Multi-agent Abductive Reasoning with Confidentiality*

(Extended Abstract)

Jiefei Ma, Alessandra Russo, Krysia Broda, Emil Lupu
Department of Computing, Imperial College London
180 Queen's Gate, London, United Kingdom, SW7 2AZ
{j.ma, a.russo, k.broda, e.c.lupu}@imperial.ac.uk

ABSTRACT

In the context of multi-agent hypothetical reasoning, agents typically have partial knowledge about their environments, and the union of such knowledge is still incomplete to represent the whole world. Thus, given a global query they need to collaborate with each other to make correct inferences and hypothesis, whilst maintaining global constraints. There are many real world applications in which the confidentiality of agent knowledge is of primary concern, and hence the agents may not share or communicate all their information during the collaboration. This extra constraint gives a new challenge to multi-agent reasoning. This paper shows how this dichotomy between "open communication" in collaborative reasoning and protection of confidentiality can be accommodated, by extending a general-purpose distributed abductive logic programming system for multi-agent hypothetical reasoning with confidentiality. Specifically, the system computes consistent conditional answers for a query over a set of distributed normal logic programs with possibly unbound domains and arithmetic constraints, preserving the private information within the logic programs.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms

Keywords

Reasoning (Multi-agent), Knowledge Representation, Distributed Problem Solving

1. INTRODUCTION

In the context of multi-agent reasoning, each agent has its own *partial knowledge* about the world together with local and/or global constraints. Given a reasoning task, agents need to interact and compute answers that are consistent with respect to the global constraints. In the case where the union of all the agent knowledge is still incomplete to represent the whole world, hypothetical reasoning is needed, and agents need to collaborate to make correct inferences and hypotheses given a global query. Previously, a general-purpose system called DAREC [3] has been developed, which combines distributed problem solving and abductive logic programming, for multi-agent hypothetical reasoning. Agent knowledge in DAREC is represented as a normal logic program, and a distributed abductive logic programming algorithm is used to coordinate the agents' local reasoning tasks. Thus, agents compute local conditional answers, by assuming the undefined knowledge that is needed to maintain their (global) constraints, and coordinate their proofs through consistency checks over their respective assumptions. DAREC is the first distributed abductive system that can compute non-ground answers and handle arithmetic constraints.

However, in DAREC all knowledge is considered public and hence during collaboration agents are free to communicate any information they may have. This assumption may not hold in application domains where confidentiality is an additional primary concern, e.g., policy analysis of a distributed network formed by devices belonging to different parties. In such problem settings, agents may contain private information that cannot be shared with others during, or after, the reasoning, and hence they must decide what to disclose between their communications. This paper addresses the new challenge of extending DAREC for multi-agent hypothetical reasoning with confidentiality. There are two main contributions. At knowledge representation level we have extended the logical language and the distributed abductive framework to allow modelling of private agent knowledge. At the algorithmic level, we have extended the distributed proof procedure with a *safe* yet efficient agent interaction protocol, which prevents private knowledge being passed between agents and allows a degree of concurrent computation. The new system may be used in several ways. For example, each abductive agent could be implemented as a reactive reasoning module of an agent (with well-known agent architectures, such as BDI) in a larger MAS to support other agent/system functionalities. Alternatively, the whole system could be implemented as a "simulator" to verify properties or behaviour of a target MAS (i.e., each agent in the target MAS is represented by an abductive agent).

2. KNOWLEDGE REPRESENTATION

Standard abductive logic programming [1] and DAREC notations are used throughout the paper. Each agent is modelled as an abductive framework $\mathcal{F} = \langle \Pi, \mathcal{AB}, \mathcal{IC} \rangle$, where

*This research is continuing through participation in the International Technology Alliance sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence.

Cite as: Multi-agent Abductive Reasoning with Confidentiality (Extended Abstract), Jiefei Ma, Alessandra Russo, Krysia Broda and Emil Lupu, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1137-1138.

$\mathcal{AB}$ is the set of all *abducible atoms*, Π is a (finite) set of rules $H \leftarrow L_1, \ldots, L_n$ ($n \geq 0$) called the *local background knowledge*, and $\mathcal{IC}$ is a (finite) set of denials $\leftarrow L_1, \ldots, L_n$ ($n > 0$) called the *integrity constraints*, where H is an atom and each L_i is a literal. In our new system, a new type of atom, called *askable*, is introduced in addition to the abducible (atoms), the *non-abducible (atoms)* and the *(arithmetic) constraint (atoms)*. An askable is $p(\vec{t})@Ag$ where $p(\vec{t})$ is a non-abducible and Ag is either a variable or a constant representing an agent identifier. Intuitively, during the collaborative reasoning process an askable sub-goal $p(\vec{t})@Ag$ means it should (only) be solved by agent Ag, or "(only) agent Ag has knowledge of/can be asked about it". Thus, a negative askable literal should be read as $\neg(p(\vec{t})@Ag)$ and not as $(\neg p(\vec{t}))@Ag$. Non-abducible are considered *private* to agents; whereas askables (as well as abducibles) are shared between agents. Only non-abducible and askable atoms can appear in the head of a rule. A *global abductive framework* is a pair $\langle \Sigma, \widehat{\mathcal{F}} \rangle$ denoting the sets of all agent identifiers and frameworks respectively, with the assumption that the set of all abducible atoms is agreed by everyone, i.e., $\mathcal{AB}_i = \mathcal{AB}_j$ for any $i, j \in \Sigma$. Given a query $\mathcal{Q}$, the task of multi-agent hypothetical reasoning with confidentiality is to compute a subset of abducibles $\Delta \subseteq \mathcal{AB}$ such that (i) $\bigcup_{i \in \Sigma} \Pi_i \cup \Delta \models \mathcal{Q}$, (ii) $\bigcup_{i \in \Sigma} \Pi_i \cup \Delta \models \bigcup_{i \in \Sigma} \mathcal{IC}_i$, and (iii) no (reasoning of) private non-abducibles of an agent are disclosed to others.

3. DISTRIBUTED ALGORITHM

From the operational point of view, our new distributed abductive algorithm is a coordinated state rewriting process, consisting of a series of *local abductive inferences* by the agents and *coordination* of these local inferences. The local inference is a top-down (goal-directed) reasoning process, where a current agent (i) solves as many sub-goals of the query as possible, using its own knowledge, and (ii) collects those sub-goals that are solvable only by other agents (i.e., the askables), and the constraints that must be satisfied by all agents to guarantee *global consistency* of the final answer. These are generated from *constructive negations* and *arithmetic constraints* during the local inference process. They can be reduced to a set of inequalities and arithmetic constraints and be handled by external Constraint Logic Programming (CLP) solvers, enabling also reasoning over unbounded domains. The collected sub-goals and constraints, together with the hypotheses made during the local inference, are encapsulated into a *token state*, which is then passed around to other agents for further processing once all private sub-goals (i.e., non-abducibles) of the current agent have been solved by the agent. This guarantees that confidential information is not included in the token state and not passed to other agents. The coordination of state-passing implements *synchronised backtracking*, whilst enabling concurrent computation between local inferences. The coordination allows two types of agent interaction: *positive* and *negative*. In the case of a positive interaction, the token state is directed to a suitable helper agent (i.e. who may help to solve some pending sub-goals), whereas for negative interactions, it is passed among all agents enforcing each to check the pending constraints. Application dependent strategies may be adopted to interleave/combine such interactions in order to reduce communication overheads.

4. APPLICATIONS AND EXPERIMENTS

The distributed abductive algorithm is proven to be *sound* with respect to the three-valued completion semantics for abductive logic programs [5], and *complete* upon termination of the execution. The terminating condition depends on the structure of the overall logic program formed by the union of all the agent frameworks, i.e., it is hierarchical or abductive acyclic [6]. The System has been implemented in YAP Prolog 6 [1]. It has also been tested for decentralised policy analysis (e.g., modality conflict detection and system behaviour simulation), where each node of a distributed systems has private security policies and domain information modelled as a normal logic program within the formal policy framework proposed by Craven et. al. [4]. Note that the policy language in the framework can guarantee the overall logic program is abductive acyclic, and hence guarantees the termination of the distributed abductive task.

5. CONCLUSION AND FUTURE WORK

Confidentiality in knowledge is one important constraint that makes a multi-agent reasoning problem challenging, and it is also a very common assumption in MAS's. Our main contributions include (1) a logical framework for modelling the distributed knowledge of a multi-agent system where the agents' background knowledge are correlated and have private information, and (2) a top-down distributed abductive algorithm which allows agents to perform collaborative hypothetical reasoning without disclosing private information. Furthermore, by limiting the set of abducible atoms to be empty, the system becomes a general purpose distributed deductive theorem prover that performs constructive negation whilst maintaining confidentiality. This is very useful when dealing with logic programs with unbound domains that cannot be implemented with a bottom-up algorithms such as answer set programming (ASP). The system has many potential applications including decentralised security policy analysis.

There are some applications where the separation between public and private hypotheses is desired in collaborative reasoning, e.g., distributed planning/scheduling with confidentiality [2]. As future work, we would like to extend our system to handle private abducible predicates, and to perform extensive benchmarking for the system.

6. REFERENCES

[1] A.C.Kakas and P.Mancarella. Abductive logic programming. In *LPNMR*, pages 49–61, 1990.

[2] J.Ma, A.Russo, K.Broda, and E.Lupu. Multi-agent planning with confidentiality. In *AAMAS (2)*, pages 1275–1276, 2009.

[3] J. Ma, K. Broda, A. Russo, and E. Lupu. Distributed abductive reasoning with constraints. In *Post-proceedings of DALT-10*, 2011.

[4] R.Craven, J.Lobo, J.Ma, A.Russo, E.C.Lupu, and A.Bandara. Expressive policy analysis with enhanced system dynamicity. In *Proc. of 4th ASIACCS*, pages 239–250, 2009.

[5] F. Teusink. Three-valued completion for abductive logic programs. *Theor. Comput. Sci.*, 165(1):171–200, 1996.

[6] S. Verbaeten. Termination analysis for abductive general logic programs. In *Int. Conf. on LP*, pages 365–379, 1999.

[1] `http://www.dcc.fc.up.pt/~vsc/Yap/`