Logic agents, dialogues and negotiation: an abductive approach

F. Sadri*; F. Toni*; P. Torroni†
* Dept. of Computing, Imperial College; 180 Queens Gate, SW7 London, UK
† DEIS, Università di Bologna; V.le Risorgimento 2, 40136 Bologna, Italy
{fs, ft}@doc.ic.ac.uk; ptorroni@deis.unibo.it

Abstract

This paper proposes a logical framework for automated negotiation processes. It describes the language, the knowledge and the reasoning that is required to build negotiation dialogues between two agents, in order to obtain resources. Such knowledge is expressed in a declarative way. In particular, we represent the beliefs of agents as logic programs and integrity constraints, as in abductive logic programming, with the dialogue performatives treated as abducibles / hypotheses. The integrity constraints play the role of dialogue / negotiation protocols. In this work we also determine some important properties of a particular set of integrity constraints, showing that the protocol is such that the interaction between agents terminates in a finite number of steps and converges to a solution.

1 Introduction

Negotiation plays a central role in e-commerce Multi-Agent applications. In multi-agents systems, agents interact in order to exchange knowledge, when their own is not enough to achieve their goals. In domains such as e-commerce, when it is natural to assume that agents are self-interested, it is most likely that they need to negotiate in order to obtain the information or resources they need. The protocols used for negotiation have to be properly designed in order to ensure that each party will have a positive payoff out of the negotiation process.

The focus of this work is the knowledge and the reasoning that is required to build dialogues between two negotiating agents, in order to exchange resources that help achieve the agents' objectives. We express such knowledge in a declarative way. In particular, we represent the beliefs of agents as logic programs and integrity constraints, as in abductive logic programming [Kakas et al. (1998)], with the dialogue performatives treated as abducibles / hypotheses. The integrity constraints play the role of dialogue / negotiation protocols.

In addition, we investigate certain interesting properties of dialogues built within the proposed declarative framework, such as termination and success.

Finally, we describe the dynamic execution of an abductive proof-procedure [Fung and Kowalski (1997); Sadri and Toni (2000)], as within the agent architecture of Kowalski and Sadri (1999), to generate dialogues between agents, conforming to the given negotation protocols.

Throughout the paper we refer to a simplified version of an example from Parsons et al. (1998). In this example, an agent a wants to hang a picture on the wall and plans to do it by using a nail and a hammer. Unfortunately, a does not have a nail and needs to negotiate with b to try to obtain it.

We make some simplifying assumptions. The agents that we describe by giving their abductive logic programs are assumed to be friendly, i.e. collaborative, although in the general framework we propose, different varieties of attitudes may be modeled by other different programs (e.g. selfish, altruistic, etc.). Also, only two agents are assumed to be participating in any dialogue, and each agent takes part in at most one dialogue at a time. Finally, agents have only one goal and they commit to only one intention at a time, with respect to that goal, with the intention including a full plan to achieve it.

The paper is organized as follows: in Section 2 we will introduce the language used to negotiate, in Section 3 we will describe the agents from a knowledge representation point of view, in Section 4 we will discuss some properties of the dialogue, and in Section 5 we will talk about the execution model. Section 6 concludes the paper.

2 Language for negotiation

The *dialogue primitives (performatives)* are of the form $tell(a, b, Move, t)$, where a and b are the sending and the receiving agents, respectively, t represents the time when the performative is uttered, and $Move$ is a *dialogue move*, recursively defined as follows (note that the given dialogue moves are a subset of those proposed in Amgoud et al. (2000)):

- $\mathbf{request}(give(R))$ is a dialogue move, used to request a resource R;
- $\mathbf{promise}(give(R), give(R'))$ is a dialogue move, used to propose and to commit to exchange deals, of resource R' in exchange for resource R.
- if $Move$ is a dialogue move, so are

accept$(Move)$, **refuse**$(Move)$ (used to accept/refuse a previous dialogue $Move$)

challenge$(Move)$ (used to ask a justification for a previous $Move$)

justify$(Move, Support)$ (used to justify a past $Move$, by means of a $Support$)

- there are no other dialogue moves, except the ones given above.

Once uttered, dialogue performatives are recorded on a blackboard (the *dialogue store*) shared amongst the agents, in the order in which they have been uttered. For example, the dialogue store might be:

$tell(a, b, request(give(nail)), 1)$
$tell(b, a, refuse(request(give(nail))), 2)$
$tell(a, b, challenge(refuse(request(give(nail)))), 3)$
$tell(b, a, justify(refuse(request(give(nail))),$
$\{\sim have(nail)\}), 4)$

where $\sim$ stands for not.

Note that times are *transaction timestamps*. Note further that the dialogue store grows monotonically over a dialogue, as new performatives are uttered. We will assume the dialogue store is reset at the end of each dialogue.

3 Knowledge

Each agent's knowledge $\mathcal{K}$ is defined as a tuple $< \mathcal{B}, \mathcal{R}, \mathcal{I}, \mathcal{D}, \mathcal{G} >$, where:

- $\mathcal{B}$ contains domain-dependent beliefs about the world, such as the information about itself and the other agents, as well as domain-independent beliefs, used to regulate the negotiation dialogues;
- $\mathcal{R}$ contains the resources owned *initially* (i.e., before the dialogue starts) by the agent (the current state of owned resources can be inferred through $\mathcal{R} \cup \mathcal{D}$, see further on);
- $\mathcal{I}$ is the agent's current intention, i.e., the plan that the agent intends to carry out in order to achieve its goal, together with the available resources and the missing resources, that need to be obtained for the plan to be executable. In $\mathcal{I}$ we also include the agent goal;
- $\mathcal{D}$ contains the current dialogue store, represented by the past (*time-stamped*) dialogue performatives. The dialogue store is shared amongst agents, so all agents involved in the same dialogue are assumed to have the same component $\mathcal{D}$.
- $\mathcal{G}$ is the agent goal;

As an example, the knowledge of some agent a can be the following $\mathcal{K}_a$ (0 stands for the initial time):

$\mathcal{B}_a$ domain-specific beliefs $\{\ is\_agent(b), i\_am(a)\ \}$, as well as domain-independent beliefs, e.g., those described in subsection 3.1;
$\mathcal{R}_a$: $\{\ have(picture, 0), have(hammer, 0),$
$have(screwdriver, 0)\ \}$;
$\mathcal{I}_a$: $\{\ available(\{hammer, picture\}, 0),$
$plan(\{obtain(nail), hit(nail),$
$hang(picture)\}, 0), goal(\{hung(picture)\}),$
$missing(\{nail\}, 0)\ \}$;
$\mathcal{D}_a$: $\emptyset$.
$\mathcal{G}_a$: $hung(picture)$;

Note that the goal is not timestamped. We assume that actions within plans are implicitly temporally ordered (left-to-right), an goals are implicitly time-stamped by the implicit time of the last action in the plan.

We assume that the agents are provided with a *cost* function that maps from the domain of the possible intentions (plans) to $\mathbb{R}^+$. Such function will have to be *acceptable*[1] in order for some results to hold, as it will be discussed in the sequel. In this paper, we will adopt as a cost function for a given intention the number of its missing resources.

We assume that plans and missing resources in intentions are generated by a planner, but we do not rely upon any specific planner: for instance we could adopt a STRIPS-, situation calculus- or event calculus-based planner, or even a plan library. Moreover, each agent might rely upon a different planner. However, we assume that, when given a goal (in $\mathcal{G}$) and the available resources (in $\mathcal{R}$), the planner of an agent provides a suitable plan in the form of an ordered sequence of actions, as well as a (possibly empty) list of missing resources (namely, it returns an intention in $\mathcal{I}$). The planner of an agent might be used to generate plans for other agents, if a resource exchange is required in order to obtain some missing resources in the agent's current plan, and to re-plan (see later in this section).

The planner has access to domain-dependent beliefs in $\mathcal{B}$, describing preconditions and effects of actions or plan libraries. We assume that agents cannot exchange rules such as '*you can have a picture hung if you have a screw and a screwdriver*'. Therefore, we assume that negotiating agents share knowledge about actions. Such knowledge is also stored in $\mathcal{B}$. This assumption allows agents to re-plan for other agents, if an exchange of resources is required.

[1] A *cost* function $|\ .\ |\ :\ \mathcal{I} \rightarrow \mathbb{R}^+$ is *acceptable* if it satisfies the following properties:

- $missing(\ \emptyset\ ) \in I\ \Rightarrow\ |\ I\ |\ =\ 0$
- $missing(\ Rs\ ) \in I, Rs \neq \emptyset\ \Rightarrow\ |\ I\ |\ >\ 0$
- $missing(\ Rs\ ) \in I, missing(\ \bar{Rs}\ ) \in \bar{I},$ $Rs = \bar{Rs}\ \Rightarrow\ |\ I\ |\ =\ |\ \bar{I}\ |$
- $missing(\ Rs\ ) \in I, missing(\ \bar{Rs}\ ) \in \bar{I},$ $Rs \supset \bar{Rs}\ \Rightarrow\ |\ I\ |\ >\ |\ \bar{I}\ |$

In the remainder of this section we will describe the domain-independent part of the belief component of the knowledge of agents. We will assume that this part is held by all agents.

3.1 Dialogue protocols

Dialogue protocols are expressed by means of if-then rules of the form:

```
% if the performative P is uttered at time
% T and conditions C hold in K at time T
% then performative P' should be uttered
% a time T+1
```

$$P(T) \wedge C(T) \Rightarrow P'(T+1)$$

The performative P plays the role of a *trigger* for the dialogue protocol. The conditions can be satisfied by retrieving information from the knowledge of the agent, e.g. the missing resources and the currently owned resources as well as the dialogue store. We will assume that this information is compiled into if-rules defining the predicates $have$ (the agent has a resource R), $miss$ (the agent has not yet obtained a resource R needed for a plan P to achieve a goal G), $need$ (the agent has a resource R needed for a plan P to achieve a goal G), and so on. Before we introduce the dialogue protocols, let us present a simplified version of the if-rules for some such predicates:

```
% rule 1: the agent has resource R at
% time T if she had it initially and has
% not given it away
```

$$\begin{aligned} &have(R,T) \leftarrow \\ &\quad have(R,0)\ \wedge 0 < T \\ &\quad \wedge \sim [gave\_away(R,T1), 0 < T1 \leq T] \end{aligned}$$

```
% rule 2: the agent has resource R at
% time T if she has obtained it and did
% not give it away
```

$$\begin{aligned} &have(R,T) \leftarrow \\ &\quad obtained(R,T1)\ \wedge T1 < T \\ &\quad \wedge \sim [gave\_away(R,T2), T1 < T2 \leq T] \end{aligned}$$

```
% rule 3: the agent obtained R at time T
% from Agent after she requested it
```

$$\begin{aligned} &obtained(R,T) \leftarrow \\ &\quad i\_am(X) \wedge tell(Agent, X, accept(request(R)), T) \end{aligned}$$

```
% rule 4: the agent gave away R at time T
% in reply to the Agent's request
```

$$\begin{aligned} &gave\_away(R,T) \leftarrow \\ &\quad i\_am(X) \wedge tell(X, Agent, accept(request(R)), T) \end{aligned}$$

```
% rule 5: agent a misses R at time T if
% it is in the missing list in I_a for a
% plan P and a goal G
```

$$\begin{aligned} &miss(R,P,G,Rs,T) \leftarrow \\ &\quad intend(I)\ \wedge plan(P,T) \in I\ \wedge goal(G) \in I \\ &\quad \wedge missing(Rs,T) \in I\ \wedge R \in Rs \end{aligned}$$

```
% rule 6: agent a needs R at time T if it
% is in the available list in I_a for a plan
% P for a goal G
```

$$\begin{aligned} &need(R,P,G,T) \leftarrow \\ &\quad intend(I)\ \wedge plan(P,T) \in I\ \wedge goal(G) \in I \\ &\quad \wedge available(Rs,T) \in I\ \wedge R \in Rs \end{aligned}$$

Note that resources are considered not to be owned if they have been promised to another agent, even if the actual delivery has not yet been carried out. Indeed, here we are not concerned with the execution of a plan, and we assume that agents will actually obtain the resources they have been promised by the time of the execution of the plan.

A dialogue protocol can be seen as a property that must be satisfied at all time, by enforcing (uttering) the conclusion of such protocol whenever it is triggered and the conditions in its premise are satisfied by the if-rules. In this sense, they behave like active rules in databases and integrity constraints in abductive logic programming [Kakas et al. (1998)]. Indeed, the knowledge of an agent can be seen as an abductive logic program, with the logic program consisting of the if-rules above plus the owned resources, the integrity constraints consisting of the dialogue protocols given below, and the abducibles (possible hypotheses) consisting of the performatives. In the sequel we will consider the following dialogue protocols. Note that, from now on, we will drop the time parameter, for simplicity.

```
% IC 1: accept a request
```

$$\begin{aligned} &i\_am(X)\ \wedge\ tell(Agent, X, request(give(R)))\ \wedge\ have(R) \\ &\wedge\ \sim need(R) \\ &\quad \Rightarrow tell(X, Agent, accept(request(give(R)))) \end{aligned}$$

This set of constraints describes the behaviour of a collaborative but selfish agent. In particular, the agent will $challenge$ the $request$ and start a negotiation thread (`IC2`) only if there exists a resource that she is actually missing, i.e., only if she can obtain a positive payoff from further dialogue.

```
% IC 2(a): challenge a request if R is
% not held
```

$$\begin{aligned} &i\_am(X)\ \wedge\ tell(Agent, X, request(give(R))) \\ &\wedge\ miss(\_,\_,\_,\_)\ \wedge\ \sim have(R) \end{aligned}$$

$\Rightarrow tell(X, Agent, challenge(request(give(R))))$

```
% IC 2(b): challenge a request if R is
% needed
```

$i\_am(X) \wedge tell(Agent, X, request(give(R))) \wedge have(R) \wedge miss(\_,\_,\_,\_) \wedge need(R)$
$\Rightarrow tell(X, Agent, challenge(request(give(R))))$

```
% IC 3(a): refuse a request if not
% convenient to propose an exchange
```

$i\_am(X) \wedge tell(Agent, X, request(give(R))) \wedge \sim miss(\_,\_,\_,\_) \wedge \sim have(R)$
$\Rightarrow tell(X, Agent, refuse(request(give(R))))$

```
% IC 3(b): refuse a request if not
% convenient to propose an exchange
```

$i\_am(X) \wedge tell(Agent, X, request(give(R))) \wedge \sim miss(\_,\_,\_,\_) \wedge need(R)$
$\Rightarrow tell(X, Agent, refuse(request(give(R))))$

```
% IC 4: justify a request
```

$i\_am(X) \wedge tell(Agent, X, challenge(request(give(R)))) \wedge miss(R, Plan, Goal, Missing) \wedge Support = \{missing(Missing), plan(Plan), goal(Goal)\}$
$\Rightarrow tell(X, Agent, justify(request(give(R)), Support))$

In IC5 and IC6, the policy adopted is that when X cannot give R to $Agent$, X tries to be helpful to $Agent$ and looks for an alternative plan for $Agent$'s $Goal$, in that $Agent$ changes intention (R is no longer required, but the new plan requires a new resource R'' that X has and can give) and X gets a resource R' that she is missing. In doing that, X looks for an exchange that is favorable to both, which could be thought of as a *selfish* behaviour. If no such alternative can be found, or if X does not miss anything at all, as in IC3, X refuses the request.

```
% IC 5: refuse a request after a
% justification: Agent cannot find an
% alternative plan for X's goal that could
% be proposed for a deal
```

$i\_am(X) \wedge tell(Agent, X, justify(request(give(R)), Support)) \wedge goal(Goal) \in Support \wedge miss(R', \_,\_,\_) \wedge \sim exists\_alternative\_plan(Goal, without(\{R', R\}))$
$\Rightarrow tell(X, Agent, refuse(give(R)))$

```
% IC 6: propose R'' in exchange for R':
% X finds a different plan for Agent's goal
% that makes use of the resource R'' that
% she has and does not need, and that
% requires neither of {R,R'}, while she
% needs R', therefore can propose a deal:
% R'' in exchange of R'
```

$i\_am(X) \wedge tell(Agent, X, justify(request(give(R)), Support)) \wedge goal(Goal) \in Support \wedge miss(R', \_,\_,\_) \wedge have(R'') \wedge \sim need(R'') \wedge choose\_alternative\_plan(Goal, NewPlan, without(\{R, R'\}), with(\{R''\}))$
$\Rightarrow tell(X, Agent, promise(give(R''), give(R')))$

In order to evaluate a proposal, the agent that receives the *promise* must check if it allows for a lower cost plan. To that purpose, the predicate $choose\_better\_plan$ will have to compare the old plan and the new one with respect to the cost function. It is clear now that such function plays a central role in the whole negotiation process, ensuring monotonicity and termination. In particular, such cost function must be acceptable, with respect to the criteria given above. We will discuss this in more detail in the next section.

```
% IC 7: accept a deal: X finds the deal
% proposed by Agent convenient (the predi-
% cate choose_better_plan is successful
% only if NewPlan costs less than OldPlan)
```

$i\_am(X) \wedge tell(X, Agent, request(give(R))) \wedge tell(Agent, X, promise(give(R''), give(R'))) \wedge need(R, OldPlan, Goal) \wedge have(R') \wedge \sim need(R') \wedge choose\_better\_plan(Goal, NewPlan, OldPlan, without(\{R, R'\}), with(\{R''\}))$
$\Rightarrow tell(X, Agent, accept(promise(give(R''), give(R'))))$

```
% IC 8: refuse a deal
```

$i\_am(X) \wedge tell(X, Agent, request(give(R))) \wedge tell(Agent, X, promise(give(R''), give(R'))) \wedge need(R, OldPlan, Goal) \wedge \sim choose\_better\_plan(Goal, NewPlan, OldPlan, without(\{R, R'\}, with(\{R''\}))$
$\Rightarrow tell(X, Agent, refuse(promise(give(R''), give(R'))))$

In the sequel, we will denote with $\mathcal{S}$ the set of integrity constraints in $\mathcal{B}$, and with $\bar{\mathcal{S}}$ this particular set, $\{IC_1, \ldots, IC_8\}$, describing this particular agent behaviour. In the next section we will introduce some definitions, with respect to a generic set of integrity constraints, and some properties that hold for $\bar{\mathcal{S}}$ in particular.

4 Agent dialogues and properties

A dialogue between two agents can be formally defined as follows, given that the agents are equipped with knowledge as described in the previous section.

Definition 1 (*Dialogue*) A *dialogue* between two agents a and b is a sequence of performatives, $\{p_0, p_1, p_2, \ldots\}$, such that:

- $\forall\ i \geq 0$, if p_i is uttered by agent a (viz. b), then

p_{i+1} (if any) is uttered by agent b (viz. a);

- $\forall\, i \geq 0$, if p_i is uttered by agent $x \in \{a,b\}$, then p_{i+1} (if any) is uttered by agent $y \in \{a,b\} \setminus \{x\}$ such that there exists a (ground) integrity constraint $ic \in \mathcal{S}, body(ic) \Rightarrow head(ic)$, where
  - i) $\mathcal{K}_y \cup p_i$ entails $body(ic)$, and
  - ii) $p_{i+1} = head(ic)$;
- $\forall\, i,j \geq 0$, if $i \neq j$ then $p_i \neq p_j$.[2]

A *request dialogue* with respect to a resource R and an intention I of agent x is a dialogue $\{p_0, p_1, p_2, \ldots\}$ such that $p_0 = tell(x, y, request(give(R)))$, where $x, y \in \{a,b\}, x \neq y, missing(Rs) \in I$ and $R \in Rs$.

In the sequel, by *dialogue* we will always mean a *request dialogue*.

Definition 2 (*Termination*) A dialogue $\{p_0, p_1, \ldots, p_n\}$, $n \geq 0$, is *terminated* when, given that x utters p_n, there exists no possible performative that y can utter, i.e., for all $ic \in \mathcal{S}, body(ic) \Rightarrow head(ic)$, $\mathcal{K}_y \cup p_n$ does not entail $body(ic)$.

It is time now to introduce a first property of the integrity constraint set $\bar{\mathcal{S}}$.

Property 1 (*Termination of the request dialogue*) A request dialogue between two agents, a and b, terminates in a finite number of steps, provided the rules in $\mathcal{B}$ (including the planner) constitute a terminating program, and the ICs in $\mathcal{K}$ are a subset of $\bar{\mathcal{S}}$.

Proof. (*sketch*) Since the dialogue is logically inferred within an abductive framework (and in particular, this framework is based on the IFF proof-procedure), where the abductive derivation is performed interleaving resolution steps made by the two agents, a dialogue will terminate if and only if such resolution terminates. In particular, the if-definitions in $\mathcal{B}$ (including those of the planner) constitute a terminating program by hypothesis, and the integrity constraints in $\bar{\mathcal{S}}$ are such that the same constraint cannot be selected for propagation a second time in the same dialogue, with a different grounding of the body. Finally, the form of the integrity constraints is such that there cannot be a cycle involving an if-definition and an integrity constraint. Therefore, since $\bar{\mathcal{S}}$ is a finite set, for any grounded query (i.e. for any resource request) it is possible to draw a finite IFF-tree. ∎

A request dialogue with respect to a resource could terminate with or without success:

[2]By $p_i \neq p_j$, we mean that the two dialogue moves are different, *independently* of the time parameter.

Definition 3 (*Successful request dialogue*) A terminated request dialogue $\{p_0, p_1, \ldots, p_n\}$ between agents a and b, with respect to an intention I and a resource R of agent a, is *successful* when one of the following situations holds:

$$p_n = tell(b, a, accept(request(give(R))))$$
$$p_n = tell(a, b, accept(promise(give(R'), give(R))))$$
$$p_n = tell(b, a, accept(promise(give(R), give(R'))))^3$$

In other words, a request dialogue is successful when the resource is obtained, possibly in exchange for another resource. In Figure 1 two possible dialogues are sketched: the former terminates successfully while the latter does not.

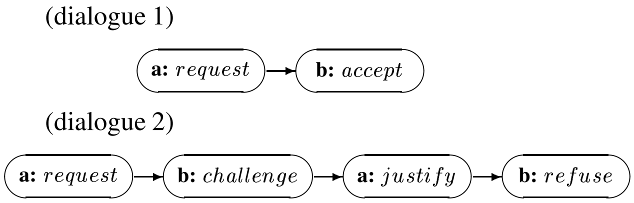

Figure 1: Two dialogues, a successful one (dialogue 1) and an unsuccessful one (dialogue 2)

A request dialogue that does not terminate successfully can still result in a step forward towards the achievement of the agent's goal. This situation occurs when there exists an alternative intention with a plan that does not require the resource in question and has a lower *cost* than the original intention. In this case, the old intention becomes obsolete, and the agent embraces the new one. This form of *conditional* successful dialogue is defined as follows:

Definition 4 (*c-successful request dialogue*) A terminated request dialogue $\{p_0, p_1, \ldots, p_n\}$ between agents a and b, initiated by a, with respect to an intention I of agent a and a resource R, is *c-successful* if b will give R'' to a in exchange for R', namely one of the following situations holds:

i) $p_n = tell(a, b, accept(promise(give(R'), give(R''))))$
`% a accepts b's proposal and commits`
`% to giving R' in exchange for R''`
ii) $p_n = tell(b, a, accept(promise(give(R''), give(R'))))$
`% b accepts a's proposal and commits`
`% to giving R'' in exchange for R'`

and given that $goal(G) \in I, missing(Rs) \in I, R \in Rs$, there exists an intention $\bar{I}$ such that $goal(G) \in \bar{I}$, $missing(\bar{Rs}) \in \bar{I}, R \notin \bar{Rs}, R' \notin \bar{Rs}, R'' \in \bar{Rs}$ and

[3]If both arguing agents are programmed with $\bar{\mathcal{S}}$, this performative will never be uttered, since there is no derivation tree that can reach it. However, we aim at giving here a most general definition of termination, that may hold if we provide agents with a different set of integrity constraints.

$\mid \bar{I} \mid \; < \; \mid I \mid$.

Namely, the agents agreed on an exchange of resources (R'' in exchange of R') that makes the request for R obsolete for a. In Figure 2 we have an example of c-successful terminated dialogue.

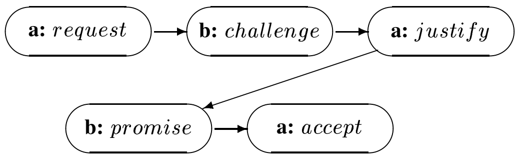

Figure 2: c-successful dialogue: on receiving the justification from a, b promises a new resource that makes the original request obsolete

The use of the $<$ relation rather than $\leq$ ensures the strict monotonicity of the negotiation procedure: for we can say that a goal in an intention $\mathcal{I}$, having initially n missing resources can be carried out after at most n successful *or* c-successful dialogues (in the case of at least one c-successful dialogue, the plan, and therefore the intention, may be modified during the negotiation phase).

We will call *unsuccessful* dialogues those which are terminated but are neither successful nor c-successful.

Property 2 (*Convergence of the request dialogue*) Given an acceptable cost function $\mid . \mid \; : \; \mathcal{I} \rightarrow \mathbb{R}^+$, for every terminated request dialogue with respect to an intention I and a resource R, between agents a and b, whose integrity constraints are a subset of $\bar{\mathcal{S}}$, there exists an intention $\bar{I}$ such that $\mid \bar{I} \mid \leq \mid I \mid$.

Proof. (*sketch*) By definition, a dialogue can be *successful*, *c-successful*, or *unsuccessful*. In the first case, given that $missing(Rs) \in I, R \in Rs$, $\bar{I}$ is such that $missing(\bar{Rs}) \in \bar{I}$, therefore $Rs \supset \bar{Rs} \Rightarrow |I| > |\bar{I}|$. If the dialogue is c-successful, we have that this property holds by definition: again, we have $|I| > |\bar{I}|$. Finally, if the dialogue is unsuccessful, after it has terminated the agent intention has not changed, therefore we have $I = \bar{I}$. ■

Intuitively, there is no rule that will force a change of intention if it does not produce a positive payoff in terms of cost. Therefore, a request dialogue will never result in an increased cost of the plan. In particular, if the dialogue is successful, it is possible to substitute the '$\leq$' relation with '$<$' (and in the case of our particular cost function, we have $\mid \bar{I} \mid = \mid I \mid -1$).

When adopting other sets of integrity constraints, different from $\bar{\mathcal{S}}$, we can ensure their effectiveness for negotiation purposes, by ensuring that property 2 holds for them.

5 The execution model

The formulation of the knowledge of agents as given in section 3 paves the way to the use of the agent architecture in Kowalski and Sadri (1999) to model the behaviour of each agent engaged in a dialogue, compliant with the dialogue protocols expressed by the integrity constraints in $\mathcal{S}$.

The elements introduced so far are: a knowledge base $\mathcal{K}$ including a set $\mathcal{S}$ of integrity constraints, a planner and a shared blackboard for communication. The agent architecture that we propose is therefore the combination of a communication layer, a planning module, and a reasoning module. The *communication layer* implements the shared blackboard that contains the dialogue. We assume that once an agent writes a performative on the dialogue store, the recipient can read it. The writing onto the dialogue store is assumed to be incremental (at least within a dialogue, until it terminates). The *planning module* is initially used to build an intention for an agent. The *reasoning module* is in charge of activating a sequence of dialogues until either there are no more missing resources in the intention, or the intention turns out not to be viable because it is impossible to obtain all its missing resources.

While the initial plan can be obtained statically, the dialogue must be performed dynamically. Therefore, in order to select a resource, trigger the suitable $request$ move, and carry on with the dialogue itself, the reasoning must be part of an *agent cycle* that combines rationality with reactivity. To this purpose we adopt an observe-think-act agent cycle à la Kowalski-Sadri [Kowalski and Sadri (1999)], where observations are blackboard events, thinking is the reasoning part that makes use of $\mathcal{S}$ and acting is writing a dialogue move. These agents adopt the abductive proof-procedure of Fung and Kowalski (1997) as their underlying engine, with the if-then-rules in $\mathcal{S}$ used for forward reasoning, and the given rules (if-definitions) used backwards.

6 Discussion

This paper builds on work on argumentation by Amgoud et al. (2000) and proposes a logical framework for automated negotiation processes. In that, it fills the gaps of previous work, describing in detail the language, the knowledge and the reasoning that is required to build negotiation dialogues between two agents, in the domain of resource achievement, in order to determine some important properties. On the other hand, Amgoud et al. (2000) is more general in that the proposed framework addresses the much broader set of dialogues classified in Walton and Krabbe (1995).

Our work combines automated negotiation issues such as languages and protocol properties with automated reasoning issues (in particular, argumentation systems). Related work in the area of negotiation is enormous, and it is beyond our aims to survey it; those interested could refer to Rosenschein and Zlotkin (1994) for an introduction. A combination of negotiation and argumentation, relevant to our work, is that of Sycara (1989), and more recently Kraus et al. (1998), where the authors adopt a modal logic approach and focus on the mental states of agents that move towards an agreement, and on the way to persuade a counterpart in order to foster cooperation. The authors' formalization of dialogues makes use of rules that, similarly to our abductive logic programs, are triggered when a performative is received, and may lead to another dialogue move under certain conditions, within an agent cycle. The execution is intertwined with the dialogue, allowing the agents to express threats, promises of rewards, etc. and to maintain a mental model of the other partners. Such an approach, focused on persuasion, leads to different results, being somewhat more descriptive, and not aiming to determine specific properties of the negotiation protocol.

In Zlotkin and Rosenschein (1996) the authors adopt a game theory approach to the study of automated negotiation. In such work it is interesting to see how the authors discuss the issue of private information and goal declaration strategy; however, their work – as most of other that are game theory-based – puts a strong stress on the fact that agents are not cooperative, and focuses on conflict resolution rather than collaboration. Our work is somewhat orthogonal, aiming at obtaining the resources the agents need, without taking into account the problems that could arise from disclosing private information. We understand that this aspect is very important, though, especially in an e-commerce context.

From a language point of view, much work has been done in the area of speech act theory, by which our negotiation language is indirectly inspired [Cohen and Perrault (1979); Cohen and Levesque (1995)], and on the semantic issues that arise, especially in terms of verifiability [Wooldridge (2000)] and guaranteed success [Wooldridge and Parsons (2000)]. Our work is again orthogonal to these, and we make the assumption that both agents involved in the dialogue share a common semantics of the speech acts (our performatives), and that they can interpret them within the IFF proof-procedure. Also, our approach is totally independent of the planner, while Cohen and Perrault (1979), aiming to integrate dialogue and plans, make some important assumptions, that they call *heuristic principles of purposeful behaviour*, describing some requirements that a planner must fulfill. On the other hand, the work in Wooldridge (2000) becomes even more relevant as soon as we want to study the properties of the protocol when the set of integrity constraints changes, and thus the behaviour of the agents changes.

As far as argumentation, in our work we mostly use previous results; many exhaustive references on the abductive-argumentative reasoning, on the IFF proof-procedure, and on the KS agent cycle can be found in literature, and in particular in the work cited in the paper.

Our future work is aimed at determining properties of *sequences* of dialogues with respect to an intention. Also, we would like to explore different sets of integrity constraints, exhibiting different behaviours, and to study the applicability of our approach to other application domains.

Acknowledgments

This work has been supported by the M.U.R.S.T. project on *Intelligent Agents: Interaction and Knowledge Acquisition*, and by the E.P.S.R.C. project on *Logic-Based Multi-Agent Systems*. We would like to thank the anonymous referees, Paola Mello, Evelina Lamma, Anna Ciampolini, and Maurelio Boari for their precious comments about a previous version of this work, and Yannis Xanthakos for his help in sketching the proof of the termination property.

References

L. Amgoud, S. Parsons, and N. Maudet. Arguments, dialogue and negotiation. In W. Horn, editor, *Proceedings 14th European Conference on Artificial Intelligence, Berlin, Germany*. IOS Press, August 2000.

P. R. Cohen and H. J. Levesque. Communicative actions for artificial agents. *Proceedings First International Conference on Multiagent Systems, San Francisco, California*, June 1995.

P. R. Cohen and C. R. Perrault. Elements of a plan-based theory of speech acts. *Cognitive Science*, 3(3), 1979.

T. H. Fung and R. A. Kowalski. The iff proof procedure for abductive logic programming. *Journal of Logic Programming*, 1997.

A. C. Kakas, R. A. Kowalski, and F. Toni. The role of abduction in logic programming. *Handbook of Logic in AI and Logic Programming*, 5:235–324, 1998.

R. A. Kowalski and F. Sadri. From logic programming to multi-agent systems. *Annals of Mathematics and AI*, 1999.

S. Kraus, K. Sycara, and A. Evenchik. Reaching agreements through argumentation; a logical model and implementation. *Artificial Intelligence*, 104:1–69, 1998.

S. Parsons, C. Sierra, and N. R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.

J. S. Rosenschein and G. Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers*. MIT Press, Cambridge, Massachusetts, 1994.

F. Sadri and F. Toni. Abduction with negation as failure for active and reactive rules. In E. Lamma and P. Mello, editors, *Proceedings AI*IA 99, 6th Congress of the Italian Association for Artificial Intelligence*, number 1792 in LNAI, pages 49–60. Springer Verlag, 2000.

K. P. Sycara. Argumentation: Planning other agents' plans. In *Proceedings 11th International Joint-Conference on Artificial Intelligence*, pages 517–523. Morgan Kaufman, 1989.

D. N. Walton and E. C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. State University of New York Press, Albany, NY, 1995.

M. J. Wooldridge. Semantic issues in the verification of agent communication languages. *Journal of Autonomous Agents and Multi-Agent Systems*, 2000.

M. J. Wooldridge and S. Parsons. Languages for negotiation. In W. Horn, editor, *Proceedings 14th European Conference on Artificial Intelligence, Berlin, Germany*. IOS Press, August 2000.

G. Zlotkin and J. S. Rosenschein. Mechanisms for automated negotiation in state oriented domains. *Journal of Artificial Intelligence Research*, 5:163–238, 1996.