Distributed Medical Diagnosis with Abductive Logic Agents

A. Ciampolini, P. Mello, S. Storari [1]

Abstract. We describe the application of a multi-agent system for the distributed diagnosis of infections within an hospital. Diagnosing infections within an hospital is a complex task that may require to collect data (e.g. analysis results, details on patient's clinical history, diagnosis hypotheses) from several information sources (such as, for example, analysis laboratories, hospital wards. These sources act autonomously and they often have a partial knowledge about patients health, their clinical history, and medical information in general. As a natural consequence, this may lead a single entity (e.g., a specialist) to formulate incorrect diagnosis. In such a context, to obtain a correct diagnosis on the basis of information coming from different sources, a coordination mechanism is needed for the integration of collected data into a final diagnosis which should be compatible both with patient's anamnesis and other knowledge (possibly distributed over the system) related to the clinical case. In the paper we face this problem by using abduction, which is a reasoning mechanism for formulating hypotheses in the case of incomplete knowledge, suitably etended to a multi-agent setting. In particular, we first apply *ALIAS* abductive agents to distributed diagnosis and show how the coordination mechanisms provided by such system are well suited when composing several (possibly partial) diagnosis into a final response, which is consistent with the knowledge of involved agents (i.e., hospital entities or specialist doctors). In the second part of the paper, we extend basic ALIAS coordination mechanisms towards probabilistic abduction. In this way, several (possibly partial) diagnosis obtained by probabilistic abductive reasoning can be merged into a final set of abductive diagnosis, each marked with a probability value.

1 Introduction

In recent years, the interest for intelligent agents has considerably grown from both theoretical and practical point of view [12]. The agent paradigm, in fact, is well suited to represent applications that merge features inherited both from the distributed systems area (such as locality, distribution, interaction, mobility etc.) and from artificial intelligence (such as reasoning, adaptation, etc.). In particular, intelligent agents require both deductive and reasoning capabilities, and social capabilities, which make possible interaction, collaboration and competition among different agents.

Traditionally, the medical field has been a valuable testbed for artificial intelligence techniques, and up to now a lot of medical applications have taken advantage from this kind of approach. Agents in medicine could bring an added value due to the capability to deal with distributed, heterogeneous, multiple and possibly incomplete knowledge bases. This could be the case, for instance, of diagnosis in an hospital, where the clinical problems of one patient could be analysed by several distinct (possibly remote) experts. Each expert (e.g., a specialist doctor) can reason on given symptoms using its own local (and possibly partial) knowledge base. Each expert is autonomous, in the sense that she/he can provide a diagnosis independently of others. It could be useful, however, to *coordinate* the reasoning of two (or more) doctors, in order to join or compare different diagnostic responses. This paper indeed addresses this issue: we propose the application of ALIAS multiple abductive agents [3] to medical diagnosis. ALIAS is an agent architecture where agents are logic-based and capable to perform abductive reasoning. ALIAS agents can also be coordinated following two schemes: collaboration and competition.

Abduction has been widely recognized as a powerful mechanism for hypothetical reasoning in presence of incomplete knowledge [5, 7, 9]. Abduction is generally understood as reasoning from effects to causes, and also captures other important issues such as reasoning with defaults and beliefs (see for instance [11, 14]). For these reasons abduction is often applied to diagnosis.

The multi-agent approach we are proposing relies on several agents each representing a distinct expert. Each agent encloses an abductive reasoning mechanism that, given a set of observed symptoms, allows single agents to formulate their own diagnosis. Moreover, agent reasoning can be coordinated following several patterns (namely, *collaboration and competition*), thus making possible the joining of diagnosis into a unique set of answers.

Let us consider, as an example for the collaborative case, a group of medical doctors, each one expert in a particular area (e.g. gastroenterology, ematology, etc.) who have to collaborate in order to formulate a diagnosis for a given set of symptoms. Each expert can be modeled by an abductive agent whose task is to find an hypothesis (i.e., a disease) as an explanation for a sub-part of the symptoms (i.e., those relevant for his/her area) given as observations to the agent. In some cases, the hypotheses raised by an agent can generate an inconsistency with other hypotheses raised by a collaborative agent (for instance, a certain symptom s does not occur when

[1] DEIS, Università di Bologna, Viale Risorgimento 2, 40136 Bologna, Italy {aciampolini, pmello, sstorari}@deis.unibo.it

the disease d is present).

As an example of the competitive case, let us consider again a group of medical doctors, each one expert in a particular area who have to find a diagnosis for the symptom. Each expert can be modelled by an abductive agent whose task is to find an hypothesis (i.e., a diagnosis) as an explanation for the symptoms given as observations to the agents. Different, alternative diagnoses can thus be proposed by the different agents, and the *best* one can be chosen according to some policy (for instance, the one with major incidence, or the most plausible one, with respect to the clinical history of the patient).

Therefore, in the *collaborative* case, the solution of a (possibly complex) task is distributed among several agents following a "divide-and-conquer" style; although each collaborative agent performs a local reasoning, the solution of the initial problem (obtained joining the abductive explanations obtained for each sub-problem) must be "consistent" with the knowledge of all agents involved. In the *competitive* case, instead, several agents could be asked to solve the same problem. Thus, each competing agent, reasoning on its own knowledge, possibly finds an abductive solution for the given problem: among all the available solutions some of them could be selected following a pre-defined policy.

However, we are aware that the ALIAS multi-agent abductive logical framework here discussed is too limited if we want to use it in a real context of distributed medical diagnosis. In particular, even if logic-based abduction is considered a suitable framework for diagnosis, a way of associating probability with hypothesis could be very useful for determining the "best" diagnosis. This need is more evident in a multi-agent context where different knowledge bases collaborate and compete in order to find a consistent diagnosis. The possibility of merging logical and probabilistic notions of of evidential reasoning in a unifying computational framework based on abduction has been the subject of a lot of work in literature. We consider here, as reference,[16] the work of David Poole about probabilistic Horn Abduction and show how its basic notions of probability and combination of probability can be easy introduced in the multi-agent context of ALIAS. This allows us (under certain conditions) to generate different "global" sets of consistent hypothesis (representing the alternative results of the distributed diagnosis) each one characterized by a probability value that can be used for determining the "best" diagnosis.

The paper is organized as follows. Section 2 presents abduction and the ALIAS multi-agent architecture. Section 3 shows how the problem of clinical diagnosis can be suitably faced within the system. In section 4 the integration of multi-agent ALIAS coordination mechanisms with Poole's Probabilistic Horn abduction is presented, together with a medical diagnosis example showing the advantages provided by this approach. Conclusions and future work follow.

2 The *ALIAS* multi-agent system

In this section, we introduce *ALIAS* (Abductive LogIc AgentS) [2, 3] , an agent architecture where several intelligent agents, each enclosing a local knowledge base, can either autonomously reason using its own local knowledge base or dynamically join other agents to cooperatively solve goals using abductive reasoning. The reasoning of agents could be coordinated following either a *competitive* or a *collaborative* model, or a combination of them. The inner structure of each agent, shown in figure 1, is basically composed of two modules: the *Abductive Reasoning Module* ($\mathcal{ARM}$, for short) and the *Agent Behavior Module* ($\mathcal{ABM}$). Each agent is characterized by two knowledge bases each situated at a different abstraction level: the *abductive* knowledge base and the *behavior* knowledge base. The first KB is represented by an abductive logic program (details are given in section 2) used to perform abductive reasoning; the behavior KB is a set of logic clauses that describe the behavior (i.e. actions and interactions) of the agent within the environment. In particular, the abductive knowledge base is encapsulated within the abductive reasoning module.

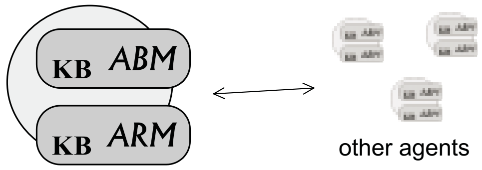

Figure 1. *The structure of an* ALIAS *agent*

The $\mathcal{ABM}$, instead, handles the behavior knowledge base, which is expressed by means of the *LAILA* language (presented in section 2.3). In particular, the social behavior of each agent can be expressed within the $\mathcal{ABM}$, by means of collaborative/competitive queries submitted to other agents. Sometimes the agent's behavior may require the abductive explanation of a goal: this situation requires an interaction between local $\mathcal{ABM}$ and $\mathcal{ARM}$, in order to locally start an abductive computation.

In this framework each ALIAS multi-agent application is represented by a set of abductive agents, each modeled by its knowledge bases (located respectively at $\mathcal{ABM}$ and $\mathcal{ARM}$ levels), and, in particular, interacting following the rules enclosed at the upper level of their structure.

2.1 Logic-based Abduction

In *ALIAS* we use abduction as the main form of agent reasoning. To this purpose, each agent encapsulates (in its reasoning module $\mathcal{ARM}$) an *abductive logic program.* An abductive logic program is a triple $\langle P, \mathcal{A}, IC\rangle$ where P is a logic program possibly with abducible atoms in clause bodies; $\mathcal{A}$ is a set of *abducible predicates*, i.e., *open* predicates which can be used to form explaining sentences; IC is a set of integrity constraints[2]: each constraint is a denial containing at least one abducible. [3].

Given an abductive program $\langle P, \mathcal{A}, \mathcal{IC}\rangle$ and a formula G, the goal of abduction is to find a (possibly minimal) set of atoms Δ which together with P entails G. It is also required

[2] In the following, for the sake of simplicity, we consider only ground programs, thus assuming that P and IC have already been instantiated.

[3] Our language uses the Prolog conventions, and has the same definitions of variables, terms, clauses and atomic symbols.

that the program $P \cup \Delta$ is consistent with respect to IC. The following example will hopefully help the intuition in understanding the idea underlying abductive reasoning.

Let us consider the following program, inspired by [13]:

```
grass_is_wet  :- rained_last_night
grass_is_wet  :- sprinkler_was_on
shoes_are_wet :- grass_is_wet
```

with integrity constraint:

```
:-  rained_last_night, sprinkler_was_on.
```

Let predicates `rained_last_night` and `sprinkler_was_on` be abducible. The observation `shoes_are_wet` is explained by two (minimal) sets of sentences, respectively:

$$\mathtt{Abd}_1 = \{\mathtt{rained\_last\_night}, \mathtt{not\ sprinkler\_was\_on}\}$$

$$\mathtt{Abd}_2 = \{\mathtt{sprinkler\_was\_on}, \mathtt{not\ rained\_last\_night}\}$$

According to [7], negation as default, possibly occurring in clause bodies, can be recovered into abduction by replacing negated literals of the form *not a* with a new positive, abducible atom *not_a* and by adding the integrity constraint $\leftarrow a, not\_a$ to IC. We suppose that each integrity constraint in IC_i of an agent A_i has at least one abducible in the body. We suppose that abducible predicates have no definition as in [10]. As concerns integrity constraints, the user-defined ones are partitioned among the various agents, while those for handling negation as default, like the constraint $\leftarrow p, not\ p$, are left implicit and replicated in each agent's knowledge base. The set Δ contains the hypotheses so far assumed by all agents.

2.2 Extending abductive reasoning to a multi-agent perspective

In order to perform abduction in a multi-agent environment we defined an algorithm able of coping with several agents possibly arguing on the same hypotheses. In particular, we equip each agent with a distinct abductive logic program (enclosed in its $\mathcal{ARM}$).

agents can dynamically join into groups (from now on, we refer to the group of agents with the term *bunch*), with the purpose, for instance, of finding the solution of a given goal in a *collaborative* way. In this perspective, although the set of program clauses and integrity constraints might differ from agent to agent, we assume that the set of abducible predicates (default predicates included) is the same for all the agents in a bunch. This implies that when proving a given goal, if an agent A assumes a new hypothesis h, all the arguing agents (i.e., the agents belonging to the same bunch) must check the consistency of h with their own integrity constraints. These checks could possibly raise new hypotheses, whose consistency within the bunch has to be recursively checked. Therefore, in ALIAS, the abductive explanation of a goal within a bunch of agents is a set of abduced hypotheses, agreed by all agents in the bunch.

We introduced some mechanism to support agent bunches, local abduction and global consistency checks. As far as local abduction is concerned, we adopted the Kakas-Mancarella proof procedure, defined originally in [7] and further refined in [10], which is correct with respect to the abductive semantics defined in [1]. The proof procedure presented in [10] extends the basic resolution mechanism adopted in logic programming by introducing the notion of *abductive* and *consistency* derivation. In ALIAS, we adopt a two-phase algorithm, which first collects the results of local ([4] abduction processes running the Kakas-Mancarella proof procedure, and then, depending on the chosen coordination policy, checks their consistency, in order to find a δ which is consistent with all the constraints present in the knowledge bases of the agents of the bunch.

Let us consider again the situation of the example presented in section 2.1, with one additional agent; in particular, let us consider a bunch composed by two agents: A_0 and A_1. Let A_0 enclose the following (simplified) abductive logic program:

```
grass_is_wet  :- sprinkler_was_on
shoes_are_wet :- grass_is_wet
```

with no integrity constraint. Agent A_1 (arguing with A_0) knows that the sprinkler takes the water from a tank that is emptied after the sprinkler has been on; thus, A_1's knowledge base be:

```
tank_is_full  :- not sprinkler_was_on
```

with integrity constraint:

```
:-  sprinkler_was_on, tank_is_full.
```

Again, let predicates `rained_last_night` and `sprinkler_was_on` be abducible, and $\Delta = \emptyset$ the current set of hypotheses. For the sake of simplicity, in this example integrity constraints of the kind `:-A, not A` are left implicit. In order to demonstrate the goal (observation) `:-shoes_are_wet`, agent A_0 applies its clause reducing the goal to `:-grass_is_wet` and therefore to `:-sprinkler_was_on`. Then, since `sprinkler_was_on` is abducible the consistency for `sprinkler_was_on` has to be checked within the bunch $\{A_0, A_1\}$. In particular, agent A_1 tries to prove the failure of the integrity constraint `:-sprinkler_was_on, tank_is_full`. The failure of `tank_is_full` is proved in one-step derivation by reducing the goal to `:-not sprinkler_was_on`. Since `not sprinkler_was_on` is abducible but its complement `sprinkler_was_on` has already been assumed, this derivation fails, and the consistency check succeeds. Then, A_1 agrees with A_0 in assuming `sprinkler_was_on`, that is definitively added to the current set of hypotheses.

Despite the choice of using the Kakas-Mancarella abductive proof procedure for agent abductive reasoning, it is worth to notice that the ALIAS architecture is not strictly bound to it: the same high-level features of the system could exploit different abduction algorithms ([6, 8].

2.3 The LAILA Language

The *ALIAS* agent behavior is expressed by means of the *Language for AbductIve Logic Agents* (LAILA, for short). This language allows to model agent actions and interactions in a

[4] In this context, *local* means *performed within the agent that is demonstrating the query.*

logic programming style. In particular we will focus on agent social behavior, and especially on how each agent can request demonstration of goals to other agents in the system. To this purpose, we introduce two composition operators: the collaborative AND operator (`&`) and the competitive operator (`;`) that can be used by each agent to express and coordinate abductive queries to other (set of) agents. The language provides also a communication operator (`>`) that is used to submit queries to other agents. The interested reader could find in [3] formal descriptions of both syntax and semantics of LAILA language.

Let us consider, for instance, the following LAILA competitive query q, formulated by agent A_0:

$$?\,\texttt{A1} > \texttt{G}\,;\,\texttt{A2} > \texttt{G}$$

It means that A_0 asks either $A1$ *or* A_1 to demonstrate goal G. It causes the following effects:

- A_0 asks A_1 to solve G; if G succeeds in A_0, N $(N > 0)$ abductive explanations δ_{1i} $(i \in [1, \ldots N])$, consistent in the bunch $\{A_0, A_1\}$, could be obtained for G.

- A_0 asks A_2 to solve G; if G succeeds in A_1, M $(M > 0)$ abductive explanations δ_{2j} $(j \in [1, \ldots M])$ consistent in the bunch $\{A_0, A_2\}$, could be obtained for G.

- The resulting set Δ_q contains all the plausible abductive explanations for G and encloses both δ_{1i} $(i \in [1, \ldots, N])$ and δ_{2j} $(j \in [1, \ldots, M])$.

If both A_1 and A_2 fail, the competitive query fails.

Let us consider, now, the following collaborative query q, given by agent A_0:

$$?\,\texttt{A1} > \texttt{G1}\,\&\,\texttt{A2} > \texttt{G2}$$

It expresses that agent A_0 asks agent A_1 to prove G_1 *and* A_2 to prove goal G_2; the query will succeed *iif* both agents A_0 and A_1 succeed; in particular, it will cause the following effects:

- A_0 asks A_1 to solve $G1$; if $G1$ succeeds in A_0, N $(N > 0)$ abductive explanations δ_{1i} $(i \in [1, \ldots N])$, consistent in the bunch $\{A_0, A_1\}$, could be obtained for $G1$.

- A_0 asks A_2 to solve $G2$; if $G2$ succeeds in A_1, M $(M > 0)$ abductive explanations δ_{2j} $(j \in [1, \ldots M])$ consistent in the bunch $\{A_0, A_2\}$, could be obtained for $G2$.

- The abductive explanation for the collaborative query q is therefore a set of hypotheses:

  $\Delta_q = \{\delta_k \mid \delta_k = \delta_{0i} \cup \delta_{1j} \wedge \delta_k \text{ is consistent in } \{A_0, A_1, A_2\}\}$

If if either $A_1 > G_1$ or $A_2 > G_2$ fails, the query q fails.

3 Distributed Diagnosis with ALIAS: an Example

Now we provide a diagnosis example aimed at showing the practical meaning of both ALIAS agents and LAILA constructs; this example will also show their concrete applicability to the medical diagnosis problem. In medicine, as in many other real application fields, we cannot rely on the assumption about completeness of knowledge; more likely, each expert (i.e., each specialist) handle a knowledge which is often very large, but in any case it could not be considered *complete*. For this reason we implement medical diagnosis in the multi-agent ALIAS context, supporting distributed abductive reasoning. We represent each doctor as an ALIAS agent with the aim of showing that coordination connectives such as *collaboration* and *competition* could be used to suitably merge incomplete knowledge by joining two or more agents for demonstrating a LAILA query.

The following example aims to represent the common relation between a general medicine doctor and some specialized doctors. Usually a patient goes to the general medicine doctor and asks him/her for a diagnosis capable of explaining his/her symptoms. The general medicine doctor possibly requires the opinions of specialized doctors in order to identify the suitable diagnosis for the given symptoms. This task is performed asking to them the explanation of the symptoms that they are capable of handling. Each specialist, tries on his own to explain one ore more symptoms and returns back, if possible, a set of plausible diagnoses. These diagnoses may be represented, for example, by a set of suspect diseases that could cause the observed symptoms.

Let us consider, for instance, the case of a patient exhibiting a lower limbs suffering. In this case, for instance, the general medicine doctor could refer to four doctors, each specialized in a particular area: an osteologist, a neurologist, a cardiologist and an angiologist).

We choose to map each doctor on a distinct ALIAS agent and to translate his/her specific knowledge into a logic abductive program, to be enclosed within agent's local $\mathcal{ARM}$. According to this approach, in the following we report ALIAS agents KBs[5]:

Osteologist:

$\mathcal{ABM}$:

SD(Gender) $\leftarrow$ $\downarrow$ SD(Gender)

$\mathcal{ARM}$:

```
SD(Gender) :- MC(yes), SPO(yes, Gender)
SD(Gender) :- MC(yes), SPO(no, Gender)
SD(Gender) :- MC(no)  , SPO(yes, Gender)
:- MC(yes), MC(no).
:- SPO(yes,M), SPO(no,M).
:- SPO(yes,F), SPO(no,F).
```

[5] For the sake of synthesis, in agents KBs we denoted symptoms and diseases with the following acronyms:
```
LLSS = Lower Limbs Suffering under Stress
LLSR = Lower Limbs Suffering in Repose
LLSW = Lower limbs Swelling
SS = Spine Suffering
SD = Skeletal Deformity
HP = Heart Problems
EM = Embolus
THR = Thrombosis
SPO = Spondylitis
MC = Calcium Deficiency
SCIA = Sciatica
DIA = Diabetes
```

Neurologist:

$\mathcal{ABM}$:

```
SS(Gender) ←   ↓ SS(Gender)
```

$\mathcal{ARM}$:

```
LLSR :- SCIA(yes)
SS :- SCIA(yes), SPO(yes, Gender)
SS :- SCIA(yes), SPO(no, Gender)
SS :- SCIA(no) , SPO(yes, Gender)
:- SCIA(yes), SCIA(no).
:- SPO(yes,M), SPO(no,M).
:- SPO(yes,F), SPO(no,F).
```

Angiologist:

$\mathcal{ABM}$:

```
LLSS ←   ↓ LLLS
```

$\mathcal{ARM}$:

```
LLSS :- EM(yes), THR(yes), DIA(yes)
LLSS :- EM(yes), THR(yes), DIA(no)
LLSS :- EM(yes), THR(no) , DIA(yes)
LLSS :- EM(yes), THR(no) , DIA(no)
LLSS :- EM(no) , THR(yes), DIA(yes)
LLSS :- EM(no) , THR(yes), DIA(no)
LLSS :- EM(no) , THR(no) , DIA(yes)
LLSR :- DIA(yes)
:- EM(yes), EM(no).
:- THR(yes), THR(no).
:- DIA(yes), DIA(no).
```

Cardiologist:

$\mathcal{ABM}$:

```
LLSS ←   ↓ LLSS
```

$\mathcal{ARM}$:

```
LLSS ← HP(yes), EM(yes), THR(yes)
LLSS ← HP(yes), EM(yes), THR(no)
LLSS ← HP(yes), EM(no) , THR(yes)
LLSS ← HP(yes), EM(no) , THR(no)
LLSS ← HP(no) , EM(yes), THR(yes)
LLSS ← HP(no) , EM(yes), THR(no)
LLSS ← HP(no) , EM(no) , THR(yes)
LLSW ← HP(yes)
:- HP(yes), HP(no)
:- EM(yes), EM(no).
:- THR(yes), THR(no)
```

Usually, there are two different query types that the general medicine doctor may issue to these doctors: a collaborative query and a competitive query. When the same symptom may be explained by two or more different doctors, it may be useful to ask to all of them for explanation, in competition, in order to collect all the available explanations and then to identify the more suitable ones among them. When we have two different symptoms explainable by two different specialized doctor, it is necessary a cooperation between them in order to identify and merge the set of diagnosis capable of explaining both symptoms, provided that they are consistent.

3.1 ALIAS Collaborative diagnosis.

As an example of the collaborative case, let us consider a patient who complains of having a skeleton deformity with spine suffering. The patient is a man. In this case the general medicine agent may issue the following query q:

`Osteologist > SD(m)& Neurologist > SS(m).`

The Osteologist agent returns the following diagnoses (each represented by a set of hypotheses δ_{Oi}):

δ_{O1} = { MC(yes), SPO(yes,m) }
δ_{O2} = { MC(yes), SPO(no,m) }
δ_{O3} = { MC(no) , SPO(yes,m) }

Similarly, the Neurologist agent returns the following diagnoses:

δ_{N1} = { SCIA(yes), SPO(yes,m) }
δ_{N2} = { SCIA(yes), SPO(no,m) }
δ_{N3} = { SCIA(no), SPO(yes,m) }

As described in section 2.3, therefore, the resulting set of diagnoses Δ_q associated with the collaborative query q contains the following consistent diagnoses:

δ_{q1} = { MC(yes), SPO(yes,m), SCIA(yes) }
δ_{q2} = { MC(no) , SPO(yes,m), SCIA(yes) }
δ_{q3} = { MC(yes), SPO(no,m) , SCIA(yes) }
δ_{q4} = { MC(yes), SPO(yes,m), SCIA(no) }
δ_{q5} = { MC(no) , SPO(yes,m), SCIA(no) }

Each final diagnosis δ_{qi} is obtained by merging two distinct partial diagnoses (respectively for SD and SS) into a consistent union set.

3.2 ALIAS Competitive diagnosis.

As an example of the competitive case, let us consider, now, a male patient who complains of having a lower limbs suffering under stress (represented by the LLSS predicate). In this case, the general medicine agent may fire the following competitive query:

`Angiologist > LLSS(m) ; Cardiologist > LLSS(m)`

The Angiologist agent explains LLSS with the following diagnoses:

δ_{A1} = { EM(yes), THR(yes), DIA(yes) }
δ_{A2} = { EM(yes), THR(yes), DIA(no) }
δ_{A3} = { EM(yes), THR(no) , DIA(yes) }
δ_{A4} = { EM(yes), THR(no) , DIA(no) }
δ_{A5} = { EM(no) , THR(yes), DIA(yes) }
δ_{A6} = { EM(no) , THR(yes), DIA(no) }
δ_{A7} = { EM(no) , THR(no) , DIA(yes) }

In a similar way, the Cardiologist agent explain LLSS with the following deltas:

δ_{C1} = { HP(yes), EM(yes), THR(yes) }
δ_{C2} = { HP(yes), EM(yes), THR(no) }
δ_{C3} = { HP(yes), EM(no) , THR(yes) }
δ_{C4} = { HP(yes), EM(no) , THR(no) }
δ_{C5} = { HP(no) , EM(yes), THR(yes) }
δ_{C6} = { HP(no) , EM(yes), THR(no) }
δ_{C7} = { HP(no) , EM(no) , THR(yes) }

As explained in 2.3, the general medicine agent will collect all these diagnoses into an unique set Δ_q associated with the competitive query. These diagnoses practically represent alternative explanations for the given symptom LLSS.

4 Multi-Agent Probabilistic Horn abduction

We are aware that the multi-agent abductive logical framework here discussed is too limited if we want to use it in a real context of distributed medical diagnosis. In particular, even if logic-based abduction is considered a suitable framework for diagnosis, a way of associating probability with hypothesis could be very useful for determining the "best" diagnosis. This need is more clear in a multi-agent context where different knowledge bases collaborate and compete in order to find a consistent diagnosis. The possibility of merging logical and probabilistic notions of evidential reasoning in unifying computational framework based on abduction has been the subject of a lot of work in literature [15, 16]. We consider here as reference the work of David Poole about probabilistic Horn Abduction [16] and show how its basic notions of probability and combination of probability can be easy introduced in the multi-agent context of ALIAS. This allows us (under certain conditions) to generate different "global" sets of consistent hypothesis (representing the alternative results of the distributed diagnosis) each one characterized by a probability that can be used for determining the "best" diagnosis.

4.1 Probabilistic Horn Abduction

In the following we will summary the main features of the Poole's work on Probabilistic Horn Abduction (see [16]), where a formulation of abduction with probabilities associated with the Hypotheses is presented in a logic programming context. In particular, abduction is restricted to definite clauses with simple forms of integrity constraints. The language is that of pure Prolog (i.e., definite clauses) with special disjoint declarations that specify a set of disjoint hypotheses with associated probabilities. The main design considerations were to make a language the simplest extension to pure Prolog that also included probabilities. In the porposed framework, any probabilistic information can be represented using only independent hypotheses. If there is any dependence amongst hypotheses, new hypothesis may be intruduced to explain that dependency. The disjoint declaration have the following formulation:
disjoint($[h_1 : p_1,\ \ldots,\ h_n : p_n]$*).*
where the h_i are atoms, and the p_i are real numbers $0 \leq p_i \leq 1$ such that $p_1 + \ldots + p_n = 1$. The h_i will be referred to as **hypotheses.** Any ground instance of a hypothesis belongs to one disjoint declaration, cannot be an instance of another hypothesis in any of the disjoint declarations (either the same declaration or a different declaration) nor can it be an instance of the head of any clause. This restricts the language so that we cannot encode arbitrary clauses as disjoint declarations. Each instance of each disjoint declaration is independent of all of the other instances of disjoint declarations and the clauses.
The probabilistic Horn abduction theory is defined as a collection of definite clauses and disjoint declarations. In this theory: F is the set of Horn clauses, F' is the set of ground instances of elements of F, H is a set of atoms representing the possible hypotheses and H'is the set of ground instances of elements of H. Considering this theory, [14] if g is a closed formula, an **explanation** of g from (F, H) is a set $D$of elements of H' such that

$$F \cup D| = g$$

and

$$F \cup D| \neq false$$

The first condition says that D is a sufficient cause for g, and the second says that D is possible. Only the **minimal explanations** of g are considered: an explanation of g is minimal when no strict subset is an explanation of g.

4.1.1 Assumptions about the rule base

Probabilistic Horn abduction also contains some assumptions about the rule base. In the following we cite just some of them that will be subject to further discussion in the multi-agent context. For a complete and detailed presentation and discussion see [16].

The **first assumption (A1)** says that the rules in F' for a ground nonassumable atom are covering. That is, if the rules for a in F' are:
$a \leftarrow B_1$
$a \leftarrow B_1$
....
$a \leftarrow B_m$
if a is true, one of the B_i is true. Thus Clark's completion [4] is valid for every nonassumable.

The **second assumption (A2)** says that the bodies of the rules in F for an atom are mutually exclusive. Given the above rules for a, this means that for each $i \neq j$, B_i and B_j cannot both be true in the domain under consideration. This assumption may be satisfied adding extra conditions to the rules to make sure they are disjoint.

Associated with each possible hypothesis is a prior probability. We use this prior probability to compute arbitrary probabilities. Our final assumption is to assume that logical dependencies impose the only statistical dependencies on the hypotheses. The **third assumption (A3)** say that ground instances of hypothesis that are not inconsistent are probabilistically independent.

Thus different disjoint declarations define independent hypotheses. Different instances of hypotheses are also independent.

See [16] for more justification of these assumptions.

4.1.2 Some Consequents of the Assumptions

Under these assumptions if *exp1(g, T) is* the set of minimal explanations of conjunction of atoms g from probabilistic Horn abduction theory T:

$$P(g) = P\left(\underset{e_i \in expl(g,T)}{V} e_i\right) = \sum_{e_i \in expl(g,T)} P(e_i)$$

Thus to compute the prior probability of any g we sum the probabilities of the explanations of g.

To compute arbitrary conditional probabilities, we use the definition of conditional probability:

$$P(\alpha|\beta) = \frac{P(\alpha \wedge \beta)}{P(\beta)}$$

Thus to find arbitrary conditional probabilities $P(\alpha|\beta)$, we find $P(\beta)$, which is the sum of the explanations of β, and $P(\alpha \wedge \beta)$ which can be found by explaining α from the explanations of β. Thus arbitrary conditional probabilities can be computed from summing the prior probabilities of explanations.

Finally, let us recall how to compute the prior probability of an explanation D of g.

Under assumption A3, if $\{h_1, h_2, \ldots, h_n\ ,\}$ are part of a minimal explanation, then

$$P\left(h_1 \wedge h_2 \wedge ... \wedge h_n\right) = \prod_{i=1}^{n} P\left(h_i\right)$$

To compute the prior of the minimal explanation we multiply the priors of the hypotheses. The posterior probability of the explanation is proportional to this.

In summary, for the probabilistic analysis we have to assume that the explanations were disjoint and covering. Thus, to compute the prior probability of any g we sum the probabilities of the explanations of g. Moreover, we assume also that ground instances of hypotheses that are consistent are probabilistically independent. Therefore, to compute the prior of the minimal explanation we multiply the priors of the hypotheses. Notice that these assumptions are not intended to be enforced by the system. It is up to the user (or some other system) to enforce these constraints. If these assumptions are violated then we make no guarantees about the "good" behavior of the system.

4.2 Integrating Probabilistic Horn Abduction within ALIAS

In this section we will show how we extended basic ALIAS coordination mechanisms towards probabilistic abduction. In particular, we transformed ALIAS into a multi-agent system where single agents adopt Poole's probabilistic Horn abduction to formulate explanations of given symptoms (for this reason, from now on, we will refer agents with the term *Poole's agents*, or PA). Thus each PA, given a goal (or symptom) s, will autonomously demonstrate it by calculating a set of plausible explanations *Expl(s)*. For each explanation δ in the set *Expl(s)* a probability value *P(δ)* can be calculated.

The novelty, with respect to Poole's work, are that we coordinate several abductive reasoners (the PAs), each enclosing its own KB, by means of ALIAS coordination mechanisms (i.e., collaboration and competition). With respect to the ALIAS system, instead, the novelty is that here agents reason using probabilistic horn abduction, and therefore produce explanations labeled with a probability value; this allows us to represent uncertainty associated to plausible explanations.

To this purpose we have extended Poole's rules for goal probability computation to the case of collaborative/competitive queries. In this way, explanations associated to competitive/collaborative queries could be labeled with a probability value, that could drive the selection of most plausible hypotheses.

Let us consider, first, the case of a collaborative query $q1$. As presented in 2.3 it could be expressed by agent A_0, using LAILA sintax, in the form:

$$A_1 > G_1 \ \&\ A_2 > G_2.$$

The demonstration of the query $q1$ will produce the following effects:

- A_0 asks A_1 to solve G_1; if G_1 succeeds, N $(N > 0)$ abductive explanations $\delta_{1i}^{P_{1i}}$ $(i \in [1, \ldots, N])$ for G_1 are obtained, where P_{1i} is the probability value associated with the explanation δ_{1i}.

- A_0 asks A_2 to solve G_2 ; if G_2 succeeds, M $(M > 0)$ abductive explanations $\delta_{2j}^{P_{2j}}$ $(j \in [1, \ldots, M])$ for G_2 are obtained, where P_{2j} is the probability value associated with the explanation δ_{2j}.

- The abductive explanation for the collaborative query $q1$ is therefore a set of hypotheses: $\Delta_{q1} = \{\delta_k \mid \delta_k = \delta_{0i} \cup \delta_{1j} \wedge \delta_k \text{is consistent} \wedge \delta_k \text{is minimal}\}$

The collaborative query $q1$ expresses an extended goal whose explanations are collected into the set Δ_{q1}: so,the probability associated with the collaborative goal $q1$ can be calculated as follows:

$$P(q1) = \sum_{\delta_k \in \Delta_{q1}} P(\delta_k) \tag{1}$$

As usual, the probability associated to an explanation formulated by a single PA is calculated as follows:

$$P_{ij} = P(\delta_{ij}) = \prod_{h_n \in \delta_{ij}} P(h_n)$$

Then, if we want to know the conditional probability of a illness h given a conjunction of symptoms $s_0 \wedge s_1$, whose explanation is obtained by the collaborative query q:

$$A_o > s_0 \& A_1 > s_1.$$

we have to calculate:

$$P(h \mid q) = \frac{P(h \wedge s_0 \wedge s_1)}{P(q)}$$

where $P(q)$ can be calculated using equation 1 and $P(h \wedge s_0 \wedge s_1)$ can be obtained by summing probabilities of the explanations in Δ_q that contain the hypothesis h.

Let us consider, now, the case of a competitive query $q2$. As in 2.3 it could be expressed, using LAILA sintax, in the form:

$$A_1 > G; A_2 > G.$$

The demonstration of the query $q2$ will produce the following effects:

- A_0 asks A_1 to solve G; if G succeeds in A_1, N $(N > 0)$ abductive explanations $\delta_{1i}^{P_{1i}}$ $(i \in [1, \ldots N])$, consistent in the bunch $\{A_0, A_1\}$, could be obtained for G, where P_{1i} is the probability associated with the explanation δ_{1i}.

- A_0 asks A_2 to solve G; if G succeeds in A_2, M $(M > 0)$ abductive explanations $\delta_{2j}^{P_{2j}}$ $(j \in [1, \ldots M])$ consistent in the bunch $\{A_0, A_2\}$, could be obtained for G, where P_{2j} is the probability associated with the explanation δ_{2j}.

- The resulting set Δ_{q2} contains all the plausible abductive explanations for G and encloses both δ_{1i} $(i \in [1, \ldots, N])$ and δ_{2j} $(j \in [1, \ldots, M])$.

As in the collaborative case,the probability associated with the competitive goal $q2$ can be calculated as follows:

$$P(q2) = \sum_{\delta_k \in \Delta q2} P(\delta_k)$$

The basic property of our system is that the probabilities calculated in a distributed manner by using the local KBi of the agent ai involved, should be equivalent to the ones determined by the Poole algorithm if we have all the knowledge in a single and global KB obtained as the union of the KBi involved.

Notice that, since we work in a multi-agent framework, Poole's assumptions about disjointness and covering of hypotheses should be guaranteed with respect to the union of the knowledge bases of the agents involved in the computation and not necessary locally in a single knowledge base. For instance, due to the incompleteness of the local knowledge bases the covering principle cannot be locally enforced.

4.3 Example of multiagent environment

In order to better show the additional features of the probabilistic multi-agent approach, we will refer to the example presented in section 3 an show how it could be faced in this context. In the following we list knowledge base of specialist agents, adapted to Poole's notation:

Osteologist:
```
SD ← MC(yes), SPO(yes, Gender)
SD ← MC(yes), SPO(no, Gender)
SD ← MC(no) , SPO(yes, Gender)

Disjoint([MC(yes) : 0.12, MC(no) : 0.88])
Disjoint([SPO(yes,M): 0.21, SPO(no,M): 0.79])
Disjoint([SPO(yes,F): 0.15, SPO(no,F): 0.85])
```

Neurologist:
```
LLSR ← SCIA(yes)
SS ← SCIA(yes), SPO(yes, Gender)
SS ← SCIA(yes), SPO(no, Gender)
SS ← SCIA(no) , SPO(yes, Gender)

Disjoint([SCIA(yes) : 0.31, SCIA(no) : 0.69])
Disjoint([SPO(yes,M): 0.21, SPO(no,M): 0.79])
Disjoint([SPO(yes,F): 0.15, SPO(no,F): 0.85])
```

Angiologist:
```
LLSS ← EM(yes), THR(yes), DIA(yes)
LLSS ← EM(yes), THR(yes), DIA(no)
LLSS ← EM(yes), THR(no) , DIA(yes)
LLSS ← EM(yes), THR(no) , DIA(no)
LLSS ← EM(no) , THR(yes), DIA(yes)
LLSS ← EM(no) , THR(yes), DIA(no)
LLSS ← EM(no) , THR(no) , DIA(yes)
LLSR ← DIA(yes)

Disjoint([EM(yes) : 0.06, EM(no) : 0.94])
Disjoint([THR(yes): 0.07, THR(no) : 0.93])
Disjoint([DIA(yes): 0.39, DIA(no): 0.61])
```

Heart specialist:
```
LLSS ← HP(yes), EM(yes), THR(yes)
LLSS ← HP(yes), EM(yes), THR(no)
LLSS ← HP(yes), EM(no) , THR(yes)
LLSS ← HP(yes), EM(no) , THR(no)
LLSS ← HP(no) , EM(yes), THR(yes)
LLSS ← HP(no) , EM(yes), THR(no)
LLSS ← HP(no) , EM(no) , THR(yes)
LLSW ← HP(yes)

Disjoint([HP(yes) : 0.28, HP(no) : 0.72])
Disjoint([EM(yes) : 0.06, EM(no) : 0.94])
Disjoint([THR(yes): 0.07, THR(no): 0.93])
```

4.3.1 Example of collaborative query

Let us consider the collaborative query q shown in 3.1:

`Osteologist > SD(m)& Neurologist > SS(m).` As in the case of LAILA agents, the demonstration of the query q produces the following set of diagnoses:

δ_1 = { MC(yes), SPO(yes,m), SCIA(yes) }
δ_2 = { MC(no) , SPO(yes,m), SCIA(yes) }
δ_3 = { MC(yes), SPO(no,m) , SCIA(yes) }
δ_4 = { MC(yes), SPO(yes,m), SCIA(no) }
δ_5 = { MC(no) , SPO(yes,m), SCIA(no) }

With respect to the ALIAS version, this multi-agent probabilistic approach allows us to compute two type of evaluations on the final set of diagnoses obtained as result of the issued query.

In the first type of evaluation, it is possible to identify the diagnosis with the highest probability which represent the more reliable combination of hypotheses. The probability of each diagnosisδ_i, as described in section 4, may be computed multiplying the prior probability of each hypothesis belonging to δ_i. For example:

$P(\delta_1) = 0.12 * 0.21 * 0.31 = 0.008$
$P(\delta_2) = 0.88 * 0.21 * 0.31 = 0.057$
$P(\delta_3) = 0.12 * 0.79 * 0.69 = 0.065$
$P(\delta_4) = 0.12 * 0.21 * 0.69 = 0.017$
$P(\delta_5) = 0.88 * 0.21 * 0.69 = 0.127$

The second type of evaluation is useful for determining the probability of a single hypothesis h (representing, for instance, a disease), given the conjunction of symptoms $SD(m) \wedge SS(m)$. According to usual notation, this could be denoted as:

$$P(h|SD(m) \wedge SS(m))$$

Applying the Bayes rules, $P(h \mid SD(m) \wedge SS(m))$ may be calculated as

$$P(h \mid SD(m) \wedge SS(m)) = \frac{P(h \wedge SD(m) \wedge SS(m))}{P(q)}$$

We can calculate $P(q)$ using equation 1:

$$P(q) = P(\delta_1) + P(\delta_2) + P(\delta_3) + P(\delta_4) + P(\delta_5) = 0.274$$

To compute $P(h \wedge SD(m) \wedge SS(m))$ we sum the probabilities of those δ_i which contain h; for example, if $h = \text{SPO(yes)}$ we obtain:

$$P(\text{SPO(yes)} \ \wedge \ SD(m) \ \wedge \ SS(m)) =$$

$$P(\delta_1) + P(\delta_2) + P(\delta_4) + P(\delta_5) = 0.209$$

Therefore:

$$P(h \mid SD(m) \ \wedge \ SS(m)) = \frac{0.209}{0.274} = 0.76$$

4.3.2 Example of competitive query

Let us consider the competitive query q shown in 3.2:

`Angiologist` > `LLSS(m) ; Cardiologist` > `LLSS(m)`

As in the case of LAILA agents, the demonstration of the query q produces the following set of diagnoses:

δ_{A1} = { EM(yes), THR(yes), DIA(yes) }
δ_{A2} = { EM(yes), THR(yes), DIA(no) }
δ_{A3} = { EM(yes), THR(no) , DIA(yes) }
δ_{A4} = { EM(yes), THR(no) , DIA(no) }
δ_{A5} = { EM(no) , THR(yes), DIA(yes) }
δ_{A6} = { EM(no) , THR(yes), DIA(no) }
δ_{A7} = { EM(no) , THR(no) , DIA(yes) }
δ_{C1} = { HP(yes), EM(yes), THR(yes) }
δ_{C2} = { HP(yes), EM(yes), THR(no) }
δ_{C3} = { HP(yes), EM(no) , THR(yes) }
δ_{C4} = { HP(yes), EM(no) , THR(no) }
δ_{C5} = { HP(no) , EM(yes), THR(yes) }
δ_{C6} = { HP(no) , EM(yes), THR(no) }
δ_{C7} = { HP(no) , EM(no) , THR(yes) }

Similarly to the case of collaboration (shown in section 4.3.1), we can calculate, for instance, $P(\text{HP(yes)} \mid \text{LLSS})$:

$$P(\text{HP(yes)} \mid \text{LLSS}) = 0.334$$

5 Conclusions and Future Work

In this paper we presented a novel multi-agent approach for implementing medical diagnosis. Diagnosis is a complex task that often requires the knowledge of several experts: we choose to represent each expert with a distinct agent. Each expert (i.e., each agent) can formulate its own diagnoses by means of abduction. Each expert, however, has a partial knowledge that even could be, in some cases, non consistent with the knowledge of other experts. For this reason, in order to achieve a more complete view (but consistent) of the problem domain, we coordinate experts by means of proper coordination protocols: collaboration and competition.

With respect to agents internal reasoning, we focused on abduction, since it is well suited to the problem of diagnosis. In particular, in section 3 we have shown how a medical diagnosis problem that involves several specialists, could be solved within ALIAS. ALIAS, in fact, is a system where agents adopt a logic-based algorithm to formulate consistent explanations for a given set of symptoms, by means of the dynamic composition of several KBs (through competitive and collaborative agent coordination).

However, we are aware that the ALIAS multi-agent abductive logical framework is too limited if we want to use it in a real context of distributed medical diagnosis. In particular, even if logic-based abduction is considered a suitable framework for diagnosis, a way of associating probability with hypothesis could be very useful for determining the "best" diagnosis. This need is more clear in a multi-agent context where different knowledge bases collaborate and compete in order to find a consistent diagnosis. For this reason we integrated within ALIAS, the Poole's probabilistic Horn abduction algorithm, as agents internal reasoning mechanism. In this way, we achieve a refinement of diagnosis, by generating (under certain conditions) different "global" sets of consistent hypothesis (representing the alternative diagnoses) each one characterized by a probability value that can be used for determining the "best" diagnosis.

A further refinement could be achieved by taking into consideration Poole's work on Bayesian networks and abduction [15], since it allows to represent in an abductive framework not only probabilistically independent hypotheses, but also the probabilistic dependencies between symptoms and hypotheses. As a matter of future work we plan to study how to apply this approach in a multi-agent context.

We are aware that examples shown in the paper cannot be considered *real* cases, but rather they represent naive abstractions of more complex real ones. We intend to test our approach with more complex cases, defined with the support of real medical experts.

The ALIAS system exploited for the applications shown in this paper is currently implemented in the non-probabilistic version. In the near future we also plan to implement on it the probabilistic approach presented in this work.

6 Acknowledgements

This work was supported by the SOCS project ,funded by the CEC, contract IST-2001-32530. The information provided is the sole responsibility of the authors and does not reflect the Community's opinion. The Community is not responsible for any use that might be made of data appearing in this publication. We want also to thank Dr. Giordana Barioni for her precious help in defining medical examples discussed in the paper.

REFERENCES

[1] A. Brogi, E. Lamma, P. Mancarella, and P. Mello. A Unifying View for Logic Programming with Non-Monotonic Reasoning. In *Theoretical Computer Science*, Vol. 184, 1–49, North Holland, 1997.

[2] A. Ciampolini, E. Lamma, P. Mello and P. Torroni. An Implementation for Abductive Logic Agents. *Proceedings AI*IA99*, Pitagora Editore, Bologna, Italy, 1999.

[3] A. Ciampolini, E. Lamma, P. Mello and P. Torroni. LAILA: A Language for Coordinating Abductive Reasoning Among Logic Agents. In *Computer Languages* 27(4):137–161,Elsevier Science, 2001.

[4] K. L. Clark. Negation as Failure. In H. Gallaire and J. Minker, eds., *Logic and Databases.* Plenum Press, New York, 1978.

[5] P. T. Cox and T. Pietrzykowski. Causes for events: Their computation and applications. In *Proc. CADE-86*, 608, 1986.

[6] M. Denecker and D. De Schreye SLDNFA: an abductive procedure for abductive logic programs. *Journal of Logic Programming*, 34(2):111–167, Elsevier, 1998.

[7] K. Eshgi and R. A. Kowalski. Abduction compared with negation by failure. In G. Levi and M. Martelli, editors, *Proc. 6th International Conference on Logic Programming*, 234. MIT Press, 1989.

[8] A. C. Kakas, ACLP: integrating abduction and constraint solving. *Proc. NMR'00*, Breckenridge, CO, 2000.

[9] A. C. Kakas and P. Mancarella, Generalized stable models: a semantics for abduction. In *Proc. 9th European Conference on Artificial Intelligence*. Pitman Pub., 1990.

[10] A. C. Kakas and P. Mancarella, On the relation between Truth Maintenance and Abduction. In *Proc. PRICAI90*, 1990.

[11] R. A. Kowalski, Problems and promises of computational logic. In *Proc. Symposium on Computational Logic*, 1-36. Springer-Verlag, Nov. 1990.

[12] N. R. Jennings, M. J. Wooldridge, eds., *Agent Technology*. Springer-Verlag, 1998.

[13] J. Pearl, Embracing causality in formal reasoning. *Proc. National Conference on Artificial Intelligence*, Seattle, WA, pages 369-373, 1987.

[14] D. L. Poole, A logical framework for default reasoning. *Artificial Intelligence*, 36:27. Elsevier, 1988.

[15] D. L. Poole, Probabilistic Horn Abduction and Bayesian Networks. *Artificial Intelligence*, 64(1), 81-129, Elsevier, 1993.

[16] D. L. Poole, Logic Programming, Abduction and Probability: a top down anytyme algorithm for estimating prior and posterior probabilities. *New Generation Computing*, 11(3-4), 377-400, Elsevier, 1993.