Protocol Conformance for Logic-based Agents

Ulrich Endriss and **Nicolas Maudet** and **Fariba Sadri** and **Francesca Toni**
Department of Computing, Imperial College London
180 Queen's Gate, London SW7 2AZ, UK
{ue,maudet,fs,ft}@doc.ic.ac.uk

Abstract

An agent communication protocol specifies the "rules of encounter" governing a dialogue between agents in a multiagent system. In non-cooperative interactions (such as negotiation dialogues) occurring in open societies it is crucial that agents are equipped with proper means to check, and possibly enforce, conformance to protocols. We identify different levels of conformance (weak, exhaustive, and robust conformance) and explore how a specific class of logic-based agents can exploit a new representation formalism for communication protocols based on simple if-then rules in order to either check conformance *a priori* or enforce it at runtime.

1 Introduction

A *protocol* specifies the "rules of encounter" governing a dialogue between two or more communicating agents [Rosenschein and Zlotkin, 1994]. It specifies which agent is allowed to say what in any given situation. It will usually allow for several alternative utterances in every situation and the agent in question has to choose one according to its *strategy*. The protocol is *public*, while each agent's strategy is *private*. Protocols can help to define a suitable standardised semantics for agent communication languages in order to allow actual interoperability in open environments. Instead of being related to some (virtually inaccessible) private mental state of the agent as proposed by KQML [Labrou and Finin, 1998] and FIPA [2002], the meaning of a dialogue move refers to some publicly agreed and verifiable conversational state. For instance, one may equate the meaning of a dialogue move with the set of possible responses to this dialogue move as defined by the public protocol [Pitt and Mamdani, 1999]. Of course, this does not disqualify the idea of a private semantics, but simply emphasises that standardisation cannot be achieved at this level. Following these ideas, Pitt and Mamdani [1999] have established the notion of a layered semantics for agent communication languages which integrates both private and public levels, and each agent must implement both of them to actually be able to converse with other agents.

When considering interactions that are not necessarily cooperative (typically negotiation), it cannot be safely assumed that agents will always follow the rules of encounter specified by a particular protocol. It is then crucial to provide proper means of evaluating how well agents are adapted to these protocols, in a sense that has to be precisely defined in the context of the interaction. In this paper, we introduce three different *levels of conformance* (weak, exhaustive, and robust conformance) and show that a logical representation of the protocols for logic-based agents greatly facilitates determining (both *a priori* and at runtime) whether or not these agents behave in conformance to the protocols. We also show how to *enforce* conformance at runtime for such agents and suggest a preliminary definition for the notion of *protocol competence*, an attempt to measure how well an agent is adapted to a given protocol beyond the basic requirements of conformance.

The remainder of this paper is organised as follows. In Section 2 we introduce a new formalism for the representation of protocols based on if-then-rules. Section 3 motivates the introduction of our three levels of conformance. Abductive logic programming has recently been used as a basis for defining (private) strategies for agents negotiating over the distribution of resources [Sadri *et al.*, 2001]. Section 4 introduces these agents, illustrates how to check protocol conformance for them, and shows how the agents themselves can ensure their conformance without requiring any extra reasoning machinery. Section 5 concludes the paper.

2 Representing Protocols

In this paper, we will assume some restrictions on the kind of dialogues that we want to deal with. The dialogues we are considering only involve *two agents* which *sequentially alternate* dialogue moves. These restrictions (notably avoiding concurrency) allow us to concentrate on a particular class of protocols, namely those representable by means of *deterministic finite automata* (DFAs), of which there are numerous examples to be found in the literature [Pitt and Mamdani, 1999; Dignum and Greaves, 2000]. Our aim for this section is to introduce a protocol representation formalism based on simple if-then-rules and to show that it is adequate to represent the kinds of protocols that we are interested in here.

We recall here that a DFA consists of (i) a set of states (including an initial state, and a set of final states), (ii) a set of events, and (iii) a transition function δ which maps pairs of states and events to states. Figure 1 shows an example, taken from [Pitt and Mamdani, 1999]. Events are occurrences of dialogue moves; states are the possible stages of the conver-

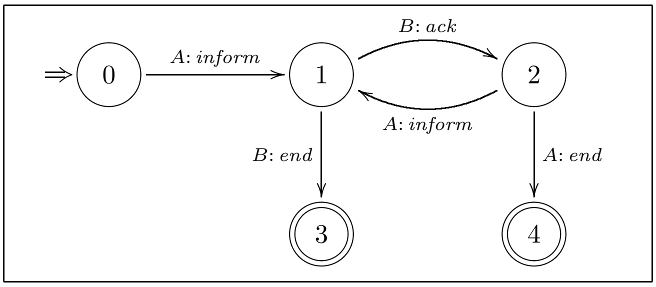

Figure 1: The continuous update protocol

sation. The protocol, entitled "continuous update protocol", aims at continuously updating the other agent about the value of some proposition. Note that it is always the same agent A (the initiator) who informs the other agent B. The *legal* communicative behaviour of both agents is captured by such a protocol. We make this notion of legality precise in the following definition:

Definition 1 (Legality) *Given a DFA with transition function δ, a dialogue move P is a legal continuation wrt. a state S iff there exists a state S' such that $S' = \delta(S, P)$.*

We shall refer to *legal inputs* (respectively *outputs*) for an agent X as those legal continuations where X is the receiver (respectively the utterer) of the dialogue move.

Protocols such as that of Figure 1 can alternatively be represented as sets of if-then-rules which specify the set of correct responses for a particular incoming dialogue move. For example, to express that agent B could react to an *inform* move sent by A either by sending an acknowledgement or by terminating the dialogue, we may use the following rule:

$$\begin{aligned} \mathit{tell}(X, Y, \mathit{inform}(P), D, T) \quad \Rightarrow \quad & \mathit{tell}(Y, X, \mathit{end}, D, T{+}1) \vee \\ & \mathit{tell}(Y, X, \mathit{ack}(P), D, T{+}1) \end{aligned}$$

Note that variables are implicitly universally quantified, except those that only appear on the righthand side of an implication, which are existentially quantified. In general, in this representation, dialogue moves are instances of the schema

$$\mathit{tell}(X, Y, \mathit{Subject}, D, T),$$

where X is the utterer, Y is the receiver ($X \neq Y$), D the identifier of the dialogue, and T the time when the move is uttered. *Subject* is the type of the dialogue moves, i.e. a performative (such as *end*) of the communication language, possibly together with a content (as in *inform*(P)). We shall mostly use the abbreviated form $P(T)$ to dialogue moves, omitting the parameters not relevant in the discussion. For the sake of simplicity, we will assume that the start of the protocol is triggered by some external event *START*—it is possible to conceive this as the result of some meta-level negotiation process to agree on a particular protocol. The start signal *START* is sent by the system to exactly one agent and exactly once during a dialogue. Similarly, a dialogue ends once one of the agents sends the signal *STOP* to the system. Dialogue inputs for the agents are either dialogue moves sent by other agents or a *START* signal sent by the system.

Going back to the example of Figure 1, we observe that this automaton in fact represents two subprotocols, one for the initiator, and one for its partner. We will refer to these two subprotocols as $\mathcal{P}_A$ and $\mathcal{P}_B$. They can be translated into a set (composed of two subsets) of if-then-rules as illustrated in Table 1. In general, given a protocol $\mathcal{P}$, we shall refer to the subprotocol guiding the behaviour of agent α as $\mathcal{P}_\alpha$.

This example suggests that we can simply translate protocols into if-then-rules where we have a single performative on the lefthand side. We call protocols that permit such a straightforward translation *shallow*. Shallow protocols correspond to DFAs where it is possible to determine the next state of the dialogue on the sole basis of the previous event. Of course, this is not always the case since it may be necessary to refer to the current state of the dialogue to determine the new state (think of two transitions with the same label leaving two different states and leading to two different states). In principle, any automata-based protocol can be transformed into a protocol that is shallow in this sense (by simply renaming any duplicate transitions). In fact, we have observed that many of the automata-based protocols proposed in the multiagent systems literature happen to be shallow already or could at least be made shallow by renaming only a small number of transitions.

Now suppose $P_1, ..., P_n$ are the dialogue moves of our communication language (including *START* and *STOP*). In the light of the above remarks, we will more generally translate DFAs into two sets of rules of the form

$$P_j(T) \Rightarrow \bigvee_{i \in I} P_i(T{+}1) \text{ with } 1 \leq j \leq n \text{ and } I \subseteq \{1, \ldots, n\}$$

where the righthand side of the rule defines the possible continuations wrt. the protocol after the input P_j (that we will sometimes refer to as the *trigger* of the rule). To ensure that this protocol is *well-formed*, we will require that the two sets of rules meet some requirements (R1–R5): there has to be at least one rule with *START* on the lefthand side in the protocol, and *START* may never occur on the righthand side (R1a, initial); there is at least one rule with *STOP* on the righthand side and *STOP* never occurs on the lefthand side (R1b, final); any dialogue move occurring on the righthand side of the first subprotocol also occurs on the lefthand side of the second one, and vice versa (R2, matching); every subprotocol includes the additional rule

$$\mathit{tell}(X, Y, S_1, T, D) \wedge \mathit{tell}(X, Y, S_2, T, D) \wedge S_1 \neq S_2 \Rightarrow \bot$$

to avoid concurrent moves (R3, non-concurrency); for each rule occurring in a subprotocol, if X is the receiver and Y the utterer of the dialogue move occurring on the lefthand side, it must be the case that X is the utterer and Y the receiver of every dialogue move occurring on the righthand side (R4, alternating); all dialogue moves occurring on the lefthand side of the rules of each subprotocol are distinct from each other (R5, distinct triggers).

The meaning of each rule which appears in a protocol is intuitively clear: it specifies for any expected dialogue move the set of correct responses the agent may utter in reply. The following definitions make these notions more precise:

Definition 2 (Expected inputs) *The set of expected inputs for an agent α wrt. a protocol $\mathcal{P}$ is defined as:*

$$\{P_j \mid [P_j(T) \Rightarrow \bigvee P_i(T{+}1)] \in \mathcal{P}_\alpha\}$$

$$\mathcal{P}_A : \left\{ \begin{array}{lll} \mathit{START}(X,Y,D,T) & \Rightarrow & \mathit{tell}(Y,X,\mathit{inform}(P),D,T{+}1) \\ \mathit{tell}(X,Y,\mathit{ack}(P),D,T) & \Rightarrow & \mathit{tell}(Y,X,\mathit{inform}(P'),D,T{+}1) \ \vee \mathit{tell}(Y,X,\mathit{end},D,T{+}1) \\ \mathit{tell}(X,Y,\mathit{end},D,T) & \Rightarrow & \mathit{STOP}(Y,X,D,T{+}1) \end{array} \right.$$

$$\mathcal{P}_B : \left\{ \begin{array}{lll} \mathit{tell}(X,Y,\mathit{inform}(P),D,T) & \Rightarrow & \mathit{tell}(Y,X,\mathit{ack}(P),D,T{+}1) \ \vee \mathit{tell}(Y,X,\mathit{end},D,T{+}1) \\ \mathit{tell}(X,Y,\mathit{end},D,T) & \Rightarrow & \mathit{STOP}(Y,X,D,T{+}1) \end{array} \right.$$

Table 1: The continuous update protocol as a set of if-then-rules

Definition 3 (Correct responses) *The set of correct responses for an agent α wrt. a protocol $\mathcal{P}$ and a dialogue move P_j is defined as:*

$$\{P_i \mid [P_j(T) \Rightarrow \bigvee P_i(T{+}1)] \in \mathcal{P}_\alpha\}$$

Intuitively it is not sufficient for a dialogue move to be merely *expected* in order to be considered *legal;* it is also necessary that the move takes place at the right moment in the history of the dialogue. But note that it follows from the property of matching (R2) that the set of legal *inputs* for an agent at a certain stage of a dialogue is a subset of its expected inputs.

3 Levels of Conformance

Taking for granted that agents share the same language of communication, we are now going to define three different *levels of conformance* to a protocol. Note that we define these notions on the basis of the *observable* conversational behaviour of the agents (i.e. what they *utter*) without making further assumptions on how they actually come to *generate* these utterances. We start with the notion of *weak conformance:*

Definition 4 (Weak conformance) *An agent is weakly conformant to a protocol $\mathcal{P}$ iff it never utters any illegal dialogue moves (wrt. $\mathcal{P}$).*

The following theorem shows that, in the context of our shallow protocols, the concept of legality is reducible to that of correctness, meaning that we can avoid to inspect the legality of the input and thus avoid to deal with the complete dialogue history.

Theorem 1 *An agent that never utters an incorrect response in reply to an expected input of a shallow protocol $\mathcal{P}$ is weakly conformant to $\mathcal{P}$.*

Proof. (sketch) For shallow protocols, the current dialogue state is uniquely identifiable given the latest move in the dialogue. Hence, the notions of correct response and legal move coincide, i.e. an agent that never utters an incorrect response will never choose an illegal continuation and will therefore be weakly conformant to the protocol. □

It is clear that any application governed by a protocol at least requires the level of weak conformance—otherwise it would not make sense to define a protocol in the first place. The notion of weak conformance captures that the agent does *not* utter any illegal moves, but does not actually require that the agent utters any dialogue move at all. For interactions where "silent moves" are undesirable, a stronger version of conformance is usually required. We make this idea precise with the notion of *exhaustive conformance:*

Definition 5 (Exhaustive conformance) *An agent is exhaustively conformant to a protocol $\mathcal{P}$ iff it is weakly conformant to $\mathcal{P}$ and it will utter at least one legal output move for any legal input of $\mathcal{P}$ it receives.*

Exhaustive conformance is certainly what is intuitively expected in *most* interactions—it is indeed often preferred to avoid considering silent moves as part of a protocol, at least to avoid confusion with lost messages. One may then argue that exhaustive conformance should be the minimum requirement for any interaction. We believe, however, it is worth making the distinction between weak and exhaustive conformance. The first reason is that there are examples where the lack of response can be considered to be part of the protocol. In such circumstances, it can be sufficient to design a weakly conformant agent, provided that silent moves will not have undesirable consequences. For instance, in a Dutch auction process "when there is no signal of acceptance from the other parties in the auction (other agents in the negotiation) the auctioneer makes a new offer which he believes more acceptable (by reducing the price). Here, because of the convention (protocol) under which the auction operates, a lack of response is sufficient feedback for the auctioneer to infer a lack of acceptance." [Jennings *et al.*, 1998]. In this case, the agent can safely be designed to react appropriately only to the proposals it is ready to accept. But if we consider recent argumentation-based protocols inspired by dialectical models it is sometimes assumed that "silence means consent" [Amgoud *et al.*, 2000]. In this case, a lack of response can commit the receiver to some propositions—this is a typical case where it is crucial that agents are exhaustively conformant. The second reason for our distinction of weak and exhaustive conformance is that they are *conceptually* different since weak conformance only involves *not* uttering (any illegal moves), while exhaustive conformance involves uttering (some legal move). This implies substantially different approaches when the issues of checking and enforcing conformance are raised, as we shall see below.

Another important problem of agent communication is the need to deal with illegal incoming messages, and to react appropriately to recover from such violations. For instance, any FIPA-compliant communicative agent has to integrate a performative `not-understood` as part of its language [FIPA, 2002]. This motivates us to introduce the following notion of *robust conformance:*

Definition 6 (Robust conformance) *An agent is robustly conformant to a protocol $\mathcal{P}$ iff it is exhaustively conformant to $\mathcal{P}$ and for any illegal input move it will utter a special dialogue move (such as* `not-understood`*).*

Robust conformance goes a step further than exhaustive conformance since it requires that an appropriate response is uttered also in reply to illegal moves. Technically, this necessitates that the agent is able to identify the legality of an incoming dialogue move, i.e. it needs to be able to check conformance wrt. the *other* agent's subprotocol.

Note also that in the case where all agents in the society are known to be weakly conformant, it is theoretically unnecessary to deal with robust conformance (since no agent will ever utter an illegal move). Such an assumption would, however, somewhat contradict the "spirit" of an open society. We should also point out that in dialogues with a very high contingent of illegal utterances the additional `not-understood` moves may in fact burden communication channels unnecessarily and simply ignoring illegal moves would in fact be a better strategy.

4 Logic-based Agents

We are now going to consider the case of a specific class of agents based on abductive logic programming that have recently been used in the context of negotiation scenarios [Sadri *et al.*, 2001]. The communication *strategy* $\mathcal{S}$ of such an agent (which forms part of its so-called *knowledge base* $\mathcal{K}$) is represented as a set of integrity constraints of the following form:

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

On receiving dialogue move P at time T, an agent implementing this rule would utter P' at time $T+1$, provided condition C is entailed by its (*private*) knowledge base. Again, variables are understood to be implicitly quantified in the same way as our protocol-rules.

4.1 Checking Conformance

There are different ways of checking conformance. One way is to check conformance *on-the-fly*, i.e. to check step by step that every dialogue move uttered by an agent is conformant to the protocol. Another way—much more interesting when one has to design an agent to take part in an interaction—is to check conformance *a priori*, i.e. to check conformance by examining the specification of the agent rather than an actual dialogue. In general, this is a difficult task, because (i) the behaviour of the agent depends on some hardly tractable notions (e.g.. beliefs and intentions), and (ii) conformance may depend on the history of the dialogue. We are now going to discuss some simple sufficient conditions for weak conformance (in the context of the logic-based agents introduced earlier) that may be checked *a priori*.

To begin with, we introduce the notion of *response space*. Intuitively, the response space of an agent specifies the possible moves that the agent can make when using a given strategy $\mathcal{S}$, *without* considering the specific conditions relating to its private knowledge base.

Definition 7 (Response space) *The response space of an agent with strategy $\mathcal{S}$ (henceforth noted $\mathcal{S}^*$) wrt. a communication language $\mathcal{L}$ is defined as the following set:*

$$\{P_j(T) \Rightarrow \bigvee\{P_i(T{+}1) \mid [P_j(T) {\wedge} C \Rightarrow P_i(T{+}1)] \in \mathcal{S}\} \mid P_j \in \mathcal{L}\}$$

with $\bigvee\{\} = \bot$.

That is, the response space is, essentially, the set of protocol-constraints we get by first dropping all private conditions C and then conjoining implications with identical antecedents by collecting the corresponding consequents into a single disjunction. The reason why we define the disjunction of the empty set as $\bot$ will become clear when we consider the next theorem, which offers a very simple way to check weak conformance *a priori* for a logic-based agent. In particular, it avoids dealing with the dialogue history, and it does not make any assumptions on the content of the agent's knowledge base (except to require that it is possible to extract the response space, as previously described).

Theorem 2 *Let $\mathcal{P}$ be a protocol and let $\mathcal{S}^*$ be the response space of an agent A wrt. the language $\mathcal{L}$ of moves occurring in $\mathcal{P}$. If $\mathcal{S}^* \models \mathcal{P}$, then agent A is weakly conformant to $\mathcal{P}$.*

Proof. Let $\mathcal{S}^*$ be the response space of agent A wrt. the language $\mathcal{L}$ of protocol $\mathcal{P}$. Then for every rule $P \Rightarrow P_1 \vee \cdots \vee P_n$ in $\mathcal{P}$, $\mathcal{S}^*$ will contain an implication $P \Rightarrow P'_1 \vee \cdots \vee P'_m$ (with the same antecedent P)—and possibly a number of additional implications for dialogue moves in the language $\mathcal{L}$ that do not occur as triggers in $\mathcal{P}$. (All the antecedents of the implications in $\mathcal{S}^*$ are distinct.) Now suppose $\mathcal{S}^* \models \mathcal{P}$ holds. Observe that a formula of the form $P \Rightarrow P_1 \vee \cdots \vee P_n$ in $\mathcal{P}$ can only be the consequence of an implication in $\mathcal{S}^*$ with the same antecedent P. But $P \Rightarrow P_1 \vee \cdots \vee P_n$ will only follow from $P \Rightarrow P'_1 \vee \cdots \vee P'_m$ provided we have $\{P'_1, \ldots, P'_m\} \subseteq \{P_1, \ldots, P_n\}$.

In other words, whenever $\mathcal{S}^* \models \mathcal{P}$ holds, then $\mathcal{S}^*$ will have the shape of a protocol that is a "syntactic restriction" of the protocol $\mathcal{P}$, possibly together with a number of irrelevant rules (with triggers not present in $\mathcal{P}$). Furthermore, by construction, any agent will be weakly conformant to the "protocol" represented by its response space. Hence, agent A will also be weakly conformant to $\mathcal{P}$, because any dialogue continuation that would be legal wrt. $\mathcal{S}^*$ would certainly have to be legal wrt. $\mathcal{P}$. □

The opposite direction of Theorem 2 does not hold, because, looking at the form of strategies, it is clear that *private* conditions may prevent the agent from uttering a particular dialogue move. In other words, it could be the case that $\mathcal{S}^* \not\models \mathcal{P}$ but that the agent is still weakly conformant to $\mathcal{P}$ because of its specific knowledge base.

The same argument prevents us from having a theorem similar to Theorem 2 in the case of *exhaustive conformance* (and of course for robust conformance). Here we would have to look more precisely at how dialogue moves are actually being generated. The basic idea would be to check that, for every expected input in the protocol, the disjunction of the conditions C related to this move in the agent's strategy is entailed by the agent's knowledge base.

4.2 Enforcing conformance

Even when Theorem 2 is not applicable and an agent cannot be proven to be weakly conformant *a priori*, it is still possible to constrain the agent's behaviour in such a way as to simply prevent it from uttering illegal moves. The problem of *enforcing* conformance (referred to as *regimentation* by Jones

$$\mathcal{S}_1 : \begin{cases} \mathit{tell}(X, a, \mathit{inform}(P), D, T) \wedge \neg\mathit{friend}(a, X) & \Rightarrow \quad \mathit{tell}(a, X, \mathit{end}, D, T+1) \\ \mathit{tell}(X, a, \mathit{inform}(P), T) \wedge \mathit{friend}(a, X) & \Rightarrow \quad \mathit{tell}(a, X, \mathit{ack}, D, T+1) \\ \mathit{tell}(X, a, \mathit{ack}, D, T) & \Rightarrow \quad \mathit{tell}(a, X, \mathit{ack}, D, T+1) \\ \mathit{tell}(X, a, \mathit{question}(P), T) \wedge \mathit{friend}(a, X) & \Rightarrow \quad \mathit{tell}(a, X, \mathit{inform}(P), D, T+1) \\ \mathit{tell}(X, a, \mathit{end}, D, T) & \Rightarrow \quad \mathit{STOP}(a, X, D, T+1) \end{cases}$$

$$\mathcal{S}_2 : \begin{cases} \mathit{tell}(X, a, \mathit{inform}(P), D, T) \wedge \mathit{enemy}(a, X) & \Rightarrow \quad \mathit{tell}(a, X, \mathit{end}, D, T+1) \\ \mathit{tell}(X, a, \mathit{inform}(P), D, T) \wedge \mathit{friend}(a, X) & \Rightarrow \quad \mathit{tell}(a, X, \mathit{ack}, D, T+1) \\ \mathit{tell}(X, a, \mathit{inform}(P), D, T) \wedge \neg\mathit{friend}(a, X) \wedge \neg\mathit{enemy}(a, X) & \Rightarrow \quad \mathit{tell}(a, X, \mathit{challenge}(P), D, T+1) \\ \mathit{tell}(X, a, \mathit{ack}, D, T) & \Rightarrow \quad \mathit{tell}(a, X, \mathit{ack}, D, T+1) \\ \mathit{tell}(X, a, \mathit{challenge}(P), D, T) & \Rightarrow \quad \mathit{tell}(a, X, \mathit{end}, D, T+1) \\ \mathit{tell}(X, a, \mathit{end}, D, T) & \Rightarrow \quad \mathit{STOP}(a, X, D, T+1) \end{cases}$$

Table 2: Examples of dialogue strategies

and Sergot [1993]) is then to try to find easy (and hopefully automatic) ways to ensure that an agent will always be conformant to a given protocol.

We will now show how a simple filtering function may be defined in the context of our logic-based agents. Since this relies on the concrete behaviour of our agents, this requires some details about the operational model which underlies these agents. By a generated response we now mean a response produced by the abductive IFF proof procedure of Fung and Kowalski [1997]. In our case, the abducibles are the dialogue move(s) that the agent will actually generate as the consequence of the observation of another agent's dialogue move (see [Sadri *et al.*, 2002] for details).

Theorem 3 *An agent generating its dialogue moves wrt. the knowledge base $\mathcal{K} \cup \mathcal{P}$ will be weakly conformant to $\mathcal{P}$.*

Proof. (sketch) Let us assume that $P(T)$ is a legal input (wrt. protocol $\mathcal{P}$) that our agent receives, and that $P'(T+1)$ is a move generated as a response by the agent. Since the move is legal, it is expected in $\mathcal{P}$. The proof procedure will produce a tree and each branch of this tree will have one of the correct responses $P_i(T+1)$ together with the constraints of the dialogue strategy and the additional condition of non-concurrency (R3). Now if $P'(T+1)$ is a generated response of the proof procedure, $P'(T+1)$ must also be a conjunct on one of these branches. But such a branch will evaluate to false if $P'(T+1) \neq P_i(T+1)$, precisely because of the condition of non-concurrency. This means that the procedure will never generate an incorrect response. Therefore, by Theorem 1, our agent will be weakly conformant to $\mathcal{P}$. □

Thus the filtering function is provided by the protocol itself. This result has very practical consequences. For instance, before entering an interaction, an agent can ensure its weak conformance to the public protocol which regulates the interaction (and thereby avoid possible penalties imposed by society), by adding the protocol to its own knowledge base. Now, while we can prevent an agent from uttering illegal dialogue moves, it is difficult to see how we could force it to utter some dialogue move (exhaustive conformance) without considering its private knowledge base and modifying its private strategy. We therefore believe that exhaustive conformance cannot be enforced automatically and should ideally be ensured during the specification of the agent by the designer.

Note that this filtering function will not block correct responses generated by the agent, unless two or more moves are being generated at the same time. This is again due to the non-concurrency condition (R3).

We should also point out that it is possible that an agent with knowledge base $\mathcal{K}$ would not utter any dialogue move given a particular input $P(T)$, while an agent with knowledge base $\mathcal{K} \cup \mathcal{P}$ (i.e. the same agent after "downloading" the protocol $\mathcal{P}$) *would* utter a (legal) move. This may, for instance, be the case if $P(T)$ never occurs on the lefthand side in the agent's original strategy and $\mathcal{P}$ includes a "deterministic" protocol rule such as $P(T) \Rightarrow P'(T+1)$. If this type of behaviour is not intended, we have to ensure that the communication languages of the agent and the protocol match (for instance, by adding the constraint $P(T) \Rightarrow \bot$ to the agent's strategy for every move P that is an expected input in $\mathcal{P}$).

4.3 Examples

We are now going to illustrate some of the points made earlier by means of example. Consider again the continuous update protocol of Table 1, and more specifically subprotocol $\mathcal{P}_B$. The set of constraints given in Table 2 represents two possible strategies for an agent a. Here, the $\neg$-operator is understood as negation as failure. We assume that the predicates *friend* and *enemy* are fully defined in the knowledge base of our agent (that is, it can determine whether or not an agent X is a friend or an enemy), and we also assume that an integrity constraint stipulates that it is not possible to be at the same time friend and enemy of the same agent ($\mathit{friend}(a, X) \wedge \mathit{enemy}(a, X) \Rightarrow \bot$).

Dialogue strategy $\mathcal{S}_1$ relies upon the communication language $\mathcal{L}_1$ which is based on the set $\{\mathit{inform}(P), \mathit{question}(P), \mathit{ack}, \mathit{end}\}$. It is interesting to note that no rule can be triggered after a *question* about P if it is the case that the agent a does not consider X as a friend. However, the agent a is still weakly and exhaustively conformant to $\mathcal{P}_B$ since this move can never be legally uttered within the protocol $\mathcal{P}$ by its partner (it is not a legal input for subprotocol $\mathcal{P}_B$).

Next consider dialogue strategy $\mathcal{S}_2$, which relies upon the communication language $\mathcal{L}_2$ based on the set $\{\mathit{inform}(P), \mathit{challenge}(P), \mathit{ack}, \mathit{end}\}$. Agent a is not weakly conformant to $\mathcal{P}_B$ since, in some situations (namely, when it does not consider X as friend, neither as an enemy), it may want to *challenge* the initiator X. This move is not expected in the protocol $\mathcal{P}_A$. If the agent a decides to enforce weak conformance as described earlier in Theorem 3, it will remain silent in the situation previously described.

Finally, let us consider an agent with the response space $\{P \Rightarrow \perp \mid P \in \mathcal{L}\}$, that is, an agent that never utters any dialogue moves at all. It is clear that such an agent will be weakly conformant to any protocol. This certainly suggests that the notion of conformance *alone* is not sufficient to evaluate how well an agent is adapted to a protocol.

Another application of the notion of response space could be to assess how well an agent can explore a given protocol beyond the minimal requirement of being able to conform to it. Intuitively, given a protocol $\mathcal{P}$, we would expect a "competent" agent to have a response space that (almost) "covers" $\mathcal{P}$, namely it has the potential to utter as many dialogue moves as the protocol allows. This would offer a notion complementary to that of conformance.

Note also that none of the agents discussed here is robustly conformant. One of the reasons for this latter remark is the fact that these agents cannot notice that the dialogue move they receive is not legal if it is an expected input (for instance if their partner starts the dialogue with an *ack*). They would consequently react "appropriately" to these moves—which can be quite confusing.

5 Conclusion

This paper has introduced different levels of conformance as basic notions to check and enforce that the behaviour of an agent is adapted to a public protocol regulating the interaction in a multiagent system. These notions have been explored in the context of the logic-based agents of Sadri *et al.* [2001]. Our approach starts from on an alternative representation formalism for communication protocols based on if-then-rules for the kinds of protocols that can be represented as DFAs. In particular, we argue that it is typically not necessary to consider the history of the dialogue besides the previous move to determine the possible legal dialogue continuations (*shallowness*). This allows us to show that a sufficient condition for an agent to be weakly conformant is to never give an incorrect response in reply to an expected input. In the context of logic-based agents we have used this result to propose a simple way to check a sufficient condition for weak conformance, which has the great advantage of ignoring the conditions related to the (private) knowledge base (by identifying an agent's *response space*), and avoids to consider the dialogue history. We have then shown how the concrete behaviour of the IFF proof procedure can be used to define a regimentation policy to enforce weak conformance. Finally, we have illustrated these notions and their limitations by means of examples.

A number of results have previously been obtained in the context of agents based on abductive logic programming, e.g. the termination of certain types of negotiation dialogues [Sadri *et al.*, 2001]. On the one hand, because we mostly ignore the details of the knowledge bases of agents and do not restrict ourselves to any specific protocol here, we cannot obtain such strong results. On the other hand, our work provides more general tools for logic-based agents communicating by means of (various) public protocols.

The work presented here can be extended in a number of ways. In particular, we only deal with very specific kinds of dialogues (although they can be generated by a wide range of protocols). Amongst other things, we plan to explore whether our formalism is expressive enough to capture more complex dialogues, such as those taking place between more than two agents or those not representable using DFAs (e.g. dialogues where concurrency is allowed or where checking conformance requires reference to the content of a dialogue move rather than just the communicative act itself).

Acknowledgements. We would like to thank the IJCAI referees for their helpful comments. This research has been funded by the European Union as part of the SOCS project (Societies Of ComputeeS), IST-2001-32530.

References

[Amgoud *et al.*, 2000] L. Amgoud, N. Maudet, and S. Parsons. Modelling Dialogues using Argumentation. In *Proc. ICMAS-2000*, 2000.

[Dignum and Greaves, 2000] F. Dignum and M. Greaves. Issues in Agent Communication: An Introduction. In F. Dignum and M. Greaves, editors, *Issues in Agent Communication*. Springer-Verlag, 2000.

[FIPA, 2002] Foundation for Intelligent Physical Agents (FIPA). *Communicative Act Library Specification*, 2002. http://www.fipa.org/specs/fipa00037/.

[Fung and Kowalski, 1997] T. H. Fung and R. Kowalski. The IFF Proof Procedure for Abductive Logic Programming. *Journal of Logic Programming*, 33(2):151–165, 1997.

[Jennings *et al.*, 1998] N. Jennings, S. Parsons P. Noriega, and C. Sierra. On Argumentation-based Negotiation. In *Proc. IWMAS-1998*, 1998.

[Jones and Sergot, 1993] A. Jones and M. Sergot. On the Characterisation of Law and Computer Systems: The Normative Systems Perspective. In *Deontic Logic in Computer Science: Normative System Specification*. John Wiley and Sons, 1993.

[Labrou and Finin, 1998] Y. Labrou and T. Finin. Semantics and Conversations for an Agent Communication Language. In M. Huhns and M. Singh, editors, *Readings in Agents*. Morgan Kaufmann Publishers, 1998.

[Pitt and Mamdani, 1999] J. Pitt and A. Mamdani. A Protocol-based Semantics for an Agent Communication Language. In *Proc. IJCAI-1999*. Morgan Kaufmann, 1999.

[Rosenschein and Zlotkin, 1994] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter*. MIT Press, 1994.

[Sadri *et al.*, 2001] F. Sadri, F. Toni, and P. Torroni. Dialogues for Negotiation: Agent Varieties and Dialogue Sequences. In *Proc. ATAL-2001*. Springer-Verlag, 2001.

[Sadri *et al.*, 2002] F. Sadri, F. Toni, and P. Torroni. An Abductive Logic Programming Architecture for Negotiating Agents. In *Proc. JELIA-2002*. Springer-Verlag, 2002.