# Towards a Logical Model of Induction from Examples and Communication

Santiago ONTAÑÓN [a,1], Pilar DELLUNDE [b,a], Lluís GODO [a], and Enric PLAZA [a]
[a] *Artificial Intelligence Research Institute, IIIA-CSIC*
[b] *Universitat Autònoma de Barcelona, UAB*

**Abstract.** This paper focuses on a logical model of induction, and specifically of the common machine learning task of inductive concept learning (ICL). We define an *inductive derivation* relation, which characterizes which hypothesis can be induced from sets of examples, and show its properties. Moreover, we will also consider the problem of communicating inductive inferences between two agents, which corresponds to the multi-agent ICL problem. Thanks to the introduced logical model of induction, we will show that this communication can be modeled using computational argumentation.

**Keywords.** Induction, Logic, Argumentation, Machine Learning

## Introduction

Inductive inference is the basis for all machine learning methods which learn general hypotheses or models from examples. However, there has been little effort in finding a logical characterization of inductive inference, except for a few proposals such as [6]. This paper focuses on a logical model of inductive inference, and specifically of the common machine learning task of inductive concept learning (ICL).

The lack of a formal logical model of induction has hindered the development of approaches that combine induction with other forms of reasoning, such as the defeasible reasoning used in computational argumentation. In this paper, we define an inductive derivation relation (denoted by $\mid\!\sim$), which characterizes which hypotheses can be induced from sets of examples, and show the properties of this inductive derivation relation. We will focus both in the single-agent inductive concept learning process as well as in a multi-agent setting. To consider multi-agent settings, we will show that the problem of communicating inductive inferences can be modeled as an argumentation framework. Since inductive inference is a form of defeasible inference we will see that our inductive derivation relation can be easily combined with an argumentation framework, constituting a coherent model of multi-agent inductive concept learning.

The remainder of this paper is organized as follows. Section 2 introduces the problem of inductive concept learning as typically framed in the machine learning literature. Then, Section 3 introduces a logical model of induction and proposes an inductive deriva-

[1] Corresponding Author: IIIA (Artificial Intelligence Research Institute), CSIC (Spanish Council for Scientific Research), Campus UAB, 08193 Bellaterra, Catalonia (Spain), santi@iiia.csic.es.

tion relation. Section 4 then focuses on the multi-agent induction problem, framing it as an argumentation process. Finally, the paper closes with related work and conclusions.

## 1. Inductive Concept Learning

Concept learning [10] using inductive techniques is not defined formally, rather it is usually defined as a task, as follows:

**Given** 1. A set of instances $X$ expressed in a language $\mathcal{L}_I$
2. A space of hypotheses or generalizations $H$ (expressions in a language $\mathcal{L}_H$)
3. A target concept $c$ defined as a function $c : X \to \{0, 1\}$
4. A set $D$ of training examples, where a training example is a pair $\langle x_i, c(x_i)\rangle$

**Find** a hypothesis $h \in H$ such that $\forall x \in X : h(x) = c(x)$

This strictly Boolean definition is usually weakened to allow the equality $h(x) = c(x)$ not being true for all examples in $X$ but just for a percentage, and the difference is called the *error* of the learnt hypothesis. This definition, although widespread, is unsatisfactory and leave several issues without a precise characterization. For example, the space of hypotheses $H$ usually is expressed only by conjunctive formulas. However, most concepts need more than one conjunctive formula (more than one generalization) but this is “left outside” of the definition and is explained as part of the strategy of an inductive algorithm. For instance, the set-covering strategy, where one definition $h_1$ is found but covers only part of the positive examples in $D$, proceeding then to eliminate the covered examples and obtain a new $D'$ that will be used in the next step.

Another definition of inductive concept learning (ICL) is that used in Inductive Logic Programing (ILP) [9], where the background knowledge, in addition to the examples, has to be taken into account. Nevertheless, ILP also defines ICL as a task to be achieved by an algorithm, as follows:

**Given** 1. A set of positive $E^+$ and negative $E^-$ examples of a predicate $p$
2. A set of Horn rules (background knowledge) $B$
3. A hypothesis language $\mathcal{L}_H$ (a sublanguage of Horn logic language)

**Find** A hypothesis $H \in \mathcal{L}_H$ such that

- $\forall e \in E^+ : B \wedge H \models e$ ($H$ is complete)
- $\forall e \in E^- : B \wedge H \not\models e$ ($H$ is consistent)

In this paper our goal is to provide a logical model of inductive inference in ICL that covers the commonly held but informally defined task of learning concept description by induction in Machine Learning.

## 2. Inductive Inference for Concept Learning

In order to present our model of induction, let us start by describing the language we will use, which corresponds to a small fragment of first order logic and is built as follows. For the sake of simplicity we assume to work with two disjoint finite sets of unary predicates: a set of predicates to describe attributes $Pred\_At = \{P_1, \ldots, P_n\}$ and a set of predicates to denote concepts to be learnt $Pred\_Con = \{C_1, \ldots, C_m\}$. To simplify notation, for each $C \in Pred\_Con$, we will write $\overline{C}(\cdot)$ to denote $C(\cdot)$ or $\neg C(\cdot)$; moreover, we will write $\neg\overline{C}(\cdot)$ to denote $\neg C(\cdot)$ if $\overline{C}(\cdot) = C(\cdot)$, and $\neg\overline{C}(\cdot)$ to denote $C(\cdot)$ if $\overline{C}(\cdot) = \neg C(\cdot)$. Moreover we assume a finite domain of constants $D = \{a_1, \ldots, a_m\}$ which will be used as example identifiers. For instance, if $P \in Pred\_At$, $C \in Pred\_Con$ and $a \in D$, then $P(a)$ will denote that example $a$ has the attribute $P$, and $C(a)$ will denote that the concept $C$ applies to $a$. Our formulas will be of two kinds:

- *Examples* will be conjunctions of the form $\varphi(a) \wedge \overline{C}(a)$, where $\varphi(a) = Q_1(a) \wedge \ldots \wedge Q_k(a)$, with $Q_i(a)$ being of the form $P_i(a)$ or $\neg P_i(a)$. A *positive example* of $\overline{C}$ will be of the form $\varphi(a) \wedge \overline{C}(a)$; a *negative example* of $\overline{C}$ will be of the form $\varphi(a) \wedge \neg\overline{C}(a)$.
- *Rules* will be universally quantified formulas of the form $(\forall x)(\varphi(x) \to \overline{C}(x))$, where $\varphi(x) = Q_1(x) \wedge \ldots \wedge Q_l(x)$, with $Q_i(x)$ being of the form $P_i(x)$ or $\neg P_i(x)$.

The set of examples will be noted by $\mathcal{L}_e$ and the set of rules by $\mathcal{L}_r$, and the set of all formulas of our language will be $\mathcal{L} = \mathcal{L}_e \cup \mathcal{L}_r$. In what follows, we will use the symbol $\vdash$ to denote derivation in classical first order logic. By *background knowledge* we will refer to a finite set of formulas $K \subset \mathcal{L}_r$, although sometimes we will consider $K$ as the conjunction of its formulas.

**Definition 1** ***(Covering)*** *Given background knowledge $K$, we say that a rule $r := (\forall x)(\alpha(x) \to \overline{C}(x))$* covers an example $e = \varphi(a) \wedge \widehat{C}(a)$ *when $\varphi(a) \wedge K \vdash \alpha(a)$.*

These notions allow us to define inductive inference of rules from examples.

**Definition 2** ***(Inductive Derivation)*** *Given background knowledge $K$, a set of examples $\Delta \subseteq \mathcal{L}_e$ and a rule $r = (\forall x)(\alpha(x) \to \overline{C}(x))$, the* inductive derivation $\Delta \mid\!\sim_K (\forall x)(\alpha(x) \to \overline{C}(x))$ *holds iff:*
*1)* ***(Explanation)*** *$r$ covers at least one positive example of $\overline{C}$ in $\Delta$,*
*2)* ***(Consistency)*** *$r$ does not cover any negative example of $\overline{C}$ in $\Delta$*

Notice that if we have two conflicting formulas in $\Delta$ of the form $\varphi(a) \wedge C(a)$ and $\psi(b) \wedge \neg C(b)$ where the example $a$ has more (or less) description attributes than example $b$, then no rule $(\forall x)(\alpha(x) \to C(x))$ covering either example can be inductively derived from $\Delta$. The next definition identifies when a set of examples is free of these kind of conflicts.

**Definition 3** ***(Consistency)*** *A set of examples $\Delta$ is said to be* consistent with respect to a concept $\overline{C}$ and background knowledge $K$ *when: if $\varphi(a) \wedge \overline{C}(a)$ and $\psi(b) \wedge \neg\overline{C}(b)$ belong to $\Delta$, then both $K \not\vdash (\forall x)(\varphi(x) \to \psi(x))$ and $K \not\vdash (\forall x)\psi((x) \to \varphi(x))$.*

**Definition 4** ***(Inducible Rules)*** *Given a set of examples $\Delta$ and background knowledge $K$, we call $IR_K(\Delta) = \{(\forall x)(\varphi(x) \to \overline{C}(x)) \mid \Delta \mid\!\sim_K (\forall x)(\varphi(x) \to \overline{C}(x))\}$ the set of all rules that can be induced from $\Delta$ and $K$.*

We will assume in the rest of the paper that $IR_K(\Delta)$ is finite. Next we show some interesting properties of the inductive inference $\mid\!\sim_K$.

**Lemma 1** *The inductive inference $\mid\!\sim_K$ satisfies the following properties:*

1. *Reflexivity: if* $\Delta$ *is consistent w.r.t.* $C$ *and* $K$*, then if* $\varphi(a) \wedge \overline{C}(a) \in \Delta$ *then* $\Delta \mathrel{|\!\!\sim}_K (\forall x)(\varphi(x) \rightarrow \overline{C}(x))$.
2. *Positive monotonicity:* $\Delta \mathrel{|\!\!\sim}_K (\forall x)(\alpha(x) \rightarrow \overline{C}(x))$ *implies* $\Delta \cup \{\varphi(a) \wedge \overline{C}(a)\} \mathrel{|\!\!\sim}_K (\forall x)(\alpha(x) \rightarrow \overline{C}(x))$
3. *Negative non-monotonicity:* $\Delta \mathrel{|\!\!\sim}_K (\forall x)(\alpha(x) \rightarrow \overline{C}(x))$ *does not imply* $\Delta \cup \{\varphi(a) \wedge \neg\overline{C}(a)\} \mathrel{|\!\!\sim}_K (\forall x)(\alpha(x) \rightarrow \overline{C}(x))$
4. *If* $K \vdash (\forall x)(\varphi(x) \rightarrow \alpha(x))$ *then,*
   $\Delta \mathrel{|\!\!\sim}_K (\forall x)(\alpha(x) \rightarrow \overline{C}(x))$ *does not imply* $\Delta \mathrel{|\!\!\sim}_K (\forall x)(\varphi(x) \rightarrow \overline{C}(x))$
5. *If* $\Delta \mathrel{|\!\!\sim}_K (\forall x)(\alpha(x) \rightarrow \overline{C}(x))$ *and* $\vdash (\forall x)(\alpha(x) \rightarrow \varphi(x))$ *then* $\Delta \not\mathrel{|\!\!\sim}_K (\forall x)(\varphi(x) \rightarrow \neg\overline{C}(x))$
6. *If* $\Delta \mathrel{|\!\!\sim}_K (\forall x)(\alpha(x) \rightarrow \overline{C}(x))$ *and* $\vdash (\forall x)(\varphi(x) \rightarrow \alpha(x))$ *then* $\Delta \not\mathrel{|\!\!\sim}_K (\forall x)(\varphi(x) \rightarrow \neg\overline{C}(x))$
7. *Falsity preserving: let* $r = (\forall x)(\alpha(x) \rightarrow \overline{C}(x))$ *such that it covers a negative example from* $\Delta$*, hence* $r \notin IR_K(\Delta)$*; then* $r \notin IR_K(\Delta \cup \Delta')$ *for any further set of examples* $\Delta'$.
8. $IR_K(\Delta_1 \cup \Delta_2) \subseteq IR_K(\Delta_1) \cup IR_K(\Delta_2)$

*Proof:*

1. Since $\varphi(a) \wedge \overline{C}(a) \in \Delta$ and we obviously have $\varphi(a) \wedge K \vdash \varphi(a)$, explanation trivially holds. Now assume $\psi(a) \wedge \neg\overline{C}(a) \in \Delta$. Then, since $\Delta$ is consistent w.r.t. $C$ and $K$, $\psi(a) \wedge K \nvdash \varphi(a)$, hence consistency also holds.
2. Trivial
3. The reason is that nothing prevents that $\varphi(a) \wedge K \vdash \alpha(a)$ may hold.
4. The reason is that, since $\varphi$ is more specific than $\alpha$, it may not cover any example.
5. Let us assume that $\vdash (\forall x)(\alpha(x) \rightarrow \varphi(x))$ and $\Delta \mathrel{|\!\!\sim}_K (\forall x)(\varphi(x) \rightarrow \neg\overline{C}(x))$. Then, by consistency, for all $\psi(a) \wedge \overline{C}(a) \in \Delta$ we have $\psi(a) \wedge K \nvdash \varphi(a)$, and hence $\psi(a) \wedge K \nvdash \alpha(a)$ as well. Then clearly, $\Delta \not\mathrel{|\!\!\sim}_K (\forall x)(\alpha(x) \rightarrow \overline{C}(x))$.
6. Let us assume now that $\vdash (\forall x)(\varphi(x) \rightarrow \alpha(x))$ and $\Delta \mathrel{|\!\!\sim}_K (\forall x)(\varphi(x) \rightarrow \neg\overline{C}(x))$. Then, by explanation, there exists $\psi(a) \wedge \neg\overline{C}(a) \in \Delta$ such that $\psi(a) \wedge K \vdash \varphi(a)$. But then we have $\psi(a) \wedge K \vdash \alpha(a)$ as well, so again $\Delta \not\mathrel{|\!\!\sim}_K (\forall x)(\alpha(x) \rightarrow \overline{C}(x))$.
7. Notice that if $r$ covers a negative example of $\Delta$, that particular example will remain in $\Delta \cup \Delta'$.
8. Let $R \in IR_K(\Delta_1 \cup \Delta_2)$. It means that $R$ at least covers a positive example $e^+ \in \Delta_1 \cup \Delta_2$ and covers no negative example of $\Delta_1 \cup \Delta_2$, so it covers no negative example of both $\Delta_1$ and $\Delta_2$. Now, if $e^+ \in \Delta_1$ then clearly $R \in IR_K(\Delta_1)$; otherwise, if $e^+ \in \Delta_2$, then $R \in IR_K(\Delta_2)$, hence in any case $R \in IR_K(\Delta_1) \cup IR_K(\Delta_2)$.

□

Let us now examine the intuitive interpretation of the properties in Lemma 1 from the point of view of ICL; for this purpose we will reformulate some notions into the vocabulary commonly used in ICL. The first property, Reflexivity, transforms (or *lifts*) every example in $e \in \Delta$ into a rule $r_e$ where constants have been substituted by variables. This *lifting* is usually called in ICL literature the "single representation trick," by which an example in the language of instances is transformed into an expression in the language of generalizations.

Property 2 states that adding a positive example $e^+$ does not invalidate any existing induced rule, i.e. $IR_K(\Delta)$ does not decrease; notice that it can increase since now there are induced rules that explain $e^+$ that were not in $IR_K(\Delta)$ that are in $IR_K(\Delta \cup \{e^+\})$. Property 3 states that adding a negative example $e^-$ might invalidate existing induced rules in $IR_K(\Delta)$, i.e. $IR_K(\Delta \cup \{e^-\}) \subseteq IR_K(\Delta)$. Property 4 states that specializing an induced rule does not imply it is still in $IR_K(\Delta)$, since it may not explain any example in $\Delta$. Properties 5 and 6 state that by generalizing (resp. specializing) an induced rule will never conclude the negation of the target concept.

Property 7 states the well known fact that inductive inference is falsity preserving, i.e. once we know some induced rule is not valid, it will never be valid again. This is related to Property 3, since once a negative example defeats an induced rule $r$, we know $r$ will never be valid regardless of how many examples are added to $\Delta$, i.e. it will never be in $IR_K(\Delta \cup \Delta')$. Property 8 states that the rules that can be induced from the union of two sets of examples are a subset of the union of the rules that can be induced from each of the sets.

The notions of inductive derivation and inducible rules allows us to define next an inductive theory for a concept as a set of inducible rules which, together with the background knowledge, explain all positive examples.

**Definition 5** ***(Inductive Theory)*** *An* inductive theory *$T$ for a concept $\overline{C}$, w.r.t. $\Delta$ and $K$, is a subset $T \subseteq IR_K(\Delta)$ such that for all $\varphi(a) \wedge \overline{C}(a) \in \Delta$, it holds that $T \cup K \cup \{\varphi(a)\} \vdash \overline{C}(a)$. T is* minimal *if there is no $T' \subset T$ that is an inductive theory for $\overline{C}$.*

Since rules in $IR_K(\Delta)$ do not cover any negative example, notice that if $T$ is an inductive theory for $\overline{C}$ w.r.t. $\Delta$ and $K$, and $\psi(a) \wedge \neg\overline{C}(a) \in \Delta$ for some constant $a$, then it holds that $T \cup K \cup \{\psi(a)\} \not\vdash \overline{C}(a)$. In the remainder of this paper we will assume agents have an algorithm capable of generating inductive theories, e.g. [11].

## 3. Multi-agent Induction through Argumentation

We will consider a multi-agent system scenario with two agents $Ag_1$ and $Ag_2$ under the following assumptions: (1) both agents share the same background knowledge $K$[2] and (2) each agent has a set of examples $\Delta_1, \Delta_2 \subseteq \mathcal{L}_e$ such that $\Delta_1 \cup \Delta_2$ is consistent. The goal of each agent $Ag_i$ is to induce an inductive theory $T_i$ of a concept $\overline{C}$ such that $T_i \subseteq IR(\Delta_1 \cup \Delta_2)$ and that constitutes an inductive theory w.r.t. $\Delta_1 \cup \Delta_2$. We will call this problem *multi-agent ICL.*

A naïve approach is for both agents to share their sets of examples, but that might not be feasible for a number of reasons, like cost or privacy. In this section we will show that by communicating their inductive inferences two agents can also solve the multi-agent inductive concept learning (ICL) problem. Let us present an argumentation-based framework that can model this problem of sharing and comparing inductive inferences in order to address the multi-agent ICL problem.

### *3.1. Computational Argumentation*

Let us introduce the necessary notions of computational argumentation we will use in the rest of this paper. In our setting, an *argumentation framework* will be a pair $\mathcal{A} = (\Gamma, \twoheadrightarrow)$, where arguments are rules, i.e. $\Gamma \subseteq \mathcal{L}_r$.

[2] For simplicity, since both agents share $K$, in the rest of this paper we will drop the $K$ from the notation.

**Definition 6** *Given two rules $R, R' \in \Gamma$, an attack relation $R \twoheadrightarrow R'$ holds when $R = (\forall x)(\alpha(x) \rightarrow \overline{C}(x))$, $R' = (\forall x)(\beta(x) \rightarrow \neg\overline{C}(x))$, and $K \vdash (\forall x)(\alpha(x) \rightarrow \beta(x))$. Otherwise, $R \not\twoheadrightarrow R'$. If $R \twoheadrightarrow R'$ and $R' \not\twoheadrightarrow R$ we say that $R$* defeats *$R'$, otherwise if both $R \twoheadrightarrow R'$ and $R' \twoheadrightarrow R$ (i.e. if $K \vdash (\forall x)(\alpha(x) \leftrightarrow \beta(x))$) we say that $R$* blocks *$R'$.*

As in any argumentation system, the goal is to determine whether a given argument is acceptable (or warranted) according to a given semantics. In our case we will adopt the semantics based on dialogical trees [3,13].

**Definition 7** *Given an argumentation framework $\mathcal{A} = (\Gamma, \twoheadrightarrow)$ and $R_0 \in \Gamma$, an* argumentation line *rooted in $R_0$ in $\mathcal{A}$ is a sequence: $\lambda = \langle R_0, R_1, R_2, \ldots, R_k \rangle$ such that:*

1. *$R_{i+1} \twoheadrightarrow R_i$ (for $i = 0, 1, 2, \ldots k$),*
2. *if $R_{i+1} \twoheadrightarrow R_i$ and $R_i$ blocks $R_{i-1}$ then $R_i \not\twoheadrightarrow R_{i+1}$.*

Notice that, given Def. 6, an argumentation line has no circularities and is always finite.

We will be interested in the set $\Lambda(R_0)$ of *maximal* argumentation lines rooted in $R_0$, i.e. those that are not subsequences of other argumentation lines[3] rooted in $R_0$. It is clear that $\Lambda(R_0)$ can be arranged in the form of a tree, where all paths from the root to the leaf nodes exactly correspond to all the possible maximal argumentation lines rooted in $R_0$. In order to decide whether $R_0$ is accepted in $\mathcal{A}$, the nodes of this tree are marked U (undefeated) or D (defeated) according to the following (cautious) rules:

1. every leaf node is marked U
2. each inner node is marked U iff all of its children are marked D, otherwise it is marked D

Then the status of a rule $R_0$ in the argumentation framework $\mathcal{A}$ is defined as follows:

- $R_0$ will be *accepted* if $R_0$ is marked U in the tree $\Lambda(R_0)$
- $R_0$ will be *rejected* if $R_0$ is marked D in the tree $\Lambda(R_0)$

In this way, we decide the status of each argument and define two sets:

$$Accepted(\mathcal{A}) = \{R \in \Gamma \mid R \text{ is accepted}\} \qquad Rejected(\mathcal{A}) = \Gamma \setminus Accepted(\mathcal{A})$$

### 3.2. Argumentation-based Induction

Given a set of examples $\Delta$, and an argumentation framework $\mathcal{A} = (\Gamma, \twoheadrightarrow)$, such that $IR(\Delta) \subseteq \Gamma$, we can define the set $AIR(\Delta, \mathcal{A})$ of argumentation-consistent induced rules as those induced from $\Delta$ which are accepted by $\mathcal{A}$, i.e. $AIR(\Delta, \mathcal{A}) = IR(\Delta) \cap Accepted(\mathcal{A})$. This allows us to define argumentation-consistent inductive theories.

**Definition 8** *An argumentation-consistent inductive theory $T$ for a concept $C$, with respect to $\Delta$, and an argumentation framework $\mathcal{A} = (\Gamma, \twoheadrightarrow)$, such that $IR(\Delta) \subseteq \Gamma$, is an inductive theory of $\Delta$ such that $T \subseteq AIR(\Delta, \mathcal{A})$.*

In other words, an argumentation-consistent inductive theory is an inductive theory composed of rules which have not been defeated by the arguments known to an agent.

[3] An argumentation line $\lambda_1$ is a subsequence of another one $\lambda_2$ if the set of arguments in $\lambda_1$ is a subset of the set of arguments in $\lambda_2$.

Let us see now how can argumentation and induction be combined in order to model the multi-agent ICL problem for two agents. The main idea is that agents induce rules from the examples they know, and then they share them with the other agent. Rules are then contrasted using an argumentation framework, and only those rules which are consistent are accepted in order to find a joint inductive theory.

Thus, in addition to $K$ and the set of examples $\Delta_i$, each agent has a different argumentation framework $\mathcal{A}_i$, corresponding to its individual point of view. Let us analyze the situation where each agent $Ag_i$ communicates all its inducible rules $IR(\Delta_i)$ to the other agent. As a result, each agent will have the same argumentation framework $\mathcal{A}^* = (IR(\Delta_1) \cup IR(\Delta_2), \twoheadrightarrow)$. Given a rule $R \in \mathit{Accepted}(\mathcal{A}^*)$, clearly there are no counterexamples of $R$ in either $\Delta_1$ or in $\Delta_2$ (given the reflexivity property the arguments corresponding to those examples would defeat $R$ otherwise). Thus, if $T_1^*$ and $T_2^*$ are argumentation-consistent inductive theories of $\Delta_1$ and $\Delta_2$ respectively with respect to $\mathcal{A}^*$, then $T_1^* \cup T_2^*$ is clearly a (joint) inductive theory w.r.t. $\Delta_1 \cup \Delta_2$.

Therefore, two agents can reach their goal of finding a joint inductive theory w.r.t. $\Delta_1 \cup \Delta_2$, by sharing all of their inductive inferences $IR(\Delta_1)$ and $IR(\Delta_2)$, then computing individually an argumentation-consistent inductive theory, $T_1^*$ and $T_2^*$ respectively, and then computing the union $T_1^* \cup T_2^*$. In other words, by sharing all the inductive inferences and using argumentation, agents can also reach their goal in the same way as sharing all the examples. However, sharing the complete $IR(\Delta_i)$ is not a practical solution since it can be very large. Nevertheless, not all arguments in $IR(\Delta_i)$ need to be exchanged. We will present a process that finds a joint inductive theory w.r.t. $\Delta_1 \cup \Delta_2$ without forcing the agents to exchange all their complete $IR(\Delta_i)$.

During this process, agents will communicate rules to each other. Let us call $S_j^t$ to the set of rules that an agent $Ag_j$ has communicated $Ag_i$ at a given time $t$ during this process. Moreover, we assume that $S_j^t \subseteq IR(\Delta_j)$, i.e. that the rules communicated by the agent $Ag_j$ are rules that $Ag_j$ has been able to induce with its collection of examples. Thus, for two agents, $\mathcal{A}_1 = (IR(\Delta_1) \cup S_2, \twoheadrightarrow)$ (i.e. $Ag_1$ will have as arguments all the inducible rules for the agent plus the rules shared by the other agent $Ag_2$); and analogously $\mathcal{A}_2 = (IR(\Delta_2) \cup S_1, \twoheadrightarrow)$.

For each argument $R \in \mathit{Rejected}(\mathcal{A}_i)$, let us denote by $\mathit{Defeaters}_i(R)$ the set of undefeated children of $R$ in the argument tree $\Lambda(R)$ in $\mathcal{A}_i$ (which will be non-empty by definition). Two agents can find a *joint inductive theory* w.r.t. $\Delta_1 \cup \Delta_2$ as follows:

1. Before the first round, $t = 0, S_1^0 = \emptyset, S_2^0 = \emptyset, T_1^0 = \emptyset, T_2^0 = \emptyset$.
2. At each new round $t$, starting at $t = 1$, each agent $Ag_i$ performs two actions:
   (a) Given $Ag_i$'s argumentation framework $\mathcal{A}_i^t = (IR(\Delta_i) \cup S_j^{t-1}, \twoheadrightarrow)$, $Ag_i$ generates a argumentation-consistent inductive theory $T_i^t$ w.r.t. its examples $\Delta_i$ such that $(T_i^{t-1} \cap \mathit{Accepted}(\mathcal{A}_i^{t-1})) \subseteq T_i^t$, and $(T_i^t \cap \mathit{Rejected}(\mathcal{A}_i^{t-1})) = \emptyset$, i.e. the new theory $T_i^t$ contains all the accepted rules from $T_i^{t-1}$ and replaces the rules that were defeated in $T_i^{t-1}$ by new rules.
   (b) $Ag_i$ creates a set of attacks $\mathcal{R}_i^t$ in the following way. Let $\mathcal{D} = \{R \in \mathit{Rejected}(\mathcal{A}_i^t) \cap S_j^{t-1} \mid \mathit{Defeaters}_i(R) \cap S_i^{t-1} = \emptyset\}$. $\mathcal{D}$ basically contains all the arguments sent by the other agent which are, according to $Ag_i$, defeated but $Ag_j$ might not be aware of (since $Ag_i$ has not shared with $Ag_j$ any of the

arguments which defeats them). $\mathcal{R}_i^t$ is created by selecting a single argument (whichever) $R' \in \mathit{Defeaters}_i(R)$ for each $R \in \mathcal{D}$. That is, $\mathcal{R}_i^t$ contains one attack for each argument that $Ag_i$ considers defeated, but $Ag_j$ is not aware of.

3. Then, a new round starts with: $S_i^t = S_i^{t-1} \cup T_i^t \cup \mathcal{R}_i^t$. When $S_1^t = S_1^{t-1}$ and $S_2^t = S_2^{t-1}$, the process terminates, i.e. when there is a round where no agent has sent any further attack.

If the set $\Delta_1 \cup \Delta_2$ is consistent, when the process terminates each agent $Ag_i$ has an argumentation-consistent inductive theory $T_i^t$ w.r.t. $\Delta_i$ that is also consistent with the examples $\Delta_j$ of the other agent $Ag_j$ (but it might not be an argumentation-consistent inductive theory w.r.t. $\Delta_j$). However their union $T_1^t \cup T_2^t$ is an inductive theory w.r.t. the examples in $\Delta_1 \cup \Delta_2$ and since both agents know $T_1^t$ and $T_2^t$, both agents can have an argumentation-consistent inductive theory w.r.t. $\Delta_1 \cup \Delta_2$. Notice that $Ag_1$ can obtain from $T_1^t \cup T_2^t$ a minimal inductive theory $T' \cup T_2^t$ where $T' \subseteq T_1^t$ is the minimum set of rules that cover those examples in $\Delta_1$ not covered by $T_2^t$ (and analogously for $Ag_2$).

**Lemma 2** *If the set $\Delta_1 \cup \Delta_2$ is consistent, the previous process always ends in a finite number of rounds t, and that when it ends $T_1^t \cup T_2^t$ is an inductive theory w.r.t. $\Delta_1 \cup \Delta_2$.*

*Proof:* First, let us prove that the final theories ($T_1^t$ and $T_2^t$) are consistent with $\Delta_1 \cup \Delta_2$. For this purpose we will show that the termination condition ($S_1^t = S_1^{t-1}$ and $S_2^t = S_2^{t-1}$) implies that the argumentation-consistent inductive theory $T_i^t$ found by an agent $Ag_i$ at the final round $t$ has no counterexamples in either $\Delta_1$ nor in $\Delta_2$.

Let us assume that there is an example $a_k \in \Delta_1$ which is a counterexample of a rule $R \in T_2^t$. Because of the reflexivity property, there is a rule $R_k \in IR(\Delta_1)$ which corresponds to that example. Since $\Delta_1 \cup \Delta_2$ is consistent, there is no counterexample of $R_k$, and thus $R_k$ is undefeated. Since, by assumption $R_k \twoheadrightarrow R$, $R_k$ should have been in $S_1^{t-1}$, $R$ would have been defeated, and therefore rule $R$ could not be part of any argumentation-consistent inductive theory generated by $Ag_2$. The analogous proof can be used to prove that there are no counterexamples of $T_1^t$ in $\Delta_1 \cup \Delta_2$.

Given that $T_i^t$ is an inductive theory w.r.t. $\Delta_i$, $T_1^t \cup T_2^t$ is an inductive theory w.r.t. $\Delta_1 \cup \Delta_2$ because it has no counterexamples in $\Delta_1 \cup \Delta_2$, and every example in $\Delta_1 \cup \Delta_2$ is explained at least by one rule in $T_1^t$ or $T_2^t$.

Finally, the process has to terminate in a finite number of steps, since, by assumption, $IR(\Delta_1)$ and $IR(\Delta_2)$ are finite sets, and at each round sets $S_1^t$ and $S_2^t$ grow at least with one new argument, but since $S_i^t \subseteq IR(\Delta_i)$, there is only a finite number of new arguments that can be added to $S_1^t$ and $S_2^t$ before the termination condition holds. □

The process to find a *joint inductive theory* can be seen as composed of three mechanisms: induction, argumentation and belief revision. Agents use induction to generate general rules from concrete examples, they use argumentation to decide which of the rules sent by another agent can be accepted, and finally they use belief revision to revise their inductive theories in light of the arguments sent by other agents. The belief revision process is embodied by how the set of accepted rules $\mathit{Accepted}(\mathcal{A}_i^t)$ changes from round to round, which also determines how an agent inductive theory changes in light of arguments shared by the other agent[4].

[4] For reasons of space an example of the execution is not included in this paper, but it can be found at `http://www.iiia.csic.es/~santi/papers/IL2010_extended.pdf`

## 4. Related Work

Peter Flach [6] introduced a logical analysis of induction, focusing on hypothesis generation. In Flach's analysis induction is studied on the meta-level of consequence relations, and focuses on different properties that may be desirable for different kinds of induction, while we focus in a limited form of induction, namely inductive concept learning, extensively studied in machine learning.

Computational argumentation is often modeled using Dung's abstract approach [4], that consider arguments as atomic nodes linked through a binary relation called "attack". On the other hand there are argumentation systems [12,7,8,2] which take as basis a logical language and an associated consequence relation used to define an argument. Some of these systems, like [7] use a logic programming language defined over a set of literals and an acceptability semantics based on dialectical trees is applied in order to determine the "acceptable arguments". In our argumentation approach, we argue about the acceptability of induced rules from examples with a well defined notion of attack relation, and the semantics is based on dialectical trees.

Finally, about the use of argumentation for concept learning, let us mention two related works. Ontañón and Plaza [11] study an argumentation-based framework (A-MAIL) that allows agents to achieve a shared, agreed-upon meaning for concepts. Concept descriptions are created by agents using inductive learning and revised during argumentation until a convergent concept description is found and agreed-upon. A-MAIL integrates inductive machine learning and MAS argumentation in a coherent approach where the belief revision mechanism that allows concept convergence is sound w.r.t. induction and argumentation models.

Amgoud and Serrurier [1] propose an argumentation framework for the inductive concept learning problem. In their framework, both examples and hypotheses are considered as arguments and they define an attack relation among them following Dung's framework. However, they do not model the inductive process of generating hypotheses from examples, but assume that a set of candidate hypotheses exists.

## 5. Conclusions and Future Work

This paper has two main contributions. First, we have presented a logical characterization of the inductive inference used in inductive concept learning, a common problem in machine learning. Additionally, we have proposed an argumentation-based approach to model the process of communication of inductive inferences which appears in multi-agent inductive concept learning. This combination of induction with argumentation in a common model is the second contribution to the paper. This combination is useful in itself, as we have shown elsewhere [11], for communication in multi-agent systems and for learning from communication. But more importantly, this combination of induction with argumentation shows the usefulness of developing a logical characterization of induction; without a formal framework to model induction there would be no possibility to combine with other forms of inference and reasoning, as for example the defeasible form of reasoning that is argumentation.

Our future work will focus on moving from a Boolean approach to a graded (or weighted) approach. ICL techniques usually accept generalizations that are not 100%

consistent with the set of examples. We intend to investigate a logic model of induction where generalizations have an associated confidence measure. Integrating induction with argumentation can make use of a confidence measure, specifically by considering weighted argumentation frameworks [5], where attacks may have different weights. We intend to investigate how weighted attacks and confidence-based induction could be modeled using multivalued or graded logics.

## Acknowledgements

We are grateful to Prof. Francesc Esteva for his insights during discussions on the earlier drafts of this paper and the anonymous reviewers for their valuable comments. Research partially funded by the projects Agreement Technologies (CONSOLIDER CSD2007-0022), ARINF (TIN2009-14704-C03-03), Next-CBR (TIN2009-13692-C03-01), LoMoReVI (FFI2008-03126-E/FILO), and by the grants 2009-SGR-1433 and 2009-SGR-1434 of the Generalitat de Catalunya.

## References

[1] Leila Amgoud and Mathieu Serrurier. Arguing and explaining classifications. In *Proc. AAMAS-07*, pages 1–7, New York, NY, USA, 2007. ACM.

[2] Philippe Besnard and Anthony Hunter. *Elements of Argumentation*. The MIT Press, 2008.

[3] Carlos Chesñevar and Guillermo Simari. A lattice-based approach to computing warranted beliefs in skeptical argumentation frameworks. In *Proc. of IJCAI-07*, pages 280–285, 2007.

[4] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.

[5] Paul E. Dunne, Anthony Hunter, Peter McBurney, Simon Parsons, and Michael Wooldridge. Inconsistency tolerance in weighted argument systems. In *Proc. of the AAMAS'09*, pages 851–858, 2009.

[6] Peter A. Flach. Logical characterisations of inductive learning. In *Handbook of defeasible reasoning and uncertainty management systems: Volume 4 Abductive reasoning and learning*, pages 155–196. Kluwer Academic Publishers, Norwell, MA, USA, 2000.

[7] Alejandro J. García and Guillermo R. Simari. Defeasible logic programming an argumentative approach. In *Theory and Practice of Logic Programming*, pages 95–138. Cambridge University Press, 2004.

[8] Guido Governatori, Michael J. Maher, Grigoris Antoniou, and David Billington. Argumentation semantics for defeasible logic. *J. Log. and Comput.*, 14(5):675–702, 2004.

[9] N. Lavrač and S. Džeroski. *Inductive Logic Programming. Techniques and Applications*. Ellis Horwood, 1994.

[10] Tom Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[11] Santiago Ontañón and Enric Plaza. Multiagent inductive learning: an argumentation-based approach. In *Proc. ICML-2010, 27th International Conference on Machine Learning*, pages 839–846. Omnipress, 2010.

[12] Henry Prakken and Giovanni Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics*, 7(1), 1997.

[13] Nicolás Rotstein, Martín Moguillansky, and Guillermo Simari. Dialectical abstract argumentation: a characterization of the marking criterion. In *Proc. of IJCAI-09*, pages 898–903, 2009.