^{\gamma}_{b1}{}'\}$.

6. Related work

The main related work is [1], which the model presented in this work extends. It enables decentralized distributed reasoning based on a distributed knowledge base such that the knowledge from different knowledge bases may conflict with each other. However, their work still lacks generality in the sense that many use case scenarios that cannot be represented in their system exist. One kind of such scenarios are those that require that agents share relevant knowledge when issuing a query to others. Another kind of scenarios are those in which the bindings among agents through mapping rules are not static, such as knowledge-intensive and dynamic environments.

There is also a proposed extension to the framework that uses a partial order relation by means of a graph structure combined with the possibility of using different semantics, such as ambiguity propagation semantics and team defeat [12]. Our work takes a different approach in defining a partial relation using a function over known agents, which enables a more efficient argument strength calculation. With regard to using the different reasoning semantics proposed, our work also supports them since it impacts only the definitions of undercut and acceptable arguments, which remain unchanged in such semantics.

Other works with similar goals and approaches are peer-to-peer inference systems, such as that proposed in [2], which also provides distributed reasoning based on distributed knowledge. The works in [14,15] provide ways of resolving inconsistent knowledge between peers, and [15] specifically uses argumentation to achieve non-monotonic reasoning. However, such systems do not deal with the idea of sharing relevant context knowledge when issuing a query, in addition to other disadvantages compared to CDL, such as considering a global priority relation over the agents instead of individual preference orders for each agent, which negatively affects the dimension of perspective in contextual reasoning.

Another relevant work is distributed argumentation with defeasible logic programming [16]. However, its reasoning is not fully distributed, and it depends on a moderator agent that has part of the responsibility of constructing arguments.

Jarraya et al. [17] adopted a different approach to enable agents to reason about shared and relevant knowledge. An agent defines a goal and initiates distributed reasoning by sending all its facts to a list of observer agents, thereby initiating a bottom-up approach that starts with facts and looks for rules to apply to the facts. Their approach is more specific than that presented in this work because it defines agents with specific roles. They also did not present argumentation-based formalism.

Rakib and Haque [18] presented a logical model for resource-bounded context-aware MAS that handles inconsistent context information using non-monotonic reasoning. Their model also allows the use of communication primitives such as *Ask* and *Tell* in the rules, which enables the simulation of the role of mapping rules. However, they did not foresee the possibility of asking or telling in a broadcast manner, which could enable the simulation of the concept of schematic rules. Moreover, their approach was not based on argumentation.

Other works on MCS [19,20] proposed resolving conflicts by means of the concept of *repair* prior to the effective reasoning process to maintain a conflict-free (consistent) global state. However, this may be undesirable in systems where agents may have different points of view of the environment, which can be conflicting but may be valid in certain contexts.

7. Conclusion and future work

In this work, we presented a structured argumentation framework to enable multi-agent fully distributed contextual reasoning, including the idea of focus, which consists of sharing relevant information to cooperating agents when issuing a query. We also proposed a key feature to enable the application of the framework in open, dynamic, and knowledge-intensive multi-agent environments. The framework was built on CDL, and we demonstrated how it is more general than CDL, thereby enabling the modeling of a broader range of scenarios.

Our future work will include the presentation of a distributed query-answering algorithm that efficiently implements the argumentation framework presented. Actually, this algorithm has already been defined and has an agent-based implementation but it was not presented in this work because of lack of space. We also intend to explicitly define other strategies for rank calculation that may better reflect different scenarios. Another interesting idea is to take advantage of the exchange of arguments between agents to enable explanation and learning.

Acknowledgements

We would like to thank Editage (www.editage.com) for English language editing and the Program in Electrical and Computer Engineering of Federal University of Technology - Paraná for financially providing English language revision.

References

[1] Bikakis, A. and Antoniou, G., Defeasible contextual reasoning with arguments in ambient intelligence. IEEE Transactions on Knowledge and Data Engineering, 22(11), pp. 1492-1506, 2010. DOI: 10.1109/TKDE.2010.37

[2] Adjiman, P., Chatalic, P., Goasdoue, F., Rousset, M.C. and Simon, L., Distributed reasoning in a peer-to-peer setting: application to the semantic web. Journal of Artificial Intelligence Research, 25, pp. 269-314, 2006. DOI: 10.1613/jair.1785

[3] Benerecetti, M., Bouquet, P. and Ghidini, C., Contextual reasoning distilled. Journal of Experimental & Theoretical Artificial Intelligence, 12(3), pp. 279-305, 2000. DOI: 10.1080/09528130050111446

[4] Monte-Alto, H., Morveli-Espinoza, M. and Tacla, C., Multi-Agent systems based on contextual defeasible logic considering focus [Online]. 2020. Available at: https://arxiv.org/abs/2010.00168.

[5] Governatori, G., Maher, M.J., Antoniou, G. and Billington, D., Argumentation semantics for defeasible logic. Journal of Logic and Computation, 14(5), pp. 675-702, 2004. DOI: 10.1007/3-540-44533-1_7

[6] 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), pp. 321-357, 1995. DOI: 10.1016/0004-3702(94)00041-X

[7] Castelfranchi, C. and Falcone, R., Trust theory: a socio-cognitive and computational model. John Wiley & Sons, 2010. DOI: 10.1002/9780470519851

[8] Modgil, S., Reasoning about preferences in argumentation frameworks. Artificial Intelligence, 173(9-10), pp. 901-934, 2009. DOI: 10.1016/j.artint.2009.02.001

[9] Bechhofer, S., Van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F. and Stein, L.A. Owl web ontology language reference. W3C Recommendation, [online]. 10(2), 2004. Available at: https://www.w3.org/TR/owl-ref

[10] Miller, G.A., Wordnet: a lexical database for english. Communications of the ACM, 38(11), pp. 39-41, 1995. DOI: 10.1145/219717.219748

[11] Sycara, K., Widoff, S., Klusch, M. and Lu, J., Larks: dynamic matchmaking among heterogeneous software agents in cyberspace. Autonomous Agents and Multi-Agent Systems, 5(2), pp. 173-203, 2002. DOI: 10.1023/A:1014897210525

[12] Bikakis, A. and Antoniou, G., Partial preferences and ambiguity resolution in contextual defeasible logic, in: Proceedings of the International Conference on Logic Programming and Nonmonotonic Reasoning, 2011, pp. 193-198.

[13] Panisson, A.R., Melo, V.S. and Bordini, R.H., Using preferences over sources of information in argumentation-based reasoning, in: Proceedings of the 5th Brazilian Conference on Intelligent Systems (BRACIS), 2016, pp. 31-36.

[14] Chatalic, P., Nguyen, G.H. and Rousset, M.C., Reasoning with inconsistencies in propositional peer-to-peer inference systems, in: Proceedings of the 17th European Conference on Artificial Intelligence (ECAI), 2006, pp. 352-356.

[15] Binas, A. and McIlraith, S.A., Peer-to-peer query answering with inconsistent knowledge, in: Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning, 2008, pp. 329-339.

[16] Thimm, M. and Kern-Isberner, G., A distributed argumentation framework using defeasible logic programming, in: Proceedings of the 2008 Conference on Computational Models (COMMA), 2008, pp. 381-392.

[17] Jarraya, A., Arour, K., Borgi, A. and Bouzeghoub, A., Distributed cooperative reasoning in ambient environment, in: Proceedings of the 30th International Conference on Advanced Information Networking and Applications (AINA), 2016, pp. 1085-1092.

[18] Rakib, A. and Mahfooz, Ul. and Haque, H., A logic for context-aware non-monotonic reasoning agent, in: Proceedings of the 13th Mexican International Conference on Artificial Intelligence (MICAI), 2014, pp.453-471.

[19] Dao-Tran, M., Eiter, T., Fink, M. and Krennwallner, T., Dynamic distributed nonmonotonic multi-context systems, in: Nonmonotonic Reasoning, Essays Celebrating its 30th Anniversary, Studies in Logic, College Publications, Lexington, Kentucky, USA, 2011.

[20] Brewka, G., Ellmauthaler, S., Gonçalves, R., Knorr, M., Leite, J. and Puhrer, J., Reactive multi-context systems: heterogeneous reasoning in dynamic environments. Artificial Intelligence, 256, pp. 68-104, 2018. DOI: 10.1016/j.artint.2017.11.007

H.H.L.C. Monte-Alto, received his MSc. on Computer Science in 2014 from the State University of Maringá (UEM), Brazil. Currently, he is an assistant professor at Federal University of Paraná (UFPR), Brazil and is pursuing his Dr. on Electrical Engineering and Industrial Informatics from the Federal University of Technology - Paraná (UTFPR), Brazil, researching argumentation frameworks and multi-agent systems.
ORCID: 0000-0003-3136-0378

A.T. Possebom, received his Dr. on Electrical Engineering and Industrial Informatics in 2018 from UTFPR, Brazil. Currently, he is a full professor in informatics at the Paraná Federal Institute of Education, Science and Technology (IFPR), Brazil. His research interests include: multi-agent systems, argumentation, and decision making.
ORCID: 0000-0002-1347-5852

M. Morveli-Espinoza, received her Dr. on Electrical Engineering and Industrial Informatics in 2018 from UTFPR, Brazil. From 2007 to 2010, she did research at the Artificial Intelligence Research Institute of Barcelona.
ORCID: 0000-0002-7376-2271

C.A. Tacla, received his PhD. in Informatics in 2003 from the University of Technology of Compiègne (UTC), France. Currently, he is full professor in the program in Electrical and Computer Engineering at UTFPR, Brazil. His research interests include: multi-agent systems, argumentation, ontologies, and cooperative systems.
ORCID: 0000-0002-8244-8970