A Dialectic Approach to Problem-Solving

Eric Martin, Jean Sallantin

A dialectic approach to problem-solving

Eric Martin[1] and Jean Sallantin[2]

[1] School of Computer Science and Eng., UNSW Sydney NSW 2052, Australia,
emartin@cse.unsw.edu.au
[2] LIRMM, CNRS UM2, 161, rue Ada, 34 392 Montpellier Cedex 5, France,
js@lirmm.fr

Abstract. We analyze the dynamics of problem-solving in a framework which captures two key features of that activity. The first feature is that problem-solving is a social game where a number of problem-solvers interact, rely on other agents to tackle parts of a problem, and regularly communicate the outcomes of their investigations. The second feature is that problem-solving requires a careful control over the set of hypotheses that might be needed at various stages of the investigation for the problem to be solved; more particularly, that any incorrect hypothesis be eventually refuted in the face of some evidence: all agents can expect such evidence to be brought to their knowledge whenever it holds. Our presentation uses a very general form of logic programs, viewed as sets of rules that can be activated and fire, depending on what a problem-solver is willing to explore, what a problem-solver is willing to hypothesize, and what a problem-solver knows about the problem to be solved in the form of data or background knowledge.
Our framework supports two fundamental aspects of problem-solving. The first aspect is that no matter how the work is being distributed amongst agents, exactly the same knowledge is guaranteed to be discovered eventually. The second aspect is that any group of agents (with at one end, one agent being in charge of all rules and at another end, one agent being in charge of one and only one rule) might need to sometimes put forward some hypotheses to allow for the discovery of a particular piece of knowledge in finite time.

1 Introduction

The last century has seen the advent of a number of frameworks that place rationality at the heart of the process of scientific discovery; still none of those frameworks has endowed epistemology with a definitive mathematical foundation. The seminal research of Herbert Simon on the logical theorist and McCarthy's research on commonsense reasoning are two prominent examples of general attempts at interfacing the thinking of rational agents and the dynamics of scientific discovery, before more specific approaches, tackling more specific problems, have appeared. Departing more or less from Boole's framework, a plethora of logics, mainly developed in the AI community, have grounded various rational approaches to problem solving; some of them have been validated

by the implementation of tools, successfully applied to the resolution of a broad class of problems. Formal Learning Theory, PAC learning and Query learning, developed around the prominent work of Gold [4], Valiant [8] and Angluin [2], offer theoretical concepts, rooted in recursion theory, statistics and complexity theory, to describe the process of data generalization. Induction has been studied from different angles, in particular by Pierce and Suppes, before the Inductive Logic Programming community suggested a more practical approach. Numerous investigations on automatic or semi-automatic scientific discovery have taken place [5, 7, 3, 1]. Finally, let us mention the more recent work on the relationships between scientific discovery and game theory; but these pointers do not by far exhaust the whole body of work on the relationship between rationality and scientific discovery.

Our approach is based on an extension of Parametric logic [6], a new framework that unifies logic and formal learning theory, developed along three dimensions, two of which are illustrated in this paper.

- The first dimension is formal. It equates logical discovery with theorem proving, in a logical setting where the work of scientists boils down to inferring a set of theorems. We consider two binary categories of agents. The first category opposes *independent agents*, who work alone, to *social agents*, who share the work. The second category opposes *theoretical agents*, who have no time nor space restrictions on the inferences they can perform, including the ability to perform transfinite inferences, to *empirical agents*, whose inferences must be performed in finite (but unbounded) time with finite (but unbounded) memory.
- The second dimension is cognitive, and applies to the way theorems can be derived, based on two key notions: *postulates* and *hypotheses*. Postulates are what agents use when they organize their work; they represent statements whose validity will be assessed "later." Postulates allow for particular scheduling or "outsourcing" of the work. Hypotheses are what agents assume in order to seed or activate a proof. Hypotheses can turn out to be confirmed or refuted, they can end up being plausible or paradoxical.
  The categorization of agents is based on how they deal with postulates and hypotheses. Theoretical agents do not need hypotheses whereas empirical agents might. Independent agents might not need postulates whereas social agents do.

2 The logical framework

2.1 An illustrative example

Imagine the following game. Countably many copies of every card in a deck of 52 cards are available to a *game master*. The game master chooses a particular ω-sequence of cards. For instance, she might choose the sequence consisting of nothing but the ace of spades. Or she might choose the sequence where the queen of hearts alternates with the four of spades, starting with the latter. A number of

players, who do not know which sequence has been chosen by the game master, can make requests and ask her to reveal the nth card in the sequence, for some natural number n. The players aim at eventually discovering which sequence of cards has been chosen by the game master, or to discover some of its properties.

The game illustrates the process of *scientific discovery*, with the game master playing the role of Nature, and the players the role of the scientists. A feature of the game is that unless the game master has explicitly ruled out a large number of possible sequences, the players usually cannot, at any point in time, know whether their guesses are correct: they might at best be able to *converge in the limit* to correct guesses. We have not precisely defined what a "guess" is. There has to be a language where some properties of a sequence of cards can be described, and the expressive power of the language is crucial in circumscribing what the players can or cannot achieve. Let us refer to such a description as a *theory*, in analogy to the work of scientists whose aim is to discover theories that correctly describe or predict some aspects of the field of study. In this paper, we will let *logic programs* play the role of theories.

2.2 Logical background

$\mathbb{N}$ denotes the set of natural numbers and Ord the class of ordinals. We consider a finite *vocabulary* $\mathcal{V}$ consisting of a constant $\overline{0}$, a unary function symbol s, the *observational* predicate symbols, namely, the unary predicate symbols

$$\text{hearts spades diamonds clubs ace two } \ldots \text{ ten jack queen king}$$

and a number of other predicate symbols. For all nonzero $n \in \mathbb{N}$, we denote by $\overline{n}$ the term obtained from $\overline{0}$ by n successive applications of s; $\overline{n}$ will refer to the nth card. We denote by $\mathrm{Prd}(\mathcal{V})$ the set of predicate symbols in $\mathcal{V}$. Given $n \in \mathbb{N}$, we denote by $\mathrm{Prd}(\mathcal{V}, n)$ the set of members of $\mathrm{Prd}(\mathcal{V})$ of arity n. We fix a countably infinite set of (first-order) variables and a repetition-free enumeration $(v_i)_{i\in\mathbb{N}}$ of this set. We need a notation for the set of all possible sequences of cards.

Definition 1. *We call* possible game *any set T of closed atoms such that:*

- *for all $n \in \mathbb{N}$, T contains one and only one member of hearts($\overline{n}$), spades($\overline{n}$), diamonds($\overline{n}$), clubs($\overline{n}$);*
- *for all $n \in \mathbb{N}$, T contains one and only one member of ace($\overline{n}$), two($\overline{n}$), ..., ten($\overline{n}$), jack($\overline{n}$), queen($\overline{n}$), king($\overline{n}$);*
- *T contains no other atom.*

We consider a notion of logical consequence that is best expressed on the basis of a forcing relation $\Vdash$, based on both principles that follow.

- The intended interpretations are *Herbrand structures*: every individual has a unique name (a numeral); this is because intended interpretations are ω-sequences of cards—$\overline{n}$ being the name of the nth card in the sequence.

– Disjunction and existential quantification are constructive: an agent will derive a disjunction iff she has previously derived one of the disjuncts, and she will derive an existential sentence iff she has previously derived one of the closed instances of the sentence's matrix.

We denote by $\mathcal{L}_{\omega\omega}(\mathcal{V})$ the set set *sentences*, that is, closed first-order formulas over $\mathcal{V}$. Given two sets S and T of sentences, we write $S \Vdash T$ iff S forces all members of T.

2.3 Logic programs and occurrence markers

A formal logic program provides, for every $n \in \mathbb{N}$ and $\wp \in \mathrm{Prd}(\mathcal{V}, n)$, two rules: one whose head is $\wp(v_1, \ldots, v_n)$, and one whose head is $\neg\wp(v_1, \ldots, v_n)$. This is at no loss of generality since the left hand side of the rules can contain equality and the intended interpretations are Herbrand. So to define a formal logic program, we only need the left hand side of both rules associated with a predicate symbol and its negation. It is convenient, and fully general as well, to assume that all variables that occur free on the left hand side of a rule also occur on the right hand side of the rule.

Definition 2. *A* logic program (over $\mathcal{V}$) *is defined as a family of pairs of formulas over $\mathcal{V}$ indexed by* $\mathrm{Prd}(\mathcal{V})$*, say* $((\varphi^+_\wp, \varphi^-_\wp))_{\wp\in\mathrm{Prd}(\mathcal{V})}$*, such that for all* $n \in \mathbb{N}$ *and* $\wp \in \mathrm{Prd}(\mathcal{V}, n)$*,* $\mathrm{fv}(\varphi^+_\wp) \cup \mathrm{fv}(\varphi^-_\wp)$ *is included in* $\{ v_1, \ldots, v_n \}$.

An important particular kind of logic program is the folllowing.

Definition 3. *Let a logic program* $\mathcal{P} = ((\varphi^+_\wp, \varphi^-_\wp))_{\wp\in\mathrm{Prd}(\mathcal{V})}$ *be given. We say that* $\mathcal{P}$ *is* symmetric *iff for all* $\wp \in \mathrm{Prd}(\mathcal{V})$*,* $\varphi^-_\wp = {\sim}\varphi^+_\wp$.

To distinguish between agents, we need the key notion of *occurrence marker*, which intuitively is a function that selects some occurrences of literals in some formulas. Let ψ be a nullary predicate symbol or the negation of a nullary predicate symbol. An agent could select an occurrence o of ψ in a formula φ because she wants to (provisionally) assume that ψ is either true or false, at least in the particular context of ψ occurring in φ at occurrence o. We will see that social agents will make use of the opportunity of assuming that ψ is false, whereas empirical agents will make use of the opportunity of assuming that ψ is true. Actually, ψ does not have to be nullary for these ideas to be developed (we will need more generality anyway), so the definitions that follow deal with arbitrary literals, not only literals built from a nullary predicate symbol. The underlying idea is the same, though it was more easily explained under the assumption that ψ is nullary.

We want to be able to select occurrences of literals in the left hand sides of the rules of a logic program. This justifies the definition that follows.

Definition 4. *Let a logic program* $\mathcal{P} = ((\varphi^+_\wp, \varphi^-_\wp))_{\wp\in\mathrm{Prd}(\mathcal{V})}$ *be given. An* occurrence marker for $\mathcal{P}$ *is a sequence of the form* $((O^+_\wp, O^-_\wp))_{\wp\in\mathrm{Prd}(\mathcal{V})}$ *where for all members* $\wp$ *of* $\mathrm{Prd}(\mathcal{V})$*,* $O^+_\wp$ *and* $O^-_\wp$ *are sets of occurrences of literals in* $\varphi^+_\wp$ *and* $\varphi^-_\wp$*, respectively.*

What we need is to be able to replace some occurrences of literals in some formulas by some other formulas. Given a formula φ and a partial function ρ from the set of occurrences of literals in φ to $\mathcal{L}_{\omega\omega}(\mathcal{V})$, we denote by $\varphi[\rho]$ the result of applying ρ to φ. For instance, if ρ is the function that maps the first occurrence of p in $\varphi = p \wedge (q \vee p)$ to $r \wedge s$, then $\varphi[\rho] = (r \wedge s) \wedge (q \vee p)$.

3 Independent and social agents

Let a logic program $\mathcal{P}$ and an occurrence marker Ω for $\mathcal{P}$ be given. Suppose that $\mathcal{V}$ contains n predicate symbols for some nonzero $n \in \mathbb{N}$, so there are $2n$ rules in $\mathcal{P}$, n positive rules and n negative rules, say $R_0, \ldots R_{2n-1}$. Imagine that for all $m < 2n$, R_m is 'under the responsability' of some agent A_m (a single agent might be responsible for many rules in $\mathcal{P}$, possibly all of them). Let $m < 2n$ be given. Some occurrences of literals in R_m might be marked by Ω. Intuitively, these are the occurrences of literals that A_m 'does not bother to' or 'is not able to' directly deal with: a marked occurrence of literal in R_m is assumed by A_m to be false *unless* A_m *is told otherwise* (expectedly by another agent, but possibly by himself...), for instance because those literals are not under A_m's responsibility—they are instances of rules whose right hand side are under the responsibility of other agents. The definitions that follow formalize these ideas.

Definition 5. *Let a formula φ, a set O of occurrences of literals in φ, and a set E of literals be given. Let ρ be the function from O into $\mathcal{L}_{\omega\omega}(\mathcal{V})$ such that for all $o \in O$, $n \in \mathbb{N}$, $\wp \in \mathrm{Prd}(\mathcal{V}, n)$ and terms $t_1, \ldots, t_n$,*[3]

$$\rho(o) = \begin{cases} \bigvee\{ \bigwedge_{1\le i\le n} t_i = t'_i \mid \wp(t'_1,\ldots,t'_n) \in E \} & \text{if } \wp(t_1,\ldots,t_n) \in o, \\ \bigvee\{ \bigwedge_{1\le i\le n} t_i = t'_i \mid \neg\wp(t'_1,\ldots,t'_n) \in E \} & \text{if } \neg\wp(t_1,\ldots,t_n) \in o. \end{cases}$$

We let $\odot^O_E\varphi$ denote $\varphi[\rho]$.

Definition 6. *Let a logic program $\mathcal{P} = ((\varphi^+_\wp, \varphi^-_\wp))_{\wp\in\mathrm{Prd}(\mathcal{V})}$, a possible game T, and an occurrence marker $\Omega = ((O^+_\wp, O^-_\wp))_{\wp\in\mathrm{Prd}(\mathcal{V})}$ for $\mathcal{P}$ be given. We inductively define a family $([\mathcal{P}, T, \Omega]_\alpha)_{\alpha\in\mathrm{Ord}}$ of sets of closed literals as follows. For all ordinals α, $[\mathcal{P}, T, \Omega]_\alpha$ is the $\subseteq$-minimal set of literals that contains T and such that for all $n \in \mathbb{N}$, $\wp \in \mathrm{Prd}(\mathcal{V}, n)$ and closed terms $t_1, \ldots, t_n$,*

- $\wp(t_1,\ldots,t_n) \in [\mathcal{P}, T, \Omega]_\alpha$ *iff* $[\mathcal{P}, T, \Omega]_\alpha \Vdash \odot^{O^+_\wp}_{\bigcup_{\beta<\alpha}[\mathcal{P},T,\Omega]_\beta} \varphi^+_\wp[t_1/v_1,\ldots,t_n/v_n]$;
- $\neg\wp(t_1,\ldots,t_n) \in [\mathcal{P}, T, \Omega]_\alpha$ *iff* $[\mathcal{P}, T, \Omega]_\alpha \Vdash \odot^{O^-_\wp}_{\bigcup_{\beta<\alpha}[\mathcal{P},T,\Omega]_\beta} \varphi^-_\wp[t_1/v_1,\ldots,t_n/v_n]$.

We set $[\mathcal{P}, T, \Omega] = \bigcup_{\alpha\in\mathrm{Ord}}[\mathcal{P}, T, \Omega]_\alpha$.

The independent agent does everything by herself; she does not rely on anyone. If we assume that she works 'nonstop' then her behavior is captured by the empty occurrence marker.

[3] In case $n = 0$, the replacing expression is $\bigvee\{ \bigwedge\varnothing \}$ if $\wp \in E$, and $\bigvee\varnothing$ if $\wp \notin E$. Note that $\bigvee\{ \bigwedge\varnothing \}$ is logically equivalent to $\bigwedge\varnothing$.

Definition 7. *Let a logic program* $\mathcal{P} = ((\varphi^+_\wp, \varphi^-_\wp))_{\wp\in\mathrm{Prd}(\mathcal{V})}$ *and a possible game* T *be given. Let* $\Omega = ((O^+_\wp, O^-_\wp))_{\wp\in\mathrm{Prd}(\mathcal{V})}$ *be the occurrence marker for* $\mathcal{P}$ *such that for all* $\wp \in \mathrm{Prd}(\mathcal{V})$, $O^+_\wp$ *and* $O^-_\wp$ *are empty. We write* $[\mathcal{P}, T]$ *for* $[\mathcal{P}, T, \Omega]$.

The next result shows that social agents, irrespective of how their responsibility has been defined, discover the same information, no less, not more, as the independent agent.

Proposition 1. *For all logic programs* $\mathcal{P}$, *possible games* T *and occurrence markers* Ω *for* $\mathcal{P}$, $[\mathcal{P}, T, \Omega] = [\mathcal{P}, T]$.

4 Theoretical and empirical agents

In the previous section, we have allowed agents to interact transfinitely many times: in $[\mathcal{P}, T, \Omega]_\alpha$, we allow α to be an infinite ordinal. In this section, we tackle the following issue: is it possible to derive all derivable information in finite time, irrespective of how social agents share their work, or of how single agents organize their work? Obviously, this requires a way of 'working' different to what the concepts that have been defined so far accept. In this section, we will allow agents to make *hypotheses*. If an agent can assume that some literals in the bodies of some rules are true, she might be able to speed up the derivations she can perform. Such hypotheses should abide stringent conditions. We suggest that a hypothesis should eventually either be *confirmed*, that is, proved correct, or *refuted*, that is, proved wrong. Let us first precisely define what 'making a hypothesis' means. A pleasant feature of this notion is that it is again based on the notion of occurrence marker. This time, we use occurrence markers to select some occurrences of literals on the left hand side of some rules to make them the targets of some hypotheses.

Definition 8. *Let a formula* φ, *a set* O *of occurrences of literals in* φ, *and a set* E *of literals be given. Let* ρ *be the function from* O *into the set of formulas such that for all* $o \in O$, $n \in \mathbb{N}$, $\wp \in \mathrm{Prd}(\mathcal{V}, n)$ *and terms* $t_1, \ldots, t_n$, $\rho(o)$ *is equal to* $\bigvee\{\, \wp(t_1, \ldots, t_n), \bigwedge_{i=1}^n t_i = t'_i \mid \wp(t'_1, \ldots, t'_n) \in E \,\}$ *if* $\wp(t_1, \ldots, t_n) \in o$, *and to* $\bigvee\{\, \neg\wp(t_1, \ldots, t_n), \bigwedge_{i=1}^n t_i = t'_i \mid \neg\wp(t'_1, \ldots, t'_n) \in E \,\}$ *if* $\neg\wp(t_1, \ldots, t_n) \in o$. *We let* $\circledcirc^O_E\varphi$ *denote* $\varphi[\rho]$.

An agent willing to assume that the literals in E are true provided that they occur on the left hand side of the rules of a logic program $\mathcal{P}$, as selected by the occurrence marker Ω for $\mathcal{P}$, essentially decides to work on the basis of the logic program $\mathcal{P} +_\Omega E$ introduced in the definition that follows.

Definition 9. *Let a logic program* $\mathcal{P}$ *and an occurrence marker* Ω *for* $\mathcal{P}$ *be given. Write* $\mathcal{P} = ((\varphi^+_\wp, \varphi^-_\wp))_{\wp\in\mathrm{Prd}(\mathcal{V})}$ *and* $\Omega = ((O^+_\wp, O^-_\wp))_{\wp\in\mathrm{Prd}(\mathcal{V})}$. *Given a set* E *of literals, the sequence* $((\circledcirc^{O^+_\wp}_E \varphi^+_\wp, \circledcirc^{O^-_\wp}_E \varphi^-_\wp))_{\wp\in\mathrm{Prd}(\mathcal{V})}$ *is denoted* $\mathcal{P} +_\Omega E$.

Our aim is to show that making hypotheses can pay off.

Definition 10. *A logic program* $\mathcal{P} = ((\varphi^+_\wp, \varphi^-_\wp))_{\wp \in \mathrm{Prd}(\mathcal{V})}$ *is* acceptable *iff the following holds. Let* $\mathcal{V}^*$ *be* $\mathcal{V}$ *without the observational predicate symbols.*

- *For all possible games* T, $[\,\mathcal{P}, T\,]$ *is a complete set of literals.*
- *The restriction of* $\mathcal{P}$ *to* $\mathcal{V}^*$ *is symmetric.*
- *For all* $\wp \in \mathrm{Prd}(\mathcal{V})$,
  - *if* $\wp$ *is observational then both* $\varphi^+_\wp$ *and* $\varphi^-_\wp$ *are equal to* $\bigvee \varnothing$,
  - *either* $\wp$ *is nullary or no quantifier occurs in* $\varphi^+_\wp$, *and*
  - *all quantified formulas that occur in* $\varphi^+_\wp$ *have one quantifier only.*

Here is an example of part of an acceptable logic program.

$$\begin{aligned}
\forall v_1\,((\mathrm{hearts}(v_1) \vee \mathrm{diamonds}(v_1)) &\rightarrow \mathrm{red}(v_1))\\
\forall v_1\,((\mathrm{spades}(v_1) \vee \mathrm{clubs}(v_1)) &\rightarrow \mathrm{black}(v_1))\\
\forall v_0\,(\mathrm{red}(v_0) \leftrightarrow \mathrm{black}(s(v_0))) &\rightarrow \mathrm{alternatedColors}\\
(\forall v_0\,\mathrm{red}(v_0) \vee \exists v_0\,(\mathrm{queen}(v_0) \wedge \mathrm{clubs}(v_0))) &\rightarrow \mathrm{allRedsOrAQofC}
\end{aligned}$$

The proposition that follows shows that it is possible to enrich $\mathcal{V}$ into a vocabulary $\mathcal{V}'$, transform $\mathcal{P}$ into a logic program $\mathcal{P}'$ over $\mathcal{V}'$, and make some assumptions such that all possible games T, all members of $[\,\mathcal{P}, T\,]$ can be derived after a finite number of steps. Moreover, $\mathcal{P}'$ is such that it is safe to make any set of assumptions; indeed, any set of assumptions that is inconsistent with $\mathcal{P}'$ and a possible game will proved inconsistent after finitely many inferences.

Proposition 2. *Let* $\mathcal{V}^*$ *be* $\mathcal{V}$ *without the observational predicate symbols. For all acceptable logic programs* $\mathcal{P}$, *there exists a finite set* E *of nullary predicate symbols that do not belong to* $\mathcal{V}$ *and there exists a logic program* $\mathcal{P}'$ *over* $\mathcal{V} \cup E$ *whose restriction to* $\mathcal{V}^* \cup E$ *is symmetric such that for all possible games* T, *there exists an occurrence marker* Ω *for* $\mathcal{P}'$ *with the following properties.*

- $[\,\mathcal{P}, T\,]$ *and the restrictions of* $[\,\mathcal{P}', T\,]$ *and* $[\,\mathcal{P}' +_\Omega E, T\,]$ *to* $\mathcal{V}$ *are equal;*
- *for all occurrence markers* Ω' *for* $\mathcal{P}'$, $[\,\mathcal{P}' +_\Omega E, T\,] = \bigcup_{n\in\mathbb{N}} [\,\mathcal{P}' +_\Omega E, T, \Omega'\,]_n$;
- *for all possible games* T *and for all occurrence markers* Ω' *and* Ω'' *for* $\mathcal{P}'$, *if* $[\,\mathcal{P}' +_{\Omega''} E, T\,] \neq [\,\mathcal{P}', T\,]$ *then* $\bigcup_{n\in\mathbb{N}} [\,\mathcal{P}' +_{\Omega''} E, T, \Omega'\,]_n$ *is inconsistent.*

The transformation of $\mathcal{P}$ to $\mathcal{P}'$ amounts to replacing some complex formulas in the bodies of some rules of $\mathcal{P}$ by some new nullary predicate symbols, themselves defined thanks to a new pair of rules—a form of predicate invention—that can play the role of hypotheses and enjoy a refutation property. With the previous example of acceptable logic program, E could consist of two nullary predicate symbols, say p and q, and $\mathcal{P}'$ could be defined as

$$\begin{aligned}
\forall v_1\,((\mathrm{hearts}(v_1) \vee \mathrm{diamonds}(v_1)) &\rightarrow \mathrm{red}(v_1))\\
\forall v_1\,((\mathrm{spades}(v_1) \vee \mathrm{clubs}(v_1)) &\rightarrow \mathrm{black}(v_1))\\
\forall v_0\,(\mathrm{red}(v_0) \leftrightarrow \mathrm{black}(s(v_0))) &\rightarrow p\\
p &\rightarrow \mathrm{alternatedColors}\\
\forall v_0\,\mathrm{red}(v_0) &\rightarrow q\\
(q \vee \exists v_0\,(\mathrm{queen}(v_0) \wedge \mathrm{clubs}(v_0))) &\rightarrow \mathrm{allRedsOrAQofC}
\end{aligned}$$

An agent would then have four options, depending on whether she would assume p or q in the bodies of the 4th and 6th rules, respectively. For any possible game T, one of these options would be appropriate and allow the agent to discover whether T is a sequence of cards where black and red alternate, or whether T is a sequence consisting of nothing but red cards, unless it contains a queen of clubs. Any wrong set of hypotheses would be guaranteed to be eventually refuted in the limit on the basis of a finite subset of T.

5 Conclusion

We have presented a framework where fundamental questions about the nature of scientific discovery can be formulated and studied. The basic working hypothesis is that a purely logical approach to scientific discovery and problem solving is possible, in a way that can shed light on the nature of those activities. We believe that our approach can address a whole range of questions related to the nature of scientific discovery or problem solving, always within the boundaries of a pure logical setting. For instance, Angluin proposes a binary categorization of agents, with learners and teachers, and she proves robustness results about their interaction; how does this categorization translate into our setting? Starting from a fixed language, we have to a certain extent accounted for predicate invention in the last proposition, allowing agents to make a rational use of hypotheses expressed in an extension of the original language, but how does predicate invention relate to postulates? Surely, logic is not an iron collar, but it can potentially strive far beyond the territories where it has been confined to.

References

1. M. Afshar, C. Dartnell, D. Luzeaux, and J. Sallantin. Aristotle's square revisited to frame discovery science. *Journal of Computers*, 2(5):54–66, 2007.
2. D. Angluin and M. Krikis. Learning from different teachers. *Machine Learning*, 51(2):137–163, 2003.
3. D. Chavalarias and J.-P. Cointet. Bottom-up scientific field detection for dynamical and hierarchical science mapping—methodology and case study. *Scientometrics*, 75(1), 2008.
4. M. E. Gold. Language identification in the limit. *Information and Control*, 10(5):447–474, 1967.
5. P. W. Langley, G. L. Bradshaw, and H. A. Simon. Rediscovering chemistry with the BACON system. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*. Springer, 1984.
6. E. Martin, A. Sharma, and F. Stephan. Deduction, induction and beyond in parametric logic. In Michele Friend, Norma B. Goethe, and Valentina S. Harizanov, editors, *Induction, Algorithmic Learning Theory, and Philosophy*, volume 9 of *Logic, Epistemology and the Unity of Science*. Springer, 2007.
7. L. N. Soldatova, A. Clare, A. Sparkes, and R. D. King. An ontology for a robot scientist. *Bioinformatics*, 22(14):e464–e471, 2006.
8. L. L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.