Defeasible Contextual Reasoning with Arguments in Ambient Intelligence

Antonis Bikakis and Grigoris Antoniou

***Abstract*—The imperfect nature of context in Ambient Intelligence environments and the special characteristics of the entities that possess and share the available context information render contextual reasoning a very challenging task. The accomplishment of this task requires formal models that handle the involved entities as autonomous logic-based agents and provide methods for handling the imperfect and distributed nature of context. This paper proposes a solution based on the Multi-Context Systems paradigm, in which local context knowledge of ambient agents is encoded in rule theories (*contexts*), and information flow between agents is achieved through mapping rules that associate concepts used by different contexts. To handle *imperfect* context, we extend Multi-Context Systems with non-monotonic features; local defeasible theories, defeasible mapping rules, and a preference ordering on the system contexts. On top of this model, we have developed an argumentation framework that exploits context and preference information to resolve potential conflicts caused by the interaction of ambient agents through the mappings, and a distributed algorithm for query evaluation.**

***Index Terms*—Ambient Intelligence, contextual reasoning, defeasible reasoning, argumentation systems**

I. Introduction

COMPUTING is moving towards pervasive, ubiquitous environments in which devices, software agents and services are all expected to seamlessly integrate and cooperate in support of human objectives, anticipating needs and delivering services in an anywhere, any-time and for-all fashion [1]. *Pervasive Computing* and *Ambient Intelligence* are considered to be key issues in the further development and use of Information and Communication technologies, as evidenced, for example, by the IST Advisory Group [2].

The study of Ambient Intelligence environments and pervasive computing systems has introduced new research challenges in the field of Distributed Artificial Intelligence. These are mainly caused by the imperfect nature of context and the special characteristics of the entities that possess and share the available context information. Henricksen and Indulska characterize in [3] four types of imperfect context: *unknown*, *ambiguous*, *imprecise*, and *erroneous*. The agents that operate in ambient environments are expected to have different goals, experiences and perceptive capabilities and use distinct vocabularies to describe their context. Due to the highly dynamic and open nature of the environment and the unreliable wireless communications that are restricted by the range of transmitters, ambient agents do not typically know a priori all other entities that are present at a specific time instance nor can they communicate directly with all of them.

A. Bikakis and G. Antoniou are with the Institute of Computer Science, FO.R.T.H., Greece, e-mail: {bikakis,antoniou}@ics.forth.gr.

So far, Ambient Intelligence systems have not managed to efficiently handle all these challenges. As it has been already surveyed in [4], most of them follow classical reasoning approaches that assume perfect knowledge of context, failing to deal with cases of missing, inaccurate or inconsistent context information. Regarding the distribution of reasoning tasks, a common approach followed in most systems assumes the existence of a central entity, which is responsible for collecting and reasoning with all the available context information. However, Ambient Intelligence environments have much more demanding requirements. The dynamics of the network and the unreliable and restricted wireless communications inevitably call for the decentralization of the reasoning tasks.

In previous works, we presented a formal approach for representing imperfect and distributed context in Ambient Intelligence environments [5], and described application scenarios of our approach in the Ambient Intelligence domain [6]. Our approach is based on a formalism from Distributed Artificial Intelligence, known as *Multi-Context Systems* (*MCS*). Specifically, we model the involved entities as autonomous logic-based agents; the knowledge possessed by an agent as a local context rule theory; and the associations between the context knowledge possessed by different ambient agents as mappings between their respective context theories. To support cases of missing or inaccurate local context knowledge, we represent contexts as local theories of Defeasible Logic; and to handle potential inconsistencies caused by the interaction of contexts, we extend the MCS model with defeasible mapping rules, and with a preference ordering on the system contexts. The main contribution of this paper is to provide a semantic characterization of the model using *arguments*. The use of arguments is a natural choice in multi-agent systems, as evidenced for example in [7], and aims at a more formal and abstract description of our approach. Conflicts that arise from the interaction of mutually inconsistent contexts are captured through *attacking* arguments, and conflict resolution is achieved by *ranking* arguments according to a preference ordering on the system contexts. We also provide an operational model in the form of a distributed algorithm for query evaluation. The algorithm has been implemented in Java and evaluated in a simulated P2P system, and the results are available in [8]. Here, we focus more on its formal properties.

The rest of the paper is organized as follows: next section describes a motivating scenario that highlights the special characteristics of contextual reasoning in Ambient Intelligence. Section III presents background information and related work on contextual reasoning. Section IV presents the proposed representation model, while Section V describes its argumentation semantics. Section VI presents the distributed algorithm for query evalaution and studies its properties, while last section summarizes and discusses future work.

II. Motivating Scenario

In this section we describe an imaginary scenario from the Ambient Intelligence domain and discuss the challenges of reasoning with the available context information.

A. Context-Aware Mobile Phone in an Ambient Classroom

The scenario involves a context-aware mobile phone that has been configured by Dr. Amber to decide whether it should ring (in case of incoming calls) based on his preferences and context. Dr. Amber has the following preferences: His phone should ring in case of an incoming call, unless it is in silent mode or he is giving a lecture.

Consider the case that Dr. Amber is currently located in the 'RA201' university classroom. It is class time, but he has just finished with a course lecture, and still remains in the classroom reading his emails on his laptop. The mobile phone receives an incoming call, while it is in normal mode.

The phone cannot decide whether it should ring based only on its local context knowledge, which includes information about incoming calls and the mode of the phone, as it is not aware of other important context parameters (e.g. Dr. Amber's current activity). Therefore, it attempts to contact through the wireless network of the university other ambient agents that are located nearby, import from them further context information, and use this information to reach a decision.

In order to determine whether Dr. Amber is currently giving a lecture, the mobile phone uses two rules. The first rule states that if at the time of the call there is a scheduled lecture, and Dr. Amber is located in a university classroom, then he is possibly giving a lecture. Information about scheduled events is imported from Dr. Amber's laptop, which hosts Dr. Amber's calendar. According to this, there is a scheduled class event. Information about Dr. Amber's location is imported from the wireless network localization service, which at the time of the call localizes Dr. Amber (actually his mobile phone) in 'RA201' university classroom. The second rule states that if there is no class activity in the classroom, then Dr. Amber is rather not giving a lecture. Information about the state of the classroom is imported from the classroom manager, a stationary computer installed in 'RA201'. In this case, based on local information about the state of the projector (it is off), and information about the presence of people in the classroom that it imports from an external person detection service, which in this case detects only one person, the classroom manager infers that there is no class activity.

The overall information flow in the scenario is depicted in Figure 1. Eventually, the mobile phone will receive ambiguous information from the various ambient agents operating in the classroom. Information imported from Dr. Amber's laptop and the localization service leads to the conclusion that Dr. Amber is currently giving a lecture, while information imported from the classroom manager leads to the contradictory conclusion that Dr. Amber is not currently giving a lecture. To resolve this conflict, the mobile phone must be able to evaluate the information it receives from the various sources. For example, in case it is aware that the information derived from the classroom manager is more accurate than the information imported from Dr. Amber's laptop, it will determine that Dr. Amber is not currently giving a lecture, and therefore reach the 'ring' decision.

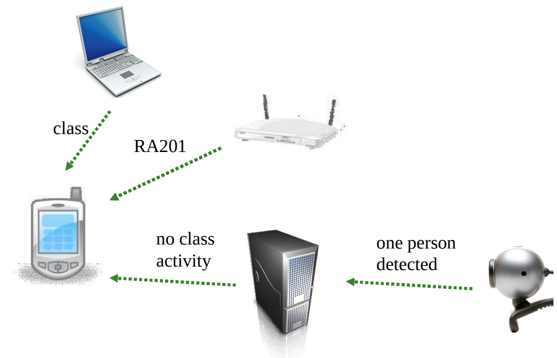

Fig. 1. Context Information Flow in the Scenario

B. Assumptions and Challenges

In the scenario presented above, we have implicitly made the following assumptions:

- There is an available means of communication through which an ambient agent can communicate and exchange context information with a subset of the other available ambient agents.
- Each agent is aware of the type and quality of knowledge that each of the other agents possesses, and has specified how part of this knowledge relates to its local knowledge.
- Each agent has some computing and reasoning capabilities that it enables it conducting simple reasoning tasks.
- Each agent is willing to disclose and share part of its local knowledge.

The challenges of contextual reasoning in the described scenario include:

- Local context knowledge is incomplete, meaning that none of the agents involved in the scenario has immediate access to all available context information.
- Context knowledge is ambiguous; specifically, the information imported from the laptop, the localization service and the classroom manager leads to a conflict about Dr. Amber's current activity.
- Context knowledge is imprecise; e.g. Dr. Amber's schedule may be inaccurate.
- The computational capabilities of some of the devices are restricted (e.g. the mobile phone), so the overhead imposed by the reasoning tasks must not be too heavy.
- The communication load must not also be too heavy, so that the mobile phone can quickly reach a decision.
- Real-time requirements on overall system performance.

III. Background & Related Work

A. Multi-Context Systems

Since the seminal work of McCarthy [9] on *context* and *contextual reasoning*, two main formalizations have been proposed to formalize context: the *Propositional Logic of Context* (*PLC* [10], [11]), and the *Multi-Context Systems* (*MCS*) introduced in [12], which later became associated with the *Local Model Semantics* [13]. MCS have been argued to be most adequate with respect to the three dimensions of contextual reasoning (*partiality*, *approximation*, *proximity*) and shown to be technically more general than PLC [14]. A MCS consists of a set of *contexts* and a set of inference rules (known as *mapping* rules) that enable information flow between different contexts. A context can be thought of as a logical theory (a set of axioms and inference rules) that models local knowledge. Different contexts are expected to use different languages, and although each context may be locally consistent, global consistency cannot be required or guaranteed.

The MCS paradigm has been the basis of two recent studies that were the first to deploy non-monotonic features in contextual reasoning: (a) the non-monotonic rule-based MCS [15], which supports default negation in the mapping rules allowing to reason based on the absence of context information; and (b) the multi-context variant of Default Logic (*ConDL* [16]), which additionally handles the problem of mutually inconsistent information provided by two or more different sources using default mapping rules. However, *ConDL* does not provide ways to model the quality of imported knowledge, nor preference between different information sources, leaving the conflicts that arise in such cases unresolved.

The use of Multi-Context Systems as a means of specifying and implementing agent architectures has been proposed in some recent studies (e.g. [17], [18]). Specifically, they propose *breaking* the logical description of an agent into a set of contexts, each of which represents a different component of the architecture, and the interactions between these components are specified by means of bridge rules. A similar approach is followed in [19], where *contextual deliberation* of cognitive agents is achieved using a special extension of Defeasible Logic. Here, we follow a different approach; a context does not represent a logical component of an agent, but rather the viewpoint of each different agent in the system.

B. Peer Data Management Systems

Peer data managements systems can be viewed as special cases of MCS, as they consist of autonomous logic-based entities (*peers*) that exchange local information using *bridge rules*. Two prominent recent works that handle the problem of peers providing mutually inconsistent information are: (a) the approach of [20], which is based on non-monotonic epistemic logic; and (b) the propositional P2P Inference System of [21]. A major limitation of both approaches is that conflicts are not actually resolved using some external preference information; they are rather isolated. Our approach enables resolving such conflicts using a preference ordering on the information sources. Building on the work of [21], [22] proposed an argumentation framework and algorithms for inconsistency resolution in P2P systems using a preference relation on system peers. However, their assumptions of a single global language and a global preference relation are in contrast with the dimension of perspective in MCS. In our approach, each agent uses its own vocabulary to describe its context and defines its own preference ordering.

IV. Representation Model

We model a MCS C as a collection of distributed context theories C_i: A context is defined as a tuple of the form (V_i, R_i, T_i), where V_i is the vocabulary used by C_i, R_i is a set of rules, and T_i is a preference ordering on C.

V_i is a set of positive and negative literals. If q_i is a literal in V_i, $\sim q_i$ denotes the complementary literal, which is also in V_i. If q_i is a positive literal p then $\sim q_i$ is $\neg p$; and if q_i is $\neg p$, then $\sim q_i$ is p. We assume that each context uses a distinct vocabulary.

R_i consists of two sets of rules: the set of local rules and the set of mapping rules. The body of a local rule is a conjunction of *local* literals (literals that are contained in V_i), while its head contains a local literal. There are two types of local rules:

- Strict rules, of the form

$$r_i^l : a_i^1, a_i^2, ...a_i^{n-1} \rightarrow a_i^n$$

  They express sound local knowledge and are interpreted in the classical sense: whenever the literals in the body of the rule ($a_i^1, a_i^2, ...a_i^{n-1}$) are strict consequences of the local theory, then so is the conclusion of the rule (a_i^n). Strict rules with empty body denote factual knowledge.
- Defeasible rules, of the form

$$r_i^d : b_i^1, b_i^2, ...b_i^{n-1} \Rightarrow b_i^n$$

  They are used to express uncertainty, in the sense that a defeasible rule (r_i^d) cannot be applied to support its conclusion (b_i^n) if there is adequate contrary evidence.

Mapping rules associate literals from the local vocabulary V_i (*local literals*) with literals from the vocabularies of other contexts (*foreign literals*). The body of each such rule is a conjunction of local and foreign literals, while its head contains a single local literal. Mapping rules are modeled as defeasible rules of the form:

$$r_i^m : a_i^1, a_j^2, ...a_k^{n-1} \Rightarrow a_i^n$$

r_i^m associates local literals of C_i (e.g. a_i^1) with local literals of C_j (a_j^2), C_k (a_k^{n-1}) and possibly other contexts. a_i^n is a local literal of the theory that has defined r_i^m (C_i).

Finally, each context C_i defines a strict total preference ordering T_i on C to express its confidence on the knowledge it imports from other contexts. This is of the form:

$$T_i = [C_k, C_l, ..., C_n]$$

According to T_i, C_k is preferred to C_l by C_i, if the rank of C_k is lower than the rank of C_l in T_i. The strict total preference ordering enables resolving all potential conflicts that may arise from the interaction of contexts through their mapping rules. An alternative choice, which is closer to the needs of ambient applications, is partial ordering. However, this would add complexity to the reasoning tasks, and would enable resolving certain conflicts only.

We have deliberately chosen to use the simplest version of Defeasible Logic, and disregard *facts*, *defeaters* and the *superiority relation* between rules, which are used in fuller versions of Defeasible Logic [23], to keep the discussion and technicalities simple. Besides, the results of [23] have shown that these elements can be simulated by the other ingredients of the logic. We should also note that the proposed model may also support overlapping vocabularies and enable different contexts to use elements of common vocabularies (e.g. URIs) by adding a context identifier, e.g. as a prefix in each such word, adding the modified words in the vocabularies of the contexts, and using appropriate mappings to associate them.

Example. The representation model described above is applied as follows to the scenario described in Section II. The local knowledge of the mobile phone (denoted as C_1) is encoded in strict local rules $r^l_{1,1}$ and $r^l_{1,2}$, while local defeasible rules $r^d_{1,3}$ and $r^d_{1,4}$ encode Dr. Amber's preferences.

$r^l_{1,1} : \rightarrow incoming\_call_1$
$r^l_{1,2} : \rightarrow normal\_mode_1$
$r^d_{1,3} : incoming\_call_1, \neg lecture_1 \Rightarrow ring_1$
$r^d_{1,4} : silent\_mode_1 \Rightarrow \neg ring_1$

Mapping rules $r^m_{1,5}$ and $r^m_{1,6}$ encode the associations of the local knowledge of the mobile phone with the context knowledge of Dr. Amber's laptop (C_2), the localization service (C_3), and the classroom manager (C_4).

$r^m_{1,5} : classtime_2, location\_RA201_3 \Rightarrow lecture_1$
$r^m_{1,6} : \neg class\_activity_4 \Rightarrow \neg lecture_1$

The local context knowledge of the laptop (C_2), the localization service (C_3), the classroom manager (C_4) and the person detection service (C_5) is encoded in rules $r^l_{2,1}$, $r^l_{3,1}$, $r^l_{4,1}$ and $r^l_{5,1}$, respectively. To import this information from the person detection service, the classroom manager uses rule $r^m_{4,2}$.

$r^l_{2,1} : \rightarrow classtime_2$
$r^l_{3,1} : \rightarrow location\_RA201_3$
$r^l_{4,1} : \rightarrow projector(off)_4$
$r^m_{4,2} : projector(off)_4, detected(1)_5 \Rightarrow \neg class\_activity_4$
$r^l_{5,1} : \rightarrow detected(1)_5$

The mobile phone is configured to give highest priority to information imported by the classroom manager and lowest priority to the person detection service. This is described in preference order $T_1 = [C_4, C_3, C_2, C_5]$.

V. Argumentation Semantics

The argumentation framework described in this section extends the argumentation semantics of Defeasible Logic presented in [24] (which in turn is based on the grounded semantics of Dung's abstract argumentation framework [25]) with the notions of distribution of the available knowledge, and preference among system contexts. The framework uses arguments of local range, in the sense that each one contains rules of a single context only. Arguments of different contexts are interrelated in the *Support Relation* (defined below) through mapping rules. The Support Relation contains triples that represent proof trees for literals in the system. Each proof tree is made of rules of the context that the literal in its root is defined by. In case a proof tree contains mapping rules, for the respective triple to be contained in the Support Relation, similar triples for the foreign literals in the proof tree must have already been obtained. We should note that, for sake of simplicity, we assume that there are no loops in the local context theories, and thus proof trees are finite. Loops in the local knowledge bases can be easily detected and removed without needing to interact with other agents. However, even if there are no loops in the local theories, the global knowledge base may contain loops caused by mapping rules. We should also note that even if only classical (*strict*) negation is used in the underlying object language, we can simulate *negation as failure* using elements of the underlying language following a technique based on auxiliary predicates first presented in [26].

Let $C = \{C_i\}$ be a MCS. The *Support Relation* of C (SR_C) is the set of all triples of the form (C_i, PT_{p_i}, p_i), where $C_i \in C$, $p_i \in V_i$, and PT_{p_i} is the proof tree for p_i based on the set of local and mapping rules of C_i. PT_{p_i} is a tree with nodes labeled by literals such that the root is labeled by p_i, and for every node with label q:

1) If $q \in V_i$ and $a_1, ..., a_n$ label the children of q then
   - If $\forall a_i \in \{a_1, ..., a_n\}$: $a_i \in V_i$ then there is a local rule $r_i \in C_i$ with body $a_1, ..., a_n$ and head q
   - If $\exists a_j \in \{a_1, ..., a_n\}$ such that $a_j \notin V_i$ then there is a mapping rule $r_i \in C_i$ with body $a_1, ..., a_n$ and head q
2) If $q \in V_j \neq V_i$, then this is a leaf node of the tree and there is a triple of the form (C_j, PT_q, q) in SR_C
3) The arcs in a proof tree are labeled by the rules used to obtain them.

An *argument* A for a literal p_i is a triple (C_i, PT_{p_i}, p_i) in SR_C. Any literal labeling a node of PT_{p_i} is called *a conclusion* of A. However, when we refer to *the conclusion* of A, we refer to the literal labeling the root of PT_{p_i} (p_i). We write $r \in A$ to denote that rule r is used in the proof tree of A. A (*proper*) *subargument* of A is every argument with a proof tree that is (proper) subtree of the proof tree of A.

A *local argument* of a context C_i is an argument that contains only local literals of C_i. If a local argument A contains only strict rules, then A is a *strict local argument*; otherwise it is a *defeasible local argument*. A is a *mapping argument* if its proof tree contains at least one foreign literal. $Args_{C_i}$ is the set of arguments derived from context C_i. $Args_C$ is the set of all arguments in C: $Args_C = \bigcup Args_{C_i}$.

The conclusions of all strict local arguments in $Args_{C_i}$ are local logical consequences of C_i. Distributed logical consequences are derived from a combination of local and mapping arguments in $Args_C$. In this case, we should also consider conflicts between competing rules, which are modeled as *attacks* between arguments, and preference orderings, which are used in our framework to *rank* mapping arguments.

The *rank of a literal* p in context C_i (denoted as $R(p, C_i)$) equals 0 if $p \in V_i$. If $p \in V_j \neq V_i$, then $R(p, C_i)$ equals the rank of C_j in T_i. The *rank of an argument* A in C_i

(denoted as $R(A, C_i)$) equals the maximum between the ranks in C_i of the literals contained in A. It is obvious that for any three arguments A_1, A_2, A_3: If $R(A_1, C_i) \leq R(A_2, C_i)$ and $R(A_2, C_i) \leq R(A_3, C_i)$, then $R(A_1, C_i) \leq R(A_3, C_i)$; namely the preference relation $<$ on $Args_C$, which is build according to ordering T_i, is transitive.

The definitions of *attack* and *defeat* apply only for local defeasible and mapping arguments. An argument A *attacks* a local defeasible or mapping argument B at p_i, if p_i is a conclusion of B, $\sim p_i$ is a conclusion of A, and the subargument of B with conclusion p_i is not a strict local argument. An argument A *defeats* an argument B at p_i, if A attacks B at p_i, and for the subarguments of A, A' with conclusion $\sim p_i$, and of B, B' with conclusion p_i: $R(A', C_i) \leq R(B', C_i)$.

To link arguments through the mapping rules that they contain, we introduce the notion of *argumentation line*. An *argumentation line* A_L for a literal p_i is a sequence of arguments in $Args_C$, constructed in steps as follows:

- In the first step add in A_L one argument for p_i.
- In each next step, for each distinct literal q_j labeling a leaf node of the proof trees of the arguments added in the previous step, add one argument with conclusion q_j; the addition should not violate the following restriction.
- An argument B with conclusion q_j can be added in A_L only if A_L does not already contain a different argument D with conclusion q_j.

The argument for p_i added in the first step is called the *head argument* of A_L. If the number of steps required to build A_L is finite, then A_L is a finite argumentation line. Infinite argumentation lines imply loops in the global knowledge base. Arguments contained in infinite lines participate in *attacks* against counter-arguments but may not be used to support the conclusion of their argumentation lines.

The notion of *supported* argument is meant to indicate when an argument may have an active role in proving or preventing the derivation of a conclusion. An argument A is *supported* by a set of arguments S if: (a) every proper subargument of A is in S; and (b) there is a finite argumentation line A_L with head A, such that every argument in $A_L - \{A\}$ is in S.

A local defeasible or mapping argument A is *undercut* by a set of arguments S if for every argumentation line A_L with head A, there is an argument B, such that B is supported by S, and B defeats a proper subargument of A or an argument in $A_L - \{A\}$. That an argument A is *undercut* by a set of arguments S means that we can show that some premises of A cannot be proved if we accept the arguments in S.

An argument A is *acceptable* w.r.t a set of arguments S if:

1) A is a strict local argument; or
2) A is supported by S and every argument defeating A is undercut by S

Intuitively, that an argument A is *acceptable* w.r.t. S means that if we accept the arguments in S as valid arguments, then we feel compelled to accept A as valid. Based on the concept of acceptable arguments, we define *justified arguments* and *justified literals*. J_i^C is defined as follows:

- $J_0^C = \emptyset$;
- $J_{i+1}^C = \{A \in Args_C \mid A \text{ is acceptable w.r.t. } J_i^C\}$

The set of *justified arguments* in a MCS C is $JArgs^C = \bigcup_{i=1}^{\infty} J_i^C$. A literal p_i is *justified* if it is the conclusion of an argument in $JArgs^C$. That an argument A is justified means that it resists every reasonable refutation. That a literal p_i is justified actually means that it is a logical consequence of C.

Finally, we also introduce the notion of *rejected arguments* and *rejected literals* for the characterization of conclusions that are not derived by C. An argument A is *rejected* by sets of arguments S, T when:

1) A is not a strict local argument, and either
2) a) a proper subargument of A is in S; or
   b) A is defeated by an argument supported by T; or
   c) for every argumentation line A_L with head A there exists an argument $A' \in A_L - \{A\}$, such that either a subargument of A' is in S; or A' is defeated by an argument supported by T

That an argument is rejected by sets of arguments S and T means that either it is supported by arguments in S, which can be thought of as the set of already rejected arguments, or it cannot overcome an attack from an argument supported by T, which can be thought of as the set of justified arguments. Based on the definition of rejected arguments, we define R_i^C as follows:

- $R_0^C = \emptyset$;
- $R_{i+1}^C = \{A \in Args_C \mid A \text{ is rejected by } R_i^C,\ JArgs^C\}$

The set of *rejected arguments* in a MCS C is $RArgs^C = \bigcup_{i=1}^{\infty} R_i^C$. A literal p_i is *rejected* if there is no argument in $Args^C - RArgs^C$ with conclusion p_i. That a literal is rejected means that we are able to prove that it is not a logical consequence of C.

Lemmata 1-3 describe some formal properties of the framework. Their proofs are presented in the Appendix. Lemma 1 refers to the monotonicity in J_i^C and $R_i^C(T)$.

Lemma 1: The sequences of sets of arguments J_i^C *and* $R_i^C(T)$ *are monotonically increasing.*

Lemma 2 represents the fact that no argument is both "believed" and "disbelieved".

Lemma 2: In a defeasible Multi-Context System C*:*

- *No argument is both justified and rejected.*
- *No literal is both justified and rejected.*

If we assume consistency in the local strict rules of a context (two complementary conclusions may not be derived as strict local consequences of a context theory), then using the previous Lemma, it is easy to prove that the entire framework is consistent; this is described in the following Lemma.

Lemma 3: If the set of justified arguments in a MCS C*,* $JArgs^C$*, contains two arguments with complementary conclusions, then both arguments are strict local arguments.*

Example (continued). The arguments that are derived from the MCS C of the example described in the previous section are the arguments depicted in Figure 2 and their subarguments.

A_1, B_1 and D_1 are in $Args_{C_1}$; the first two of them are mapping arguments, while D_1 is a strict local argument of C_1. B_2, B_3 and A_5 are strict local arguments of C_2, C_3 and C_5, respectively, while A_4 is a mapping argument of C_4.

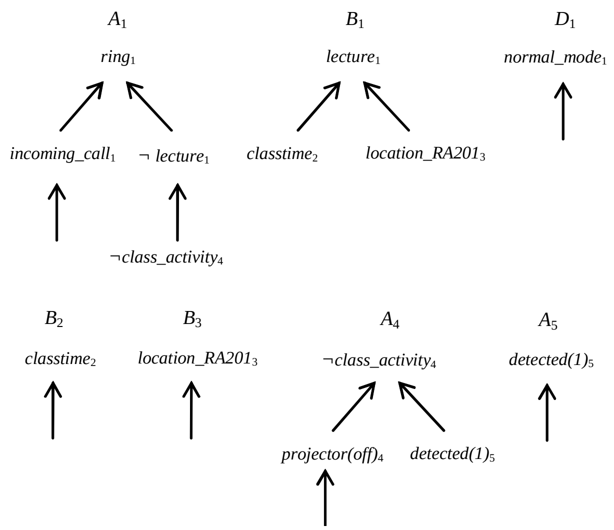

Fig. 2. Arguments in the Ambient Intelligence Scenario

J_0^C contains no arguments, while J_1^C contains the strict local arguments of the system; namely A'_1, D_1, B_2, B_3, A'_4 and A_5, where A'_1 is the subargument of A_1 with conclusion $incoming\_call_1$, and A'_4 is the subargument of A_4 with conclusion $projector(off)_4$.

J_2^C additionally contains A_4, since A_4 and A_5 form an argumentation line (A_{L4}) with head A_4, both A_5 and A'_4 (the only proper subargument of A_4) are in J_1^C, and there is no argument that attacks A_4.

J_3^C additionally contains A''_1 (the subargument of A_1 with conclusion $\neg lecture_1$), since A_5, A_4 and A''_1 form an argumentation line with head A''_1, A_4 and A_5 are both in J_2^C, and the only argument attacking A''_1, B_1, does not defeat A''_1, as according to $T_1 = [C_4, C_3, C_2, C_5]$, $R(A''_1, C_1) = 1 < R(B_1, C_1) = 3$.

Finally, J_4^C additionally contains A_1, as all proper subarguments of A_1 (A'_1 and A''_1) are in J_3^C, there is an argumentation line with head A_1, which consists of arguments A_1, A_4 and A_5, and both A_4 and A_5 are in J_3^C, while the only attacking argument, B_1, does not defeat A_1.

J_4^C actually constitutes the set of justified arguments in C ($JArgs^C = J_4^C$), as there is no other argument that can be added in the next steps of J_i^C. Hence, all the literals defined in the system except $lecture_1$ and $silent\_mode_1$ are justified.

On the other hand, $R_0^C(JArgs^C)$ contains no arguments, while $R_1^C(JArgs^C)$, which equals $RArgs^C(JArgs^C)$ contains only one argument, B_1, as B_1 is defeated by A_1, which is in $JArgs^C$. Hence, the conclusion of B_1, $lecture_1$, is rejected, since there is no other argument with this conclusion. $silent\_mode_1$ is also rejected, since it is not the conclusion of any argument is $Args_C$.

VI. Distributed Query Evaluation

$P2P\_DR$ is a distributed algorithm for query evaluation that implements the proposed argumentation framework. The specific problem that it deals with is: *Given a MCS C, and a query about literal p_i issued to context C_i, compute the truth value of p_i*. For an arbitrary literal p_i, $P2P\_DR$ returns one of the values: (a) *true*; indicating that p_i is justified in C; (b) *false*; indicating that p_i is rejected in C; or (c) *undefined*; indicating that p_i is neither justified nor rejected in C.

A. Algorithm Description

$P2P\_DR$ proceeds in four main steps. In the first step (lines 1-8 in the pseudocode given below), $P2P\_DR$ determines whether p_i, or its negation $\sim p_i$ are consequences of the local strict rules of C_i, using $local\_alg$, a local reasoning algorithm, which is described later in this section. If $local\_alg$ computes *true* as an answer for p_i or $\sim p_i$, $P2P\_DR$ returns *true / false* respectively as an answer for p_i and terminates.

In the second step (lines 9-12), $P2P\_DR$ calls $Support$ (described later in this section) to determine whether there are *applicable* and *unblocked* rules with head p_i. We call *applicable* those rules that for all literals in their body $P2P\_DR$ has computed *true* as their truth value, while *unblocked* are the rules that for all literals in their body $P2P\_DR$ has computed either *true* or *undefined* as their truth value. $Support$ also returns two data structures for p_i: (a) the set of foreign literals used in the most preferred (according to T_i) chain of applicable rules for p_i (SS_{p_i}); and (b) the set of foreign literals used in the most preferred chain of unblocked rules for p_i (BS_{p_i}). If there is no unblocked rule for p_i, the algorithm returns *false* as an answer and terminates.

In the third step (lines 13-14), $P2P\_DR$ calls $Support$ to compute the respective constructs for $\sim p_i$.

In the last step (lines 15-24), $P2P\_DR$ uses the constructs computed in the previous steps and the preference order defined by C_i (T_i), to determine the truth value of p_i. In case there is no unblocked rule for $\sim p_i$ ($unb_{\sim p_i} = false$), or SS_{p_i} is computed by $Stronger$ (described later in this section) to be *stronger* than $BS_{\sim p_i}$, $P2P\_DR$ returns *true* as an answer for p_i. That SS_{p_i} is *stronger* than $BS_{\sim p_i}$ means that the chains of applicable rules for p_i involve information from contexts that are preferred by C_i to the contexts that are involved in the chain of unblocked rules for $\sim p_i$. In case there is at least one applicable rule for $\sim p_i$, and BS_{p_i} is *not stronger* than $SS_{\sim p_i}$, $P2P\_DR$ returns *false* as an answer for p_i. In any other case, the algorithm returns *undefined*.

The context that is called to evaluate the query for p_i (C_i) returns through Ans_{p_i} the truth value for p_i. SS_{p_i} and BS_{p_i} are returned to the querying context (C_0) only if the two contexts (C_0 and C_i) are actually the same context. Otherwise, the empty set is assigned to both SS_{p_i} and BS_{p_i} and returned to C_0. In this way, the size of the messages exchanged between different contexts is kept small. $Hist_{p_i}$ is a structure used by $Support$ to detect loops in the global knowledge base. The algorithm parameters are:

- p_i: the queried literal (input)
- C_0: the context that issues the query (input)
- C_i: the context that defines p_i (input)
- $Hist_{p_i}$: the list of pending queries ($[p_1, ..., p_i]$) (input)
- T_i: preference order of C_i (input)
- SS_{p_i}: a set of foreign literals of C_i denoting the Supportive Set of p_i (output)
- BS_{p_i}: a set of foreign literals of C_i denoting the Blocking Set of p_i (output)

- Ans_{p_i}: the answer returned for p_i (output)

P2P_DR$(p_i, C_0, C_i, Hist_{p_i}, T_i, SS_{p_i}, BS_{p_i}, Ans_{p_i})$

```
call local_alg(p_i, localAns_{p_i})
if localAns_{p_i} = true then
    Ans_{p_i} ← true, SS_{p_i} ← ∅, BS_{p_i} ← ∅
    terminate
call local_alg(∼ p_i, localAns_{∼p_i})
if localAns_{∼p_i} = true then
    Ans_{p_i} ← false, SS_{p_i} ← ∅, BS_{p_i} ← ∅
    terminate
call Support(p_i, Hist_{p_i}, T_i, sup_{p_i}, unb_{p_i}, SS_{p_i}, BS_{p_i})
if unb_{p_i} = false then
    Ans_{p_i} ← false, SS_{p_i} ← ∅, BS_{p_i} ← ∅
    terminate
Hist_{∼p_i} ← (Hist_{p_i} − {p_i}) ∪ {∼ p_i}
call
Support(∼ p_i, Hist_{∼p_i}, T_i, sup_{∼p_i}, unb_{∼p_i}, SS_{∼p_i}, BS_{∼p_i})
if sup_{p_i} = true and (unb_{∼p_i} = false
or Stronger(SS_{p_i}, BS_{∼p_i}, T_i) = SS_{p_i}) then
    Ans_{p_i} ← true
    if C_0 ≠ C_i then
        SS_{p_i} ← ∅, BS_{p_i} ← ∅
else if sup_{∼p_i} = true and
Stronger(BS_{p_i}, SS_{∼p_i}, T_i) ≠ BS_{p_i} then
    Ans_{p_i} ← false, SS_{p_i} ← ∅, BS_{p_i} ← ∅
else
    Ans_{p_i} ← undefined
    if C_0 ≠ C_i then
        SS_{p_i} ← ∅, BS_{p_i} ← ∅
```

$local\_alg$ is called by $P2P\_DR$ to determine whether the truth value of the queried literal can be derived from the local strict rules of a context theory (R^s).

local_alg$(p_i, localAns_{p_i})$

```
for all r_i ∈ R^s[p_i] do
    for all b_i ∈ body(r_i) do
        call local_alg(b_i, localAns_{b_i})
    if for all b_i: localAns_{b_i} = true then
        return localAns_{p_i} = true and terminate
return localAns_{p_i} = false
```

$Support$ is called by $P2P\_DR$ to determine whether there are applicable and unblocked rules for p_i. In case there is at least one applicable rule for p_i, $Support$ returns $sup_{p_i} = true$; otherwise, it returns $sup_{p_i} = false$. Similarly, $unb_{p_i} = true$ is returned when there is at least one unblocked rule for p_i; otherwise, $unb_{p_i} = false$.

$Support$ also returns two data structures for p_i: SS_{p_i}, the Supportive Set for p_i; and BS_{p_i}, the Blocking Set for p_i. To compute these structures, it checks the applicability of the rules with head p_i, using the truth values of the literals in their body, as these are evaluated by $P2P\_DR$. To avoid loops, before calling $P2P\_DR$, it checks if the same query has been issued before during the running call of $P2P\_DR$. For each applicable rule r_i, $Support$ builds its Supportive Set, SS_{r_i}; this is the union of the set of *foreign literals* contained in the body of r_i with the Supportive Sets of the local literals contained in the body of the rule. Similarly, for each unblocked rule r_i, it computes its Blocking Set BS_{r_i} using the Blocking Sets of its body literals. $Support$ computes the Supportive Set of p_i, SS_{p_i}, as the *strongest* rule Supportive Set SS_{r_i}; and its Blocking Set, BS_{p_i}, as the *strongest* rule Blocking Set BS_{r_i}, using the $Stronger$ function. The parameters of $Support$ are:

- p_i: the queried literal (input)
- $Hist_{p_i}$: the list of pending queries ($[p_1, ..., p_i]$) (input)
- T_i: the preference ordering of C_i (input)
- sup_{p_i}, indicating whether p_i is supported in C (output)
- unb_{p_i}, indicating whether p_i is unblocked in C (output)
- SS_{p_i}: the Supportive Set of p_i (output)
- BS_{p_i}: the Blocking Set of p_i (output)

Support$(p_i, Hist_{p_i}, T_i, sup_{p_i}, unb_{p_i}, SS_{p_i}, BS_{p_i})$

```
sup_{p_i} ← false
unb_{p_i} ← false
for all r_i ∈ R[p_i] do
    cycle(r_i) ← false
    SS_{r_i} ← ∅
    BS_{r_i} ← ∅
    for all b_t ∈ body(r_i) do
        if b_t ∈ Hist_{p_i} then
            cycle(r_i) ← true
            BS_{r_i} ← BS_{r_i} ∪ {d_t} {where d_t is literal b_t if b_t ∉ V_i;
            otherwise d_t is the first foreign literal of C_i added in
            Hist_{p_i} after b_t}
        else
            Hist_{b_t} ← Hist_{p_i} ∪ {b_t}
            call
            P2P_DR(b_t, C_i, C_t, Hist_{b_t}, T_t, SS_{b_t}, BS_{b_t}, Ans_{b_t})
            if Ans_{b_t} = false then
                stop and check the next rule
            else if Ans_{b_t} = undefined or cycle(r_i) = true then
                cycle(r_i) ← true
                if b_t ∉ V_i then
                    BS_{r_i} ← BS_{r_i} ∪ {b_t}
                else
                    BS_{r_i} ← BS_{r_i} ∪ BS_{b_t}
            else
                if b_t ∉ V_i then
                    BS_{r_i} ← BS_{r_i} ∪ {b_t}
                    SS_{r_i} ← SS_{r_i} ∪ {b_t}
                else
                    BS_{r_i} ← BS_{r_i} ∪ BS_{b_t}
                    SS_{r_i} ← SS_{r_i} ∪ SS_{b_t}
    if unb_{p_i} = false or Stronger(BS_{r_i}, BS_{p_i}, T_i) = BS_{r_i}
    then
        BS_{p_i} ← BS_{r_i}
    unb_{p_i} ← true
    if cycle(r_i) = false then
        if sup_{p_i} = false or Stronger(SS_{r_i}, SS_{p_i}, T_i) = SS_{r_i}
        then
            SS_{p_i} ← SS_{r_i}
            sup_{p_i} ← true
```

$Stronger(A, B, T_i)$ returns the *strongest* between two sets of literals, A and B, according to preference order T_i. A literal a_k (defined in C_k) is *preferred* to literal b_j (defined in C_l), if C_k precedes C_l in T_i. The strength of a set is determined by the least preferred literal in this set.

Stronger(A, B, T_i)

```
if ∃b_l ∈ B: ∀a_k ∈ A: C_k has lower rank than C_l in T_i
or (A = ∅ and B ≠ ∅) then
    Stronger = A
else if ∃a_k ∈ A: ∀b_l ∈ B: C_l has lower rank than C_k in T_i
or (B = ∅ and A ≠ ∅) then
    Stronger = B
else
    Stronger = None
```

Example (continued) Given a query about $ring_1$, $P2P\_DR$ proceeds as follows. It fails to compute an answer based on C_1's local theory, and uses rules $r^m_{1,5}$ and $r^m_{1,6}$ to compute an answer for $\neg lecture_1$. Using the local rules of C_2, C_3, C_4 and C_5, it computes positive answers for $classtime_2$, $location\_RA201_3$ and $\neg class\_activity_4$ respectively, determines that both $r^m_{1,5}$ and $r^m_{1,6}$ are applicable, and computes their Supportive Sets: $SS_{r^m_{1,5}} = \{classtime_2, location\_RA201_3\}$ and $SS_{r^m_{1,6}} = \{\neg class\_activity_4\}$. As C_4 precedes C_2 in T_1, $P2P\_DR$ determines that $SS_{r^m_{1,6}}$ is stronger, computes a positive answer for $\neg lecture_1$, and eventually (using rule $r^d_{1,3}$) returns *true* as an answer for $ring_1$.

B. Properties of the Algorithm

Below, we describe some formal properties of $P2P\_DR$ regarding its termination, soundness and completeness w.r.t. the argumentation framework, complexity and the ability to create an equivalent defeasible theory from the distributed context theories. The proofs for Propositions 2, 3, which refer to soundness and completeness, are presented in the Appendix. The rest of the proofs are available at www.csd.uoc.gr/∼bikakis/thesis.pdf.. Proposition 1 is a consequence of the cycle detection process within the algorithm.

Proposition 1: The algorithm is guaranteed to terminate returning one of the values true, false and undefined as an answer for the queried literal.

Propositions 2 and 3 associate the results computed by $local\_alg$ and $P2P\_DR$ with concepts of the argumentation framework.

Proposition 2: For a Multi-Context System C *and a literal* p_i *in* $C_i \in C$, *local_alg returns*

1) $localAns_{p_i} = true$ *iff there is a strict local argument for* p_i *in* $JArgs^C$
2) $localAns_{p_i} = false$ *iff there is no strict local argument for* p_i *in* $JArgs^C$

Proposition 3: For a Multi-Context System C *and a literal* p_i *in* C, $P2P\_DR$ *returns:*

1) $Ans_{p_i} = true$ *iff* p_i *is justified in* C
2) $Ans_{p_i} = false$ *iff* p_i *is rejected in* C
3) $Ans_{p_i} = undefined$ *iff* p_i *is neither justified nor rejected in* C

Propositions 4 and 5 are consequences of two structures that $P2P\_DR$ retains for each context, which keep track of the incoming and outgoing queries of the context. The worst case that both propositions refer to is when all rules of C_i contain either p_i (the queried literal) or $\sim p_i$ in their head and all system literals in their bodies.

Proposition 4: The total number of messages exchanged between the system contexts for the evaluation of a query is, in the worst case, $O(n \times \sum P(n,k))$, *where* n *stands for the total number of literals in the system,* $\sum$ *expresses the sum over* $k = 0, 1, ..., n$, *and* $P(n,k)$ *stands for the number of permutations with length* k *of* n *elements. In case, there are no loops in the global knowledge base, the number of messages is polynomial to the size of the global knowledge base.*

Proposition 5: The number of operations imposed by one call of $P2P\_DR$ *for the evaluation of a query for literal* p_i *is, in the worst case, proportional to the number of rules in* C_i*, and to the total number of literals in the system.*

Proposition 6 describes the ability to create an equivalent global defeasible theory from the distributed context theories in case there are no loops in the global knowledge base. In this theory, local strict rules of the system contexts are modeled as strict rules, local defeasible and mapping rules are modeled as defeasible rules, and preference orderings are used to derive priorities between competing rules. A complete description of the procedure is described in [27]. This property enables resorting to centralized reasoning by collecting all the available information in a central entity. In addition, this result is typical of other works in the area of Peer-to-Peer reasoning, in which the distributed query evaluation algorithm is related to querying a single knowledge base that can be constructed (see, e.g. [28]).

Proposition 6: There is a standard process that takes as input the distributed context theories and their preference orderings and creates a global unified theory of Defeasible Logic, which in case there are no loops in the global theory, produces equivalent results with $P2P\_DR$*, under the proof theory of [23].*

VII. Conclusion

The challenges of reasoning with the available context knowledge in Ambient Intelligence environments have not yet been successfully addressed by the existing Ambient Intelligence systems. Most of them are either ad-hoc or make simplifying assumptions: perfect knowledge of context, centralized context, and unbounded computational and communicating capabilities. The requirements, though, are much different in such environments. The uncertainty of context and its distribution to heterogeneous devices with restricted capabilities, impose the need for different reasoning approaches.

This paper proposes a totally distributed approach for reasoning in Ambient Intelligence environments based on the representation of context knowledge shared by ambient agents as context theories in Multi-Context Systems. Using a total preference ordering on the system contexts, our approach enables resolving all potential conflicts that may arise from the interaction of contexts through their mappings. The paper also provides an argumentation-based semantic characterization of the model, and a distributed algorithm for query evaluation that implements the argumentation framework.

Our ongoing work involves: (a) defining *average case* scenarios and studying the performance of the query evaluation algorithm in such cases; (b) studying alternative methods for conflict resolution, which differ in the way that agents evaluate the imported context information; (c) studying the relation between our reasoning model and loop checking variants of Defeasible Logic, such as those described by Nute in [29], [30]; and (d) implementing real-world applications of our approach in Ambient Intelligence environments, with contexts lying on a variety of stationary and mobile devices (such as PDAs or cell phones), and communicating through wireless communications. We have already described some possible application scenarios of our approach in [6].

REFERENCES

[1] M. Weiser, "The computer for the 21st century," *Scientific American*, vol. 265, no. 3, pp. 94–104, September 1991.
[2] ISTAG, "Ambient Intelligence: from vision to reality," Information Society Technologies Advisory Group (ISTAG), Tech. Rep., September 2003.
[3] K. Henricksen and J. Indulska, "Modelling and Using Imperfect Context Information," in *Proceedings of PERCOMW '04*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 33–37.
[4] A. Bikakis, T. Patkos, G. Antoniou, and D. Plexousakis, "A Survey of Semantics-based Approaches for Context Reasoning in Ambient Intelligence," in *Constructing Ambient Intelligence*, ser. Communications in Computer and Information Science, M. Mühlhäuser, A. Ferscha, and E. Aitenbichler, Eds. Springer, 2008, pp. 14–23.
[5] A. Bikakis and G. Antoniou, "Distributed Reasoning with Conflicts in a Multi-Context Framework," in *AAAI*, D. Fox and C. P. Gomes, Eds. AAAI Press, 2008, pp. 1778–1779.
[6] A. Bikakis and G. Antoniou, "Distributed Defeasible Contextual Reasoning in Ambient Computing," in *AmI*, 2008, pp. 308–325.
[7] P. McBurney and S. Parsons, *Argumentaion in Artificial Intelligence*. Springer US, 2009, ch. Dialogue Games for Agent Argumentation, pp. 261–280.
[8] A. Bikakis, G. Antoniou, and P. Hassapis, "Alternative Strategies for Conflict Resolution in Multi-Context Systems," in *Proceedings of the 5th IFIP Conference on Artificial Intelligence Applications and Innovations (AIAI'2009)*, ser. IFIP Advances in Information and Communication Technology, vol. 296. Springer, 2009.
[9] J. McCarthy, "Generality in Artificial Intelligence," *Communications of the ACM*, vol. 30, no. 12, pp. 1030–1035, 1987.
[10] S. Buvac and I. A. Mason, "Propositional Logic of Context." in *AAAI*, 1993, pp. 412–419.
[11] J. McCarthy and S. Buvač, "Formalizing Context (Expanded Notes)," in *Computing Natural Language*, A. Aliseda, R. van Glabbeek, and D. Westerståhl, Eds. Stanford, California: CSLI Publications, 1998, pp. 13–50.
[12] F. Giunchiglia and L. Serafini, "Multilanguage hierarchical logics, or: how we can do without modal logics," *Artificial Intelligence*, vol. 65, no. 1, 1994.
[13] C. Ghidini and F. Giunchiglia, "Local Models Semantics, or contextual reasoning=locality+compatibility," *Artificial Intelligence*, vol. 127, no. 2, pp. 221–259, 2001.
[14] L. Serafini and P. Bouquet, "Comparing formal theories of context in AI," *Artificial Intelligence*, vol. 155, no. 1-2, pp. 41–67, 2004.
[15] F. Roelofsen and L. Serafini, "Minimal and Absent Information in Contexts," in *IJCAI*, 2005, pp. 558–563.
[16] G. Brewka, F. Roelofsen, and L. Serafini, "Contextual Default Reasoning," in *IJCAI*, 2007, pp. 268–273.
[17] J. Sabater, C. Sierra, S. Parsons, and N. R. Jennings, "Engineering Executable Agents using Multi-context Systems," *Journal of Logic and Computation*, vol. 12, no. 3, pp. 413–442, 2002.
[18] A. Casali, L. Godo, and C. Sierra, "A Logical Framework to Represent and Reason about Graded Preferences and Intentions," in *KR*, 2008, pp. 27–37.
[19] M. Dastani, G. Governatori, A. Rotolo, I. Song, and L. van der Torre, "Contextual Deliberation of Cognitive Agents in Defeasible Logic," in *AAMAS*, 2007, p. 148.
[20] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati, "Inconsistency Tolerance in P2P Data Integration: an Epistemic Logic Approach," in *DBPL-05*, ser. LNCS, vol. 3774. SV, 2005, pp. 90–105.
[21] P. Chatalic, G. H. Nguyen, and M.-C. Rousset, "Reasoning with Inconsistencies in Propositional Peer-to-Peer Inference Systems," in *ECAI*, 2006, pp. 352–356.
[22] A. Binas and S. A. McIlraith, "Peer-to-Peer Query Answering with Inconsistent Knowledge," in *KR*, 2008, pp. 329–339.
[23] G. Antoniou, D. Billington, G. Governatori, and M. J. Maher, "Representation results for defeasible logic," *ACM Trans. Comput. Logic*, vol. 2, no. 2, pp. 255–287, 2001.
[24] G. Governatori, M. J. Maher, G. Antoniou, and D. Billington, "Argumentation Semantics for Defeasible Logic," *J. Log. and Comput.*, vol. 14, no. 5, pp. 675–702, 2004.
[25] P. M. Dung, "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games," *Artificial Intelligence*, vol. 77, pp. 321–357, 1995.
[26] G. Antoniou, M. J. Maher, and D. Billington, "Defeasible Logic versus Logic Programming without Negation as Failure," *Journal of Logic Programming*, vol. 42, no. 1, pp. 47–57, 2000.
[27] A. Bikakis and G. Antoniou, "Local and Distributed Defeasible Reasoning in Multi-Context Systems," in *RuleML*, 2008, pp. 135–149.
[28] P. Adjiman, P. Chatalic, F. Goasdoue', M.-C. Rousset, and L. Simon, "Distributed Reasoning in a Peer-to-Peer Setting: Application to the Semantic Web," *Journal of Artificial Intelligence Research*, vol. 25, pp. 269–314, 2006.
[29] D. Nute, *Defeasible Deontic Logic*. Kluwer, 1997, ch. Apparent Obligation, pp. 288–315.
[30] D. Nute, "Defeasible Logic," in *Proceedings of the 14th International Conference on Applications of Prolog*, 2001, pp. 87–114.

APPENDIX

Lemma 1. The sequences of sets of arguments J_i^C and $R_i^C(T)$ are monotonically increasing.

Proof. We prove the Lemma by induction on i. The inductive base is trivial in both cases since $J_0^C = \emptyset$ and $R_0^C(T) = \emptyset$ and thus $J_0^C \subseteq J_1^C$ and $R_0^C(T) \subseteq R_1^C(T)$.

By definition strict local arguments are acceptable w.r.t. every set of arguments; thus they are in every J_i^C.

Let A be an argument in J_n^C and let B be an argument defeating A. By definition, B is undercut by J_{n-1}^C; namely for every argumentation line B_L with head B, there is a literal q and an argument D, such that D is supported by J_{n-1}^C and D defeats a proper subargument of B or an argument in $B_L - \{B\}$ at q. By inductive hypothesis $J_{n-1}^C \subseteq J_n^C$; hence D is also supported by J_n^C. Consequently, B is undercut by J_n^C. Since A is an argument in J_n^C, by definition A is supported by J_{n-1}^C, and by inductive hypothesis, A is also supported by J_n^C. Therefore A is acceptable w.r.t. J_n^C, and $A \in J_{n+1}^C$.

We consider now the sequence of rejected arguments. Let A be an argument is $R_n^C(T)$. By definition, A is not a strict local argument and one of the three following conditions hold: (a) A proper subargument of A, A' is in $R_{n-1}^C(T)$. By inductive hypothesis $R_{n-1}^C(T) \subseteq R_n^C(T)$; hence $A' \in R_n^C(T)$ and $A \in R_{n+1}^C(T)$; (b) for every argumentation line A_L with head A, a subargument A' of an argument in $A_L - \{A\}$ is in $R_{n-1}^C(T)$, and by inductive hypothesis $A' \in R_n^C(T) \Rightarrow A \in R_{n+1}^C(T)$; or (c) a proper subargument of A or an argument in $A_L - \{A\}$ is defeated by an argument supported by T. In this case $A \in R_i^C(T)$ for every i, and therefore $A \in R_{n+1}^C(T)$.

Lemma 2. In a Multi-Context System C:

1. No argument is both justified and rejected.
2. No literal is both justified and rejected.

Proof Suppose that there is an argument that is both justified and rejected. Let *n* be the smallest index such that for some argument A, $A \in RArgs^C(JArgs^C)$ and $A \in J_n^C$. Since $A \in J_n^C$, it holds that either (a) A is a strict local argument; or (b) A is supported by J_{n-1}^C and every argument defeating A is undercut by J_{n-1}^C. Since $A \in RArgs^C(JArgs^C)$, (a) does not hold. Hence, there is an argumentation line A'_L with head A such that for every subargument of A or argument in $A'_L - \{A\}$, A', it holds that $A' \in J_{n-1}^C$, and by Lemma 1 $A' \in J_n^C$. By definition, every argument defeating A' is undercut by J_{n-1}^C.

Since $A \in RArgs^C(JArgs^C)$, it holds by definition that for every argumentation line A_L with head A either (c) there exists an argument B that is supported by $JArgs^C$ and defeats a subargument of A or an argument in $A_L - \{A\}$; or (d) a subargument of A or an argument in $A_L - \{A\}$ is in $RArgs^C(JArgs^C)$. However, we have already proved that $A' \in J^C_{n-1}$, and by supposition n is the smallest index such that for some argument A, $A \in RArgs^C(JArgs^C)$ and $A \in J^C_n$; therefore (d) does not hold.

By (b) and (c), there exists an argument B', such that B' defeats A', and is supported by $JArgs^C$ and undercut by J^C_{n-1}. Hence, for every argumentation line B_L with head B' there is an argument D that is supported by J^C_{n-1} and defeats an argument in $B_L - \{B'\}$ or a proper subargument of B'. By definition of supported, there is an argumentation line B'_L with head B' such that every argument defeating an argument in $B'_L - \{B\}$ or a proper subargument of B' is undercut by $JArgs^C$. Hence D is undercut by $JArgs^C$; namely, for every argumentation line D_L with head D there is an argument E that is supported by $JArgs^C$ and defeats an argument in $D_L - \{D\}$ or a proper subargument of D. Since D is supported by J^C_{n-1}, there is an argumentation line D'_L with head D s.t. for every subargument of D or argument in $D'_L - \{D\}$, D', $D' \in J^C_{n-1}$. However, since D is undercut by $JArgs^C$, D' is defeated by an argument E' supported by $JArgs^C$; therefore $D' \in RArgs^C(JArgs^C)$ and $D' \in J^C_{n-1}$, which contradicts the assumed minimality of n. Hence the original supposition is false, and no argument is both justified and rejected.

The second part follows easily from the first: if the literal p is justified, then there is an argument A for p in $JArgs^C$. From the first part, $A \in Args^C - RArgs^C(JArgs^C)$. Thus if p is justified then it is not rejected.

Lemma 3. If the set of justified arguments of C, $JArgs^C$ contains two arguments with conflicting conclusions, then both are strict local arguments.

Proof. Let the two arguments be A and B. Suppose B is a strict local argument. Then, for A to be acceptable with respect to every S, A must also be a strict local argument (otherwise B would defeat A, and B cannot be undercut by S). Thus, by symmetry, either A and B are both strict local arguments, or they are both defeasible local or mapping arguments. Suppose that both are defeasible local or mapping arguments and B defeats A. Then A must be rejected because it is defeated by an argument supported by $JArgs_C$, and is justified by assumption. By Lemma 2, this is not possible. Similarly, if we assume that A defeats B, we will conclude that B is both justified and rejected. Therefore, the two arguments are strict local arguments.

Proposition 2. For a MCS C and a literal p_i in $C_i \in C$, $local\_alg$ returns:

1. $localAns_{p_i} = true$ iff there is a strict local argument for p_i in $JArgs^C$
2. $localAns_{p_i} = false$ iff there is no strict local argument for p_i in $JArgs^C$

Proof (1, $\Rightarrow$). We use induction on the number of calls of $local\_alg$ that are required to produce the answer for p_i.

Inductive Base. Suppose that $local\_alg$ returns $localAns_{p_i} = true$ in one call. This means that there is a local strict rule with head p_i in C_i, r_i, such that $body(r_i) = \emptyset$. Using r_i we can build a strict local argument for p_i.

Inductive Step. Suppose that $n+1$ calls of $local\_alg$ are required to compute $localAns_{p_i} = true$. This means that there is a strict local rule with head p_i (say r_i) such that $\forall a_i \in body(r_i)$, $local\_alg$ returns $localAns_{p_i} = true$ in n or less calls. By inductive hypothesis, for every a_i there is a strict local argument for a_i in $Args_C$. Using the arguments for a_i and rule r_i we can build a strict local argument for p_i.

(1, $\Leftarrow$). We prove the left to right part of (1) using induction on the height of strict local arguments for p_i in $Args_C$.

Inductive Base. Suppose that there is a strict local argument for p_i in $Args_C$ (say A) with height 1. This means that there is a strict local rule with head p_i with empty body in C_i; hence $local\_alg$ will return $localAns_{p_i} = true$.

Inductive Step. Suppose that A is a strict local argument for p_i with height $n+1$ in $Args_C$. Then, there is a strict local rule with head p_i (r_i) in C_i, such that for every literal a_i in its body there is a strict local argument with height $\leq n$ in $Args_C$. By inductive hypothesis, $local\_alg$ returns $localAns_{a_i} = true$ for every $a_i \in body(r_i)$. Consequently $local\_alg$ will return $localAns_{p_i} = true$.

(2, $\Rightarrow$). By the definition of $local\_alg$ it is trivial to verify that $local\_alg$ cannot return both $true$ and $false$ as an answer for a literal p_i. Suppose that $local\_alg$ returns $localAns_{p_i} = false$. Suppose that there is a strict local argument for p_i in $Args_C$. Then (by the first part of the Proposition) $localAns_{p_i} = true$, which contradicts our original hypothesis. Consequently there is no strict local argument for p_i in $Args_C$.

(2, $\Leftarrow$). Similarly (for the right to left part) we suppose that there is no strict local argument for p_i in $Args_C$. Supposing that $local\_alg$ returns $localAns_{p_i} = true$, we conclude (by the first part of the Proposition) that there is a strict local argument for p_i in $Args_C$, which contradicts our original hypothesis.

Auxiliary Lemma. For a MCS C and a literal p_i in C:

1. If $P2P\_DR$ returns $Ans_{p_i} = true$ and $SS_{p_i} = \Sigma$, then there is an argument A for p_i in $Args_C$, such that A uses applicable rules, and $R(A, C_i)$ equals 0 in case $\Sigma = \emptyset$, or $max_{a \in \Sigma}(R(a, C_i))$ otherwise, and for any other argument B for p_i in $Args_C$, such that B uses applicable rules: $R(A, C_i) \leq R(B, C_i)$.
2. If $P2P\_DR$ returns $Ans_{p_i} = true$ or $Ans_{p_i} = undefined$ and $BS_{p_i} = \Sigma$, then there is an argument A for p_i in $Args_C$, such that A uses unblocked rules, and $R(A, C_i)$ equals 0 in case $\Sigma = \emptyset$, or $max_{a \in \Sigma}(R(a, C_i))$ otherwise, and for any other argument B for p_i in $Args_C$, such that B uses unblocked rules: $R(A, C_i) \leq R(B, C_i)$.

Proof (1). We use induction on the number of calls of $P2P\_DR$ that are required to compute Ans_{p_i} and SS_{p_i}.

Inductive Base. $Ans_{p_i} = true$ derives in one call of $P2P\_DR$. This means that either (a) $localAns_{p_i} = true$ and $SS_{p_i} = \emptyset$, and by Proposition 2, there is a strict local argument A for p_i in $Args_C$. For all literals a in the body of the rules contained in A, $localAns_a = true$. Hence, A uses only applicable rules. Since A is a local argument, $R(A, C_i) = 0$. Hence, there is no argument B such that $R(B, C_i) < R(A, C_i)$; or (b) there is a local defeasible rule with empty body and head p_i in C_i. Using this rule, we can build an argument A for p_i such that $R(A, C_i) = 0$; therefore, there is no argument B such that $R(B, C_i) < R(A, C_i)$.

Inductive Step. $Ans_{p_i} = true$ and $SS_{p_i} = \Sigma$ derives in $n+1$ calls of $P2P\_DR$. This means that there is a rule r_i with head p_i in C_i, such that $\forall a \in body(r_i)$: $P2P\_DR$ returns $Ans_a = true$ and SS_a in at most n calls, and $\Sigma = SS_{r_i}$. By inductive hypothesis, for all a there is an argument A_a for a in $Args_C$ such that A_a uses applicable rules, and $R(A_a, C_j)$ equals 0 in case $SS_a = \emptyset$ or $max_{a' \in SS_a}(R(a', C_j))$ otherwise (where C_j is the context such that $a \in V_j$), and for any other argument B_a for a in $Args_C$ that uses applicable rules: $R(B_a, C_j) \geq R(A_a, C_j)$.

Using the arguments A_a and rule r_i we build an argument A for p_i as follows: The subset of the arguments A_a that support local literals of C_i (denoted as A_{a_i}) are used as proper subarguments of A, and r_i is used in A to support p_i, which labels the root of A. By the definition of rank of arguments:

$$R(A, C_i) = max(max_{A_{a_i}}(R(A_{a_i}, C_i)), max_{a_j}(R(a_j, C_i)))$$

where a_j are the literals in the body of r_i such that $a_j \notin V_i$. By inductive hypothesis:

$$R(A, C_i) =$$
$$max(max_{a' \in \bigcup SS_{a_i}}(R(a', C_j)), max_{a_j}(R(a_j, C_i)))$$
$$\Rightarrow R(A, C_i) = max_{d \in (\bigcup SS_{a_i}) \cup (\bigcup a_j)}(R(d, C_i))$$
$$\Rightarrow R(A, C_i) = max_{d \in SS_{r_i}}(R(d, C_i))$$

and for any other argument A' for p_i in $Args_C$ that uses rule r_i and applicable rules to support p_i, $R(A, C_i) \leq R(A', C_i)$. In case $\Sigma = SS_{r_i} = \emptyset$, which means that there is no foreign literal in the body of r_i, and for every $a_i \in body(r_i)$: $SS_{a_i} = \emptyset$, using inductive hypothesis it is easy to verify that $R(A, C_i) = 0$.

By the definition of $P2P\_DR$, it also holds that for any other rule t_i with head p_i in C_i, either (a) there is a literal γ in the body of t_i such that $P2P\_DR$ returns either $Ans_\gamma = undefined$ or $Ans_\gamma = false$ - in this case t_i is not applicable; or (b) $\forall \gamma \in body(t_i)$: $Ans_\gamma = true$ and $Stronger(\Sigma, SS_{t_i}, T_i) \neq SS_{t_i}$. The latter results are obtained in n or less calls of $P2P\_DR$. By inductive hypothesis, the argument for p_i that uses rule t_i and applicable rules to support p_i with the lowest rank w.r.t. C_i is F with rank: $R(F, C_i) = max_{f \in SS_{t_i}}(R(f, C_i))$, and by the definition of *Stronger* it holds that there is a literal f' in SS_{t_i} such that for all d in $\Sigma = SS_{r_i}$, $R(f', C_i) \geq R(d, C_i)$. Therefore $R(A, C_i) \leq R(F, C_i)$. Overall, the rank of A is equal or lower than the rank of any other argument in $Args_C$ that uses applicable rules to support p_i.

(2). We use induction on the number of calls of $P2P\_DR$ that are required to compute Ans_{p_i} and BS_{p_i}.

Inductive Base. As there are no loops in the local context theories, at least two calls of $P2P\_DR$ are required to return *undefined* as an answer for p_i. Hence, the Inductive Base for the case that $Ans_{p_i} = true$ is that this answer is returned by $P2P\_DR$ in one call. Similarly with the first part of the Lemma, we can prove that $BS_{p_i} = \emptyset$, and there is an argument $A \in Args_C$ for p_i, such that $R(A, C_i) = 0$, and A uses only applicable (and therefore unblocked) rules.

The Inductive Base for the case that $Ans_{p_i} = undefined$ and $BS_{p_i} = \Sigma$ is two calls of $P2P\_DR$. Since we assume that there are no loops in the local context theories, there are no rules such that the literal in their head also belongs to the body of the rule. Hence, the following conditions must hold: (a) $localAns_{p_i} = false$; by Proposition 2 this means that there is no strict local argument for p_i in $Args_C$; (b) there is no rule with head $\sim p_i$ in C_i; and (c) there is only one rule r_i with head p_i in C_i, with one literal in its body (say q_j), for which it holds (c_1) $q_j \notin V_i$; (c_2) there is no rule with head $\sim q_j$ in C_j; and (c_3) there is only one rule with head q_j (say r_j) in C_j, such that p_i is the only literal in the body of t_j. Hence, the only argument for p_i (A) can be obtained using rule r_i, and the only argument for q_j (A') can be obtained using rule r_j. Neither r_i nor r_j are blocked since there are no rules with contradictory conclusions, and $\Sigma = BS_{r_i} = \{q_j\}$. Therefore A uses unblocked rules, $R(A, C_i) = R(q_j, C_i) = max_{a \in \Sigma} R(a, C_i)$ and there is no other argument for p_i in $Args_C$.

Inductive Step. $Ans_{p_i} = true$ or $Ans_{p_i} = undefined$ and $BS_{p_i} = \Sigma$ derive in $n+1$ calls of $P2P\_DR$. This means that there is a rule r_i with head p_i in C_i, such that $\forall a \in body(r_i)$: $P2P\_DR$ returns either $Ans_a = true$ or $Ans_a = undefined$ and BS_a in at most n calls, and $\Sigma = BS_{r_i}$. By inductive hypothesis, for all a there is an argument A_a for a in $Args_C$ such that A_a uses unblocked rules, and $R(A_a, C_j)$ equals 0 in case $SS_a = \emptyset$ or $max_{a' \in SS_a}(R(a', C_j))$ otherwise, and for any other argument B_a for a in $Args_C$ that uses unblocked rules: $R(B_a, C_j) \geq R(A_a, C_j)$.

Similarly with the first part of the Lemma, using the arguments for a and rule r_i, we can build an argument A for p_i, such that A uses unblocked rules, and

$$\Rightarrow R(A, C_i) = max_{d \in BS_{r_i}}(R(d, C_i))$$

and for any other argument A' for p_i in $Args_C$ that uses unblocked rules to support p_i, $R(A, C_i) \leq R(A', C_i)$. In case $\Sigma = BS_{r_i} = \emptyset$, using inductive hypothesis it is easy to verify that $R(A, C_i) = 0$.

Proposition 3. For a MCS C and a literal p_i in C_i, $P2P\_DR$ returns:

1. $Ans_{p_i} = true$ iff p_i is justified
2. $Ans_{p_i} = false$ iff p_i is rejected by $JArgs^C$
3. $Ans_{p_i} = undefined$ iff p_i is neither justified nor rejected by $JArgs^C$

Proof. (⇒). We prove the left to right part of the proposition using induction on the calls of $P2P\_DR$.

Inductive Base. (1) $P2P\_DR$ returns $Ans_{p_i} = true$ in one call. This means that either (a) $localAns_{p_i} = true$ - then, by Proposition 2, there is a strict local argument A for p_i in $Args_C$. Hence, $A \in JArgs^C$ and p_i is justified; or (b) there is a local defeasible rule r_i in C_i such that $body(r_i) = \emptyset$ and there is no rule with head $\sim p_i$ in C_i. Therefore, there is an argument A for p_i in $Args_C$ with root p_i, which contains only rule r_i, and there is no argument attacking A. Since A has no proper subarguments and it is not attacked by any argument, $A \in JArgs^C$; therefore p_i is justified.

(2). $P2P\_DR$ returns $Ans_{p_i} = false$ in one call. This means that $localAns_{p_i} = false$ (by Proposition 2 this means that there is no strict local argument for p_i in $Args_C$) and either (a) $localAns_{\sim p_i} = true$ - there is a strict local argument B for $\sim p_i$ in $Args_C$, which by definition is supported by $JArgs^C$, it defeats any non-strict argument for p_i in $Args_C$, and is not undercut by $JArgs^C$, and therefore p_i is rejected by $JArgs^C$; or (b) there is a local defeasible rule s_i with head $\sim p_i$ in C_i, such that $body(s_i) = \emptyset$. Therefore, there is an argument B for $\sim p_i$ in $Args_C$, with root p_i, which contains only rule s_i. For B it holds that it has no proper subarguments - therefore it is supported and not undercut by $JArgs^C$ - and $R(B, C_i) = 0$ - therefore it defeats any non-strict argument for p_i. Since there is no strict local argument for p_i in $Args_C$, every argument for p_i is defeated by B; therefore p_i is rejected by $JArgs^C$.

(3). At least two calls of $P2P\_DR$ are required to compute *undefined* as an answer for p_i. Since we assume that there are no loops in the local context theories, there are no rules such that the literal in their head also belongs to the body of the rule. Hence, the following conditions must hold: (a) $localAns_{p_i} = false$; by Proposition 2 this means that there is no strict local argument for p_i in $Args_C$; (b) there is no rule with head $\sim p_i$ in C_i, which means that there is no argument in $Args_C$ attacking the arguments for p_i at their root; and (c) there is only one rule r_i with head p_i in C_i, with one literal in its body (say q_j), for which it holds (c_1) $q_j \notin V_i$; (c_2) there is no rule with head $\sim q_j$ in C; and (c_3) there is only one rule with head q_j (say r_j) in C, such that p_i is the only literal in the body of r_j. Hence, the only argument for p_i (A) can be obtained using rule r_i, and the only argument for q_j (A') can be obtained using rule r_j. None of the two arguments is neither justified by $JArgs^C$ nor rejected by $JArgs^C$ (since there are not attacking arguments). Therefore, p_i is neither justified nor rejected by $JArgs^C$.

Inductive Step. (1). $P2P\_DR$ returns $Ans_{p_i} = true$ in $n+1$ calls. The following conditions must hold:

(a) there is a rule r_i with head p_i in C_i, such that for all literals a in its body it holds that $Ans_a = true$ is returned by $P2P\_DR$ in at most n calls. By inductive hypothesis, for every a, there is an argument A_a with conclusion a in $JArgs^C$. Therefore, for every A_a it holds that either A_a is a local argument, or it is the head of an argumentation line A_{La}, such that every argument in A_{La} is in $JArgs^C$. Using arguments A_a, argumentation lines A_{La} and rule r_i, we can build an argument A for p_i and an argumentation line A_L with head A, such that every proper subargument of A and every argument in $A_L - \{A\}$ are in $JArgs^C$ - in other words, A is supported by $JArgs^C$.

(b) $localAns_{\sim p_i} = false$ - by Proposition 2, there is no strict local argument for $\sim p_i$ in $Args_C$

(c) for all rules s_i with head $\sim p_i$ in C_i, either (c_1) there is a literal b in the body of s_i for which $P2P\_DR$ returns $Ans_b = false$ in n calls. By inductive hypothesis, b is rejected by $JArgs^C$, which means that every argument for b is defeated by an argument supported by $JArgs^C$. Hence, every argument B using rule s_i in $Args_C$ is undercut by $JArgs^C$; or (c_2) $\forall b \in body(s_i)$: $P2P\_DR$ returns either *true* or *undefined* as an answer for b (in at most n calls) and $Stronger(SS_{r_i}, BS_{s_i}, T_i) = SS_{r_i}$. By Auxiliary Lemma, we conclude that there is an argument A for p_i in $Args_C$, which uses rule r_i and applicable rules to support p_i, and has rank $R(A, C_i) = max_{d \in SS_{r_i}}(R(d, C_i))$, and for every argument B for $\sim p_i$ in $Args_C$ that uses unblocked rules and rule s_i to support $\sim p_i$, it holds that $R(A, C_i) < R(B, C_i)$; therefore every such argument B does not defeat A at p_i.

Suppose that one of these arguments B defeats a proper subargument of A, D at q_i. Since A uses applicable rules, for q_i $P2P\_DR$ returns $Ans_{q_i} = true$ in n or less calls. Therefore, by inductive hypothesis, there is an argument D' for q_i in $JArgs^C$. B is not a strict local argument, as in that case q_i would be rejected. Suppose B defeats D' at q_i. Since D' is in $JArgs^C$, B is undercut by $JArgs^C$. In case B attacks but cannot defeat D', by definition it holds that $R(D', C_i) < R(B, C_i)$. But since we have already supposed that B defeats D; $R(B, C_i) \leq R(D, C_i)$. Therefore, $R(D', C_i) < R(D, C_i)$ and $R(A', C_i) < R(A, C_i)$, where A' is the argument for p_i that derives from A by replacing D with D'. Following the same process for every subargument D of A, we can obtain an argument A' for p_i, such that A' is supported by $JArgs^C$ and every argument B, such that B uses unblocked rules and B defeats a proper subargument of A', B is undercut by $JArgs^C$. And since $R(A', C_i) < R(A, C_i)$, it holds that for every such argument B, $R(A', C_i) < R(B, C_i)$; therefore B does not defeat A neither at its inner nodes nor at its root.

Suppose that an argument B for $\sim p_i$ in $Args_C$ uses a rule s_i that is not unblocked. By inductive hypothesis, for some literal b in B, it holds that b is rejected; hence B is undercut by $JArgs^C$.

Overall, using A' and the justified argumentation lines for the foreign literals in the body of r_i, we can obtain an argument for p_i, which is supported by $JArgs^C$, and every argument defeating A' is undercut by $JArgs^C$; therefore A' is acceptable w.r.t. $JArgs^C$, and p_i is justified.

(2). $P2P\_DR$ returns $Ans_{p_i} = false$ in $n+1$ calls. The following two conditions must hold: (a) $localAns_{p_i} = false$; hence there is no strict local argument for p_i in $Args_C$; and (b) for every rule r_i with head p_i, either (b_1) there is a literal a in the body of r_i, such that $P2P\_DR$ returns $Ans_a = false$ in at most n calls. By inductive hypothesis, this means that a is rejected, and therefore if $a \in V_i$, every argument A using r_i is defeated by an argument supported by $JArgs^C$, while if $a \notin V_i$, every argumentation line with head A contains an argument that is defeated by an argument supported by $JArgs^C$. In any of the two cases, the arguments using r_i are rejected by $JArgs^C$; or (b_2) there is a rule s_i with head $\sim p_i$ in C_i, such that $P2P\_DR$ returns $Ans_b = true$ for any literal b in the body of s_i, and for all literals a in the body of r_i, $P2P\_DR$ returns *true* or *undefined* in at most n calls, and $Stronger(BS_{r_i}, SS_{s_i}, T_i) \neq BS_{r_i}$. By inductive hypothesis and Auxiliary Lemma, in the same way with before, using rule s_i we can build an argument B for $\sim p_i$ such that B is supported by $JArgs^C$ and has lower or equal rank than any argument A for p_i that uses unblocked rules and rule r_i; therefore B defeats any such argument for p_i.

Consider now the arguments for p_i in $Args_C$ that use at least one rule that is not unblocked. In the same way with before, we can prove that these arguments are defeated by an argument supported by $JArgs^C$.

Therefore, for every argument A for p_i it holds that either A or an argument in every argumentation line with head A is defeated by an argument supported by $JArgs^C$; therefore p_i is rejected by $JArgs^C$.

(3). $P2P\_DR$ returns $Ans_{p_i} = undefined$ in $n+1$ calls. The following conditions must hold:

(a) $localAns_{p_i} = false$ and $localAns_{\sim p_i} = false$; by Proposition 2, there are no strict local arguments for p_i and $\sim p_i$ in $Args_C$;

(b) for all rules r_i with head p_i in C either (b_1) there is a literal a in the body of r_i such that $P2P\_DR$ returns either *false* or *undefined* as an answer for a in n or less calls; or (b_2) for all a, $P2P\_DR$ returns *true* as an answer for a in n or less calls, but there is a rule s_i with head $\sim p_i$ in C_i, such that for every literal b in the body of s_i, $P2P\_DR$ returns either *true* or *undefined* as an answer for b in n or less calls, and $Stronger(SS_{r_i}, BS_{s_i}, T_i) \neq SS_{r_i}$. For the case described in (b_1), using inductive hypothesis, a is not justified; therefore there is no argument for p_i in $Args_C$ that is supported by $JArgs^C$. For the case of (b_2), by inductive hypothesis and Auxiliary Lemma, there is an argument B in $Args_C$ that uses s_i to support $\sim p_i$, such that B uses unblocked rules, and for every argument A in $Args_C$ that uses applicable rules and r_i to support p_i, it holds that $R(B, C_i) \leq R(A, C_i)$, which means that B defeats A at p_i. In the same way with before, we can prove that there is an argument B' in $Args_C$, which also uses rule s_i, and has lower rank than B w.r.t. C_i, and an argumentation line B_L with head B', such that no subargument of B or argument in B_L is defeated by an argument supported by $JArgs^C$. Therefore B' is not undercut by $JArgs^C$, and defeats any non-strict argument with applicable rules with head p_i. Since there is no strict local argument for p_i, and for the arguments for p_i that use rules that are not applicable w.r.t. C, it is easy to verify that they are not supported by $JArgs^C$, we reach to the conclusion that for every argument A for p_i in $Args_C$, A is either not supported by $JArgs^C$, or it is attacked by an argument in $Args_C$, which is not undercut by $JArgs^C$; therefore p_i is not justified.

(c) for all rules s_i with head $\sim p_i$ in C_i either (c_1) there is a literal b in the body of s_i, such that $P2P\_DR$ returns either *false* or *undefined* as an answer for b in n or less calls; or (c_2) for all b, $P2P\_DR$ returns *true* as an answer for b in n or less calls, but there is a rule r_i with head p_i in C_i, such that for every literal a in the body of r_i, $P2P\_DR$ returns either *true* or *undefined* as an answer for a in n or less calls, and $Stronger(BS_{r_i}, SS_{s_i}, T_i) = BS_{r_i}$. In the same way with before, we can reach to the conclusion that there is an argument A in $Args_C$, which uses rule r_i, and an argumentation line A_L with head A, such that there is no argument B that is supported by $JArgs^C$ and defeats an argument in A_L. Therefore p_i is not rejected by $JArgs^C$.

$\Leftarrow$ (1). We use induction on the stage of acceptability of arguments with conclusion p_i in $Args_C$.

Inductive Base. Suppose that an argument A for p_i in $Args_C$ is acceptable w.r.t. J_0^C. This means that either: (a) A is a strict local argument for p_i; in this case, by Proposition 2, $P2P\_DR$ will return $localAns_{p_i} = true$, and therefore $Ans_{p_i} = true$; or (b) A is a defeasible local argument in $Args^{C_i}$ that is supported by J_0^C, and every argument defeating A is undercut by J_0^C. Since A is supported by J_0^C, A contains one defeasible rule with head p_i (say r_i) with empty body. Suppose that there is a rule s_i with head $\sim p_i$ in C_i, such that for all literals b in its body, $P2P\_DR$ returns either $Ans_b = true$ or $Ans_b = undefined$ and $Stronger(SS_{r_i}, BS_{s_i}, T_i) = BS_{s_i}$. This means, by Auxiliary Lemma, that for all arguments for p_i using applicable rules and rule r_i, there is an argument B' that uses rule s_i and unblocked rules, which has lower rank than A in C_i. But, since $R(A, C_i) = 0$ (A is a local argument of C_i), $R(B', C_i) < 0$, which is not possible. Therefore for every rule s_i with head $\sim p_i$ in C_i, either (c) there is a literal b in the body of s_i for which $P2P\_DR$ returns $Ans_b = false$, or (d) $Stronger(SS_{r_i}, BS_{s_i}, T_i) \neq SS_{s_i}$. Suppose that (d) holds and $Stronger(SS_{r_i}, BS_{s_i}, T_i) = none$. By definition of $Stronger$ and Auxiliary Lemma, this means that there is an argument B for $\sim p_i$ in $Args_C$ that uses unblocked rules and rule s_i and $R(B, C_i) = 0$. Therefore B defeats A, and for every rule used in B it holds that for all literals in its body $Ans_b \neq false$. By the first part of the Proposition, this means that there is an argumentation line B_L with head B, such that and no argument in B_L is defeated by an argument supported by J_0^C. However, since B defeats A, B is undercut by J_0^C. Hence for every argumentation line B_L with head B, there is an argument D that is supported by J_0^C and defeats a proper subargument of B or an argument in $B_L - \{B\}$; the latter conclusion contradicts our previous conclusion that no argument in B_L is attacked by J_0^C. Therefore our supposition that $Stronger(SS_{r_i}, BS_{s_i}, T_i) = none$

does not hold. Consequently, for every rule s_i with head $\sim p_i$ in C_i, either (a) there is a literal b in the body of s_i for which $P2P\_DR$ returns $Ans_b = false$, or (b) $Stronger(SS_{r_i}, BS_{s_i}, T_i) = SS_{r_i}$. Overall, $P2P\_DR$ will compute $sup_{p_i} = true$, and either $unb_{\sim p_i} = false$ or $Stronger(SS_{p_i}, BS_{\sim p_i}, T_i) = SS_{p_i}$, and eventually will return *true* as an answer for p_i.

Inductive Step. Suppose that A is an argument for p_i in $Args_C$ that is acceptable w.r.t. J^C_{n+1}. This means that either: (a) A is a strict local argument for p_i - in this case, by Proposition 2, $P2P\_DR$ will return $localAns_{p_i} = true$, and $Ans_{p_i} = true$; or (b) A is supported by J^C_{n+1} and every argument defeating A is undercut by J^C_{n+1}. That A is supported by J^C_{n+1} means that every proper subargument of A is acceptable w.r.t. J^C_n, and there is an argumentation line A_L with head A such that every argument in A_L is acceptable w.r.t. J^C_n. By inductive hypothesis, there is a rule r_i with head p_i in C_i, such that for every literal a in the body of r_i, $P2P\_DR$ returns $Ans_a = true$. Suppose that B is an argument in $Args_C$ that defeats A. By definition, B is undercut by J^C_{n+1}; namely, for every argumentation line B_L with head B, there is an argument D in $Args_C$, which is supported by J^C_{n+1}, and defeats a proper subargument of B or an argument in $B_L - \{B\}$. Since D is supported by J^C_{n+1}, every subargument of D is acceptable w.r.t. J^C_n, and there is an argumentation line D_L with head D, such that every argument in D_L is acceptable w.r.t. J^C_n. Suppose that D defeats B' at q_j (where B' is either a proper subargument of B or an argument in an argumentation line with head B). By inductive hypothesis, there is a rule t_j for q_j in C_j, such that for all literals d in the body of t_j, $P2P\_DR$ returns $Ans_d = true$. It also holds that $R(D, C_j) \leq R(B', C_j)$. Suppose that there is a rule s_j with head q_j in C_j such that for all literals b in the body of s_j, $P2P\_DR$ returns $Ans_b \neq false$ and $Stronger(BS_{s_j}, SS_{t_j}, T_i) = BS_{s_j}$. By Auxiliary Lemma, for every argument D' for $\sim q_j$ in $Args_C$ that uses applicable rules and rule t_j, there is an argument E for q_j in $Args_C$, which uses unblocked rules and rule s_j s.t. $R(E, C_j) < R(D', C_j)$. As E uses unblocked rules, it easy to verify by the first part of the Proposition, that E contains no literals that are rejected by $JArgs^C$, and that there is an argumentation line E_L for q_j such that no argument in E_L is defeated by an argument supported by $JArgs^C$. Following the same process, for every literal that B' is undercut at, we can build an argumentation line B_L for $\sim p_i$, such that no argument in B_L is defeated by an argument supported by $JArgs^C$. This contradicts the fact that every argument defeating A is undercut by $JArgs^C$. Therefore, for all rules s_j with head q_j in C_j, either there is a literal b' in the body of s_j, such that $Ans'_b = false$, or there is a rule t_j with head q_j in C_j, such that for all literals d in the body of t_j, $P2P\_DR$ returns $Ans_d = true$ and $Stronger(BS_{s_j}, SS_{t_j}, T_i) \neq BS_{s_j}$. These conditions suffice for $P2P\_DR$ to return *false* as an answer for q_j. Therefore, for rule s_i, which is used in B to support $\sim p_i$, it holds that there is a literal b in the body of s_i, such that $Ans_b = false$.

Consider now a rule s_i with head $\sim p_i$ in C_i, which is contained in an argument in $Args_C$, which does not defeat A. Suppose that for all literals b in the body of s_i, $P2P\_DR$ returns either *true* or *undefined* as an answer for b, and $Stronger(SS_{r_i}, BS_{s_i}, T_i) \neq SS_{r_i}$. Then, by Auxiliary Lemma and by the first part of the Proposition, we can verify that there is an argument B and an argumentation line B_L with head B, such that no argument in B_L is defeated by an argument supported by $JArgs^C$, and for every argument A' that uses applicable rules and r_i to support p_i, $R(A', C_i) \geq R(B, C_i)$. Using the same reasoning with before, we conclude that p_i is not justified, which contradicts our original supposition. Therefore, for every rule s_i with head $\sim p_i$ in C_i, which is contained in an argument B that does not defeat A, either there is a literal b in the body of s_i, such that $P2P\_DR$ returns *false* as an answer for b, or $Stronger(SS_{r_i}, BS_{s_i}, T_i) \neq SS_{r_i}$.

Overall, there is a rule r_i with head p_i in C_i, such that for every literal a in the body of r_i, $P2P\_DR$ returns *true* as an answer for a, and for for every rule s_i for $\sim p_i$ in C_i, either there is a literal b in the body of s_i, such that $P2P\_DR$ returns *false* as an answer for b, or $Stronger(SS_{r_i}, BS_{s_i}, T_i) \neq SS_{r_i}$. Therefore, $P2P\_DR$ will compute $sup_{p_i} = true$, and either $unb_{\sim p_i} = false$ or $Stronger(SS_{p_i}, BS_{\sim p_i}, T_i) = SS_{p_i}$, and eventually will return *true* as an answer for p_i.

$\Leftarrow$ (2). Suppose that for a literal p_i that is rejected by $JArgs^C$, it holds that $P2P\_DR$ returns either *true* or *undefined* as an answer for p_i. By definition, either (a) $localAns_{p_i} = true$, which means there is a strict local argument for p_i in $Args_C$, and leads to the conclusion that p_i is justified, which by Lemma 2 contradicts our original supposition that p_i is rejected by $JArgs^C$; or (b) there is a rule r_i with head p_i in C_i, such that for all literals a in the body of r_i, $P2P\_DR$ returns either *true* or *undefined* as an answer for a, and for all rules s_i with head $\sim p_i$ in C_i, either there is a literal b in the body of s_i, such that $Ans_b \neq true$ or $Stronger(BS_{r_i}, BS_{s_i}, T_i) = BS_{r_i}$. By Auxiliary Lemma and the first part of the Proposition, this implies that there is an argument A for p_i in $Args_C$ and an argumentation line A_L with head A, such that no argument in $A_L - \{A\}$ and no proper subargument of A is defeated by an argument supported by $JArgs^C$, and for every argument B for $\sim p_i$ in $Args_C$, either B is not supported by $JArgs^C$, or there is an argument D for p_i in $Args_C$ and an argumentation line D_L with head D, such that no argument in $D_L - \{D\}$ and no proper subargument of D is defeated by an argument supported by $JArgs^C$, and $R(D, C_i) < R(B, C_i)$. This leads to the conclusion that p_i is not rejected, which contradicts our original supposition. Therefore, $P2P\_DR$ will return *false* as an answer for p_i.

$\Leftarrow$ (3) For a literal p_i, which is neither justified nor rejected by $JArgs^C$, suppose that $Ans_{p_i} = true$. By the first part of the theorem, p_i is justified (contradiction). Suppose that $Ans_{p_i} = false$. By the first part of the theorem, this means that p_i is rejected by $JArgs^C$ (contradiction). Therefore, for p_i, $P2P\_DR$ will return $Ans_{p_i} = undefined$.