AIAI'09

IJCAI'09 International Workshop on Abductive and Inductive Knowledge Development

PRE WORKSHOP PROCEEDINGS

In collaboration with the
21st International Joint Conference on Artificial Intelligence
IJCAI'09

http://www.cs.bris.ac.uk/∼oray/AIAI09/
http://ijcai-09.org/

Pasadena, California, USA
July 12, 2009

Introduction

Abduction and induction are forms of logical inference with many applications in AI. In isolation, abduction is a type of explanatory-based reasoning appropriate for tasks such as planning and diagnosis, while induction is a type of generality-based reasoning appropriate for tasks like classification and learning. In combination, previous studies have shown that abduction and induction can be usefully integrated in order to provide a more powerful reasoning framework. But recent experience in applying such hybrid approaches to real-world problems is beginning to suggest that even greater benefits might be achieved by incorporating these techniques into a more refined cycle of knowledge development where domain theories evolve incrementally in response to continued feedback from the user or domain being modelled.

This Workshop will explore new ways of using abduction and induction to assist the evolution of knowledge in real-world problems In particular, it aims to better understand how abduction and induction can be used within a continuous cycle of knowledge development. Our goal is to examine possible applications and to investigate how the revision cycle can be better supported by computational tools. Drawing on analogies with methodology of science, we will explore and build upon recent work in the mechanisation of theory revision, hypothesis evaluation, and the design of experiments. In this way, we will provide an inter-disciplinary forum for researchers seeking to formalise and automate such knowledge development cycles in various disciplines (scientific, applied, cognitive or philosophical).

Workshop Chair

Oliver Ray (University of Bristol, UK)

Organising Committee

Peter Flach (University of Bristol, UK)
Antonis Kakas (University of Cyprus, Cyprus)
Oliver Ray (University of Bristol, UK)

Programme Committee

Michael Agar (The University of Maryland and Ethknoworks LLC, USA)
Atocha Aliseda-Llera (Universidad Nacional Autonoma de Mexico, Mexico)
Chitta Baral (Arizona State University, USA)
Peter Flach (University of Bristol, UK)
Jerry Hobbs (University of Southern California, USA)
Katsumi Inoue (National Institute of Informatics, Japan)
John Josephson (The Ohio State University and Aetion Technologies LLC, USA)
Antonis Kakas (University of Cyprus, Cyprus)
Ross King (University of Aberystwyth, UK)
Oliver Ray (University of Bristol, UK)

Programme

08.45 OPEN

09.00 Toward an abductive foundation of semantic science
David Poole

09.30 Towards a model of collective knowledge discovery
Gauvain Bourgne and Katsumi Inoue

10.00 COFFEE

10.30 Invited Talk: Four insights about mechanisms for abductive processing
John Josephson

11.15 Higher-Order Logic Learning and Lambda-Progol
Niels Pahlavi and Stephen Muggleton

11.45 Multimodal Abduction in Knowledge Development
Lorenzo Magnani

12.15 LUNCH

14.00 Towards the automation of scientific method
Oliver Ray, Amanda Clare, Maria Liakata, Larisa Soldatova, Ken Whelan and Ross King

14.30 Explaining Effects of Host Gene Knockouts on Brome Mosaic Virus Replication
Deborah Chasman, Brandi Gancarz, Paul Ahlquist and Mark Craven

15.00 COFFEE

15.30 Invited Talk: The Role of Openness in Scientific Automation: a case for Open Notebook Science
Jean-Claude Bradley

16.15 Knowledge Evolution in Geologic Mapping
Boyan Brodaric

16.45 The Reasoning Processes underlying Claude Bernard's Scientific Discoveries
Bassel Habib and Jean-Gabriel Ganascia

17:15 PANEL DISCUSSION

18.15 CLOSE

Table of Contents

Towards an abductive foundation of semantic science

David Poole
Department of Computer Science,
University of British Columbia,
2366 Main Mall, Vancouver, B.C., Canada V6T 1Z4
poole@cs.ubc.ca
http://cs.ubc.ca/~poole/

Abstract

The aim of semantic science is to have scientific data and scientific theories represented and published in machine understandable form. There is much work on developing scientific ontologies and representing scientific data in terms of these ontologies. The next step is to publish scientific theories that can make predictions on the published data and can be used for prediction on new cases. This can be used to advance the development of science and to provide useful predictions that can be evaluated according to all available data.

To make a prediction for a particular case, we need to use an ensemble of theories that fit together (are consistent) and make a prediction on a particular case. We argue that this is a form of abduction, that has similarities and differences to the standard definitions of abduction. This is preliminary work, presenting pre-theoretic foundations of the field.

Introduction

The basic idea of semantic science [Poole et al., 2008] is:

- Information is published using well defined ontologies [Smith, 2003b] to allow semantic interoperability.
- People publish data [Fox et al., 2006; McGuinness et al., 2007] described using the vocabulary specified by the ontologies. Part of this data includes metadata about what the data is about and how it was generated. Data repositories include the Community Data Portal (http://cdp.ucar.edu/) and the Virtual Solar-Terrestrial Observatory (http://vsto.hao.ucar.edu/index.php).
- Scientists publish theories that make predictions on data. These theories make reference to ontologies. These predictions can be tested on the published data. As part of each theory is information about what data this theory is prepared to make predictions about.
- New data can be used to evaluate, and perhaps update, the theories that make predictions on this data. Predictions on new data can be used to judge the theories as well as find outliers in the data, which can be statistical anomalies, fraudulent data or some new, little understood phenomenon.
- The descriptions of competing theories can be used to devise experiments that will distinguish the theories.
- If someone wants to make a prediction for a new case (e.g., a patient in a diagnostic setting, or predicting a landslide), they can use the best theories to make the prediction. They would either use the best theory or theories, or average over all theories weighted by they ability to predict this phenomenon of interest. The use will be able to ask for what evidence there is for the theory.
- There is no central authority to vet as to what counts as legitimate scientific theories. Each of us can choose to make decisions based on the whichever theories we want. We will be able to judge theories by their predictions on unseen data and other criteria.
- We expect semantic science search engines to be developed. Given a theory, a search engine would be able to find data that can be used to evaluate or tune the theory. Given data, a search engine would be able to find the theories that make predictions on the data.

The relationship amongst ontologies, data and theories is given in Figure 1. The data depends on the world and the ontology. The theories depend on the ontology, indirectly on the world (if a human is designing the theory), and directly on some of the data (as we would expect that the best theories would be based on as much data as possible). Given a new case, theories make predictions about that case that can be used for decision making. The ontologies, data sets and theories, evolve in time.

The term "science" is meant to be as broad as possible. We can have scientific theories about any natural or artificial phenomenon. We could have scientific theories about traditional disciplines such as earth sciences, physics, chemistry, biology, medicine and psychology but we would also imagine theories as diverse as predicting which companies will be most profitable, predicting where the best parties are, or predicting who will win football games. The only criteria is that a scientific theory must put itself at risk by making predictions about observable phenomenon.

Semantic science has no prior prejudice about the source or the inspiration of theories; as long as the theories are prepared

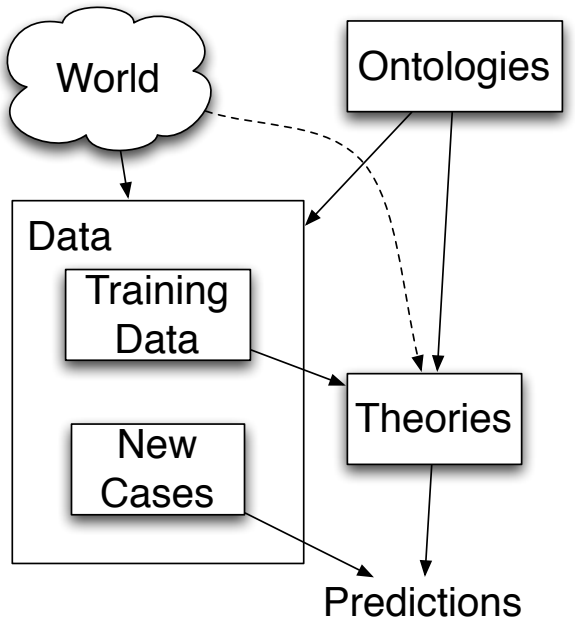

Figure 1: Role of Ontologies, Data and Theories in Semantic Science

to make predictions about unseen data, they can be included. We are not, a priori, excluding religion, astrology, or other areas that make claim to the truth; if they are prepared to make predictions about what will be observed, we can test how well their predictions fit the available data, and use their predictions for other data.

Semantic science is trying to be broad and bottom-up. It should serve to democratize science in allowing not just the elite to create data and theories. Like scientists themselves, it should be skeptical of all of the information it is presented with.

We anticipate that the most useful theories will make probabilistic predictions, however theories can make whatever predictions they like. Users of the theories can choose to adopt theories based on whatever criteria they like, e.g., some combination of fit to the existing data and simplicity or prior plausibility. Users can also choose to ignore theories that don't make the sort of predictions they like.

This is like the machine learning vision, where the data sets are heterogeneous and published with respect to formal ontologies. The theories also persist and can be compared when new data arrives. We expect the highest standards to be used in evaluation of the theories. For the foreseeable future virtually all theories will be a mix of human generated and machine learned; humans define the structure and parameter space and the machines optimizes these with respect to fit to data and learning biases. Semantic science provides a mechanism for Bayesian inference, where we need to condition on *all* relevant information that was not part of building the model. A semantic science search engine should allow us to find all of the relevant data on which to condition.

To make this project manageable, we can define four levels of semantic science:

0. Deterministic semantic science where all of the theories make definitive predictions. This class includes both propositional and first-order theories. This has been studied under the umbrella of abductive logic programming [Kakas and Denecker, 2002].
1. Feature-based semantic science, where there are non-deterministic[1] predictions are about feature values of data. This is the most common form of machine learning. Such theories can be specified in terms of random variables that represent the values of features.
2. Relational semantic science, where the predictions are about the properties of objects and relationships among objects. In this case, the values of properties may be meaningless names; the structure of the relationships is used to make predictions. This is what has been studied in relational learning [Getoor and Taskar, 2007; De Raedt et al., 2008].
3. First-order semantic science, where the aim is to make predictions about the existence of objects or predictions about universally quantified statements. This is more challenging as conditioning is not well-defined [Poole, 2007]. We may not know which object in the world the theory is making a prediction about, as the theory may refer to the existence of object filling a role, but we may know which object fills the role.

In the rest of this paper, we only consider the first two of these. We will describe these in terms of features. Features generalize propositions, as a proposition is a Boolean feature. Features can also be seen as properties of a single individual under consideration. There can also be global features that are not about any individual. Seeing the features as properties allow for a correspondence with the work on ontologies. An attribute is a feature-value (or property-value) pair, for example that rock's age is 50 million years is an attribute of the rock.

Predictions from Theories

Scientific theories are typically narrow; they don't make predictions on arbitrary sets of data. For example, someone may develop a theory for the prognosis of a particular type of lung cancer. To use this theory for a prediction of a particular patient, we first predict whether the patient has this form of lung cancer, then use this theory to predict the prognosis. We need other theories about the prognosis for the possibility that the patient has a different form of lung cancer, or doesn't have lung cancer.

A set of theories that fit together to make a prediction for a particular case is called a *theory ensemble*. The theories in a theory ensemble must be consistent, in a way we describe below, and must be enough to make a prediction in a particular case.

There is a correspondence between theory ensembles and explanations in abduction. There will be a direct match for

[1] Non-deterministic can mean many things. Here we consider just the case where there are probabilistic predictions. But there are many alternatives, such as qualitative predictions, probability ranges or fuzzy predictions.

level 0 semantic science. The other levels will be more complicated when the theories make probabilistic predictions, and when we can't even be sure that a theory makes semantic sense for a particular data set.

We don't assume that theories are explicitly specified, in the sense that theories can can be arbitrarily complex and use arbitrary computation to make predictions.

Ontologies

In AI, an ontology [Smith, 2003b; Noy and Hafner, 1997; Gómez-Pérez et al., 2004] is a specification of the meaning of vocabulary used by an information system. Ontologies form the backbone of the Semantic Web [Berners-Lee et al., 2001]. There has recently been much work in standardizing ontologies, such as using the Web Ontology Language OWL [McGuinness and van Harmelen, 2004]. Science is one of the areas where ontology development and deployment is well under way [Smith et al., 2007].

Ontologies can be very complicated, as would be expected in a world where language has evolved to be useful and new terminology is invented to describe what was not easy to describe using previous terminology.

We have been advocating a structure for ontologies using what we call Aristotelian definitions [Smith, 2003a; Poole et al., 2009], based on the idea of Aristotle [350 B.C.] that each class should be described in terms of a super-class (the genus) and property values (the differentia) that differentiate this class from other subclasses of the genus. Defining all classes in terms of properties, as opposed to specifying subclass relationships directly, simplifies reasoning as we only need to give the values of properties and the class structure logically follows. It is also a natural way to define concepts in many cases. Simple Aristotelian definitions often give rise to complicated subclass relationships, but simple subclass relationships give simple Aristotelian definitions.

For the rest of this paper, we will thus ignore classes, and consider only features (conflating features and properties as we are only considering feature-based semantic science). Properties, however, have domains; they are only defined in the context where other properties have particular values. Properties are not defined in when their domain does not hold.

Data

We assume that data is published referring to the ontologies used. As part of each data set, for the purpose of this paper, assume the following meta-data is specified:

- The context in which the data was collected. This is a proposition made up of assignments to features. For example, if the data was of people who have a certain type of cancer, the context would be the attributes that define the people and the attributes that define the type of cancer.
- The features that this data makes predictions about (what is often called the dependent variables).
- The features that were controlled for in the data (the independent variables).

To predict such data, a theory needs to predict the values of the dependent variables as a function of the context and the independent variables.

Theories

Each theory makes predictions about some feature values or property values of an individual.

We assume a theory has three components:

- A context in which specifies preconditions of when it can be applied. This is a proposition that must be true for the theory to make sense.
- A set of input features about which it does not make predictions.
- A set of output features about which is can make a prediction (as a function of the input features).

For example, the ideal gas law is a theory that makes predictions about the pressure P, volume V, number of particles n and the temperature in the context of a gas, namely that $PV \propto nT$. It makes predictions that can be judged against data. There are alternative theories are be more accurate for real gasses, e.g., when the pressure is high, and the gas molecules are heterogeneous. This theory is not applicable to rocks or to lung cancer.

Theories are not universally applicable; for example we can't use a theory about the prognosis of people with cancer on rocks. Theories have preconditions that specify what they make predictions about. These preconditions are of three different sorts:

- Conditions which define when the theory makes sense. When these conditions are false, the theory is nonsense. The conditions are the domains of the features used in the theories.
- Conditions which define the intended scope of the theory. These conditions specify what the theory was designed to predict.
- Conditions which specify when the theory will be used in a particular theory ensemble.

For example, a theory that makes predictions of the prognosis of patients with lung cancer may be applicable for arbitrary people. In a particular theory ensemble, it may only be used for the patients with lung cancer who have not had some particular drug, as the theory ensemble may use another theory that makes predictions in that case.

One class of theories that is of particular interest is the "null hypothesis". There is a null hypothesis for each feature. This theory says that the feature has randomly distributed values, with probabilities that are independent of the other features. It is important as it is always applicable, and gives a base case upon which to compare other theories.

Theory Ensembles

To make a prediction, we need more than a single theory. We need to use multiple theories that fit together to make a prediction. We call such a collections of theories a *theory ensemble*. We expect a formal definition of theory ensembles to

be quite complex to cover the richness of real theories. However, there does seem some to be properties that we can define independently of any formalism.

A theory ensemble T needs to satisfy the following properties:

- T is coherent: it does not rely on the value of a feature in a context where the features is not defined (i.e., outside of the domain of the feature). Thus if feature f has domain d, it has to be used in a context where d is true. For example, writing $d \wedge f$, which is false if d is false, and has the value of f otherwise, would satisfy coherence.
- T is consistent: it does not make different predictions for any feature in any context.
- T is predictive: it makes a prediction in every context that is possible. Thus if we have a theory that includes $a \rightarrow b$, and we need to make a prediction on b, then we need to have our theory ensemble imply a, or also predict b in the context of $\neg a$.
- T is minimal in that it does not include theories that are not required to be predictive.

For level-0 semantic science, this corresponds to the standard definition of abduction. The predictive condition corresponds to being able to prove the goal. Coherence is also needed for theories that use ontologies, but if we make the domain of a property as a precondition for the property, coherence is entailed by the other three properties for the deterministic case.

For type 1 semantic science, the situation is more complex, and there is still much more research required to get a satisfactory definition of a theory ensemble. A simplistic notion of a theory ensemble for a particular piece of data (that contains a context, values for its independent variables and values for its dependent variables) consists of a set of $\langle c, t\rangle$ pairs where t is a theory and c is a proposition which implies the domains of the properties used in t. The pair $\langle c, t\rangle$ specifies that theory t will be used for predictions in the context c. The following example shows how this notion of a theory can be used with the properties defined above to give a prediction:

Example 1 Suppose we have data about a person who coughs, and we want to make predictions about their prognosis. We have the following Boolean random variables (we will use the lower case variant as the proposition that the variable is true):

- *Person* is true if the object is a person.
- *L* is true if the person will live for more than a year (it gives the prognosis of a person).
- *HC* is true if the person has cancer
- *HLC* is true if the cancer the person has is lung cancer
- *Coughs* is true if the person coughs

Suppose the background ontology specifies the *person* is the domain of the properties *L*, *HC* and *Coughs*. The domain of *HLC* is *hc* (i.e., we can only talk about the value of *HLC* when *HC* is true).

Suppose we have the following theories that have been published:

- T_1 is about the prognosis of people with lung cancer.
- T_2 is about the prognosis of people with cancer.
- T_3 is the null hypothesis that gives the prognosis of people in general.
- T_4 predicts (probabilistically) whether people with cancer have lung cancer, as a function of coughing (i.e., *hc* is the context, *Coughs* is the independent variable and *HLC* is the dependent variable).
- T_5 predicts (probabilistically) whether people have cancer

A possible theory ensemble is $\{\langle person, T_5\rangle, \langle \neg hc, T_3\rangle, \langle hc, T_4\rangle, \langle hlc, T_1\rangle, \langle hc \wedge \neg hlc, T2\rangle\}$

In this ensemble, although T_2 can make predictions for anyone with cancer, it is only used for those without lung cancer. Similarly T_3 is only used when the person does not have cancer.

If T_4 made definitive predictions about lung cancer, only one of $\langle hlc, T_1\rangle$ and $\langle hc \wedge \neg hlc, T2\rangle$ would be in the theory ensemble.

This example has ignored many of the details of real theories. Theories can make predictions about many features and an ensemble may not need to use all of them. We need to treat conditions differently depending of whether they are part of the context, whether they are observed in the data and when they are not observed in the data. We also need to be concerned with how the theories interact with the ontologies.

Frequently Asked Questions

There are a number of questions that have been asked. Some for which I have a reasonable answer are here.

Will this replace peer review?

There is a related question of "What is the role of humans in semantic science?" In some sense the goal of semantic science is to let the computers do what that are good at, and let humans do what they are good at. This is true for evaluation too; we should use computers to evaluate what computers are good at evaluating and let humans evaluate what they are good at evaluating. Computers are (should be) good at evaluating how well theories fit data. But fit to data isn't the only property we want of a scientific theory. We also want insight; computers are not as good at evaluating this. We also want a notion of simplicity and elegance of theories, which may be hard to formally specify for a computer. So semantic science will not replace peer review, but will give extra tools for which to evaluate science.

What is to prevent fraudulent data and theories?

It might seem that the enterprise will break down on fraudulent data, as people publishing fraudulent data can make their theories look good. Suppose someone was to post fraudulent data. First, existing theory ensembles that make predictions on that data will be surprised by the data; they will conclude that the data is very unlikely. Conditioned on the new data, the theory ensembles will become less likely. Next someone could propose a theory that the data is anomalous or fraudulent (or perhaps such a hypothesis could always automatically

available). This could have a small prior probability that may depend on the source of the data. The theory ensembles will then split into those that adopt the theory that the data is fraudulent, and those that do not. These new theory ensembles can be evaluated. A theory ensemble that concludes that all of the data is anomalous will not be very likely. A theory ensemble that can account for anomalous data will be become more likely. Thus the general mechanism of evaluating theory ensembles can handle anomalous and fraudulent data.

As theories can make any predictions, it seems as though there could not be fraudulent theories. However, there are two cases that need to be taken into account.

We may expect that theories would declare what data they used to learn from. However a theory could lie. This may be problematic if the theory is trained on the data used to evaluate theories. Such a theory would look good. However, such theories will not continue to look good when tested on brand new data.

The other way a theory can be fraudulent is to use other theories without acknowledgment. If not all theories are open, it is possible to steal parts of other theories. The use of theory ensembles is meant to mitigate any advantage that could be obtained by doing this. Reusing other theories is a legitimate part of semantic science, so there seems to be no advantage of stealing other theories. Theory ensembles also give credit where credit is due.

How does it relate to ensemble learning?

Ensemble learning [Dietterich, 2002] is a common technique for combining multiple learners to get a better prediction. The ensembles are typically a combination of predictions of a target feature based on the input features. Combining predictions for a single feature is definitely allowed as part of a theory ensemble. There are however reasons why this may not become commonplace. First, we don't expect the theories to be developed independently; theories are developed building on previous theories, trying to improve them. Much of the work on ensemble learning is about how to sensibly generate the classifiers that will be combined, whereas here we have predictors that are not designed to be combined. Second, the producers of such theories will want to make the best theories possible. The tools that are available to them include ensemble learning. It doesn't seem that cascaded ensemble learning will work well.

Theory ensembles can be an arbitrary combination of theories. This will range from the linear averaging of bagging to the conditional application of the example given above. A specification language for theory ensembles will allow all such combinations.

Theory ensembles are also related to algorithm portfolios [Xu et al., 2008], which learn which algorithms to run based on features of the problem being solved. We expect that similar learning could be used to choose which theories make the best predictions under which circumstances. Just as the algorithm portfolio learning can use algorithms that were not designed to be used with the portfolio, we expect that scientists will develop theories without needing to be concerned about how they will be used.

How can data, theories, ensembles and ontologies evolve in time?

We expect data to be continually published. Ontologies evolve to accommodate new categories of observations. Theories also improve. Theories can use whatever internal computations they like to make predictions. However, if one theory wants to use the some internal feature of an another theory, or if that feature is added to the data, the vocabulary to describe that feature needs to be added to the ontology. Ontologies can be evaluated by whether the distinctions they describe are useful in making predictions. Thus as theories become more sophisticated, the distinctions they need to make their predictions are added to the vocabulary, and are incorporated into the data.

How do we get there?

There is currently much work on developing scientific ontologies and publishing data with respect to these ontologies [Smith et al., 2007; Fox et al., 2006; McGuinness et al., 2007]. It would seem that publishing ontologies and data will only continue to grow. Scientists want others to use their data, as do their funders. There is growing recognition of the need to develop ontologies to allow for the sharing of such data and other information. Scientists care about the language used to describe their science. Many become involved in developing scientific vocabulary, and the formal representation in ontologies, because they don't want others to define the vocabulary that will become standard.

There has been much less work on developing theories. We have built some systems in geology, for minerals exploration and landslide susceptibility [Jackson, Jr. et al., 2008; Sharma et al., 2009] that represent published theories that make predictions about a limited number of properties. These systems were quite complicated as they reasoned about the probability of existence of individuals that filled roles.

Based on this experience, we recognized that there is still work to be done on feature-based representations before we try to extend it to relational and first-order representations. We need to build future systems on solid foundations.

By first developing level-1 semantic science based on features, we should be able to develop firm foundations for this case, in much the same way that machine learning has been able to develop in this context. We can then move to relations and then to first-order semantic science.

Conclusion

This paper has sketched some pre-theoretic ideas on how theory ensembles work and their relationship to explanations in abduction. I believe that it is important to get the pre-theoretic notions correct before creating a formalism that can be studied as an abstract entity. I also expect that there will be many iterations of getting the definitions of theories and theory ensembles right.

The potential of semantic science seems huge, but there are many technical and social issues that need to be solved before it can come to maturity. The development of ontologies and the publishing of data using those ontologies has advanced greatly in recent years. The main technical issues remaining are to do with the representations of the theories and the-

ory ensembles and the infrastructure to publish and search for data and theories. To bring this vision of semantic science to fruition will require advances in many fields.

References

Aristotle (350 B.C.). *Categories*. Translated by E. M. Edghill, http://www.classicallibrary.org/Aristotle/categories/.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, pp. 28–37. URL http://www.sciam.com/article.cfm?id=the-semantic-web.

De Raedt, L., Frasconi, P., Kersting, K., and Muggleton, S.H. (Eds.) (2008). *Probabilistic Inductive Logic Programming*. Springer.

Dietterich, T.G. (2002). Ensemble learning. In M. Arbib (Ed.), *The Handbook of Brain Theory and Neural Networks,*, pp. 405–408. MIT Press, Cambridge, MA, second edition. URL http://web.engr.oregonstate.edu/~tgd/publications/hbtnn-ensemble-learning.ps.gz.

Fox, P., McGuinness, D., Middleton, D., Cinquini, L., Darnell, J., Garcia, J., West, P., Benedict, J., and Solomon, S. (2006). Semantically-enabled large-scale science data repositories. In *5th International Semantic Web Conference (ISWC06)*, volume 4273 of *Lecture Notes in Computer Science*, pp. 792–805. Springer-Verlag. URL http://www.ksl.stanford.edu/KSL_Abstracts/KSL-06-19.html.

Getoor, L. and Taskar, B. (Eds.) (2007). *Introduction to Statistical Relational Learning*. MIT Press, Cambridge, MA.

Gómez-Pérez, A., Fernández-López, M., and Corchu, O. (2004). *Ontological Engineering*. Springer.

Jackson, Jr., L.E., Smyth, C.P., and Poole, D. (2008). Hazardmatch: an application of artificial intelligence to landslide susceptibility mapping, Howe Sound area, British Columbia. In *4th Canadian Conference on Geohazards*.

Kakas, A. and Denecker, M. (2002). Abduction in logic programming. In A. Kakas and F. Sadri (Eds.), *Computational Logic: Logic Programming and Beyond*, number 2407 in LNAI, pp. 402–436. Springer Verlag. URL http://www2.cs.kuleuven.be/cgi-bin/dtai/publ_info.pl?id=39495.

McGuinness, D., Fox, P., Cinquini, L., West, P., Garcia, J., Benedict, J.L., and Middleton, D. (2007). The virtual solar-terrestrial observatory: A deployed semantic web application case study for scientific research. In *Proceedings of the Nineteenth Conference on Innovative Applications of Artificial Intelligence (IAAI-07)*. Vancouver, BC, Canada. URL http://www.ksl.stanford.edu/KSL_Abstracts/KSL-07-01.html.

McGuinness, D.L. and van Harmelen, F. (2004). OWL web ontology language overview. W3C Recommendation 10 February 2004, W3C. URL http://www.w3.org/TR/owl-features/.

Noy, N.F. and Hafner, C.D. (1997). The state of the art in ontology design: A survey and comparative review. *AI Magazine*, 18(3): 53–74. URL http://www.aaai.org/Library/Magazine/vol18.php\#Fall.

Poole, D. (2007). Logical generative models for probabilistic reasoning about existence, roles and identity. In *22nd AAAI Conference on AI (AAAI-07)*. URL http://www.cs.ubc.ca/spider/poole/papers/AAAI07-Poole.pdf.

Poole, D., Smyth, C., and Sharma, R. (2008). Semantic science: Ontologies, data and probabilistic theories. In P.C. da Costa, C. d'Amato, N. Fanizzi, K.B. Laskey, K. Laskey, T. Lukasiewicz, M. Nickles, and M. Pool (Eds.), *Uncertainty Reasoning for the Semantic Web I*, LNAI/LNCS. Springer. URL http://www.cs.ubc.ca/spider/poole/papers/SemSciChapter2008.pdf.

Poole, D., Smyth, C., and Sharma, R. (2009). Ontology design for scientific theories that make probabilistic predictions. *IEEE Intelligent Systems*, pp. 27–36. URL http://www2.computer.org/portal/web/computingnow/2009/0209/x1poo.

Sharma, R., Poole, D., and Smyth, C. (2009). A framework for ontologically-grounded probabilistic matching. *International Journal of Approximate Reasoning*, to appear.

Smith, B. (2003a). The logic of biological classification and the foundations of biomedical ontology. In D. Westerståhl (Ed.), *Invited Papers from the 10th International Conference in Logic Methodology and Philosophy of Science*. Elsevier-North-Holland, Oviedo, Spain. URL http://ontology.buffalo.edu/bio/logic_of_classes.pdf.

Smith, B. (2003b). Ontology. In L. Floridi (Ed.), *Blackwell Guide to the Philosophy of Computing and Information*, pp. 155—166. Oxford: Blackwell. URL http://ontology.buffalo.edu/smith/articles/ontology_pic.pdf.

Smith, B. et al. (2007). The OBO foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology*, 25(11): 1251–1255. URL http://www.nature.com/nbt/journal/v25/n11/pdf/nbt1346.pdf.

Xu, L., Hutter, F., Hoos, H.H., and Leyton-Brown, K. (2008). SATzilla: portfolio-based algorithm selection for SAT. *Journal of Artificial Intelligence Research*, 32: 565–606. URL http://jair.org/papers/paper2490.html.

Towards a model of collective knowledge discovery

Gauvain Bourgne and **Katsumi Inoue**
National Institute of Informatics, Tokyo, Japan
{bourgne,ki}@nii.ac.jp

Abstract

This paper presents a preliminary framework for modeling knowledge discovery while taking into account possible interactions within a group of scientists. A model is proposed, and different possible levels of interactions are detailed in the process of building hypotheses and assessing them. These process are illustrated by a simple inductive case study in which agents try to discover some laws governing visibility of a target in a basic kind of block world.

1 Introduction

Research in Knowledge Discovery has provided a rich sample of models that account well for the various forms of reasoning and inferences at work in the scientific discovery processes. Especially, a great number of models have focused on a logical perspective [Popper (1959); Curd (1980)], where inference such as induction and abduction is often central. However, following the traditional line of research in Artificial Intelligence where one is interested in understanding or simulating intelligence seen as an individual process, one rather common assumption is that what is being modeled is the thought process of one scientist tackling a research problem (see e.g. [Langley (1978); Kuipers (2001)]). Following [Bourgne and Corruble (2008); Wajnberg et al. (2004)], we argue that, this view can be seen as rather limited, or at least partial, when one looks at the growing literature, coming essentially from the Social Studies community, which study science as a fundamentally social process [Longino (1990); Latour (1979)]. In this paper, we provide a model to illustrate not only inferences of individual scientists but also interactions between them. Though we will illustrate it with an inductive case study, we expect this model to be general enough to take into account various forms of reasoning.

We shall first describe our model, and the different kinds of knowledge and processes it uses. Then we will focus on the different levels of interaction involved in the knowledge discovery process. In [Guerra-Hernandez (2003)], different levels of learning were described, depending on the activity of other agents in the process. Loosely following this gradation, we shall first describe the process of a single agent, and then, at a second level, the case in which a reasoner gets some information from the other agents in order to build its hypothesis. Agents collaborating with the reasoner can be passive, acting as a mere source of information, or take a more active role in the discovery process by criticizing the hypothesis of the main reasoner. At last, we shall consider knowledge discovery as a fully collective process, in which different reasoners might propose, refine or criticize hypotheses. Finally, we shall conclude by discussing our current model and its limitations, and provide leads for future works.

2 Basic model

We shall give here the basic elements of our model. First we will discuss knowledge representation and distribution in the context of groups of agents building hypotheses based on observations and background theories. Then we will present the different steps involved in building a hypothesis, whether by a single reasoner or by a group of reasoners.

2.1 Knowledge distribution

We consider a group of n scientists represented as agents $a_1, \ldots, a_n$. These agents have different kinds of knowledge, which will be discussed below. First, we need to distinguish between *revisable* knowledge and *non-revisable* knowledge. In the ideal case, non-revisable knowledge would be *certain* knowledge that cannot bring inconsistencies, where as revisable knowledge would be the hypothesis that agents are trying to build, and would be susceptible to revision when obtaining new informations. Here, however, we will make a weaker assumption by stating that non-revisable knowledge is knowledge in which the agent is quite confident, and will not consider revising it unless forced to do so by inconsistencies. It is a way to direct the formation of hypotheses by differentiating what is considered (at least temporarily) as certain from what can be more easily revised. Among the non-revisable knowledge, we will distinguish some set of observations, which represent acquired knowledge defining the target of the hypothesis. Observations, here, are

facts that should be explained by the rest of the knowledge. It is the inability to explain some observations that will motivate the formation of a hypothesis to complete the theory. To further distinguish between facts acquired through perception (as would be suggested by the term observation), and non self-explanatory facts that need to be explained by the theory, we shall mark the later ones as *manifestations* (in the sense of [Peng and Reggia (1990)]). A *manifestation* will thus be an observation that is not self-explanatory, and that should be a consequence of the theory and the other observations. If it is not, it means the theory is incomplete and some hypothesis should be made to complete it. At last, since we consider a group of scientists, some distinction must be made about the distribution of the knowledge. To keep the model simple for the time being, we will just consider that some knowledge is either *common* to all agents in the group or *localized* to a particular individual. More refined models would consider sub-groups in the society of agents and further distinguish between group knowledge that are shared only by member of a given sub-group and common knowledge shared by all the society.

We describe below briefly the different kinds of knowledge that will finally be used in our model, and give the associated notations. We do not describe the specific syntax of these elements as the model could apply to different representation, but an example will be given after to illustrate a possible formalization.

Common Theory $\mathcal{T}^C$. This is the common background knowledge of the society. It is considered non-revisable, and is shared by all agents in the society.

Individual Theory $\mathcal{T}_i^I$. This is the individual prior knowledge of agent a_i. It considered non-revisable during the generation of a hypothesis, though if needed, it would be easier to revise than the common theory as it is not shared with other agents.

Observations O_i. These are factual statements representing the individual experience of an agent a_i. It is acquired knowledge that is considered non-revisable and certain (sensors are assumed to be perfect). Some of the observations of an agent might not be direct, and have been obtained through communications with other agents.

Manifestations M_i. These are a special kind of observations that need to be explained. They will be the basis for checking the completeness of a theory or a hypothesis. We have $M_i \subseteq O_i$.

Full theory of an agent $\mathcal{T}_i$. It will sometimes be convenient to refer to all non-revisable knowledge of an agent a_i, given by the full theory $\mathcal{T}_i = \mathcal{T}^C \cup \mathcal{T}_i^I \cup O_i$.

Hypothesis H_i. This is individual revisable knowledge that is used to complete the theory of an agent. We consider in this paper that each individual reasoner only pursues one hypothesis H_i at the same time, though this hypothesis might be composed of several components (different rules for instance). Another perspective could be to consider a set $\mathcal{H}_i$ of possible (interesting) hypotheses, which would be evaluated in parallel.

Common Hypothesis H^C. This would be a collective hypothesis considered by a group of agents in order to complete the common background theory $\mathcal{T}^C$. It will be discussed in the collective knowledge discovery section.

2.2 Coherence and completeness

When building a hypothesis, an agent should try to ensure two properties with respect to its full theory $\mathcal{T}_i$ and its manifestations M_i. The first requirement is that its hypothesis should be *coherent* with the theory. It means that it should not be possible to derive contradictions from the theory and the hypothesis. This property is needed to ensure the soundness of the subsequent reasonings. Then, a hypothesis should also be *complete*, meaning that, when associated to the theory, it should explain all the manifestations (in a non-trivial way). More formally, it means that we should be able to derive every elements of M_i from $(\mathcal{T}_i \setminus M_i) \cup H_i$. An hypothesis that does not explain every element of M_i, but still explain some of them will be called a *partial hypothesis*. As opposed to an incoherent hypothesis, a partial hypothesis might be kept, if no complete hypothesis can be found.

We will illustrate these notions on a running example in which some agents try to discover the rules of a simple simulated environment. Agents can go to some small rooms in which they know there is a target whose position they know. However they might not be able to see it, and they are trying to discover some hypothesis about the conditions in which it is possible to see the target. *Observations* of the agents will be situations, as the ones shown in figure 1.

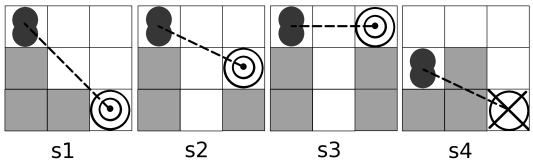

Fig. 1: Initial observations of agent a_1.

Situation s_1 is described by the following set of literals (in situation s, `Pos(`o`,`N`,`s`)` gives the absolute position N of o, `Level(`N`,`lvl`)` gives the level of the ground of absolute position N, and `See(`o`,`s`)` indicates that the agent can see o):

`Pos(ag,1,`s_1`).` `Pos(tgt,3,`s_1`).` `Level(1,high,`s_1`).`
`Level(2,mid,`s_1`).` `Level(3, low,`s_1`).` `See(tgt,`s_1`).`

The literal `See(tgt,`s_1`)`, being the observation we want to explain by the hypothesis would be considered a *manifestation*. When the target cannot be seen (as in s_4), the agent will explicitly observe $\neg$`See(tgt,`S`)`.

Then, in order to abstract more easily from a given situation, agents will use the following common theory $\mathcal{T}^C$ to derive set of predicates relative to

their position (while facing the target):

RPos(tgt,$N_2 - N_1$,S) ← Pos(ag,N_1,S), Pos(tgt,N_2,S), ($N_1 \leq N_2$).

RPos(tgt,$N_1 - N_2$,S) ← Pos(ag,N_1,S), Pos(tgt,N_2,S), ($N_1 > N_2$).

RLevel($N - N_1$,L,S) ← Pos(ag,N_1,S), Pos(tgt,N_2,S), ($N_1 \leq N_2$), Level(N,L,S).

RLevel($N_1 - N$,L,S) ← Pos(ag,N_1,S), Pos(tgt,N_2,S), ($N_1 > N_2$), Level(N,L,S).

Thus, in situation s, RPos(o,r,s) and RLevel(r,lvl,s) gives resp, the relative position r of o and the ground level lvl of relative position r. Using this theory, an agent would derive the following description of situation s_1:

RPos(tgt,2,s_1). RLevel(0,high,s_1). RLevel(1,mid,s_1). RLevel(2,low,s_1).

An hypothesis would then be a set of rules

$$\texttt{See(tgt,}S\texttt{)} \leftarrow \texttt{Lit}_1(\hat{X},S),\ldots,\texttt{Lit}_1(\hat{X},S)$$

where $\texttt{Lit}_i$ are literals whose predicate depends on the situations S. Those could be restricted by the agent to literals whose predicate is either RPos or RLevel. We will not consider absolute position or level Pos and Level in the following for the sake of simplicity. Then a *coherent* hypothesis wrt to $\mathcal{T}^C$ and O_i would be a set of rules that does not infer See(ag,S) when ¬See(ag,S) has been observed in O_i, and a *complete* hypothesis wrt to $\mathcal{T}^C$ and O_i will be a set of rules that derive See(ag,S) in every situation of O_i in which it has been observed.

2.3 Hypothesis formation cycle

Figure 2 depicts the different steps in the discovery of new knowledge through hypothesis building. Each of these processes is discussed below.

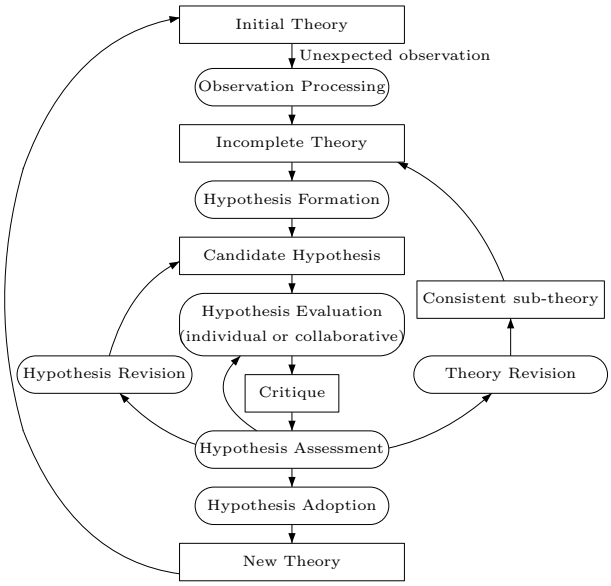

Fig. 2: Hypothesis formation cycle.

Observations Processing. First, after observing a number of facts, an agent processes the observations with its initial theory. It memorizes them, marks some of them as manifestations, and then checks coherence and completeness of the current theory. If non-revisable knowledge is really certain, then coherence should be ensured (if not, there might be a need to drop some part of the theory, as explained in theory revision process). Then if some manifestation cannot be explained, it gets an incomplete theory that should be completed by building hypotheses.

Hypothesis Formation. Having realized that its theory is not yet complete, an agent tries to improve it by building a new hypothesis. There are a number of hypothesis formation processes that can be used at this step, such as Inductive or Abductive Logic Program [Brogi et al. (1997); Muggleton (1991)], CF-Induction [Inoue (2004)], or Inductive Concept Learning algorithms. In the running example, we shall use an incremental bottom-up concept learning algorithm on the ground instance of the agent theory. A full description of this algorithm in the propositional case can be found in [Henniche (1997)].

Hypothesis Evaluation. We will explore this step more in depth in the following sections, depending of the level of interaction involved. Basically, an agent confronts its hypothesis to other observations to check its correctness. There are however different ways to get interesting observations in order to evaluate the hypothesis. After an evaluation phase, an agent gets a critique of its hypothesis, which indicate if the hypothesis is coherent and/or complete or not with the tested observations.

Hypothesis Assessment. At this step, the agent takes the critique into account to decide whether it will revise its hypothesis (e.g. if a critique indicated it was incoherent, or incomplete), continue evaluating it, or adopt it.

Hypothesis Revision. When a candidate hypothesis is rejected because of its incoherence or incompleteness, the agent needs to revise it. It can either build a new hypothesis from scratch, in which case this step is equivalent to a new hypothesis formation step, or used the discarded hypothesis as a basis for finding a better one. In this case, it needs to use some incremental hypothesis formation mechanism. The mechanism used in our example is an example of incremental process.

Theory Revision. If no hypothesis can be generated, an agent might revise its assumption that its theory and observations are non-revisable. In this case it needs to solve incoherences in them. As we rely solely on hypothesis formation to discover new knowledge, we will only withdraw some knowledge at this step. Basically, the agents tries to find a maximal coherent subset of its theory, possibly guided in this process by some heuristics. As a result it should get an incomplete theory that it will try to

improve by generating new hypotheses.

Hypothesis Adoption. If an agent has a very high confidence in its hypothesis, it might decide to consider it as non-revisable and add it to its individual theory $\mathcal{T}_i^I$. Such a process should be done carefully, as it compromises the certainty of non-revisable information. A theory revision might then be needed later on.

3 Individual knowledge discovery

In this section, we shall explain how an agent can get new observations by itself to evaluate its hypothesis. In this case, it does not use the knowledge of the other agents. It is the situation that is most often studied [Thagard (1988); Popper (1959)]. We propose here a simple approach based on passive and active learning Angluin (1987).

3.1 Pursued observation

In order to get new observations that could confirm or refute its hypothesis, an agent might just use some random set of observations, or wait for events to occur that will provide new observations. Such a *passive* approach is equivalent to postponing the evaluation of the current hypothesis until enough new data has been gathered. An agent would return to its previous activity (possibly using its current hypothesis as a basis for reasoning while doing it), memorizing new situations, and resume learning when sufficient feedback is given. It is not very interesting in a epistemic perspective, but could be useful in situations where learning is not the main task of the agent, but something that should be done in parallel to improve its effectiveness in doing its main task.

3.2 Experimentation

A more active way to gather relevant observations is to do some experiments. An agent would then set the parameters of its experiment and observe the manifestations. It can be related to *active learning*, an experiment being a slightly more general kind of *membership query*. An experiment could be conducted in order to evaluate a hypothesis in two main ways. The first way would be to check the coherence of the hypothesis by devising situation in which the hypothesis would infer some manifestations and check if the expected manifestations is indeed observed. Such experiments could refute the hypothesis by making it incoherent with the new observations. Otherwise, it would provide new observations with which the hypothesis is coherent and complete, giving it more support and thus increasing its plausibility. Another kind of experiment would test situations close to the ones in which the hypothesis would infer manifestations in order to check its completeness. If a given manifestation is observed in such contest, it would challenge its completeness, and provide ground for generalizing it. Otherwise, it would still help defining more precisely the boundaries of the hypothesis, by preventing later unsuitable generalizations. This kind of experiment seems to be more suited for inductive hypotheses.

We consider an agent a_1 whose common theory $\mathcal{T}^C$ is the one defined in first example, and whose individual theory $\mathcal{T}_1^I$ is initially empty. This agents observe a number of situations s_1, s_2, s_3, s_4 given in Figure 1. These observations do not contradict $\mathcal{T}^C$, but reveal its incompleteness, since manifestations `See(tgt,`s_i`)` with $i \in \{1, 2, 3\}$ are not explained by it. a_1 thus apply its hypothesis formation process on them and gets a hypothesis H_1 composed of a single rule: $h_{1,1}$: `See(tgt,`S`) ← RLevel(0,high,`S`), RPos(tgt,2,`S`).` It will know seek to do some experiments to check this hypothesis. These experiments are given by figure 3.

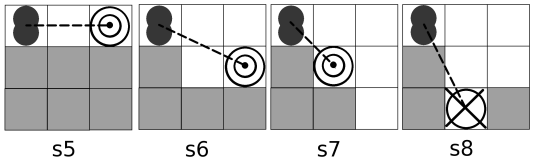

Fig. 3: Experiments of agent a_1.

First, the agent tries to generate other situations that triggers its hypothesis, that is situation in which the agent is on high ground, and the target is two cases away. It builds situation s_5 and s_6, that seems to corroborate H_1. Then, it checks if some generalization might be possible. In order to see if it would be useful to generalize the rule by removing `RPos(tgt,2)` from the body of $h_{1,1}$[1] it will generate situations in which we have `RLevel(0,high)`, but not `RPos(tgt,2)`. First of these situations is s_7, which leads it to revise its hypothesis in $H_1' = \{h_{1,1}'$:`See(tgt,`S`) ← RLevel(0,high,`S`)`$\}$. But then, checking a bit further this new hypothesis, it gets situation s_8 that contradicts it. A revision is needed, and we finally get $H_1'' = \{$
$h_{1,1}''$: `See(tgt,`S`) ← RLevel(0,high,`S`), RPos(tgt,2,`S`).`
$h_{1,2}''$: `See(tgt,`S`) ← RLevel(0,high,`S`), RLevel(1,mid,`S`), RLevel(2,low,`S`).` $\}$

4 Discovering knowledge by information sharing

If an agent is trying to discover some knowledge while being able to communicate with some fellow agents, it can then get observations and informations from them to improve its reasoning. Especially, other agents might be an easier source of information than experiments, which might need the agent to change the environment before observing what it tries to know. There are of course different ways to get information from other agents, as we shall see in the following.

4.1 Observation collection

In the same manner that an agent could get random examples from the environment, an agent could just ask

[1] Note that trying to remove `RLevel(0,high)` would not be useful because of situation s_4.

another agent for some of its observations, without any more detail. However, since the agent has to take an active role anyway in asking for observations, it might be better to give more details about what it would like. We shall rather consider the case in which an agent has some idea about what kinds of observation it would like to get. Typically, it might be considering running some experiment about it, but try to avoid that by using the experience of other agents. In both those case, the queried agent has a rather limited role, since it simply selects some observations to send according to the criterion that is given to it.

Let a_2 be another agent, sharing $\mathcal{T}^C$ with agent a_1. This agent has its own individual theory $\mathcal{T}_2^I$ which contains one rule: $\neg$`RLevel(`X`,`L_2`,`$S`)` $\leftarrow$ `RLevel(`X`,`L_1`,`$S`)`, $L_1 \neq L_2$. He uses this rule to derive explicit negation of relative levels, and consider such literals for the body of the rules of its hypothesis. a_2 however, is not able to modify the levels of the block in the environment. It has only access to one room, in which it has already observed a few situations, shown in figure 4. Note that using $\mathcal{T}^C$, situations s_9 and s_{12} have the same relative description. Using these observations and its theory, a_2 builds hypothesis H_2 which contains single rule

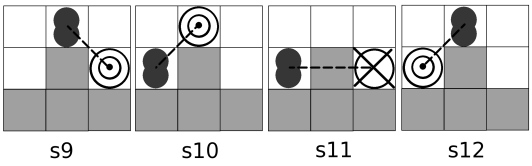

Fig. 4: Initial observations of agent a_2.

$$h_{2,1}\ \text{:}\texttt{See(tgt,}S\texttt{)} \leftarrow \texttt{RPos(tgt,1,}S\texttt{)},\ \neg\texttt{RLevel(1,low,}S\texttt{)}.$$

As it cannot do any significant experiment in its room, a_2 tries to communicate with a_1 to get new observations. First, in order to check its hypothesis, it ask for situations in which the target is on an adjacent block whose level is not low. Searching in its observations, a_1 replies with the set of observations related to s_7 which tends to confirm the hypothesis. a_2 then wants to check if it could generalize its hypothesis by removing $\neg$`RLevel(1,low)`. It asks a_1 for observations concerning situations in which the target is in the next block and this block is low. It gets the results of experiment s_8, which means that it should not generalize its rule in this way.

4.2 Critique

Rather than asking about observations that the other agents might have, an agent could propose it its hypothesis and let it decide which observations would be the more interesting to send back. The queried agent then takes an active role of *critic*. Obviously, the more interesting observations would be those that would defeat the hypothesis. If none can be found, the critic agent just confirms that it thinks this hypothesis is good. Such a process is similar to the *equivalence* queries [Angluin (1987)], with the critic agent playing the role of a limited oracle. Since two agents might have different individual theories, it is possible that a set of observations that an agent considers as a counter-example for some hypothesis is not considered as such by the first agent. In such case, in order to get a better critique, an agent will provide some of its knowledge to argue whether a set of observation defeats its hypothesis or not. Such exchanges allows the agents to learn a bit about each other's individual theories.

To demonstrate this, we consider a third agent a_3, that possess a more complex individual theory $\mathcal{T}_3^I$ (`RDiff(`x`,`y`,`N`,`$s`)` gives the relation difference of level N between relative positions x and y):

`Diff(`L`,`L`,0).` `Diff(low,mid,1).` `Diff(mid,high,1).` `Diff(low,high,2).`

`Diff(`L_1`,`L_2`,0)` $\leftarrow$ `Diff(`L_2`,`L_1`,`$N`)`.

`RDiff(`X`,`Y`,`N`,`$S`)` $\leftarrow$ `RLevel(`X`,`L_1`,`$S`)`, `RLevel(`$Y$`,`$L_2$`,`$S`)`,

`Diff(`L_1`,`L_2`,`$N`)`, $X \leq Y$.

$\neg$`RDiff(`X`,`Y`,`N_2`,`$S`)` $\leftarrow$ `RDiff(`X`,`Y`,`N_1`,`$S`)`, $N_1 \neq N_2$.

Using this theory, a_3 can make hypotheses related to the difference of levels between two blocks. The situations that a_3 has observed are given by figure 5. Its hypothesis is then

$$H_3 = \{h_{3,1}\ \text{:}\texttt{See(tgt,}S\texttt{)} \leftarrow \texttt{RPos(tgt,1,}S\texttt{)},\ \texttt{RDiff(0,1,0,}S\texttt{)},\ \neg\texttt{RDiff(0,1,1,}S\texttt{)},\ \neg\texttt{RDiff(0,1,2,}S\texttt{)}.\}$$

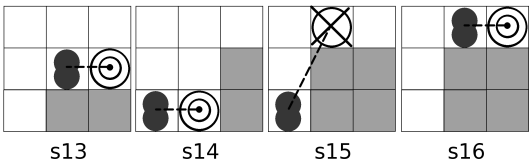

Fig. 5: Initial observations of agent a_3.

Now, a_3 proposes this hypothesis to a_2. a_2 then checks the coherence and completeness of H_3 wrt to its observations and theory. It replies with s_9, which shows the incompleteness of H_3. Using this situation, a_3 applies its theory and revise its hypothesis to get a new one :

$$H_3' = \{h_{3,1}'\ \text{:}\texttt{See(tgt,}S\texttt{)} \leftarrow \texttt{RPos(tgt,1,}S\texttt{)},\ \neg\texttt{RDiff(0,1,2,}S\texttt{)}.\}$$

When it proposes H_3' to a_2, however, a_3 might get again the same situation (or a similar one like s_{10}) since a_3 does not understand `RDiff` and cannot trigger $h_{3,1}'$ to explain e.g. `See(tgt,`s_{10}`)`. a_3 thus gives the following justification using its individual theory :

{ `Diff(mid,high,1).` `RDiff(`X`,`Y`,`N`,`$S`)` $\leftarrow$ `RLevel(`X`,`L_1`,`$S`)`,

`RLevel(`Y`,`L_2`,`$S`)`, `Diff(`$L_1$`,`$L_2$`,`$N`)`, $X \leq Y$.

$\neg$`RDiff(`X`,`Y`,`N_2`,`$S`)` $\leftarrow$ `RDiff(`X`,`Y`,`N_1`,`$S`)`, $N_1 \neq N_2$. }

After that, it will have to repeat the process with s_9 so that a_2 gets to understand `RDiff`, and finally accepts H_3'.

5 Collective knowledge discovery

In the previous sections, agents could use the knowledge of the other agents to help evaluate their individual hypotheses in order to improve their individual theory. We shall now discuss how a common hypothesis could be build, and eventually used to improve the common theory. Three main steps need to be detailed for revising the common theory through a common hypothesis. First, we have to describe what can be a candidate common hypothesis. Such a candidate could be either the hypothesis or individual theory of one of the agents (or a part of it), or some combination of the individual knowledge of all the agents. Then, this candidate must be evaluated and assessed by the group. Individual agent might express opinions about whether they think the hypothesis should be rejected (and why), further evaluated (*weak assent*), or adopted as part of the common theory (*strong assent*). An agent could also propose a revised version of the candidate instead of rejecting it. At last, this different evaluations should be aggregated in some way to make a decision about the hypothesis. If it is rejected, a new candidate common hypothesis might be proposed. If it is revised or if the final decision is a weak assent, then it should be evaluated more thoroughly to reach a more definite conclusion. Meanwhile it could be used by the agents, who might individually experiment to improve their assessment of it. Then, if the global decision is a strong assent, the hypothesis is adopted as part of the common theory. In the following we detail one possible way of building a common hypothesis in the context of rule learning, and then discuss the circumstances in which the common theory might have to be revised.

5.1 Consensual preferred hypothesis

Once each agent has a hypothesis that has been criticized by each of the other agents of the group, the agents of the group begin a collective hypothesis generation phase. First each agent proposes its hypothesis (here a set of rules). If its rules relies on its individual theory to explain manifestation, an agent will attach the relevant information to them. Then, each of them votes for one of the proposed rules, using some individual preference relation (which can be based e.g. on the conciseness of the rule, and its explanatory power with respect to its manifestations). The rule that gets the more votes will be considered as a candidate (partial) hypothesis. Then each agent expresses whether it considers that this rule is fit for adoption (strong assent), or if it has not yet enough support for it (weak assent). Since the candidate hypothesis is chosen among rules that should be coherent with every agent's information, there should be no case of rejection. If all agents of the group agree to adopt it (consensus), then this rule would be become part of the common theory. Otherwise, it will remain as part of the common hypothesis until either it is defeated or all the agent decide that it should be adopted. Meanwhile, each agent that has only given a weak assent will continue to evaluate this rule, through new experiments or communication with agents that defend this partial hypothesis. Eventually, they would either agree to adopt the rule or find some counter-example that they will share with the group. Then the candidate common hypothesis is withdrawn and a new one should be proposed in the same way.

5.2 Revising the common theory

When a agent finds incoherences between its theory and its observations, it cannot generate hypotheses and needs to revise its theory. In most cases, a local revision of its individual theory should be enough. However, it is possible that its observations are incoherent with the common theory, that it cannot revise by itself. In such situation, the agent will first try to confirm its observations by new experiments, and possibly get more observations rising incoherence. After gathering counter-examples against the current common theory, an agent will notify the other agents in the group, and share these observations. Then some collective revision should be done in order to withdraw some part of the common theory and solve the incoherence. A simple vote seems satisfactory, through more complex process could require agents to justify their choices and argue about them. Such mechanisms are however beyond the scope of this paper.

6 Conclusion

We presented in this paper a framework to model knowledge discovery within a group of scientists (represented as agents), and gave some illustration of these processes through an inductive case study. This framework is still preliminary, but we expect it to be generic enough to model other kinds of reasoning. A first step to improve it would be to check this by applying it to different forms of hypothesis generation such as CF-induction [Inoue (2004)] or abductive logic programs. Currently, each agent only considers one individual hypothesis at a time. Such a hypothesis is usually selected from a set of possible relevant ones. However, there is not always a total order available to choose one hypothesis over all the possible ones, and it would thus be worthwhile to investigate pursuing a set of preferred hypotheses rather than a single one. If each agent keeps a set of possible hypothesis, then we could use some mechanism similar to those used in [Sakama and Inoue (2005, 2006)] to generate a set of candidate common hypotheses in collective knowledge discovery. Another interesting lead would be to deal with confidence in the hypotheses. Such confidence could be derived from probabilistic or statistical consideration.

References

Angluin, D. 1987. Learning regular sets from queries and counterexamples. *Inf. Comput.* 75(2):87–106.

Bourgne, G., and Corruble, V. 2008. A framework for knowledge discovery in a society of agents. In *Proc of DS '08*, 172–184. Springer.

Brogi, A.; Lamma, E.; Mancarella, P.; and Mello, P. 1997. A unifying view for logic programming with non-monotonic reasoning. *Theor. Comp. Sc.* 184(1–2):1–59.

Curd, M. 1980. *The logic of discovery : an analysis of three approaches*. Reidel. 173–183.

Guerra-Hernandez, A. 2003. *Apprentissage d'agents rationnels BDI dans un univers Multi-Agents*. Ph.D. Dissertation, Université Paris 13 - Institut Galilée.

Henniche, M. 1997. Apprendre incrémentalement dans l'espace des généralisations maximalement spécifiques d'instances. In *JICAA'97*, 125–135. Roscoff.

Inoue, K. 2004. Induction as consequence finding. *Mach. Learn.* 55(2):109–135.

Kuipers, T. 2001. *Structures in Science – Heuristic Patterns Based on Cognitive Structures*. Kluwer Acad. Pub.

Langley, P. 1978. Bacon.1: A general discovery system. In *Proc. of Canadian AI-1978*, 173–180.

Latour, B. 1979. *Laboratory Life: The Social Construction of Scientific Facts (SAGE Library of Social Research)*. Sage Publications, Inc.

Longino, H. E. 1990. *Science as Social Knowledge: Values and Objectivity in Scientific Inquiry*. Princeton Univ. Press.

Muggleton, S. 1991. Inductive logic programming. *New Gen. Comput.* 8(4):295–318.

Peng, Y., and Reggia, J. A. 1990. *Abductive inference models for diagnostic problem-solving*. New York: Springer.

Popper, K. 1959. *The Logic of Science Discovery*. London: Routledge.

Sakama, C., and Inoue, K. 2005. Combining answer sets of nonmonotonic logic programs. In *CLIMA VI, LNAI 3900*, 320–339. Springer.

Sakama, C., and Inoue, K. 2006. Constructing consensus logic programs. In *LOPSTR, LNAI 4407*, 26–42. Springer.

Thagard, P. 1988. *Computational philosophy of science*. Cambridge: MIT Press.

Wajnberg, C.; Corruble, V.; Ganascia, J.; and Moulines, C. 2004. A structuralist approach towards computational scientific discovery. *Knowledge Discovery* 3245:412–419.

Four insights about mechanisms for abductive processing

John R. Josephson
The Ohio State University & Aetion Technologies LLC
jj@cse.ohio-state.edu

I will describe four apparent insights about mechanisms for abductive processing that I have had in recent years, stimulated by work on multiple-source information fusion designed to support military situation awareness. These four insights can be summarised as follows:

1. Because explanatory relationships are transitive, bottom-up layered abductive processing can "skip levels" if there is a chain of intermediate possibilities.

I previously described a layered-abduction model of perception where conclusions from one layer become data to be explained at the next higher layer (or layers). This makes it appear that a conclusion has to be determined before the inferencing can proceed to higher layers. However, it now seems that all that is required is a path of possibilities that haven't been ruled out. This seems to be very practically important, allowing for quite strong inferencing to take place without requiring all the details to be filled in. Moreover, it makes a lot of sense, accords with many intuitive examples, and shows up in application domains.

2. Inferring intentions appears to be quite easy, at least under certain conditions, which may be very common or typical. When these condition apply, inferring an agent's intention can be done with simple plan representations, and a simple abduction engine.

3. Theory revision can be intelligently controlled by meta-abduction. When the theory-building abduction engine encounters a failed prediction, a meta-abduction engine takes over, and tries to explain the the anomaly as being the result of a mistake in reasoning by the theory-building engine. A plausible explanation, if one can be found, is capable of removing the anomaly. A best explanation, if one can be found, is accepted, and determines a theory repair.

4. Analog spatial representations can be very useful for abductive processing, at least for "perception-like" processing that attempts to infer a world state from sensor reports. Analog representations (and analog processing on these representations) is especially useful for hypothesis generation, for determining explanatory relationships between hypotheses and data, and for testing hypotheses for how well they fit the data. Thus, for some common forms of abductive processing, digital representations are not as effective as analog representations, and "logical" representations are not as effective as "pictorial" representations.

Higher-order Logic Learning and λProgol

Niels Pahlavi and Stephen Muggleton
Department of Computing, Imperial College London,
180 Queen's Gate, London SW7 2BZ, UK
{namdp05,shm}@doc.ic.ac.uk

Abstract

This paper presents Higher-order Logic Learning (HOLL), which consists of generalizing logic-based Machine Learning, and more particularly Inductive Logic Programming (ILP), from the first-order to the higher-order context. The power of expressivity of Higher-order Logic (HOL) is used to improve significantly the learning capacity and efficiency of logic-based Machine Learning – by both allowing to learn new tasks but also to perform better than first-order Machine Learning systems on some already learnable problems. We describe a Higher-order ILP system, called λProgol, adapting the ILP system Progol and based on the HOL formalism λProlog, along with its first implementation and promising results on motivating worked examples. We intend to extend the implementation, tests and evaluation of λProgol further, and to develop a theory of HOLL. We believe the use of the expressivity of Higher-order Logics in logic-based Machine Learning like ILP would allow to extend the scope of such a formalism in order to formalize and automate the evolution of knowledge through knowledge development cycles.

1 Introduction and Motivations

Much of logic-based Machine Learning research is based on First-order Logic (FOL) and Prolog, including Inductive Logic Programming (ILP) [Nienhuys-Cheng and de Wolf, 1997]. Yet, Higher-order Logic (HOL), which allows for quantification over predicates and functions, is intrinsically more expressive than FOL and has been seldom used. According to [Lloyd, 2002], "the logic programming community needs to make greater use of the power of higher-order features and the related type systems. Furthermore, higher-order logic has generally been under-exploited as a knowledge representation language". In [Lloyd, 2002] and [Lloyd, 2003], the use of HOL in Computational Logic, which has been "advocated for at least the last 30 years" is illustrated: functional languages, like Haskell98; Higher-order programming introduced with λProlog [Miller, 1998]; integrated functional logic programming languages like Curry or Escher; or the higher-order logic interactive theorem proving environment "HOL".

As we were interested in discovering learning problems for ILP, we decided to try to adapt ILP within a HOL framework, to develop Higher-order Logic Learning (HOLL). ILP seems to be rather intuitively adaptable to a FOL formalism. It is natural when considering HOLL to both develop a theory and to implement a higher-order ILP system and to test and evaluate it. We decided to choose Higher-order Horn Clauses (HOHC) [Nadathur and Miller, 1990] as a HOL formalism, since it is one of the logical foundations of λProlog. As a ILP system, we chose to adapt Progol [Muggleton, 1995] (or a similar and slightly simpler system called Aleph), which is a popular and efficient implementation.

Higher-order Logic Learning will be tested and assessed on new problems and applications not learnable by ILP (which includes the learning of higher-order predicates), but also on how well it performs on problems already handled by ILP to compare it with existing ILP systems. It would be therefore of interest to look at learning problems not handled well by ILP. One of these is learning tasks involving recursion. According to [Malerba, 2003], "learning first-order recursive theories is a difficult learning task" in a normal ILP setting. However, we can expect a higher-order system to learn better than a first-order system on such problems, because we could use higher-order predicates as background knowledge to learn recursive theories; and it will be sounder, more natural and intuitive, hence probably more efficient than meta-logical features which come from functional languages. More generally, the expressivity of HOL would make it possible to represent mathematical properties like transitivity, symmetry or reflexivity, which would allow to handle equational reasoning and functions within a logic-based framework.

The use of the expressivity of Higher-order Logics in logic-based Machine Learning like ILP would allow to extend the scope of such a formalism and integrate it to the evolution of knowledge, within continuous cycles of knowledge development. Higher-order Logics would be useful at various steps of such cycles. First it would allow ILP to handle abduction intuitively and efficiently. Higher-order Logics could also be used for Theory Revision or the invention of predicates. The different levels of abstraction allowed in Higher-order logics can be useful in Hypothesis Generation but also in Hypothesis Evaluation. And in terms of Knowledge Representation,

Higher-order Logics would allow to use more structured data formalisms.

Outline

Section 2 presents the λProlog language and its logical foundation Higher-order Horn Clauses. Section 3 describes λProgol, which is a Higher-order ILP system and Section 4 its implementation and results obtained so far. Related Works are detailed in Section 5 and Section 6 concludes this paper and presents future works we intend to pursue in this area.

2 λ-Prolog and Higher-order Horn Clauses

λProlog has been developed since 1984 by Dale Miller and Gopalan Nadathur, their main results being described in [Miller and Nadathur, 1986], [Miller, 1987] and [Miller *et al.*, 1991]. It is a higher-order logic programming language handling polymorphic typing, scoping over names and procedures, modular programming, abstract data types, the use of lambda terms as data structures and, more importantly for this paper, higher-order programming illustrated by the predicates $mappred$ and $trans$ in Examples 1 and 2. It is based on the Simple Theory of Types (with several differences), first described in [Church, 1940] which is one of the first type theories and a typed λ-calculus. [Barendregt, 1984] is a recommended reference for λ-calculus.

λProlog is based on HOHC, which are "a generalization of Horn clauses to a higher-order logic" and a "basis for logic programming". According to [Nadathur and Miller, 1990], HOHC can be "characterized as those obtained from first-order goal formulas and definite sentences by supplanting first-order terms with the terms of a typed λ-calculus and by permitting quantification over function and predicate symbols".

In [Nadathur and Miller, 1990], a theorem proving procedure for HOHC is outlined and its soundness is proved; this result is essential for λProgol allowing to implement a higher-order logic Prolog interpreter (see Section 3). This is based on Huet's semi-decision algorithm to search for unification in typed λ-calculus [Huet, 1975].

3 λProgol: a Higher-order ILP System

In this section, λProgol, a higher-order ILP formalism is presented. It is based upon Progol and Mode-Directed Inverse Entailment as defined in [Muggleton, 1995]. However, it generalizes this approach on HOHC and λProlog. Since our implementation is in Prolog, a λProlog interpreter in Prolog is needed. The main differences in the λProgol algorithm, compared to Progol, come from this interpreter and from the fact that it requires background knowledge and examples to be not Horn clauses but λProlog clauses.

Definition 1 ***λProlog Interpreter.***

A λProlog clause is of the form $(HeadAtom \Leftarrow [BodyAtom_1, \ldots, BodyAtom_n])$*, where HeadAtom has to be rigid.*

A λProlog formula is one of the following:

1. *A variable or a constant,*
2. *(X/F), where F is a formula,*
3. *(F1@F2) where F1 and F2 are formulae,*
4. *(sigma F), where F is a formula,*
5. *(pi F), where F is a formula.*

sigma and pi represent respectively the existential and universal quantifiers.

/ represents abstraction and @ represents function application as it is defined in λ-calculus [Barendregt, 1984] and in λProlog.

Atomic formulas must have the form $(h@t_1@\ldots@t_l)$*, where* h *is either a variable or constant and* $t_1 \ldots t_l$ *are terms. If* h *is a constant, it is a rigid atom; if* h *is a variable, it is a flexible atom.*

A list is of the form $cons@el_1@\ldots@el_m@nil$*. nil is the empty list.*

Example 1 shows a λProgol input file to learn the higher-order predicate mappred, defined in [Miller, 1998], which "given a predicate of two arguments and two lists, checks that corresponding elements of these two lists are related by the given predicate", along with its bottom clause and clause learned by λProgol.

Example 1. Mappred.

Mode declarations:
modeh(*,mappred@+pred@+list@+list).
modeb(*,+list=cons@-any@-list).
modeb(*,+pred@+any@+any).
modeb(*,#pred@+pred@+list@+list).
Type declarations:
list(nil). list(cons@X@Y) :- list(Y).
any(X) :- person(X). any(X) :- integer(X).
pred(mappred). pred(age).
person(bob). person(sue). person(ned).
Background Knowledge:
age@bob@23 <= []. age@sue@24 <= [].
age@ned@23 <= [].
mappred@P@nil@nil <= [].
Positive Example:
mappred@age@(cons@ned@nil)@(cons@23@nil)
<= [].
Result: Bottom Clause generated:
mappred@A@B@C<=[B=cons@D@E,
C=cons@F@E,A@D@F,mappred@A@E@E].
Clause to be learned:
mappred@A@B@C<=[B=cons@D@E,
C=cons@F@G,A@D@F,mappred@A@E@G].

The following algorithms (Algorithms 1, 2 and 3) which constitute the λProgol algorithm are very similar to the Progol algorithms. The mode declarations and mode language are identical to Progol (Definitions 20, 21, 22 in [Muggleton, 1995]) except that the mode atoms can be different because λProlog atoms are different from FOL atoms, as it can be seen in Example 1. The construction of $\perp_i$, which is the least general element of the bounded sub-lattice for each example e is described in Algorithm 1. i represents the maximum variable depth determining how many times step 5 is executed; $Recall$ determines how many times the λProlog interpreter is

called for each instantiation of the clause in step 4. The line 5.a.i in the algorithm is specific to λProgol, it is to prevent the call of flexible atoms by the λProlog interpreter. Indeed, the type $pred$ is set to correspond to higher-order predicate, which can be uninstantiated (i.e. still variable) when called by the λProlog interpreter. The call to $pred(u)$ instantiates these variables.

Algorithm 1. Construction of $\bot_i$.

1. Given natural numbers i, λProlog clauses B, λProlog clause e and set of mode declarations M.
2. Let $k = 0$, $hash : Terms \rightarrow N$ be a hash function which uniquely maps terms to natural numbers, $\overline{e}$ be $\overline{a} \wedge b_1 \wedge \ldots \wedge b_n$, $\bot_i = \langle\rangle$ and $InTerms = \emptyset$.
3. If there is no modeh in M such that $a(m) \preceq a$ then return the empty clause $\square$. Otherwise let m be the first modeh declaration in M such that m subsumes a with substitution θ_h. For each v/t in θ_h
   (a) if v corresponds to a $\#type$ then replace v in m by t
   (b) otherwise replace v in m by v_k where $k = hash(t)$ and
   (c) add t to $InTerms$ if v corresponds to $+type$.

   Add m to $\bot_i$.
4. If $k = i$ return $\bot_i$ else $k = k + 1$.
5. For each modeb m in M, let $\{v_1, \ldots, v_n\}$ be the variables of $+type$ in m and $T(m) = T_1 \times \ldots \times T_n$ be a set of n-tuples of terms such that each T_i corresponds to the set of all terms from $InTerms$ of the type associated with v_i in m (t is tested to be of a particular type by calling $type(t)$ with the λProlog interpreter).
   (a) For each $\langle t_1, \ldots, t_n\rangle$ in $T(m)$ and $\theta = \{v_1/t_1, \ldots, v_n/t_n\}$. Repeat recall times:
      i. for every variable u in $m\theta$ of type $pred$, add the call $pred(u)$ to the λProlog interpreter
      ii. if the λProlog interpreter succeeds on goal $m\theta$ with answer substitution θ'then for each v/t in θ and θ' if v corresponds to a $\#type$ then replace v in m by t otherwise replace v in m by v_k where $k = hash(t)$ and add t to $InTerms$ if v corresponds to $-type$. Add $\overline{m}$ to $\bot_i$.
6. Goto step 4.

The search for a single clause in the subsumption lattice is described in Algorithm 2. $best(s)$, $prunes(s)$, $terminated(s)$,$\rho(s)$ are defined like in Progol to find a clause with maximal compression but other types of searches can be used like the search used in Aleph which is simpler.

Algorithm 2. Algorithm for searching $\square \preceq C \preceq \bot_i$.

1. Given λProlog clauses B, λProlog clause e, and $\bot_i$ obtained in Algorithm 1.
2. Let $Open = \{\langle\square, \emptyset, 1\rangle\}$ and $Closed = \emptyset$.
3. Let $s = best(Open)$ and $Open = Open - \{s\}$.
4. Let $Closed = Closed \cup \{s\}$.
5. If $prune(s)$ goto 7.
6. Let $Open = (Open \cup \rho(s)) - Closed$.
7. If $terminated(Closed, Open)$ then return $best(Closed)$.
8. If $Open = \emptyset$ then print "no compression" and return $\langle e, \emptyset, 1\rangle$.
9. Goto 3.

λProgol uses like Progol a simple cover set algorithm described in Algorithm 3. It generalizes the examples one by one and adds the respective generalization to the background knowledge, redundant examples being then removed. The unflattening in this algorithm is defined as in Progol (Definition 43 in [Muggleton, 1995]).

Algorithm 3.Cover set algorithm.

1. Given natural number i, λProlog clauses B, set of mode declarations M, and E is the subset of B corresponding to atoms in modeh declarations in M.
2. If $E = \emptyset$ then return B.
3. Let e be the first example in E.
4. Construct $\bot_i$ for e using Algorithm 1.
5. Construct state s from $\bot_i$ using Algorithm 2.
6. Let C' be the unflattening of $C(s)$.
7. Let $B = B \cup C'$.
8. Let $E' = \{e : e \in E \; and \; B \wedge \overline{e} \vdash_h \square\}$.
9. Let $E = E - E'$.
10. Goto 2.

4 Results and Implementation

An implementation of λProgol has been made and is available. It has been tested successfully for Algorithm 1 on several examples, including Example 1 and other cases of Higher-order Programming as defined in [Miller, 1998], where the following higher-order predicates used in the examples are defined: $mappred$, $foreach$ and $trans$. The examples include cases of learning higher-order predicates as well as cases of learning first-order recursive theories with higher-order predicates as background knowledge.

In this paper, five of these examples and the results of their tests are described including Example 1 already defined and Examples 2,3,4 and 5 detailed below. Table 1 gives a table of runtimes for learning the bottom clause for the examples. The number of mode declarations ($|M|$), the number of clauses in the background knowledge ($|B|$) and the length of the bottom clause ($|BC|$) are also given for each example.

Our first choice of implementation was based on λProlog but revealed to be too inconvenient and inefficient to use; instead the current implementation is in Prolog, which is more convenient and more efficient; a requirement is the use of a

λProlog interpreter, which was implemented using a depth-first approach.

Example 2 details a practical example, already introduced in [de Raedt and Lavrac, 1996] and used in [Malerba, 2003] showing the advantage of using HOL background knowledge in a simple learning problem involving recursion. It consists of learning the predicate ancestor given the predicate parent, and the higher-order predicate $trans$ defined in [Miller, 1998], which "given a predicate of two arguments, constructs its transitive closure". The predicates married and brother are present to prove that the system makes a genuine choice expressing ancestor with parent rather than with married or brother.

Example 2. Ancestor.

Mode declarations:
modeh(*,ancestor@+person@+person).
modeb(*,#pred@#pred@+person@+person).
Type declarations:
pred(trans). pred(parent).
pred(married). pred(brother).
person(john). person(jim). person(jane).
person(bob). person(james). person(bill).
Background Knowledge:
trans@R@X@Y <= [R@X@Y].
trans@R@X@Z <= [R@X@Y,trans@R@Y@Z].
parent@john@jim <= []. parent@john@jane <= [].
parent@jim@bob <= [].
parent@bob@james <= []. married@john@jane <= [].
brother@bob@bill <= [].
Positive Examples:
ancestor@john@bob <= []. ancestor@jim@james <= [].

In this example, both the bottom clause and the clause to be learned are:

ancestor@X@Y ⇐ [trans@parent@X@Y].

In order to obtain a first-order recursive theory of ancestor, we need the unfolding of trans which gives.

ancestor@X@Y ⇐ [parent@X@Y].
ancestor@X@Y ⇐
[parent@X@Z, trans@parent@Z@Y].

In the last clause, in order to replace trans@parent@Z@Y by ancestor@Z@Y, we need the closed world assumption, which is the presumption that what is not currently known to be true is false. Under the closed world assumption, we have trans@parent@X@Y ⇐ [ancestor@X@Y]. Therefore we have.

ancestor@X@Y ⇐ [parent@X@Y].
ancestor@X@Y ⇐ [parent@X@Z, ancestor@Z@Y].

Which, in Prolog notations, gives the following first-order theory.

ancestor(X,Y) :- parent(X,Y).
ancestor(X,Y) :- parent(X,Z), ancestor(Z,Y).

This shows how natural and efficient it is to use HOL as background knowledge to handle the learning of a first-order recursive theory.

Example 3 is similar to Example 2. It also uses the higher-order predicate $trans$ as background knowledge in order to learn the predicate less_than, which given two integers determines if the first one is smaller than the second one. The predicate successor, which given an integer N returns $N+1$, is to be used with $trans$. The predicates add_10 (which given an integer N returns $N+10$) and divide_4 (which given an integer N returns $N/4$) are present to prove that the system makes a genuine choice expressing less_than with successor rather than with add_10 or divide_4.

Example 3. Less_than.

Mode declarations:
modeh(*,less_than@+integer@+integer).
modeb(*,#pred@#pred@+integer@+integer).
Type declarations:
pred(trans). pred(successor).
pred(add_10). pred(divide_4).
Background Knowledge:
trans@R@X@Y <= [R@X@Y].
trans@R@X@Z <= [R@X@Y,trans@R@Y@Z].
successor@0@1 <= []. successor@1@2 <= [].
successor@2@3 <= []. successor@3@4 <= [].
successor@4@5 <= []. successor@5@6 <= [].
successor@6@7 <= []. successor@9@10 <= [].
add_10@3@13 <= []. add_10@4@14 <= [].
divide_4@4@1 <= []. divide_4@8@2 <= [].
Positive Example:
less_than@2@7 <= [].
Result: Bottom Clause generated:
less_than@X@Y <= [trans@successor@X@Y].
Clause to be learned:
less_than@X@Y <= [trans@successor@X@Y].

Similarly to what has been done in Example 2, by unfolding, we have the following.

less_than@X@Y ⇐ [successor@X@Y].
less_than@X@Y ⇐
[successor@X@Z, trans@successor@Z@Y].

Which gives, by closed world assumption, the following first-order recursive theory (with Prolog notations).

less_than(X,Y) :- successor(X,Y).
less_than(X,Y) :- successor(X,Z), less_than(Z,Y).

Example 4 is similar to Example 1 as it learns the higher-order predicate $foreach$, defined in [Miller, 1998], which "given a predicate of one argument and a list, checks that every element of that list satisfies that predicate".

Example 4. Foreach.

Mode declarations:
modeh(*,foreach@+pred@+list).
modeb(*,(+list)=(cons@-any@-list)).
modeb(*,+pred@+any).
modeb(*,#pred@+pred@+list).
Type declarations:
list(nil). list(cons@X@Y) :- list(Y).
any(X) :- person(X). any(X) :- integer(X).
pred(foreach). pred(age). pred(male). pred(female).
person(bob). person(sue).
person(ned). person(jane). person(bill).
Background Knowledge:

Table 1: Bottom Clause Results

Example	Predicate	$\|M\|$	$\|B\|$	$\|BC\|$	Time (sec)
1	mappred	4	13	5	0.004
2	ancestor	2	18	2	0.016
3	less_than	2	18	2	0.024
4	foreach	4	18	4	0.008
5	trans	3	21	3	0.004

male@bob <= []. male@ned <= [].
male@bill <= [].
female@sue <= []. female@jane <= [].
age@sue@24 <= []. age@ned@23 <= [].
foreach@P@nil <= [].
Positive Example:
foreach@male@
(cons@ned@(cons@bob@(cons@bill@nil))) <= [].
Result: Bottom Clause generated:
foreach@P@X
<= [X=(cons@U@L),P@U,foreach@P@L].
Clause to be learned:
foreach@P@X
<= [X=(cons@U@L),P@U,foreach@P@L].

In Example 5, the higher-order predicate $trans$ is not used as background knowledge as in Examples 2 and 3, but is the predicate to be learned.

Example 5. Trans.
Mode declarations:
modeh(*,trans@+pred@+any@+any).
modeb(*,+pred@+any@-any).
modeb(*,#pred@+pred@+any@+any).
Type declarations:
any(X) :- person(X). any(X) :- integer(X).
pred(trans). pred(parent). pred(age). pred(married).
person(bob). person(sue). person(ned).
person(jane). person(bill). person(jim).
Background Knowledge:
parent@bob@sue <= []. parent@bob@ned <= [].
parent@ned@jane <= []. parent@ned@bill <= [].
parent@bill@jim <= [].
married@ned@sue <= []. married@bill@jane <= [].
age@bob@23 <= [].
trans@R@X@Y <= [R@X@Y].
Positive Example:
trans@parent@bob@jim <= [].
Result: Bottom Clause generated:
trans@R@X@Z <= [R@X@Y,trans@R@X@Y].
Clause to be learned:
trans@R@X@Z <= [R@X@Y,trans@R@Y@Z].

---

5 Related Works

There have been attempts to use HOL for logic-based Machine Learning such as by Harao starting in [Harao, 1990], Feng and Muggleton [Feng and Muggleton, 1992] and Furukawa and Goebel [Furukawa *et al.*, 1996]. They provide different higher-order extensions of least general generalization in order to handle higher-order terms in a normal ILP setting, whereas we use λProlog, a HOL framework, as a logical foundation to extend first-order ILP to a higher-order context.

John Lloyd, in [Bowers *et al.*, 2001], [Lloyd, 2002] and [Lloyd, 2003], deals with related issues. He supports the idea of generalizing ILP within a HOL framework, and introduces examples where HOL has been used in Computational Logic (Functional languages, like Haskell98 [Peyton Jones and Hughes,], Higher-order programming with λProlog [Nadathur and Miller, 1998], integrated functional logic programming languages like Curry [Hanus,] or Escher [Lloyd, 1999], higher-order logic theorem provers like HOL [Gordon and Melham, 1993]). A main similar work is [Lloyd, 2003] by Lloyd, where he develops higher-order machine learning as well. In this "worthy and fascinating exercise", it uses a typed higher-order logic, but, although similar, "a different sublogic is used for λProlog programs than the equational theories proposed" in [Lloyd, 2003]. It details a learning system, called $ALKEMY$. As Flach's review of the book notices in [Flach, 2003], "the use of higher-order logic is elegant but not crucial, as it does not offer additional power over the first-order refinement operators or hypothesis grammars" used in ILP. A main difference is that Lloyd's approach is not based on Logic Programming and therefore on ILP. According to[Flach, 2003], "it is almost a rational reconstruction of what ILP could have been, had it used Escher-style higher-order logic rather than Prolog"; whereas we intend, through the use of higher-order Horn clauses to keep the Horn clauses foundations of LP and ILP and to extend it.

6 Conclusion and Future Works

This paper presents HOLL which consists of generalizing logic-based Machine Learning, from the first-order to the higher-order context.

We intend to provide theoretical results for HOLL. ILP theory seems to be rather intuitively adaptable within a higher-order logic framework. For λProgol, we will have to prove that higher-order inverse entailment is possible and to generalize correctness and complexity results for the Progol Bottom Clause and Search algorithms. In [Wolfram, 1994], a model-theoretic semantics for HOHC is provided. We will also extend the implementation of λProgol and test and evaluate it further. We also aim to compare λProgol with already existing ILP systems, for example by considering learning tasks where it could perform better than Progol. Then, we intend to investigate tasks and discoveries not learnable by first-order ILP. It could be of interest to look at recursion, of course, but also at higher-order logic theorem provers, or integrated functional logic programming languages.

In order to investigate how Higher-order Logics could allow to extend logic-based Machine Learning like ILP to support the evolution of knowledge, we intend to look at its use for several steps of knowledge development cycles. First it allows to use more structured data formalisms. Abduction seems to be handled more naturally and efficiently with Higher-order Logics. We also want to look at how it could be used for Theory Revision and predicate invention. Then,

we aim at using the different levels of abstraction allowed in Higher-order logics to improve Hypothesis Generation but also to treat Hypothesis Evaluation. This could be applied to areas where ILP has already been used with success like in Bioinformatics or Synthetic Biology. For example, in metabolic networks, HOLL could be used for the detection of Krebs cycles, to learn recursive cycles but also to identify symmetry in highly symmetric protein structures. We would also like to adapt ASE-Progol, which is an Active Learning ILP system which has been used to discover the function of genes and was incorporated into The Robot Scientist [King *et al.*, 2004]. HOLL could also be applied in other areas where knowledge development cycles have been formalized or automated like Decision Support or Requirements Engineering.

References

[Barendregt, 1984] Hendrik P. Barendregt. *The lambda calculus its syntax and semantics*. North-Holland, 1984.

[Bowers *et al.*, 2001] Anthony F. Bowers, Christophe G. Giraud-Carrier, and John W. Lloyd. A Knowledge Representation Framework for Inductive Learning. Available at http://users.rsise.anu.edu.au/ jwl/, 2001.

[Church, 1940] Alonzo Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5:56–68, 1940.

[de Raedt and Lavrac, 1996] Luc de Raedt and Nada Lavrac. Multiple predicate learning in two inductive logic programming settings. *Logic Journal of the IGPL*, 4(2):227–254, 1996.

[Feng and Muggleton, 1992] Cao Feng and Stephen H. Muggleton. Towards inductive generalisation in higher order logic. In D. Sleeman and P. Edwards, editors, *Proceedings of the Ninth International Workshop on Machine Learning*, pages 154–162, San Mateo, CA, 1992. Morgan Kaufmann.

[Flach, 2003] Peter Flach. Book review: Logic for learning, 2003.

[Furukawa *et al.*, 1996] Koichi Furukawa, Mutumi Imai, and Randy Goebel. Hyper least general generalization and its application to higher-order concept learning. Technical report, Keio University, 1996.

[Gordon and Melham, 1993] Michael J.C. Gordon and Thomas F. Melham. *Introduction to HOL: A Theorem Proving Environment for Higher Order Logic*. Cambridge University Press, 1993.

[Hanus,] Michael Hanus. Curry: An integrated functional logic language. Available at http://www.informatik.uni-kiel.de/ curry.

[Harao, 1990] Masateru Harao. Analogical reasoning based on higher-order unification. In *ALT*, pages 151–163, 1990.

[Huet, 1975] Gerard P. Huet. A unification algorithm for typed λcalculus. *Theoretical Computer Science*, 1:27–57, 1975.

[King *et al.*, 2004] R.D. King, K.E. Whelan, F.M. Jones, P.K.G. Reiser, C.H. Bryant, S.H. Muggleton, D.B. Kell, and S.G. Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427:247–252, 2004.

[Lloyd, 1999] John W. Lloyd. Programming in an integrated functional and logic language. *Journal of Functional and Logic Programming*, 1999(3), 1999.

[Lloyd, 2002] John W. Lloyd. Knowledge Representation, Computation, and Learning in Higher-order Logic. Available at http://users.rsise.anu.edu.au/ jwl/, 2002.

[Lloyd, 2003] John W. Lloyd. *Logic for Learning*. Springer, Berlin, 2003.

[Malerba, 2003] Donato Malerba. Learning recursive theories in the normal ilp setting. *Fundamenta Informaticae*, 57(1):39–77, 2003.

[Miller and Nadathur, 1986] Dale Miller and Gopalan Nadathur. Higher-order logic programming. In Ehud Shapiro, editor, *Proceedings of the Third International Logic Programming Conference*, pages 448–462, London, 1986.

[Miller *et al.*, 1991] Dale Miller, Gopalan Nadathur, Frank Pfenning, and Andre Scedrov. Uniform proofs as a foundation for logic programming. In *Annals of Pure and Applied Logic*, pages 125–157, 1991.

[Miller, 1987] Dale Miller. Hereditary harrop formulas and logic programming. In *Proceedings of the VIII International Congress of Logic, Methodology, and Philosophy of Science*, pages 153–156, Moscow, 1987.

[Miller, 1998] Dale Miller. *λProlog: An Introduction to the Language and its Logic*. Draft Version, 1998.

[Muggleton, 1995] Stephen H. Muggleton. Inverse entailment and Progol. *New Generation Computing*, 13:245–286, 1995.

[Nadathur and Miller, 1990] Gopalan Nadathur and Dale Miller. Higher-order horn clauses. *Journal of the ACM*, 4:777–814, 1990.

[Nadathur and Miller, 1998] Gopalan Nadathur and Dale Miller. Higher-order logic programming. In and J.A. Robinson D.M. Gabbay, C.J. Hogger, editor, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 5, pages 499–590. Oxford University Press, 1998.

[Nienhuys-Cheng and de Wolf, 1997] Shan-Hwei Nienhuys-Cheng and Ronald de Wolf. *Foundations of Inductive Logic Programming*. Springer-Verlag, Berlin, 1997. LNAI 1228.

[Peyton Jones and Hughes,] Simon L. Peyton Jones and John Hughes. Haskell98: A non-strict purely functional language. Available at http://haskell.org/.

[Wolfram, 1994] David A. Wolfram. A semantics for lambda-prolog. *Theoretical Computer Science*, 136(1):277–289, 1994.

Multimodal Abduction in Knowledge Development

Lorenzo Magnani
Department of Philosophy and Computational Philosophy Laboratory
University of Pavia, Pavia, Italy
lmagnani@unipv.it

Abstract

From the perspective of distributed cognition I will stress how abduction is essentially *multimodal*, in that both data and hypotheses can have a full range of verbal and sensory representations, involving words, sights, images, smells, etc., but also kinesthetic – related to the ability to sense the position and location and orientation and movement of the body and its parts – and motor experiences and other feelings such as pain, and thus all sensory modalities. The presence of kinesthetic and motor aspects demonstrates that abductive reasoning can be manipulative. We can also see, in this regard, how implicit factors take part in the abductive procedure, which consequently acquires the character of a kind of "thinking through doing". This paper further describes 1) the fact that hypotheses in science can be built through different cognitive mediators and so they can also model the same cognitive aspect in different ways; how they can be carriers/producers of knowledge in a *multimodal* way; 2) the problem of the possible *non-explanatory* and *instrumental* nature of abductive reasoning and the analysis of the consequences for induction; 3) the role of *manipulative* abduction in building new evidence/experiments and how they trigger smart inductive inferences.

1 Multimodal Abduction

1.1 Multimodal Abduction

Logical models and computational automation of abductive and inductive cognition have certainly tended to neglect "[. . .] important philosophical concerns relating to causality and creativity" [Ray, 2007]. Similarly, philosophical and cognitive research certainly lack the rigor of logical models and the applicative chances of the computational ones. However, I think that the dialogue can still be fruitful. To this aim I propose in this paper some considerations that deal with the problem of the *multimodality* of data and hypotheses and the distributed and dynamic character of *evidence/experiment* in abductive reasoning.

Peirce considers inferential any cognitive activity whatever, not only conscious abstract thought; he also includes perceptual knowledge and subconscious cognitive activity. For instance in subconscious mental activities visual representations play an immediate role. Many commentators criticized this Peircean ambiguity in treating abduction at the same time as inference and perception. It is important to clarify this problem, because perception and imagery are kinds of that model-based cognition which I am exploiting to explain abduction: I contend that we can render consistent the two views [Magnani, 2006b], beyond Peirce, but perhaps also within the Peircean texts, partially taking advantage of the concept of *multimodal abduction*, which depicts hybrid aspects of abductive reasoning. [Thagard, 2007] observes, that abductive inference can be visual as well as verbal, and consequently acknowledges the sentential, model-based, and manipulative nature of abduction I stressed in my previous research on the subject. For example, both data and hypotheses can be visually represented:

> For example, when I see a scratch along the side of my car, I can generate the mental image of a grocery cart sliding into the car and producing the scratch. In this case both the target (the scratch) and the hypothesis (the collision) are visually represented. [. . .] It is an interesting question whether hypotheses can be represented using all sensory modalities. For vision the answer is obvious, as images and diagrams can clearly be used to represent events and structures that have causal effects.

Indeed hypotheses can be also represented using other sensory modalities:

> I may recoil because something I touch feels slimy, or jump because of a loud noise, or frown because of a rotten smell, or gag because something tastes too salty. Hence in explaining my own behavior my mental image of the full range of examples of sensory experiences may have causal significance. Applying such explanations of the behavior of others requires projecting onto them the possession of sensory experiences that I think are like the ones that I have in similar situations. [. . .] Empathy works the same way, when I explain people's behavior in a particular situation by inferring that they are having the same kind of emotional experience that I have in similar situations [Thagard, 2007].

1.2 Ignorance Preserving Reasoning and Non-Explanatory Abduction

[Gabbay and Woods, 2005] contend that abduction presents an *ignorance preserving* (but also *ignorance mitigating*) character. Abductive reasoning is a *response* to an ignorance-problem: "One has an ignorance-problem when one has a cognitive target that cannot be attained on the basis of what one currently knows. Ignorance problems trigger one or other of three responses. In the one case, one overcomes one's ignorance by attaining some additional knowledge. In the second instance, one yields to one's ignorance (at least for the time being). In the third instance, one abduces" [Woods, 2010, chapter five].

In this perspective the general form of an abductive inference can be rendered as follows, putting T for the agent's target at a time, K for his (or its) knowledge-base at that time, K^* for an accessible successor-base of K,[1] R as the attainment relation for T, H as the agent's hypothesis; $K(H)$ as K's adaptation of H, that is the revision of K upon the addition of H and R^{pres} as the relation of presumptive attainment relative to T. The general structure can be illustrated as follows:

1. T!	[setting of T as target]
2. $\neg(R(K,T))$	[fact]
3. $\neg(R(K^*,T))$	[fact]
4. $H \notin K$	[fact]
5. $H \notin K^*$	[fact]
6. $\neg R(H,T)$	[fact]
7. $R^{pres}(K(H),T)$	[fact]
8. H meets further conditions $S_1, ...S_n$	[fact]
9. Therefore, $C(H)$	[sub-conclusion, 1-7]
10. Therefore, H^c	[conclusion, 1-8] (cf. [Woods, 2009] and [Gabbay and Woods, 2005, pp. 47–48]).

[Note: Basically, line 8. indicates that H has no more plausible or relevant rival constituting a greater degree of subjunctive attainment. $C(H)$ is read "It is justified (or reasonable) to conjecture that H" and H^c its activation."

In sum, T cannot be attained on the basis of K. Neither can it be attained on the basis of any successor K^* of K that the agent knows then and there how to construct. H is not in K: H is a hypothesis that when reconciled to K produces an updated $K(H)$. H is such that if it were true, then $K(H)$ would attain T. The problem is that H is only hypothesized, so that the truth is not assured. Accordingly Gabbay and Woods contend that $K(H)$ presumptively attains T. That is, having hypothesized that H, the agent just "presumes" that his target is now attained. Given the fact that presumptive attainment is not attainment, the agent's abduction must be considered as preserving the ignorance that already gave rise to her (or its, in the case for example of a machine) initial ignorance-problem. Accordingly, abduction does not have to be considered the "solution" of an ignorance problem, but rather a response to it, in which the agent reaches presumptive attainment rather than actual attainment. $C(H)$ expresses the conclusion that it follows from the facts of the schema that H is a worthy object of conjecture. In order to solve a problem it is not necessary that an agent actually conjectures a hypothesis, but it is necessary that she states that the hypothesis is worthy of conjecture.

The superscript in H^c is a label. It reminds us, Gabbay and Woods say, that H "[...] has been let loose on sufferance" (cf. [Woods, 2010]). Through abduction the basic ignorance – that does not have to be considered a total "ignorance" – is neither solved nor left intact: it is an ignorance-preserving accommodation of the problem at hand. As I have already stressed, even though in a defeasible way, further action can be triggered either to find further abductions or to solve the ignorance problem, possibly leading to what it is called in the literature the inference to the best explanation. It is clear that in this framework the inference to the best explanation – if considered as a truth conferring achievement – cannot be a case of abduction, because abductive inference is constitutively ignorance preserving. In this perspective the inference to the best explanation also involves the role of *induction*. Of course it can be said that the requests of originary thinking are related to the depth of the abducer's ignorance.

1.3 Non-Explanatory and Instrumental Abduction

[Gabbay and Woods, 2005] also contend – and I agree with them – that abduction *is not intrinsically explanationist*, like for example its description in terms of inference to the best explanation would suggest. Not only that, abduction can also be merely *instrumental*. This conviction constitutes the main reason for proposing the GW-schema (Gabbay-Woods), which offers a representation of abductive cases not captured by that of the AKM (Aliseda-Kuipers-Magnani-Meheus), restricted to the explanatory cases. In my previous book on abduction [Magnani, 2001] I made some examples of abductive reasoning that basically are non-explanatory and/or instrumentalist without clearly acknowledging it. Gabbay and Woods's distinction between explanatory, non-explanatory and instrumental abduction is orthogonal to mine in terms of the theoretical and manipulative (including the subclasses of sentential and model-based) and further allows us to explore fundamental features of abductive cognition. Hence, if we maintain that E explains E' *only if* the first implies the second, certainly the reverse does not hold. This means that various cases of abduction are consequentialist but not explanationist. Other cases are neither consequentialist nor explanationist.

Non-explanatory modes of abduction are clearly exploited in the "reverse mathematics" pioneered by Harvey Friedman and his colleagues, e.g., [Friedman and Simpson, 2000], where propositions can be taken as axioms because they support the axiomatic proofs of target theorems. The target of reverse mathematics is to answer this fundamental question: What are the appropriate axioms for mathematics? The prob-

[1] "K^* is an accessible successor of K to the degree that an agent has the know-how to construct it in a timely way; i.e., in ways that are of service in the attainment of targets linked to K" [Woods, 2010, chapter five, footnote 20].

lem is to discover which are the appropriate axioms for proving particular theorems in central mathematical areas such as algebra, analysis, and topology [Simpson, 1999]. The idea of reverse mathematics originates with Russell's notion of the regressive method in mathematics [Russell, 1973], and is also present in some remarks of Gödel.[2] [Gabbay and Woods, 2005, p. 128] conclude, following Russell, that regressive abduction is both instrumental and non-explanatory.

Furthermore, often in physics the target is the discovery of physical dependencies which [Gabbay and Woods, 2005, pp. 122–123] consider explanatorily undetermined. In this case abduction can exhibit an *instrumental* aspect. I have contended in [Magnani, Forthcoming, chapter two] that this character is sometimes related to the conventional nature of the involved hypotheses. Moreover, also in many AI approaches based on logic programming and belief revision explanationism tends to disappear and abduction is mainly considered as proof theoretic and algorithmic: "On this view, an H is legitimately dischargeable to the extent to which it makes it possible to prove (or compute) from a database a formula not provable (or computable) from it as it is currently structured. This makes it natural to think of AI-abduction in terms of belief-revision theory, of which belief-revision according to explanatory force is only a part" [Gabbay and Woods, 2005, p. 88]. However, the explanatory character is subsumed in these AI approaches as a philosophical conception.

In sum, Gabbay and Woods maintain we can face a kind of abduction that, basically,

- is not plausibilist

at least in the sense we consider it on the explanatory framework.

They say: "It is not uncommon for philosophers to speak of the contribution made by the hypothesis of action-at-a-distance as one of explaining otherwise unexplainable observational data. [...] Like numerous instances of D-N explanation, Newtonian explanations need convey no elucidation of their explicanda. They need confer no jot of further intelligibility to them. The action at-a-distance equation serves Newton's theory in a wholly instrumental sense. It allows the gravitational theory to predict observations that it would not otherwise be able to predict" [Gabbay and Woods, 2005, pp. 118-119]. In this case Newtonian explanations are seen as epistemically agnostic conjectures, that is they lack epistemic virtues. These abductions are secured by instrumental considerations and accepted because doing so enables one's target to be hit. They cannot be discharged because of their possible implausibility, for example on the basis of empirical disconfirmation.

2 Abduction: Multimodal Hypotheses and Heuristics

2.1 Multimodal Hypotheses through Different Cognitive Mediators

Also in scientific reasoning multimodal abduction is at work and different hypotheses can be built through different cognitive mediators so that they can model the same aspect in different ways. [Flach *et al.*, 2006, p. 21], dealing with the problem of logical modeling and automated computation of the dyad abduction/induction, contend that "Modelling a scientific domain is a continuous process of observing and understanding phenomena according to some currently available model, and using this understanding to improve the original domain model. In this process one starts with a relatively simple model which gets further improved and expanded as the process is iterated. At any given stage of its development, the current model is very likely to be *incomplete* ".

Let us start considering the abductive side of this process. The abductive construction of hypotheses and theories is certainly driven by experimental observation to improve, refine, and complete the model. However, as I have anticipated above, already at the level of abductively making hypotheses, the relationship of the cognitive agent with the "experimental" observation is first of all occurring in a continuous interplay where the cognitive process is that kind of "thinking through doing", which I have described in terms of *manipulative abduction*. It involves the repeated production of new evidence, external to the cognitive subject, which provides new fundamental data to further fuel the reasoning process.

An example which deals with a simple and modestly creative way of guessing general mathematical hypotheses in actual humans can be of help. The research has been built in the pedagogical framework concerning the need of increasing knowledge on the ways in which learners in the area of school algebra develop their abilities. [Rivera and Rossi Becker, 2007] illustrate the case of different subjects [elementary majors] who are given sequences of figural and numerical cues which taken together comprise classes of abstract objects such as even and odd numbers and related diagrams: "The accompanying questions oftentimes involve a twin calculation-encapsulation process, that is, from determining specific output values to abductively forming a viable general expression which can generate any element in the class. [...] Thus, the central purpose of generalizing tasks at the elementary level is to help learners develop an ability to generalize from particular instances and be able to express the generalization in ways that are both meaningful to them and valid from the standpoint of institutional practice" (p. 141). The task to be performed is very useful to illustrate a case of multimodal abduction at work (limited to the figural and numerical case), an abductive procedure which is looking for inductively produced generalized hypotheses as closed formulas.

The observational examples O (symbols and diagrams) presented to the subjects are *limited* and *incomplete*, like it is occurring in the case of other usual scientists' creative tasks. The subjects possess some knowledge about mathematics (we would say some theories T, in logical terms) more or less accurate, describing the model of the domain that is under investigation, and the "multimodal" information contained in the examples have to be "multimodally" managed through those available theories. The subjects have to produce new hypothetical knowledge, H, which extends their own preexistent theories such that the observations can be first of all deduced by the new abductively enriched theories. It is in the

[2] For more details about this, see [Irvine, 1989], who also compares Russell's regressive method to Peirce's abduction.

abductive process that new "experiments" providing new data are often repeatedly realized.

To make a simple example

> For instance, the sequence {2, 4, 6, 8, ...} is a class and the closed formula $2n$ is one way of describing the overall structure of each number in the sequence. Also, it is perceptually apparent that evenness is one characteristic that is common to the numbers in the sequence. The Fibonacci sequence {1, 1, 2, 3, 5, 8, 13, ...} is another example of a class that can be generally described by the recursive relation $a_{n+1} = a_{n-1} + a_n$ (where $a_1 = a_2 = 1$). Its closed formula is $(1/\sqrt{5})[((1 + \sqrt{5})/2)n - ((1 - \sqrt{5})/2)^n]$ with the additional assumption that the numerical cues would obey the stated recursive form. The arithmetic class {3, 8, 13, 18, ...} can be generalized by the direct expression $5n - 2$ under the condition that the class is an increasing sequence and where $n \geqslant 1$. Resemblance encompasses implicit (deep) and explicit (surface) properties that cues within a class have in common, and these properties are not inherently a priori (p. 142).

From the cognitive-psychological point of view we can say the subjects abduce new properties of the objects at play by projecting them onto individual elements of the class being tested: for instance employing *numerical heuristics* (for instance the "finite difference method") "[...] in order to surface properties that are or are not directly knowable due to the incompleteness of the cues presented to the learners" (*ibid.*)[3] It is typical of abduction to be able to increase knowledge (it is an ampliative reasoning) about a class of objects even if they are presented in a very incomplete way: in abductive reasoning ignorance is preserved, but weakened, as illustrated above.

2.2 Externalization as an Abductive Experimental Step

We have said that evidence presented to the subjects consists of limited and incomplete *multimodal* cues, such as the ones illustrated in the example of Figure 1, which is very useful to depict the multimodal character of abduction. The subjects abductively work on the available cues: some of them adopt a merely *numerical modality* (and related inferential routines), other a *figural* one (also a hybrid combinations of both is sometimes exploited). To perform the cognitive task they are required to draw or compute two additional cases and they do this abductively by exploiting either *figural* or *numerical hypotheses* thanks to fact they were able to perceive relations possibly leading to the abductive generalization in very different ways. They do this – and this is the main point I want to stress – with the help of a suitably built *new evidence*, that

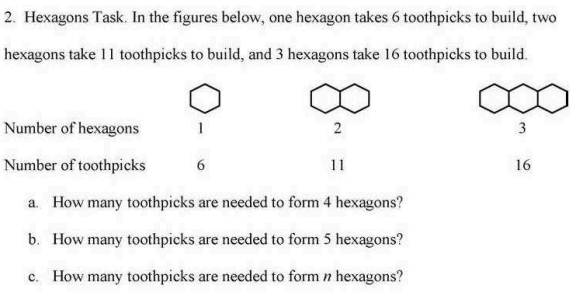

Figure 1: Hexagons Task (in [Rivera and Rossi Becker, 2007]).

is by reframing the problem in a new external mediator (in our example a sheet of paper is exploited - an example is given in Figure 2 performed thanks to their available inner mathematical knowledge and inferential procedures, other more or less simple rational reasoning devices such as guess and check, trial and error, and various capacity to draw *new* evidence and to execute visual comparison of forms, based on their stored perceptual knowledge gained through the observation of the initial cues, etc. The mediator plays the role, so to say, of a further appropriately built "experimental data", which offer chances for further knowledge.

Various intertwined *representational agents* are at play, in terms of 1) inner knowledge of humans in terms of data, theories, and inferential procedures, 2) knowledge embedded in the evidence provided at the start, 3) evidence (data) that can be picked up in the subsequent rebuilt new "experimental" evidence, to which other aspects of inner available mathematical knowledge and inferential procedures can be applied.[4] For example some subjects that used the *numerical modality* perceived relationships among the elements in the same class rather differently than the ones that used the figural one, and some of them, because of the inefficiency of the adopted abductive process, were not able to reach the final correct generalization; others instead reached a bad one.[5] Most of the 14 subjects that used the *figural modality* (performing what I have called *model-based abduction* [Magnani, 2001]) easily reached the solution of the problem at hand. They mainly based the performance on their cognitive inner capacity to draw *new* evidence on their stored perceptual knowledge gained through the observation of the initial cues.

In summary, what happens in these highly *unstable* abductive subprocesses (both from the numerical and figural perspective) is related to what I call *manipulative abduction* [Magnani, 2001]: the new evidence provided is *practical* and *situational* – that is just *ignorance mitigating* – and in this performance the subjects usually do not benefit of an explicit conceptual explanation (I have said that it is a kind of "thinking through doing"). As I have already illustrated above the

[3] Subjects can mobilize different inferential routines such as, to make an example, *guess and check* and *trial and error*, as Lakatos wonderfully analyzed in his famous book about mathematical reasoning and problem solving [Lakatos, 1976]. AI computational programs that took advantage of this Lakatosian perspective are illustrated in [Pease *et al.*, 2005].

[4] Details are illustrated in [Rivera and Rossi Becker, 2007].

[5] It is amazing to see that in the solution of this simple task, some subjects engaged complicated abductive processes without being able to reach the correct hypothesis and to trigger the subsequent induction. One subject did not guess any abductive hypothesis.

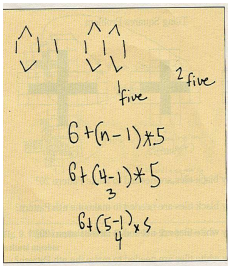

Figure 2: Written solution of the hexagons task (in [Rivera and Rossi Becker, 2007]).

results are produced through heuristics which resort to automatized well-known (and available to the subjects) mathematical procedures but also to general cognitive modalities of depicting symbols or drawings able to the aim of finding features that can possibly be further picked up and stored in the internal memory.[6] These subprocesses are constitutively cyclic: 1) they can suggest new abductive steps where new evidence is built to make available further *situated* abductive generalizations; each of these abductive *situated* generalizations can in turn generate 2) further *universal* inductive generalizations possibly to be withdrawn because of disconfirmation; in this last case a 3) further cyclic abductive-inductive process can restart. The "specificity" of abductive hypotheses is related to their ignorance-preserving character; the "generality" of inductive hypotheses is related to their truth-conferring/probability-enhancing character, at the same time endowed with an evaluative function.

2.3 Non-Explanatory Abduction and Induction

In our example, even if, so to say, some abductive steps produce new generated evidence able to transform knowledge from its tacit (in the first cues) to its explicit form, and consequently human agents are able to generate and justify them in an *explanatory* perspective, the final abduced provisional hypothesis $5n+1$ *does not* really *explain* the data, but it is a fruit of that non-explanatory abduction I have illustrated above. The inductive further step strengthens this aspect, as we will see in the following subsection.

Such explanations – made possible by the construction of new "experimental devices" (through drawings, calculations, sketches), such the one of Figure 2– generate knowledge that is always *specific* to the particular scene of the world concerning the observations explained and the given *multimodal* knowledge and routines available to the human agents. However, they allow them to predict further observable information. Building these new observations/experimental devices through manipulative abduction is central to make possible induction able to generate general knowledge, new and not reachable through abduction. In these perspective they become smart new empirical *samples* able to trigger interesting inductions.

What is the subsequent role of induction in the example illustrated in the previous section? In performing the abductive task to the general form the subjects referred to the fact they immediately saw a relationship among the drawn cues in terms of relational similarity "[...] within classes in which the focus was *not* on the individual clues in a class *per se* but on a possible invariant relational structure that was perceived between and, thus, projected onto the cues" [Rivera and Rossi Becker, 2007, p. 151]. Through the follow-up inductive stage of generalizations the subjects tested the hypotheses just examining *extensions* (new particular cases beyond what was available at the beginning of the reasoning process). This process was also able to show subjects's disconfirmation capacities: the acknowledged their mistakes in generating bad induction, which had to be abandoned, in so far as they were checked as insufficient in fully capturing in symbolic terms a general attribute that would yield the total number of toothpicks in new generated cues.

3 Manipulative Abduction: How to Build Evidence Inductively Relevant

We have seen in the mathematical example of the previous sections that abductive processes that are at play can be considered manipulative. I have introduced the concept of *manipulative abduction* - contrasted with theoretical abduction [Magnani, 2001] - to illustrate situations where we are thinking through doing and not only, in a pragmatic sense, about doing. So the idea of manipulative abduction goes beyond the well-known role of experiments as capable of forming new scientific laws by means of the results (nature's answers to the investigator's question) they present, or of merely playing a predictive role (in confirmation and in falsification). Manipulative abduction refers to an extra-theoretical behavior that aims at creating communicable accounts of new experiences to integrate them into previously existing systems of experimental and linguistic (theoretical) practices. The abductive construction of new evidence ([Gooding, 1990] call these external representational/experimental devices *construals*) in the mathematical case I have illustrated above presents an extra-theoretical behavior of this type.

As I discussed previously in [Magnani, 2006a] I think that a better understanding of manipulative abduction at the level of scientific experiment could improve our knowledge of induction, and its distinction from abduction: manipulative abduction can be considered as a kind of basis for further abductive steps but also for possible meaningful inductive generalizations. For example different generated construals can give rise to different inductive generalizations. It is difficult to grasp this distinction through present logical models of the induction/abduction puzzle.

[Josephson, 2000] maintains that "An inductive generalization is an inference that goes from the characteristics of some observed sample of individuals to a conclusion about the distribution of those characteristics in some larger populations" (p. 40).

[6] Figure 2 illustrates a further evidence built by the subject engaged in a variable-oriented abduction. In such new expressly built evidence he "experienced" "[...] the use of a variable which substitutes as a general placeholder or expression for a sequence that he perceived to continuosly grow indefinitely" [Rivera and Rossi Becker, 2007, p. 151].

I contend manipulative abduction is the correct way for describing the features of what can trigger "smart inductive generalizations", as contrasted to the trivial ones. For example, in science construals and new built evidence can shed light on this process of sample "production" and "appraisal": through construals, manipulative abduction generates abstract "specific" and "ignorance mitigating" hypotheses, which in the meantime can originate possible bases for further meaningful inductive generalizations through the identification of new samples (or of new features of already available samples). Different generated construals can give rise to different plausible inductive generalizations.

4 Conclusion

In this paper I have illustrated how abduction is essentially *multimodal*. I think the issue has some consequences for logical models and computational automation of the cognitive dyad abduction-induction in scientific reasoning. First of all scientific hypotheses expressed through different modalities can be logically and computationally taken into account and suitably represented, in so far as they model the same cognitive aspect in different ways and provide different abductive inferential chances: they are different multimodal *knowledge carriers*. Second, the problem of the *non-explanatory* and *instrumental* nature of abductive reasoning/hypotheses reveals that good inductions are favored by abductive steps, which often present *explanatory* features, but that at the end in some cases resort to an inductive *non-explanatory* result, like in the case of mathematics. Third, I have stressed that abductive steps are often occurring in a continuous interplay among the reasoner and its external cognitive environment in which new experimental data (new evidence) are built and at the same time re-offered to the reasoner in a process of *manipulative abduction*: this also refers to the role of this process of building new experimental data (new evidence) in triggering smart inductive inferences.

References

[Flach *et al.*, 2006] P. A. Flach, A. C. Kakas, and O. Ray. Abduction, induction, and the logic of knowledge development. In P. A. Flach, A. C. Kakas, L. Magnani, and O. Ray, editors, *Workshop on Abduction and Induction in AI and Scientific Modeling*, pages 21–24. Kluwer Academic Publisher, Dordrecht, 2006.

[Friedman and Simpson, 2000] H. Friedman and S. Simpson. Issues and problems in reverse mathematics. *Computability Theory and its Applications: Contemporary Mathematics*, 257:127–144, 2000.

[Gabbay and Woods, 2005] D. M. Gabbay and J. Woods. *The Reach of Abduction*. North-Holland, Amsterdam, 2005. Volume 2 of *A Practical Logic of Cognitive Systems*.

[Gooding, 1990] D. Gooding. *Experiment and the Making of Meaning*. Kluwer, Dordrecht, 1990.

[Irvine, 1989] A. Irvine. Epistemic logicism and Russell's regressive method. *Philosophical Studies*, 55:303–327, 1989.

[Josephson, 2000] J. Josephson. Smart inductive generalizations are abductions. In . P. Flach and A. Kakas, editors, *Abduction and Induction*, pages 31–44. Kluwer Academic, Dordrecht, 2000.

[Lakatos, 1976] I. Lakatos. *Proofs and Refutations. The Logic of Mathematical Discovery*. Cambridge University Press, Cambridge, 1976.

[Magnani, 2001] L. Magnani. *Abduction, Reason, and Science. Processes of Discovery and Explanation*. Kluwer Academic/Plenum Publishers, New York, 2001.

[Magnani, 2006a] L. Magnani. Hasty generalizers and hybrid abducers. external semiotic anchors and multimodal representations. In P. A. Flach, A. C. Kakas, L. Magnani, and O. Ray, editors, *Workshop on Abduction and Induction in AI and Scientific Modeling*, pages 1–8. 2006.

[Magnani, 2006b] L. Magnani. Mimetic minds. Meaning formation through epistemic mediators and external representations. In A. Loula, R. Gudwin, and J. Queiroz, editors, *Artificial Cognition Systems*, pages 327–357. Idea Group Publishers, Hershey, PA, 2006.

[Magnani, Forthcoming] L. Magnani. *Abductive Cognition. The Eco-Cognitive Dimension of Hypothetical Reasoning*. Forthcoming.

[Pease *et al.*, 2005] A. Pease, S. Colton, A. Smaill, and J. Lee. A model of Lakatos's philosophy of mathematics. In L. Magnani and R. Dossena, editors, *Computing, Philosophy and Cognition*, pages 57–85. College Publications, London, 2005.

[Ray, 2007] O. Ray. Automated abduction in scientific discovery. In L. Magnani and P. Li, editors, *Model-Based Reasoning in Science, Technology, and Medicine*, pages 103–116. Springer, Berlin, 2007.

[Rivera and Rossi Becker, 2007] F. D. Rivera and J. Rossi Becker. Abduction-induction (generalization) processes of elementary majors on figural patterns in algebra. *Journal of Mathematical Behavior*, 26:140–155, 2007.

[Russell, 1973] B. Russell. The regressive method of discovering the premises of mathematics [1907]. In D. Lackey, editor, *Essays in Analysis*, pages 45–66. George Allen and Unwin, London, 1973.

[Simpson, 1999] S. G. Simpson. *Subsystems of Second Order Arithmetic*. Springer, Berlin, 1999.

[Thagard, 2007] P. Thagard. Abductive inference: from philosophical analysis to neural mechanisms. In A. Feeney and E. Heit, editors, *Inductive reasoning: Experimental, developmental, and computational approaches* , pages 226–247. Cambridge University Press, Cambridge, 2007.

[Woods, 2009] J. Woods. Ignorance, inference and proof: abductive logic meets the criminal law. In *Proceedings of the International Conference "Applying Peirce"*, Helsinki, Finland, 2009. Forthcoming.

[Woods, 2010] J. Woods. *Seductions and Shortcuts: Error in the Cognitive Economy*. 2010. Forthcoming.

Towards the Automation of Scientific Method

Oliver Ray
Department of Computer Science
University of Bristol
Bristol, BS8 1UB, UK
oray@cs.bris.ac.uk

Amanda Clare and **Maria Liakata** and **Larisa Soldatova** and **Ken Whelan** and **Ross King**
Department of Computer Science
University of Aberystwyth
Ceredigion, SY23 3DB, UK
{afc,mal,lss,knw,rdk}@aber.ac.uk

Abstract

The automation of scientific method is a subject of increasing intellectual and practical interest, with potentially great benefits to science and society. This paper discusses four key challenges in this task and explains how they have been addressed within a functional genomics project known as the Robot Scientist. In so doing, it describes how abduction and induction have enabled the automatic revision of metabolic models through a synthesis of cutting edge artificial intelligence and laboratory robotics. Our aim is to summarise the progress which has already been made and to set out an agenda for further technological and social changes that are needed to turn the automation of science into a truly useful reality.

1 Introduction

Scientific method is the continual cycle of experiment and analysis used in all branches of science to develop rational models of empirical data. Analyses lead to models that might account for observed data. Experiments lead to data that could test inferred models. Each builds on the other.

Since the first national academies of science were founded in the latter half of the 17th century, the scientific method has transformed our understanding of ourselves and of the world around us. In so doing, it has led to unprecedented cultural developments that continue right up to the present day.

But we believe a new scientific revolution is in the making with the potential to further accelerate progress in all areas of science. It being driven by the recent explosion of robotics, computing, and the internet, which are now starting to have a major impact on the practice of science.

Many branches of science are now heavily reliant on industrial-strength experimental apparatus and high-throughput computational techniques: as can be clearly seen in a spectrum of disciplines ranging from particle physics through the biological sciences all the way up to astronomy.

In addition, the world-wide-web is changing the way that scientists communicate and collaborate. The rapid growth of on-line publishing and indexing sites has made scientific papers more accessible; and a corresponding surge in public data repositories has done the same for results.

The enthusiastic adoption of web-based tools like blogs, wikis, RSS feeds, social networks and grid computing has led to a trend dubbed *Science 2.0* [Waldrop, 2008] of greater cooperation among scientists: with some even promoting a fully public *Open Notebook* approach [Bradley, 2007].

Yet this vision of science in the 21st century is beginning to attract some intense controversy as the focus shifts from its obvious technological benefits to some of its profound social implications that fundamentally challenge the way scientists have worked for more than three hundred years.

The strong pressure on human scientists and institutions to enhance their reputations and increase their wealth through exclusive publications and patents promotes some amount of secrecy which many are duly worried may be compromised by the new agenda of transparency.

By contrast, we are concerned with just one issue: how can technology help us to do better science? We are interested in human affairs only to the extent that they help or hinder the scientific process. Our goal is to better understand and support the roles of both man and machine.

While this may well raise some sensitive issues and hard decisions, we believe the potential benefits are simply too great to ignore. Instead, we take inspiration from Alhazen who, nearly a millennium ago, wrote in defence of what we now call scientific method:

> "Truth is sought for its own sake. And those who are engaged upon the quest for anything for its own sake are not interested in other things. Finding the truth is difficult, and the road to it is rough."

In fact, we do appreciate that pragmatic considerations must play a key role in science, just as they do in any other field of endeavour. Our point is that such factors must be clearly understood and fully integrated into the scientific life-cycle in a principled and properly supported way.

From the outset, it is clear that a major obstacle to the more effective use of technology in science is due to the way that scientific results are reported and recorded; which for over three centuries has been exclusively done through published journal articles and unpublished paper notebooks.

Just representing this information in a more formal way would greatly facilitate the use of computational tools and help reduce the ambiguities of natural language. In turn, this would make it easier to store, access, inspect, query, validate and reuse analytical models and experimental data.

Moreover, the use of formal knowledge representations would allow for semantic markup and ontologies that would enable the exploitation of more sophisticated reasoning services based on emerging Semantic Web technologies [Berners-Lee *et al.*, 2001; Hendler, 2003].

This insight is leading to a view of *Semantic Science* [Poole *et al.*, 2008] which advocates the publication of theories and data in machine-readable form. The idea is that theories will be used to automatically generate predictions from data, while data will be used to automatically evaluate theories.

While the realisation of these aims would already provide significant advantages over the traditional practice of science, we believe it is only the first step in what will prove to be a far more profound integration of machine-based support into all aspects of the scientific lifecycle.

To fully exploit such technologies, the experimental and analytical phases of the scientific process must be formalised along with the resulting data and theories. This will bring rigour to the specification of scientific protocols and will enable the automation of their design and execution.

Our vision is to develop an autonomous cycle of scientific activity that combines theoretical analyses and experimental interventions in way that can be understood and guided by humans; and we argue much this is possible using methods from artificial intelligence and laboratory robotics.

This paper discusses four challenges in the automation of science and explains how they have been addressed in a project called the Robot Scientist [King *et al.*, 2004; 2009]. In so doing, it describes how abduction and induction were used to revise a state-of-the-art scientific model.

The paper is structured as follows. Section 2 introduces the Robot Scientist application domain. Section 3 outlines the techniques we have developed for formalising scientific data and models. Sections 4 and 5 describe the hard- and soft-ware we have developed for performing scientific experiments and analyses. Section 6 summarises what has been achieved and what is still to do in terms of closing the loop.

2 Selecting a Domain

To completely mechanise just one iteration of scientific method, in any domain, raises so many technical challenges that, at least for the time being, the automation of science will most likely have to be studied in the context of just a few well-defined target applications.

The Robot Scientist project focuses on a certain type of microbial growth experiment involving yeast, *S. cerevisiae*. Each such experiment measures the growth of a yeast strain (from which some genes have been removed) in a growth medium (to which some nutrients have been added).

These experiments are a classic method for understanding the function of deleted genes and the role that they play in the living cell. Furthermore, several additional factors mean this specific domain is an ideal target for automation from the perspective of both experiment and analysis:

First is the availability of yeast strains and growth nutrients needed as inputs in the experiments. Second is the availability of laboratory hardware needed to carry out the main steps of the experiments. Third is the availability of a public databases with information needed to interpret the experiments.

Even though the experimental and analytic aspects of this domain are relatively well understood, the work we describe below will show the considerable difficulties and significant benefits arising from their automation. Fortunately, much of this work will be relevant in other domains too.

3 Formalising the Domain

Having chosen an appropriate domain, any relevant data, models, experiments and analyses must be represented in enough detail for machines to perform the required tasks. As a matter of fact, this level of formality is desirable to ensure objectivity whether automation is intended or not.

While some aspects of science are highly mathematical, very often the description of experiments are treated much too informally. Moreover, even when scientific data and models are formally specified, they often leave out a lot of important assumptions and meta-data.

In the Robot Scientist project we utilise formal ontologies and logical representations to ensure the accurate encoding of all necessary information. More precisely, we have developed two separate ontologies to characterise various high-level and low-level elements of an experimental investigation.

First there is a scientific EXPeriments Ontology (EXPO) [Soldatova and King, 2006] for experimental design and methodology. Second there is an ontology of EXperimental ACTions (EXACT) [Soldatova *et al.*, 2008] for experimental protocols and physical manipulations.

Two example fragments of these ontologies are shown for a typical experiment in Figures 1 and 2, respectively. Currently, EXPO meta-data is automatically stored for every experiment conducted by the Robot Scientist.

Other interesting applications of these ontologies can be found in [Soldatova and King, 2006] and [Soldatova *et al.*, 2008] where they were used to critique three arbitrary studies on the phylogeny of solenodons, the mass of the top quark, and the preparation of yeast gene deletion cassettes.

⟨scientific experiment⟩ :	
⟨administration info⟩ :	
⟨title⟩ :	Robot scientist
⟨ID⟩ :	exp200401113-0001
⟨classification by domain⟩ :	
⟨DDC(Dewey)⟩ :	576 Microbiology
⟨research hypothesis⟩ :	
⟨natural language⟩ :	Knocked out gene named "yer152c" (= met8) has the function named "2.6.1.39" (=2-aminoadipate:2-oxoglutarate aminotransferase)
⟨artificial language⟩ :	encodes(yer152c, 2.6.1.39)
⟨null hypothesis⟩ :	
⟨artificial language⟩ :	¬encodes(yer152c, 2.6.1.39)
⟨alternative hypothesis⟩ :	
⟨natural language⟩ :	⟨time effect⟩ : maturation effect (incubator too cold) ⟨object effect⟩ : no entry of metabolite into the cells ⟨object effect⟩ : cross contamination
⟨domain model⟩ :	
⟨language⟩ :	Prolog.
⟨reference⟩ :	Whelan, K.E. & King, R.D. (2005). Using a logical model to predict the growth behaviour of yeast cell cultures. Tech. Report, UWA-DCS-05-045.
⟨experimental design⟩ :	
⟨subject⟩ :	The Robot Scientist
⟨object⟩ :	S. cerevisiae
⟨experimental model⟩ :	
⟨factor⟩ :	Strains: wild-type [Mat A, by4741]; knockout [yer152c]
⟨factor⟩ :	Metabolites: minimal media; additional compound [xxx]

Figure 1: EXPO annotation of a micro-biological experiment (a fragment, from [Soldatova and King, 2006])

⟨operating procedure⟩ :	grow yeast culture
⟨pre-condition⟩ :	sealed yeast colonies plate located_in cold room
⟨pre-condition⟩ :	YPD media bottle located_in cold room
⟨experiment action⟩ :	move 1
. . .	
⟨experiment action⟩ :	add 17
⟨component 1⟩ :	single yeast colony
⟨start volume⟩ :	small volume
⟨start container⟩ :	sealed yeast single colonies plate
⟨end container⟩ :	YPD conical flask
⟨equipment⟩ :	inoculating loop
⟨experiment action⟩ :	rename 18
⟨old name⟩ :	YPD conical flask
⟨new name⟩ :	yeast culture flask
⟨end location⟩ :	incubator
⟨experiment action⟩ :	move 19
⟨object⟩ :	yeast culture flask
⟨start location⟩ :	laminar flow hood
⟨end location⟩ :	incubator
⟨experiment action⟩ :	incubate 20
⟨object⟩ :	yeast culture flask
⟨start equipment⟩ :	shaking incubator
⟨rpm⟩ :	200
⟨tempo⟩ :	30'C
⟨time interval⟩ :	12-24h
⟨goal⟩ :	grow yeast until medium becomes cloudy
⟨post-condition⟩ :	yeast culture located_in incubator

Figure 2: EXACT annotation of a micro-biological experiment (a fragment, from [Soldatova *et al.*, 2008])

To automate the analytical phase of the scientific method, it is also necessary to obtain a formal description of the target domain. For the Robot Scientist project we have developed a detailed model of the *S. cerevisiae* metabolism [Whelan and King, 2008], part of which is shown in Figure 3.

Nodes in this figure represent metabolites involved in the transformation of the compound Glycerate-2-phosphate into the amino acids Tyrosine, Phenylalanine, and Tryptophan. Arrows denote chemical reactions converting their substrates (the tails) into products (the heads).

Each node is labelled with a KEGG identifier (in red); and each arrow is annotated with a 4-part EC number (in blue) and a set of genes (in green) called an enzyme-complex. The singly dashed line denotes the inhibition of an enzyme by a metabolite. The doubly dashed line is the cell membrane.

All reactions are assumed to take place in the cell cytosol using nutrients imported from the growth medium; and they are all assumed to proceed at a standard rate (within 1 day), except for the import of two italicised compounds, which take longer (between 1 and 2 days).

The additional nutrients and knockout genes used in each experiment are denoted by atoms of the form `additional_nutrient(e,m)` and `knockout(e,g)`, for an experiment `e`, gene `g`, and metabolite `m`. Nutrients common to all experiments are denoted `start_compound(m)`.

By definition, all start compounds and additional nutrients are in the growth medium on any day in any experiment. This is represented by two logical rules which state that a certain metabolite is in a specific compartment on a given day in a particular experiment:

```
in_compartment(Exp,Met,medium,Day) :-
    start_compound(Met).

in_compartment(Exp,Met,medium,Day) :-
    additional_nutrient(Exp,Met).
```

Reactions, genes and complexes are all assigned unique identifiers so that atoms of the form `catalyst(r,c)` and `component(g,c)` can be added to the model to in order to denote the denote the fact that complex `c` catalyses reaction `r` and the fact that gene `g` participates in complex `c`.

The inhibition of a complex `c` by a metabolite `m` is denoted `inhibitor(c,m)`. Any metabolites that are essential to the cell growth, such as the three amino acids, are specified as such by adding ground atoms of the form `essential_compound(m)`.

Cell development is arrested if an essential metabolite is not in the cytosol. But, if development is not arrested, then growth is predicted. A complex is deleted if a component gene is knocked out; and it is inhibited if some inhibitor is present (in high concentration) as an additional nutrient:

```
arrested(Exp,Day) :-
    essential_compound(Met),
    not in_compartment(Exp,Met,cytosol,Day).

predicted_growth(Exp,Day) :-
    not arrested(Exp,Day).

deleted(Exp,Cid) :-
    component(Orf,Cid),
    knockout(Exp,Orf).

inhibited(Exp,Cid) :-
    inhibitor(Cid,Met),
    additional_nutrient(Exp,Met).
```

To complete our background theory, it remains to give a logical encoding of the metabolic reactions. To facilitate the addition and removal of reactions, they are each given one of three degrees of belief: $certain$ (i.e., definitely in the model), $retractable$ (i.e., initially in the model, but can later be excluded), or $assertable$ (i.e., initially out of the model, but can later be included). Note that this allows us to consider reactions from related pathways or organisms for inclusion in a revised network; which is common practice as it ensures all newly introduced reactions are biologically feasible.

For every reaction, one rule is added to the theory for each product. Each rule states that the product will be in its compartment if (i) all substrates are in their respective compartments, (ii) there is an enzyme-complex catalysing the reaction whose activity is not inhibited and whose genes are not deleted, (iii) sufficient time has passed for the reaction to complete, and (iv) the reaction has not been excluded (if it is retractable) or it has been included (if it is assertable). As an example, the following is one of two rules produced for reaction 2.5.1.19 with id 31, assuming it is retractable:

```
in_compartment(Exp,"C01269",cytosol,Day) :-
    in_compartment(Exp,"C00074",cytosol,Day),
    in_compartment(Exp,"C03175",cytosol,Day),
    catalyst(31,Cid),
    not inhibited(Exp,Cid),
    not deleted(Exp,Cid),
    Day >= 1,
    not exclude(31).
```

As explained in [Ray *et al.*, 2009], for every start compound and additional nutrient, `m`, we assume there is an import reaction which takes `m` from the `medium` into the `cytosol`; and to each reaction with no known catalysts, we attribute an `unknown` catalyst (so all reactions are assumed to proceed in the absence of evidence to the contrary). As a result, our model of the AAA pathway actually contains 22 metabolic reactions and 23 import reactions.

4 Performing Physical Experiments

The physical mechanisation of scientific experiments poses significant technical challenges. The Robot Scientist is an autonomous laboratory platform that is able to conduct growth experiments with no human assistance. As shown by the schematic in Figure 4, the hardware includes freezers, incubators, liquid handlers, plate readers, centrifuges, washers, robot arms and environment sensors, all contained in an air-filtered enclosure. The system is able to extract particular strains of yeast from the freezer, culture them in rich growth media, harvest the cells, inoculate them into defined media, and monitor the resulting growth over a period of several days by taking regular optical density readings. In addition, it annotates all experiments with appropriate meta-data and runs them many times to ensure statistically significant results. Moreover, it schedules all these activities in a way that makes efficient use of available resources. Further details can be found in [King *et al.*, 2009].

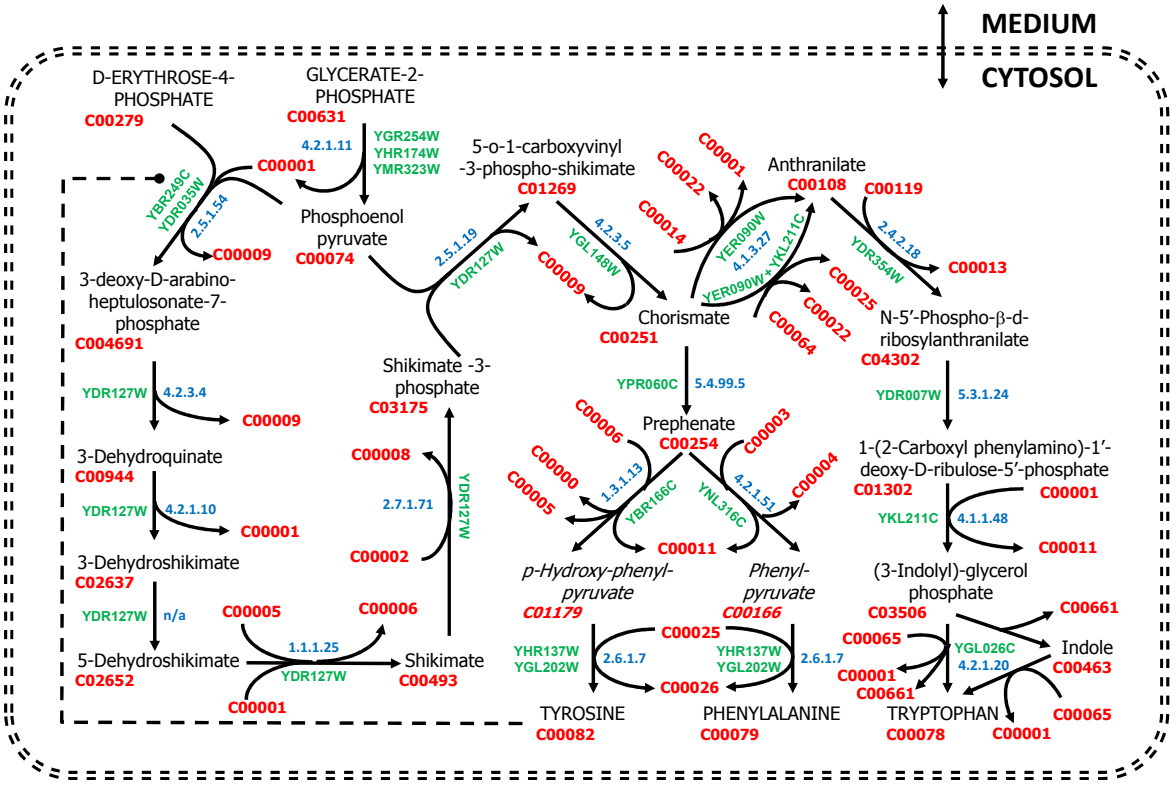

Figure 3: Aromatic Amino Acid (AAA) biosynthesis pathway of *S. cerevisiae*

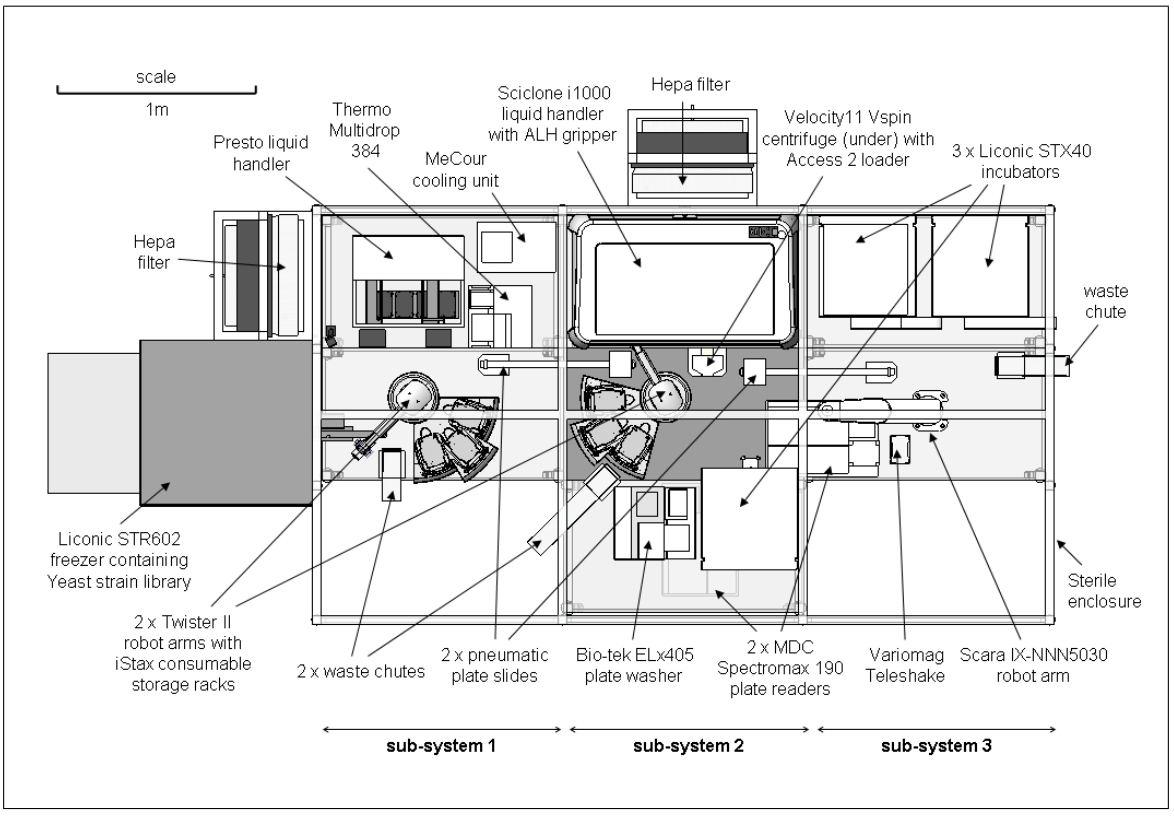

Figure 4: Schematic view of Robot Scientist hardware

5 Performing Theoretical Analyses

Over the course of time, scientific models are often revised as discrepancies begin to emerge between observed and predicted results. This is usually accomplished by hand, as in the case of the AAA pathway of Figure 3, which was mainly derived from the KEGG database, but was manually crafted in places. Our goal is to automate this revision process using computational tools.

In recent work, we have used a non-monotonic reasoning system called eXtended Hybrid Abductive Inductive Learning (XHAIL) [Ray, 2009] to do just this. The main advantage of XHAIL over other systems is that it provides well-defined semantics and proof procedure for abduction and induction over extended logic programs (which include operators for classical and default negation).

Non-monotonicity is useful as it allows to reason with defaults and exceptions and to hypothesise revisions involving the removal of information from an incorrect model as well as the addition of information into an incomplete model. XHAIL achieves this by generalising traditional techniques of language and search bias to the non-monotonic case.

First, the hypothesis space is controlled by a set of mode declarations [Muggleton, 1995] that allow the user to constrain which literals may appear in the heads and bodies of hypothesis clauses. Second, a compression heuristic [Muggleton, 1995] is used to select between competing hypotheses by preferring solutions with the fewest number of literals.

After utilising some artificial data in [Ray *et al.*, 2009] to validate the capability of XHAIL for adding and removing metabolic reactions, enzyme inhibitions or gene functions, we applied the method to try and revise our state-of-the-art AAA model in the light of real growth data collected by the Robot Scientist.

Upon submitting the observed growth results to XHAIL, we discovered they were inconsistent with the AAA model, thereby suggesting that a revision was necessary. By systematically varying the language bias, we very quickly obtained several hypotheses that achieved logical consistency between predicted and observed growth.

XHAIL produced two hypotheses of interest. The first one, was interpreted as stating that the import of anthranilate is a slow reaction (just like the import of phenyl-pyruvate and P-hydroxy-phenyl-pyruvate). The second one, was interpreted as stating that our source of Indole (`"C00463"`) is contaminated with Tryptophan (`"C00078"`). This is plausible as Indole can be synthesised from Tryptophan (by essentially reversing reaction 4.2.1.20) but any unconverted Tryptophan may be hard to separate off. Using mass spectrometry we have verified that was indeed the case.

It turns out these two hypotheses are sufficient to restore logical consistency with all of the growth measurements made by the Robot Scientist. As a result of this work, the first hypothesis has enabled us to improve our state-of-the-art AAA model; and the second hypothesis has alerted us to a source of systematic experimental error that we shall take care to avoid in future work.

6 Closing the Loop

The final step involves the integration of our hardware and software in a completely automated cycle of scientific discovery that can run indefinitely. Even though we have automated one iteration of the scientific cycle, it is not yet possible to loop arbitrarily without human assistance. In order to progress, we urgently need to develop methods for designing experiments to test competing hypotheses.

7 Related Work

Our approach builds upon earlier Robot Scientist work, which used a reasoning system called Progol5 [Muggleton and Bryant, 2000] to rediscover gene-enzyme mappings removed from the AAA pathway [King *et al.*, 2004]. However, XHAIL overcomes several key limitations of Progol5, including its inability to reason hypothetically through negation and its inability to infer more than one clause in response to any given example. For these reasons the logical model used by Progol5 employs a complex list-based representation of reactions over a logic program in which all negations are restricted to built-in predicates and where most of the code is devoted to procedural issues such as pruning the search tree, avoidance of cyclic computations, and efficient sorting of data structures. As a result, the earlier Progol5 model is restricted to the learning of new gene-enzyme mappings in single gene deletion experiments.

By contrast, the XHAIL model employs a completely declarative representation, adapted from [Whelan and King, 2008] but extended with support for enzyme-complexes, which imposes no a-priori constraints on the learning task and can be applied to multiple gene deletion experiments and simultaneously add or remove reactions, inhibitions and complexes. We note that this flexibility depends upon the ability of XHAIL to reason nonmonotinically and to infer multiple clauses in response to a single example. Moreover, XHAIL's compression heuristic is also appropriate because it ensures the revisions are in some sense minimal.

Related work in [Juvan *et al.*, 2005; Dworschak *et al.*, 2008; Papatheodorou *et al.*, 2005] and [Baral *et al.*, 2004] apply logic-based approaches to the identification of genetic regulatory networks and signalling pathways. But they do not incorporate enzyme and reaction data. Numerical techniques based on ordinary differential equations for analysing metabolic flux [Varma and Palsson, 1994] allow quantitative simulation of metabolic networks and can be used for parameter estimation. But they cannot be used to suggest structural refinements to metabolic networks.

8 Conclusions

This paper described recent progress in automating scientific method within a functional genomics domain. In particular, it described how a state-of-the-art metabolic network model was successfully revised through the abductive and inductive analysis of experimental data acquired by a Robot Scientist. This demonstrates the significant benefits to be gained by integrating abduction and induction in a biological context and it highlights the utility of non-monotonic logical reasoning to enable both the addition and removal of information.

Acknowledgments

Part of this work was funded by the BBSRC, EPSRC and RCUK. We acknowledge Wayne Aubrey for his help with Figure 2. We thank Kate Martin for performing the mass spectrometry experiments. We are grateful to Andrew Sparkes for preparing Figure 4.

References

[Baral *et al.*, 2004] C. Baral, K. Chancellor, N. Tran, N.L. Tran, A. Joy, and M. Berens. A knowledge based approach for representing and reasoning about signaling networks. In *Proc. 12th Int. Conf. on Intelligent Systems for Molecular Biology*, pages 15–22, 2004.

[Berners-Lee *et al.*, 2001] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[Bradley, 2007] J. Bradley. Open notebook science using blogs and wikis. *Nature Prepress*, 2007.

[Dworschak *et al.*, 2008] S. Dworschak, S. Grell, V. Nikiforova, T. Schaub, and J. Selbig. Modeling Biological Networks by Action Languages via Answer Set Programming. *Constraints*, 13(1/2):21–65, 2008.

[Hendler, 2003] J. Hendler. Science and the semantic web. *Science*, 299(5606):520–521, 2003.

[Juvan *et al.*, 2005] P. Juvan, J. Demsar, G. Shaulsky, and B. Zupan. GenePath: from mutations to genetic networks and back. *Nucleic Acids Res.*, 33, 2005.

[King *et al.*, 2004] R. King, K. Whelan, F. Jones, P. Reiser, C. Bryant, S. Muggleton, D. Kell, and S. Oliver. Functional Genomic Hypothesis Generation and Experimentation by a Robot Scientist. *Nature*, 427:247–252, 2004.

[King *et al.*, 2009] R. King, J. Rowland, S. Oliver, M. Young, W. Aubrey, E. Byrne, M. Liakata, M. Markham, P. Pir, L. Soldatova, A. Sparkes, K. Whelan, and A. Clare. The automation of science. *Science*, 324(5923):85–89, 2009.

[Muggleton and Bryant, 2000] S. Muggleton and C. Bryant. Theory Completion Using Inverse Entailment. In *Proc. of the 10th Int. Conf. on Inductive Logic Programming*, volume 1866 of *LNCS*, pages 130–146. Springer, 2000.

[Muggleton, 1995] S. Muggleton. Inverse Entailment and Progol. *New Generation Comp.*, 13(3-4):245–286, 1995.

[Papatheodorou *et al.*, 2005] I. Papatheodorou, A. Kakas, and M. Sergot. Inference of gene relations from microarray data by abduction. In *Proc. 8th Int. Conf. on Logic Programming and Nonmonotonic Reasoning*, volume 3662 of *LNCS*, pages 389–393. Springer, 2005.

[Poole *et al.*, 2008] D. Poole, C. Smyth, and R. Sharma. Semantic science: Ontologies, data and probabilistic theories. In *Uncertainty Reasoning for the Semantic Web I*, volume 5327 of *LNCS*, pages 26–40. Springer, 2008.

[Ray *et al.*, 2009] O. Ray, K. Whelan, and R. King. A nonmonotonic logical approach for modelling and revising metabolic networks. In *Proc. 3rd Int. Conf. on Complex, Intelligent and Software Intensive Systems*, pages 825–829. IEEE, 2009.

[Ray, 2009] O. Ray. Nonmonotonic Abductive Inductive Learning. *Journal of Applied Logic*, 3(7), 2009.

[Soldatova and King, 2006] L. Soldatova and R. King. An ontology of scientific experiments. *Journal of the Royal Society, Interface*, 3(11):795–803, 2006.

[Soldatova *et al.*, 2008] L. Soldatova, W. Aubrey, R. King, and A Clare. The EXACT description of biomedical protocols. *Bioinformatics*, 24(13):295–303, 2008.

[Varma and Palsson, 1994] A. Varma and B. Palsson. Metabolic flux balancing: Basic concepts, scientific and practical use. *Nature Biotechnology*, 12:994–998, 1994.

[Waldrop, 2008] M. Waldrop. Science 2.0: Great new tool, or great risk? *Scientific American*, 2008.

[Whelan and King, 2008] K. Whelan and R. King. Using a logical model to predict the growth of yeast. *BMC Bioinformatics*, 9(97), 2008.

Explaining Effects of Host Gene Knockouts on Brome Mosaic Virus Replication

Deborah Chasman[1,2], **Brandi Gancarz**[3], **Paul Ahlquist**[3], **Mark Craven**[2,1]
University of Wisconsin–Madison
[1]Department of Computer Sciences
[2]Department of Biostatistics and Medical Informatics
[3]Institute for Molecular Virology
chasman@cs.wisc.edu, craven@biostat.wisc.edu

Abstract

Gene products are key players in the interaction networks within a cell. We analyze an experiment in which a yeast knockout library was assayed for the effects of host gene deletion on the replication of Brome Mosaic Virus (BMV). These observations, integrated with the partially known yeast interaction network, may be used to infer which host processes and gene products are involved in the mechanism of BMV's replication. We approach this task using Inductive and Abductive Logic Programming (ILP and ALP). We use ALP to abduce causal explanations for each observation, including possible host interfaces with BMV. Some notable aspects of our task that differ from previous work using abduction in systems biology include a highly incomplete background model and a large number of observations to explain. Additionally, we expect that there are many interfaces between the host cell and the virus, and that each abduced interface will serve to explain a handful of observations. We determine that ILP is unable to identify general, informative models that characterize host-virus interactions accurately. Using ALP, however, we are able to construct causal explanations that link multiple observations to the same host interface.

Introduction

A complex network of interactions within a cell determines its response to its environment. Many of the key players in these networks are gene products; i.e., proteins and RNAs that form important structures and catalyze reactions within the cell. We have only partial knowledge of these networks. However, in some cases we can shine a light into the "black box" by selectively turning off genes and observing the resulting change in the cell's response. There exists a library of strains of the yeast *Saccharomyces cerevisiae*, for example, in which each strain is defined by having a single gene deleted or modified to allow the gene's expression to be suppressed during an experiment [Winzeler et al., 1999]. Using these deletion strains, we can perform high-throughput assays by exposing each strain to the same conditions. These observations show us the ultimate effect of the loss of a gene on the cell's response to the conditions of interest. By integrating these observations with a known partial network of intracellular gene interactions, we may be able to infer which cellular components and processes are involved in the response, and suggest a more complete model of the network.

Viral infection is one particular condition to which a cell might respond, and is one that is of particular relevance to human health. Brome Mosaic Virus (BMV) is a positive-strand RNA virus, a member of the same viral family as Hepatitis C and SARS. Understanding the BMV mechanism of replication may provide insight into the mechanisms of these high-profile, pathogenic viruses. The Ahlquist laboratory has artificially infected *Saccharomyces cerevisiae* knockouts with BMV [Kushner et al., 2003; Gancarz and Ahlquist, 2008]. By augmenting the virus with a luciferase reporter, it is possible to measure the amount of viral replication in an infected yeast colony. These assays have identified on the order of 100 host genes whose deletion or suppression significantly inhibits viral replication, and another 100 or so host genes whose knockouts encourage replication. We are interested in using computational methods to posit how these genes interact with one another, and with the virus, affecting its ability to replicate. Most genes are not directly involved in BMV's activity, but are instead part of some pathway that contains an actual interface with the virus. Our primary goal is to explain the causal chains that lead from deleted genes to potential interfaces with the virus.

The network of relationships among gene products and other molecules in *S. cerevisiae* is partially known, and is represented in various on-line databases [Christie et al., 2004; Stark et al., 2006; Pu et al., 2007]. For example, genes may encode proteins that form complexes with each other or otherwise physically interact. Or, they may catalyze reactions along the same pathway. We hypothesize that we can use these relationships to explain the BMV replication results in the yeast knockout assays. Because the background knowledge describing these networks of interactions has a rich relational structure, Inductive Logic Programming (ILP) and Abductive Logic Programming (ALP) represent compelling approaches to the problem.

As a preliminary investigation, we frame the problem as an inductive one. We wish to learn general rules for the concepts "gene deletions that significantly inhibit viral replication" and "gene deletions that significantly promote viral replication."

Dividing our observations into positive and negative examples of these concepts, we use the ILP system Aleph [Srinivasan, 2007] to hypothesize clauses for these concepts in terms of our partial interaction network for *S. cerevisiae*.

The primary focus of our study, however, is on using Abductive Logic Programming to infer explanations to account for the BMV replication observations. There are several notable aspects of our task that distinguish it from previous work using abduction for systems biology applications:

- The available background knowledge for our task is highly incomplete and likely contains some false-positive assertions.
- Our abductive task involves constructing explanations for a large number of observations (the results of hundreds of knockout/supression experiments).
- It is likely that a large number of abduced predicates are required in order to explain the observations. That is, the virus may have many interfaces to its host cell.
- Our approach forms explanations that account for multiple observations, with each explanation consisting of logical clauses that share the same abduced predicate.

In our ALP investigation, in contrast to our ILP experiment, we want to construct a specific, causal explanation for each BMV replication observation, rather than a set of general clauses. These explanations require a vital piece missing from our background information: the actual host interfaces to the virus. These interfaces may be particular gene products, protein complexes, or small molecules produced by metabolic reactions. While the ALP literature typically uses the word "explanations" to describe only the set of abduced facts, we consider an explanation to consist of the entire chain of literals describing the relationship between the knocked-out gene and the viral interface. Given that our background knowledge is so incomplete, we liberally hypothesize multiple explanations (and consequently multiple ground abducibles) for each observation. Additionally, we attempt to find explanations that serve to explain multiple observations.

Related Work

Several studies have applied ALP, ILP, or a hybrid approach to systems biology. Ong et al. (2007) apply ILP to predict gene expression regulation in yeast using time series data. Their model uses the yeast interaction network to learn clauses that cover genes showing similar expression patterns over time. Our task does not include this temporal element. Tamaddoni-Nezhad et al. (2006) use an Abductive ILP approach: they abduce the effects of a toxin on rat metabolic enzymes, and use ILP to learn general clauses covering the abduced facts. ALP is also applied in the Robot Scientist project, which contains an abductive component used to complete a model of yeast metabolic pathways [Reiser et al., 2001].

One way in which our task differs from previous work in abduction, and work integrating it with ILP, is that we do not assume the background network is complete except for the abducible predicate. Our background interactions are not necessarily active under the same conditions, and they are certainly incomplete. Additionally, we expect that the host cell and the virus interact in many unique ways. Consequently, we search for many unique, specific explanations, each covering a small group of observations, rather than a small set of general clauses.

Data and Representation

We apply two algorithms to our problem: ILP, as implemented in Aleph [Srinivasan, 2007], and ALP, as implemented in ProLogICA [Ray and Kakas, 2006], with some additions. Both algorithms require background information and examples of a target relation, which we summarize below.

Data

For our examples, we use observations of gene knockout effect on viral replication, and assign each observation to a class. The measured values represent the fold change in luciferase expression (implying viral replication) in a mutant as compared to the wildtype. The observations from the Ahlquist laboratory's BMV assays cover 4887 genes, 615 of which are essential genes [Gancarz and Ahlquist, 2008], and 4272 of which are nonessential genes [Kushner et al., 2003].

We process the measurements into one value for each assayed knockout. The nonessential gene data includes measurements from at most two successful trials; we use the average of the two as our value for that gene. The essential gene data includes up to four measurements for each assayed gene. Two separate trials were performed for each gene, with measurements taken at two time points within each trial. We average each time point over its two trials, and then take as our value the time point with the greatest magnitude. We then convert these measurements into fold changes. Finally, we assign discrete labels to each measurement based on the following two relations.

Target Relations

We consider two target relations: one classifying genes whose deletion significantly increases viral replication (`up`), and another classifying genes whose deletion significantly decreases viral replication (`down`). Previous work has defined thresholds for what fold changes are considered significant [Kushner et al., 2003; Gancarz and Ahlquist, 2008]. For essential gene knockdowns, a significant fold change has a magnitude greater than 6.0. For nonessential gene knockouts, a significant fold change has a magnitude greater than 3.0. We summarize the two relations as follows:

- **`Down.`** Positive examples of this concept are genes whose deletion or suppression *significantly* inhibits viral replication. Negative examples are observations of *any* positive fold change (fold change greater than 0). Sample encoding: `replication(ygl048c, down).` This division results in 122 positive examples and 1762 negative examples.
- **`Up.`** Positive examples of this concept are genes whose deletion or suppression *significantly* increases viral replication. Negative examples are observations of *any* negative fold change. Sample encoding:

`replication(ypl011c, up)`. This division results in 88 positive examples and 2769 negative examples.

Note that both relations exclude a set of observations with uncertain classification: those with fold-changes between 0 and the positive or negative significance threshold.

Background Knowledge

To represent the known yeast network, we assemble logical representations of gene attributes and interactions in *S. cerevisiae*. These include genetic interactions, protein-protein interactions, post-translational modifications of transcription factors, GO annotations, metabolic pathways, protein complexes, and predicted protein complexes or functional units. Each dataset is encoded using either a binary or ternary relationship among atoms. Here is a brief summary of the relationships that make up our background information, along with a sample encoding.

- **Physical and genetic interactions.** These relationships from the BioGRID database [Stark et al., 2006] describe observed physical and genetic interactions between genes. Sample encoding: `physical(GeneA, GeneB), genetic(GeneA, GeneB)`.
- **Genetic interactions from expression profiles.** These datasets from Rosetta Inpharmatics, Inc describe the quantitative effects of approximately 900 single gene knockouts on the expression levels of most other yeast genes [Hughes et al., 2000; Mnaimneh et al., 2004]. Hughes et al. include p-values calculated for each measurement. We considered a measurement to be significant if its p-value is less than or equal to .01. As Mnaimneh et al. do not include p-values for their measurements, we choose to keep those measurements with fold changes of magnitude two or greater from wildtype expression. Sample encoding: `upRegulates(GeneA, GeneB)` means that `GeneA` is necessary for transcription of `GeneB`. We observe a decrease in `GeneB` expression when `GeneA` is suppressed or deleted.
- **Post-translational modifications of transcription factors.** Transcription factors are proteins involved in the regulation of gene expression. Sometimes a transcription factor requires modification by another protein in order to activate it. The loss of the gene encoding either the transcription factor or the modifier may result in a different level of expression of the target gene, and may indirectly influence the interface with the virus. We use triplets from the PTM-Switchboard project [Everett et al., 2008]. Sample encoding: `tf_ptm(Modifier, TranscriptionFactor, TargetGene)`.
- **Metabolic pathways and protein complexes.** If a gene product is involved in a metabolic pathway, its deletion may influence activity downstream. Herrgård et al. (2008) have aggregated known pathways from multiple datasets into one consensus model. We include these pathway relationships between genes and metabolites in order to suggest a unified explanation for groups of genes whose knockouts result in similar effects and which influence the same pathway. Sample encoding: `pathForward(A, B)`. `A` and `B` may be genes, molecules, or protein complexes. For example, `A` may be a gene catalyzing the reaction that produces molecule `B`. The pathways are not linear; there may be multiple `B`s one step forward from any given `A`.
- **Protein complexes.** It is possible that the cell's interface with the virus is a protein complex. If this is the case, the knockout of a gene integral to the assemblage or functioning of the complex should inhibit viral replication. We include a collection of manually curated, literature-supported protein complexes from the CYC2008 project [Pu et al., 2008]. Sample encoding: `inComplex(GeneA, ComplexA)`.
- **GO annotations.** The Gene Ontology (GO) is a system for annotating genes with terms that describe known attributes of the genes [Ashburner et al., 2000]. The terms cover three categories: Cellular Component, Molecular Function, and Biological Process. Sample encoding: `go(GeneA, GO:1234)`.

Recent studies have predicted clusters of genes which may represent functional units. Many of these correspond to complexes previously reported in literature.

- **Predicted protein complexes.** Pu et al. (2007) have predicted a clustering based on high-throughput data of protein-protein interactions. Many of their clusters contain one or more previously reported complexes, while others predict complexes. Sample encoding: `inComplex(GeneA, YHPT_X)` means that GeneA is a member of cluster `YHPT_X`.
- **Modules and Complementing Module Pairs**. Ulitsky et al. (2008) present a method to cluster genes based on genetic interactions, endeavoring to predict functional units. Many of their resulting clusters, called modules, correspond to previously reported complexes. They also present pairs of modules, Complementing Module Pairs (CMPs), which may represent pairs of functionally redundant units. Sample encoding: `inModule(GeneA, ModuleA), cmp(ModuleA, ModuleB)`.

Hypothesizing Clauses using ILP

The first question we consider is whether our experimental observations of BMV replication can be explained by general rules induced using a learning approach such as ILP. That is, we want to assess the ability of ILP to learn meaningful clauses that characterize the `up` and `down` classes in terms of the relationships represented in our assembled background information.

We use the Aleph ILP system [Srinivasan, 2007] for the experiments reported in this section. Because the proportion of positive to negative examples is greatly skewed, we define a cost function for Aleph in which a positive example covered by a clause has twice as much weight as a negative example. We restrict Aleph to suggesting clauses covering at least three positive examples, and search for clauses up to length four.

To evaluate the ability of the learning algorithm to induce descriptions that capture meaningful generalizations of the

Table 1: Results from the ILP experiment. Shown are precision (P), recall (R), accuracy (Acc.), and F_1-measure for the target relations, based on the results of twenty-fold cross-validation. We also show the p-value for F_1 calculated from the results of the permutation test.

Relation	**P**	**R**	**Acc.**	F_1	p-**value**
up	0.040	0.441	0.670	0.073	0.05
down	0.080	0.541	0.557	0.137	0.02

positive examples, we employ a twenty-fold cross-validation methodology. In this procedure, we run the ILP algorithm twenty times, each time leaving out $\frac{1}{20}$ of the examples for a test set. We evaluate the predictive accuracy of our learned models by measuring precision, recall, and F_1 (the harmonic mean of precision and recall).

To assess whether the learned clauses represent specifically how the host genes influence viral replication, as opposed to simply characterizing groups of genes that are related in some way, we use a permutation testing methodology. Permutation testing here involves comparing the learning algorithm's predictive accuracy on the given data to its accuracy on random permutations of the observation labels. For both observation classes `up` and `down`, we randomly partition the data into positive and negative sets 100 times, keeping the class sizes in the same proportion as in the original data. We perform twenty-fold cross validation on each partition to acquire an F_1 score for that partition. The 100 resulting F_1 scores represent our null distribution.

Results

Table 1 summarizes the performance of the ILP algorithm on the test sets in terms of precision, recall, accuracy, and F_1. The rightmost column in the table shows the p-value for F_1 as determined by the permutation test. With respect to the F_1-measure, we can reject the null hypothesis at the level of $p \leq 0.05$ for both the `up` and `down` relations. This result suggests that the learned Aleph models are capturing some meaningful information about how the host genes interact with the virus. However, the low precision of the learned clauses indicates that the ILP approach is not able to characterize the host-virus interactions with high accuracy.

In light of the low precision of models induced by Aleph in these experiments, it may be more informative to examine individual clauses produced by Aleph than to consider the set of clauses as a whole. Most clauses learned for both target concepts contain only literals from the Gene Ontology. These clauses do not give us any particular explanatory advantage; a tool such as the GO::Termfinder [Boyle et al., 2004] would be better used to find enriched GO terms among our positive examples, as it also assesses the statistical significance of shared terms. Other clauses identify entire protein complexes represented by the positive examples for `up` or `down`. Some of these complexes may provide insight into the mechanism of the virus.

In summary, the results of this experiment suggest that it may not be fruitful to use a standard inductive approach to explain the viral replication observations. We conjecture that the ILP approach does not produce high-accuracy models due to (i) the degree of incompleteness in the background knowledge, and (ii) the likelihood that there are many distinct interfaces between the virus and the host cell.

Table 2: The ALP task.

- **Given:**
  - A set of observations, e^+, from the positive set of `down`
  - A logical encoding of the known partial network, *B*
  - An abducible predicate representing what is missing from the background information, `interface(X)`
  - A set of clauses for traversing *B* from an observation in e^+ to the abducible predicate
- **Do:**
  - Construct explanations for observations from e^+, using terms from *B*, and ending with grounded hypotheses for `interface(X)`

Table 3: Example explanation produced by ALP.

```
replication(efb1, down) :-
  inComplex(efb1, cyc_121),
  interface(cyc_121).
```

Hypothesizing Explanations using ALP

The goal of our second investigation is to determine if we can account for multiple `down` observations using a single abduced predicate. For this investigation, we focus on the relation `down`, because it is clinically relevant. The predicate that represents the missing piece in our understanding of the relationship between a knockout gene and viral replication is `interface(X)`, the actual host interface with the virus and the final step in an explanation. This interface may be a gene, a protein complex, or a molecule produced during a metabolic reaction. Table 2 describes the task.

An explanation takes the form of chain of relationships leading from a gene knockout to a viral interface. The example in Table 3 shows an explanation generated for the observation that viral replication is inhibited in the EFB1 knockout. In this case, the explanation is that the gene encodes a protein that is in the Complex 121 from the Cyc2008 database, and that complex is the viral interface. In EFB1-knockouts, the production of complex 121 may be prevented, thus suppressing the replication of the virus.

Background Knowledge and Model

As we would like to construct a specific, causal story for each observation, we limit our background information to terms that describe causal relationships between genes, complexes, and metabolites. We include the following in our background relations: protein complex membership, physical interactions, genetic interactions from expression profiles,

post-translational modifications of transcription factors, and metabolic pathway steps. We exclude genetic interactions for which we do not know the direction of the relationship. Similarly, we exclude GO annotations, which do not necessarily capture direct relationships between genes. Additionally, much of relevant information from the Cellular Component subontology should be redundant with the protein complex data.

We supply a simple logical model of the potential interactions between a gene and an interface. These clauses dictate how interactions from the background data may chain together to form explanations. Within these clauses, we enforce consistent behavior among genes in an explanation. For example, if we are tracing the path from a gene to the interface, no gene that appears along the path (including the final interface, if it is a gene) should belong to the set of up genes. Additionally, with respect to genetic interactions from expression profiles, we only allow those in which one knockout results in a decrease in the expression of another gene. The clauses are presented in Table 4.

Table 4: Background model used in ALP experiments.

The observed gene itself may be an interface, or the gene may be directly related to an interface.

```
replication(G, down) :-
  interface(G).
direct(A,B) :-
  tf_ptm(A,_,B), A\==B,
  not replication(B, up).
direct(A,B) :-
  tf_ptm(_,A,B), A\==B,
  not replication(B, up).
direct(A,B) :-
  upRegulates(A, B, down), A\==B,
  not replication(B, up).
direct(A,B) :-
  physical(A,B), A\=B,
  not replication(B, up).
direct(A,B) :-
  inModule(A,B).
direct(A,B) :-
  inComplex(A,B).
```

To chain multiple entities in an explanation:

```
replication(G,down) :-
  connect(G,A), interface(A).
connect(A,B) :-
  direct(A,B).
connect(A,B) :-
  not direct(A,B), direct(A,C),
  connect(C,B).
```

A special case. Pathway relationships are only causal within the context of a metabolic pathway. Once our explanation enters a pathway, all downstream entities in the explanation must be steps forward along that pathway.

```
connect(A,B) :-
  connectPathOnly(A,B).
connectPathOnly(A,B) :-
  directPath(A,B).
connectPathOnly(A,B) :-
  not directPath(A,B), directPath(A,C),
  connectPathOnly(C,B).

directPath(A,B) :-
  pathForward(A,B), A\==B,
  not replication(B, up).
```

Methods

We use ProLogICA [Ray and Kakas, 2006] to perform abduction, generating all possible explanations for each observation. We have added a slight modification to the source code so that ProLogICA outputs the grounded intermediate goals satisfied in the process of abducing facts. We then process this output into a set of coherent clauses in the form of the one shown in Table 3.

We organize the explanations based on their shared components, or tails, using a process related to the *A Priori* algorithm for association rules [Agrawal and Srikant, 1994]. The *support* of a shared tail is the number of distinct observations found in explanations sharing that tail. We refer to a set of explanations sharing a tail as an *explanation group*. An example explanation group is shown in Table 5. The output of the ALP process, then, is a set of these explanation groups. We assess the power of this approach to explain multiple genes under the same explanation. Again, we use permutation tests to determine whether we cover significantly more genes under high-coverage explanation groups as we can with explanation groups learned on randomly labelled data. We score a learned set of explanation groups (all groups generated from a set of observations) with its gene coverage. Here, we define *coverage* as the number of genes that are covered by an explanation group that covers at least a minimum number of total genes. That minimum number we call *support*.

For each of the 100 permutation tests, we randomly draw 88 genes from our observation pool to label as up, and 122 genes to label as down. For these 100 partitions, we run the ALP process as described above to acquire a set of explanation groups. We score the set of explanation groups for several values of minimum support.

Results

At a maximum clause length of five, ProLogICA constructs 19,154 explanations for the 122 positive examples of down. 85 observations generate more than one explanation. (All observations generate at least a trivial explanation - that the

Table 5: An explanation group sharing the tail `interface(ole1)`, which covers five observations.

```
replication(pre1,down):-
  physical(pre1,rpt4),
  upRegulates(rpt4,ole1,down), interface(ole1).
replication(rpt6,down):-
  physical(rpt6,rpn8),
  upRegulates(rpn8,ole1,down), interface(ole1).
replication(rpt6,down):-
  physical(rpt6,rpt4),
  upRegulates(rpt4,ole1,down), interface(ole1).
replication(rpt6,down):-
  physical(rpt6,rpt2),
  upRegulates(rpt2,ole1,down), interface(ole1).
replication(rpt6,down):-
  physical(rpt6,erv25), physical(erv25,ole1),
  interface(ole1).
replication(ubp6,down):-
  physical(ubp6,rpt4),
  upRegulates(rpt4,ole1,down), interface(ole1).
replication(ufd4,down):-
  physical(ufd4,rpt4),
  upRegulates(rpt4,ole1,down), interface(ole1).
replication(yip5,down):-
  physical(yip5,rsp5),
  tf_ptm(rsp5,spt23,ole1), interface(ole1).
```

Table 6: Results from the ALP experiment and permutation tests. For each row, explanations groups covering at least a minimum number of genes are considered; this number is in the Minimum Support column. The Coverage column displays the coverage of the set of explanation groups produced using the actual data. We calculate the p-value to be the number of random partitions with a coverage at or above that of the actual data. For minimum support of 2-13, no random partition scored higher than the actual data.

Minimum Support	**Coverage**	p**-value**
2	106	< 0.01
3	83	< 0.01
4	74	< 0.01
5	67	< 0.01
6	62	< 0.01
7	46	< 0.01
8	39	< 0.01
9	39	< 0.01
10	36	< 0.01
11	33	< 0.01
12	31	< 0.01
13	31	< 0.01
14	22	0.01

knockout gene is the interface.) With 10,769 explanations, the observation for the gene YGL048C/RPT6 generates by far the most. This gene regulates the expression of many other genes, and it has high degree in the background network. The highest number of observations sharing one tail is 14. With respect to explanation groups, we assemble 5,924 groups covering at least two genes; this collection covers 106 genes out of 122. The coverage of explanation groups at other levels of minimum support, as well as the p-values for the permutation tests, are shown in Table 6. For all values of minimum support from 2-14, $p \leq 0.05$. The number of genes we can cover using high-coverage explanation groups is significant at the 95% confidence level. This suggests that our model is capturing information about the host-virus interactions, rather than other, irrelevant interactions between genes.

Example Explanation Groups

We partially recover a subnetwork of interactions previously identified by the Ahlquist group. This subnetwork indicates that OLE1's involvement in lipid metabolism is important to BMV replication. The underlying explanation, as described by Gancarz and Ahlquist (2008), is as follows: RSP5 tags transcription factor SPT23 for activation by the proteasome. The proteasome activates SPT23, allowing SPT23 to enter the nucleus and activate the transcription of OLE1.

Our OLE1 explanation group tells part of the same story. Table 5 shows the explanation group sharing the tail `interface(ole1)`, while Figure 1 depicts the group graphically. RSP5 modifies SPT23, which activates transcription of OLE1. Several genes in the proteasome (RPT4, RPT2, RPN8) up-regulate OLE1. While we do not have observations for the effects of these genes on viral replication, they physically interact with other genes in the proteasome (RPT6, PRE1, UBP6, UFD4) that are examples of the `down` relation. Additionally, the explanation suggests that the decrease in viral replication in the YIP5 knockout may be related to its physical interaction with RSP5. It also suggests another path from the proteasome to OLE1: RPT6 has a physical interaction with ERV25, a gene with inconclusive inhibitory effect on the virus. ERV25 in turn interacts with OLE1. It is worth noting that these explanations highlight the incompleteness of our background network. Genetic interactions from expression profiles relate the proteasome to OLE1 in one step, rather than through SPT23.

As noted previously, some clauses produced by Aleph during our ILP experiment suggest complexes or genes to further investigate. However, our ALP approach may produce richer explanations than ILP. We consider, for example, the YHPT_3 complex, which contains 40 genes and is involved in transcription from the promoters of RNA polymerases I, II, and III. While the YHPT_3 complex appears in both an ILP clause and in an ALP explanation group, the latter provides a larger context.

Figure 2 provides a graphical representation of an ALP explanation group for the abducible `interface(yhpt_3)`. This group was learned under a maximum clause length of four. ILP learned the clause `replication(A,down):-inComplex(A, yhpt_3)`, which covers the knockouts for RPB3, RPA34, RPC19, SPT4, RPC19, and DST1. While

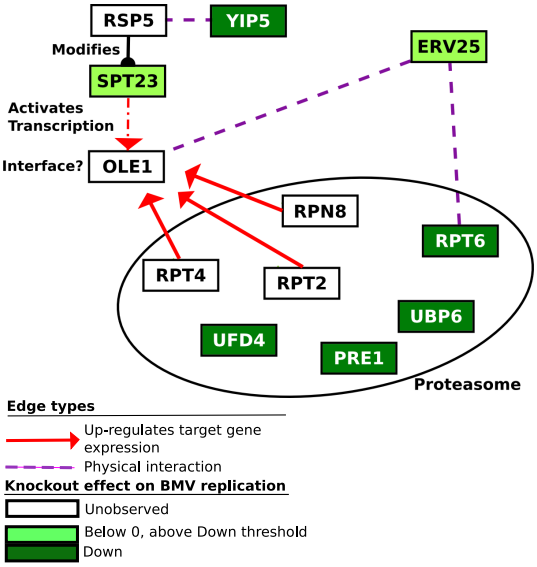

Figure 1: Graphical representation of the explanations with the tail `interface(ole1)`. Five observations (YIP5, RPT6, UBP6, PRE1, UFD4) are covered by this interface. The genes involved in the proteasome complex are outlined by an ellipse. For clarity, we do not depict all of the physical interactions among the constituents of the proteasome. It should be noted that the proteasome comprises more genes than are pictured here; this image only contains those genes that appear in the explanations produced by ALP.

both representations cover the same complex, the explanation group also indicates that the knockouts for RPT6 and DOP1 may inhibit viral replication by inhibiting the expression of the genes RPB5, RPB10, RPA43, and RPA34 in the YHPT_3 complex.

Discussion

In summary, we investigated the application of ILP and ALP to our observations of yeast knockout effect on BMV replication. Our inductive approach was unable to learn general models to characterize virus-host factor interactions with high precision. Using ALP, we abduced explanation groups linking multiple host genes to the same interface with the virus. Based on the results of permutation tests, it appears these groups capture information about virus-host interactions. We were also able to recover an explanation group previously identified by the Ahlquist group.

Our investigation thus far has suggested many ideas for future work, both in application and algorithm development. As was highlighted by the OLE1 situation, it would be worthwhile to supplement our current background information with other types of intracellular relationships. For example, we may integrate additional transcription factors and post-translational modifications. It will be necessary to further develop the ALP algorithm, improving the method for ordering and grouping explanations. We also plan to investigate the possibility of integrating other data sources into the scoring of possible explanations: for example, genetic interaction

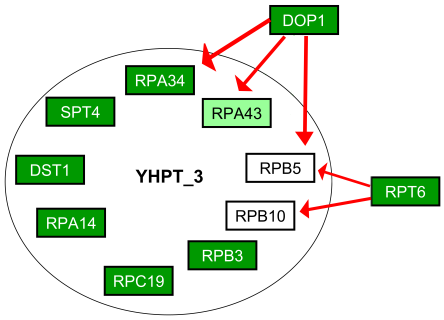

Figure 2: Graphical representation of explanations for eight `down` genes (RPB3, RPA34, RPC19, SPT4, RPC19, DST1, DOP1, RPT6) all sharing the tail `interface(yhpt_3)`. YHPT_3 contains 40 genes; only genes involved in the ALP explanation group learned with a maximum clause length of four are depicted here.

data and the quantitative measurement of a knockout's effect on viral replication. Much current research focuses on the integration of ALP and ILP. It is possible that by running ILP again on our background knowledge, plus the abduced host-virus interfaces, we may hypothesize additional clauses relating a knockout gene to a proposed interface. These clauses may be used supplement the model used in ALP. Lastly, we hope in the future to investigate our explanation groups with wet-lab experiments. We envision an exchange between the computation of hypotheses and the experimental investigation thereof; however, we currently have no official plans.

Acknowledgements

This work is supported by NIH/NLM grants T15-LM007359 and R01-LM07050.

References

[Agrawal and Srikant, 1994] Agrawal, R., and Srikant, R. 1994. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases*, Morgan Kaufmann Publishers Inc., 487–499.

[Ashburner et al., 2000] Ashburner, M., et al. 2000. Gene Ontology: tool for the unification of biology. *Nature Genetics* 25(1):25–29.

[Boyle et al., 2004] Boyle, E. I., et al. 2004. GO::TermFinder–open source software for accessing gene ontology information and finding significantly enriched gene ontology terms associated with a list of genes. *Bioinformatics* 20(18):3710–3715.

[Christie et al., 2004] Christie, K., et al. 2004. Saccharomyces genome database (SGD) provides tools to identify and analyze sequences from Saccharomyces cerevisiae and related sequences from other organisms. *Nucleic Acids Research* 32:D311–D314.

[Everett et al., 2008] Everett, L., et al. 2008. PTM-Switchboard–a database of posttranslational modifications of transcription factors, the mediating enzymes and target genes. *Nucleic Acids Research* 37:D66–D71.

[Gancarz and Ahlquist, 2008] Gancarz, B., and Ahlquist, P. 2008. Systematic identification of essential host genes affecting Brome Mosaic Virus RNA replication and gene expression. Poster.

[Herrgård et al., 2008] Herrgård, M. J., et al. 2008. A consensus yeast metabolic network reconstruction obtained from a community approach to systems biology. *Nature Biotechnology* 26(10):1155–1160.

[Hughes et al., 2000] Hughes, T. R., et al. 2000. Functional discovery via a compendium of expression profiles. *Cell* 102(1):109–126.

[Kushner et al., 2003] Kushner, D. B.; Lindenbach, B. D.; Grdzelishvili, V. Z.; Noueiry, A. O.; Paul, S. M.; and Ahlquist, P. 2003. Systematic, genome-wide identification of host genes affecting replication of a positive-strand RNA virus. *Proceedings of the National Academy of Sciences of the United States of America* 100(26):15764–15769.

[Mnaimneh et al., 2004] Mnaimneh, S., et al. 2004. Exploration of essential gene functions via titratable promoter alleles. *Cell* 118(1):31–44.

[Ong et al., 2007] Ong, I. M.; Topper, S. E.; Page, D.; and Santos Costa, V. 2007. Inferring regulatory networks from time series expression data and relational data via inductive logic programming. In *Proceedings of the Sixteenth International Conference on Inductive Logic Programming*, Springer Lecture Notes in Artificial Intelligence. 4455:366-378.

[Pu et al., 2007] Pu, S., et al. 2007. Identifying functional modules in the physical interactome of Saccharomyces cerevisiae. *Proteomics* 7(6):944–960.

[Pu et al., 2008] Pu, S., et al. 2009. Up-to-date catalogues of yeast protein complexes. *Nucleic Acids Research* 37(3):825-31.

[Ray and Kakas, 2006] Ray, O., and Kakas, A. 2006. Prologica: a practical system for abductive logic programming. In *Proceedings of the 11th International Workshop on Non-monotonic Reasoning*.

[Reiser et al., 2001] Reiser, P. G. K.; King, R. D.; Kell, D. B.; Muggleton, S. H.; Bryant, C. H.; and Oliver, S. 2001. Developing a logical model of yeast metabolism. *Electronic Transactions Articles in Artificial Intelligence* 6.

[Srinivasan, 2007] Srinivasan, A. 2007. *The Aleph Manual*. http://www.comlab.ox.ac.uk/activities/machinelearning/Aleph/aleph.html

[Stark et al., 2006] Stark, C., et al. 2006. BioGRID: a general repository for interaction datasets. *Nucleic Acids Research* 34:D535-D539.

[Tamaddoni-Nezhad et al., 2006] Tamaddoni-Nezhad, A.; Chaleil, R.; Kakas, A.; and Muggleton, S. 2006. Application of abductive ILP to learning metabolic network inhibition from temporal data. *Machine Learning* 64(1-3):209–230.

[Ulitsky et al., 2008] Ulitsky, I., et al. 2008. From E-MAPS to module maps: dissecting quantitative genetic interactions using physical interactions. *Molecular Systems Biology* 4:209-221.

[Winzeler et al., 1999] Winzeler, E., et al. 1999. Functional characterization of the Saccharomyces cerevisiae genome by gene deletion and parallel analysis. *Science* 285:901–906.

The Role of Openness in Scientific Automation: a case for Open Notebook Science

Jean-Claude Bradley
Department of Chemistry
Drexel University
bradlejc@drexel.edu

The use of Open Notebook Science to collect and make publicly available non-aqueous solubility measurements and the synthesis of anti-malarial agents will be described. ONS involves the real time sharing of all experiments and associated raw data by a community of collaborators who are geographically distributed and may have never communicated using channels other than theses shared projects. Monthly cash prizes are awarded to participating students by means of the ONS Challenge Submeta Awards for solubility measurements. The laboratory notebook pages are recorded on a public wiki and the solubility measurements, including relevant calculations, are stored in public Google Spreadsheets. A combination of ChemSpider, the GoogleDoc visualization API and web services is used to enable flexible searching and display of desired subsets of the data.

The use of such a distributed and open platform with virtually zero read/write costs for the communication of science creates new opportunities for rapid collaboration. By using a redundant information dissemination system, channels that are more human friendly can be integrated with those that are more geared to machine readability. For example a publicly editable Google Spreadsheet tied to the operation of a robotic liquid handling system opens up the possibility of integrating crowdsourced intelligence with human workflows. In another example, web services called from within a publicly editable Google Spreadsheet to perform calculations on NMR spectra can be integrated readily with manually executed steps to accelerate progress and minimize the possibility of errors.

The advantages and disadvantages of ONS and related bottom-up Open Science strategies will be discussed. The key concerns revolve around intellectual property, trust, reference-ability, publication in traditional academic vehicles and other implications for collaborations.

Knowledge Evolution in Geologic Mapping

Boyan Brodaric
Geological Survey of Canada
brodaric@nrcan.gc.ca

Abstract

To support online geologic research this paper proposes an informal model for geologic knowledge evolution that involves the interaction of induction, abduction and deduction. It also presents empirical results that support the model, and it adapts a foundational ontology for science knowledge to represent the model.

1 Introduction

It is becoming increasingly important to better understand the geologic knowledge evolution process, to facilitate online knowledge representation and reasoning in rapidly emerging e-Science environments. For example, in order to help choose amongst several competing online geologic maps in some decision-making scenario, such as nuclear waste site selection or natural hazard risk assessment, it is vital to know the underlying data and reasoning mechanisms used to determine the rock bodies depicted on each map. These rock bodies, as well as the categories used to classify them and their associated causal histories, are simultaneously inferred during a mapping campaign: the rock bodies are inferred because they are often too large and insufficiently exposed to detect as a whole; the histories are inferred because their components exist in the past; and the categories are inferred because regional variations in the Earth's dynamics and environment cause distinct rock body categories to be instantiated repeatedly within one region, but not necessarily in another region, requiring new categories to be discovered and old categories refined in every new mapping campaign. Understanding the rock bodies thus requires comprehension of their inference history.

While the general geoscience reasoning process has been examined extensively, including the role of induction, abduction, and deduction [Engelhardt & Zimmerman, 1982] as well as the general nature of geoscience theory evolution [Thagard & Nowak, 1990], and while many data schema are emerging for geologic mapping data, existing work on formal reasoning and knowledge representation does not specifically address theory development in geologic mapping [Voisard, 1999], and there exist almost no supporting empirical studies. This paper addresses these shortcomings in the following ways:

- Section 2 proposes an informal model for knowledge evolution in geological mapping, in which induction, abduction, and deduction underpin scientific model and theory development;
- Section 3 recasts previous empirical results on knowledge evolution in geologic mapping in terms of the knowledge model;
- Section 4 adapts a foundational ontology for scientific knowledge to the case of geologic mapping, as a demonstration of a machine representation for the knowledge model; and finally, Section 5 provides some concluding statements.

2 The Geologic Mapping Knowledge Model

Knowledge evolution in geologic mapping can be structured around the induction, abduction and deduction inference types [adapted from Sowa, 2006], and two key distinctions: the distinction between categories and individuals (i.e. types and tokens), and the distinction between entities existing at spatial or temporal scales than can be wholly observed by an agent, and those that cannot. As shown in Figure 1, geologists induce prototypes (categories) from observable individuals, including the internal characteristics of individual rock bodies, such as their exposed parts (e.g. at the surface), boundaries (e.g. faults), qualities (e.g. shape, color), and constituents (e.g. rock types, minerals), as well as from observed spatial relations between bodies (e.g. above, beside)—induction is at play here because these prototypes are generalizations of observed repeated patterns. A prototype for the whole rock body object is abduced from these prototypical internal and relational characteristics, plus prior knowledge—abduction is at play here because: (1) the development of a prototypical causal history for the prototypical object involves reasoning from effect to cause, and (2) the development of a description for the prototypical object additionally involves 'best-guess' hypothesis generation involving the arrangement and weighting of specific prototypical internal and relational characteristics. The full description for the prototypical object then constitutes a necessary and sufficient condition that applies locally within the geographic area of the mapping. This in turn enables further

observable individuals within the area to be deductively classified as being associated with an instance of a prototypical object—deduction is at play here because if the observables in the area satisfy the prototypical object's necessary and sufficient condition, then the observables are associated with an instance of the object. Finally, once enough observables are classified in this way, then rock body object individuals can be fully identified—abduction is at play here because a 'best-guess' is used to fill in missing information about each rock body, such as its boundaries, to enable its differentiation from others. This overall process is cyclic because a new inference at any point can trigger inferences and related revisions at other points: for example, the induction of a new observable prototype could lead to its addition to a prototypical object, and perhaps to the re-classification and revision of individuals that stimulate new observations.

In cartographic terms, identification often amounts to the development of polygons or volumes denoting the boundaries of individuals, while the categories are the entities symbolized in the map legend. In logical terms, the categories in the legend form a theory (and ontology) that is satisfied by the individuals which form a model. These notions also help distinguish this work from related efforts that focus strictly on inference in model development [Voisard, 1999], i.e. amongst individuals, exclusive of theory development.

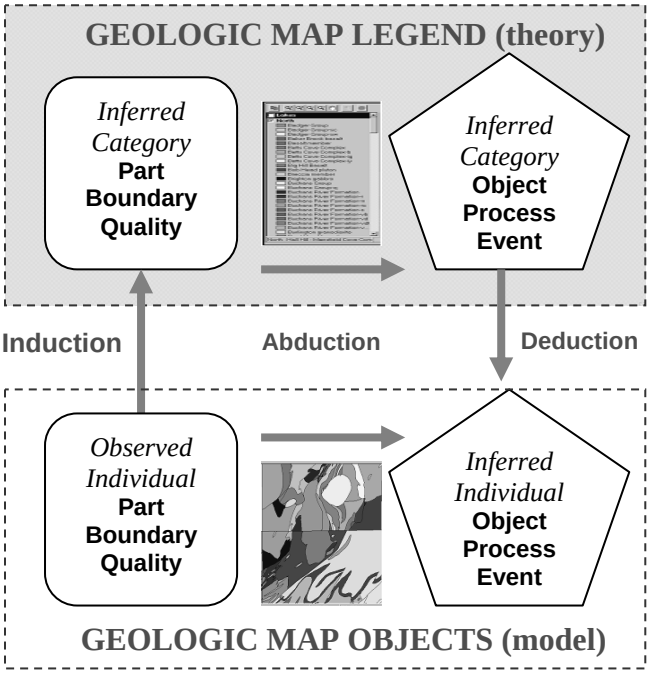

Figure 1: the geologic mapping knowledge evolution model.

3 Geologic Knowledge Evolution Case Study

In previous work an empirical study of the geologic mapping knowledge cycle revealed several trends in geologic mapping knowledge development [Brodaric & Gahegan, 2006]. These trends are briefly summarized here and discussed in relation to the stages of the knowledge model and their associated inference mechanisms.

The previous study compared geologists' observations to each other and to the final 20 prototypical object categories developed during a multi-year mapping campaign. These categories denote varieties of geologic formations, and the surface exposure of their rock body instances are drawn as polygons on the map. The geologists' initial observations were digitally recorded on-site, in the field, using handheld computers configured to ensure the geologists used common data structures and vocabularies during data collection. The geologists also conferred regularly to maximize shared understandings. In subsequent analysis, the observations were projected as points into a multi-dimensional space whose axes were denoted by internal rock characteristics. The clusters of data points for each category were compared at weekly intervals in three ways, using a Euclidean similarity metric within median-neighbor statistics: (1) between pairs of geologists, (2) between a single geologist's newly collected weekly data and all his/her previous data (Figure 2), and (3) between each geologist's data and the prototypical object description (Figures 3 and 4). Detailed analysis of two categories ("B", "C") yielded highly pertinent results:

- variously similar data were collected by the geologists for each category over time;
- at least one category ("C"), described as constituted by metamorphic rocks, was driven by data rather than other knowledge, for some geologists: this is indicated by Figure 3 where the distance of the data cluster to the prototype object is small at each weekly interval, and by Figure 2 where the size of the data cluster is larger than the distance to the prototype object. This suggests the prototype is situated at all times within the clusters of data and that prototype development is synchronized with data collection;
- at least one category ("B"), described as constituted by igneous plutonic rocks, is driven by other knowledge for all geologists: this is indicated by Figure 4 where the distance between geologists' data clusters and the prototype object is uniformly large over time. This suggests prototype development is likely influenced by theoretical factors, such as causal history.

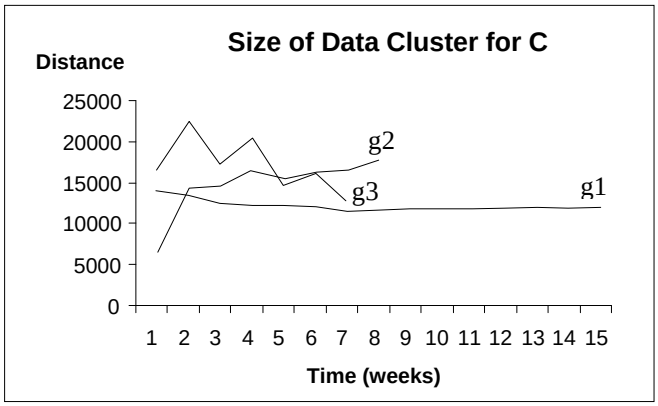

Figure 2: time-indexed comparison of data for each geologist (g_i).

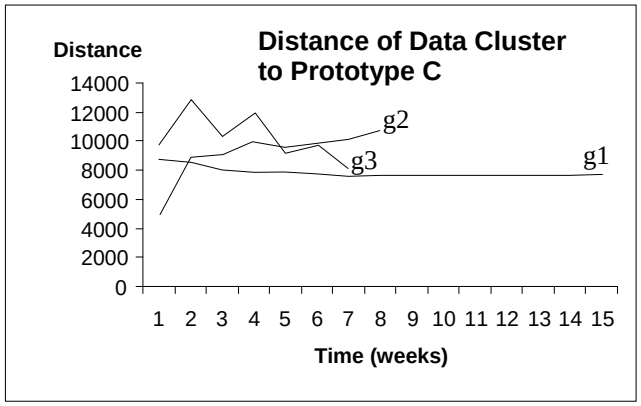

Figure 3: time-indexed comparison of each geologist's data (g1, g2, g3) to the prototype object category "C".

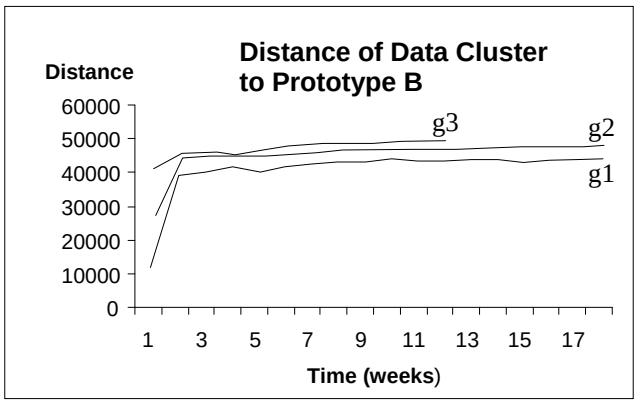

Figure 4: time-indexed comparison of each geologist's data (g1, g2, g3) to the prototype object category "B".

These results provide insights into geologic knowledge evolution that can be related to its stages and inference mechanisms. In particular, data-driven categories would seem to be dominated by the induction stage, inasmuch as the prototypical object category evolves over time in step with generalizations on observed data; the theory abduction stage would thus seem to provide only a grouping function over these generalizations. Conversely, theory-driven categories would seem to be dominated by the category abduction stage, inasmuch as knowledge other than observed data controls category development.

Data-driven categories can also help yield additional insights into the knowledge evolution process: in data-driven categories, knowledge discovery is indicated by sharp spikes in the graphs that compare new and prior data (e.g. Figure 3), because such spikes indicate when a new area of the space is being filled for the category and thus when new induced components are being added to the category description. This not only allows us to pinpoint when knowledge discovery by induction occurs, it also enables examination of the related data to evaluate and perhaps replicate category development. Figure 5 illustrates such inspection—it shows the data points associated with each geologist's knowledge discovery spikes for category "C". Apart from illuminating theory (category) development in data-driven categories, these methods can also help explain related model (individual) development: troughs in the graphs indicate periods when observations are being repeated intensely, typically at different sites, hence perhaps indicating on the one hand verification, when a category is being confirmed, and on the other hand deduction and abduction, when individual objects are being identified.

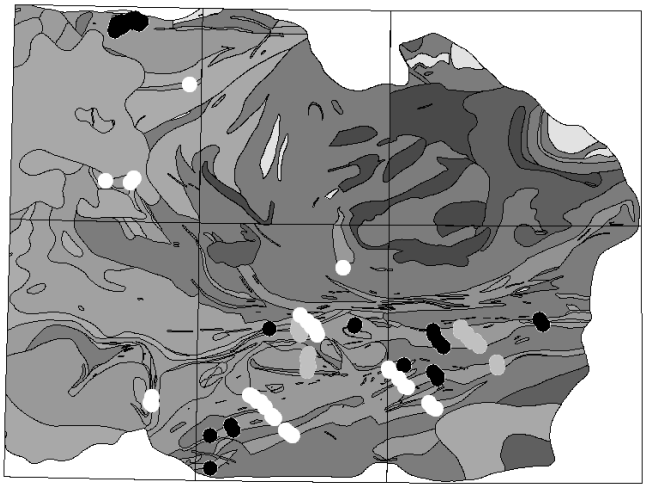

Figure 5: knowledge discovery sites for category "C" per geologist (white, grey, and black dots).

4 An Ontology for (Geo)Science Knowledge

The geoscience knowledge model and the empirical case study both suggest the need for a machine representation of the general science cycle, to help track and evaluate knowledge discovery and evolution. To address this need we have in previous work developed a formal ontology for science knowledge, the Science Knowledge Infrastructure ontology (SKIo) [Brodaric *et al.*, 2008], that specializes the DOLCE foundational ontology [Gangemi *et al.*, 2003]. SKIo is adapted here to the geologic case study as an example representation.

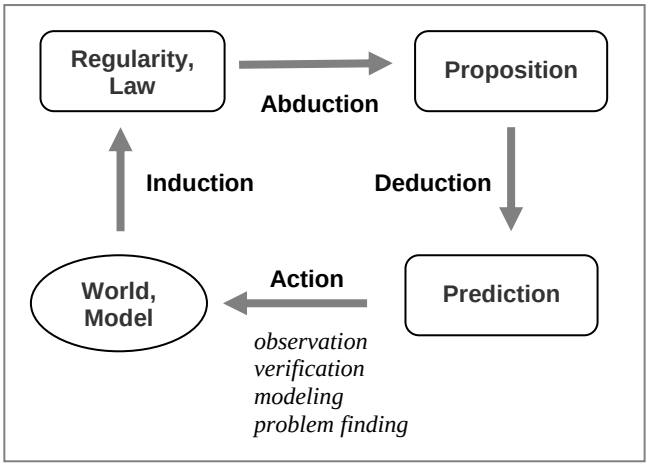

Figure 6: the science knowledge cycle [after Sowa, 2006].

SKIo specializes the DOLCE 2.1 (OWL-DL) foundational ontology with a few key science knowledge primitives that are synthesized from computational descriptions of the science knowledge cycle [mainly Ray, 2005; Sowa, 2006]. These are shown in Figure 6: empirical patterns are induced from observations or models of the world; these patterns are a `Law` if they hold universally (in all possible worlds), or they are an `Empirical Regularity` if they hold locally (in some possible worlds). A hypothetical `Proposition` is abduced from prior knowledge. A `Prediction` about the world can be deduced from empirical patterns or propositions, and these are tested against the world through scientific activities such as verification via observation. Other key scientific activities include problem finding and modeling, as well as reasoning involving `Induction`, `Abduction` and `Deduction`.

The DOLCE foundational ontology provides a rich suite of categories from which the science knowledge primitives can be specialized. An `Endurant` denotes an object-like entity such as a `Physical-Object` and the `Amount-Of-Matter` constituting it. A `Perdurant` denotes a process-like entity such as an `Event` or `Activity`. A `Quality` is a dependent characteristic such as shape or size, and a separate module denotes social artefacts such as a `Situation`, `Description`, `Theory`, or `Concept`.

As shown in Figure 7, these foundational categories anchor SKIo: science activities specialize `Activity` where each activity produces some science artefact, e.g. `Abduction` produces a `Proposition`. `ScienceTheory` specializes `Theory`, which refers to a single science idea or their collection, produced by an agent. Each `ScienceTheory` or its part can also be expressed as an information-object, such as a `SciencePublication`, and can be satisfied by some state-of-affairs situation, such as a `ScienceModel`. Satisfaction implies each individual in a `ScienceModel`, such as a rock body, must be deductively predictable from a `ScienceTheory`. Also, each part of a `ScienceTheory` plays some role in the theory, and this role can change from one theory to the next; for example a `Proposition` in one theory might be an `Assumption` in a subsequent theory: Einstein's proposed theory of special relativity is assumed in general relativity.

Figure 7 also shows how a specific prototype object category from the case study ("A") can be represented in SKIo:

- the map legend, which contains the prototype object descriptions, is an instance of `GeoScienceTheory`, and each prototype object description such as `Unit A` is an instance of a theory part, which plays the role of a `Proposition` within the theory;
- each `Proposition` is produced by an `Abduction` which is inferred from an observable prototype, and each observable prototype is itself induced from `ObservedData` produced by an `Observation`. Each prototype object `Proposition` also has parts which consist of observable prototypes, such as `Unit A Proposition` consisting of `Unit A Volcanic Rock Regularity`;
- each observed individual is a `GeoscienceObject`, either a `Perdurant`, `Endurant`, or `Quality`, and each inferred individual object is discovered by a `Proposition`, classified by an object category, and is part of a model which satisfies a `GeoScienceTheory`; e.g. the `Rock body #1` object is abduced from a proposition, classified by `Unit A Proposition`, is part of the `Macquoid Model` satisfying the `MacQuoid Legend`., and is constituted by an amount of matter such as `Basalt #1.`

5 Conclusions

This paper proposes an informal model for geologic mapping knowledge evolution that involves interacting reasoning components for induction, abduction and deduction. It also presents supporting empirical results, and adapts a foundational ontology for science knowledge to illustrate a formal representation. Outstanding questions include issues related to improved understanding of how the inference mechanisms are deployed in practice, and issues related to the utility and complexity of the representation.

References

[Brodaric & Gahegan, 2006] Brodaric, B., Gahegan, M. Experiments to examine the situated nature of geoscientific concepts. Spatial Cognition and Comp., 7(1):61-95.

[Brodaric *et al*., 2008] Brodaric, B., Reitsma, F. Qiang, Y. (2008) SKIing with DOLCE: toward an e-Science Knowledge Infrastructure. In: Eschenbach, C., Gruninger, M., (Eds.) Formal Ontology in Information Systems, Proc. of the 5th Int'l Conf., IOS Press, 208-219.

[Engelhardt & Zimmerman, 1982] Engelhardt, W., Zimmerman, J. (1982) Theory of Earth Science. New York, Cambridge, 381 pp.

[Gangemi *et al*., 2003] Gangemi, A., Guarino, N., Masolo C., Oltramari, A. (2003). Sweetening WordNet with DOLCE, AI Magazine, 24(3):13-24.

[Ray, 2005] Ray, O. (2005). Hybrid Abductive Inductive Learning. PhD Thesis, Imperial College, University of London, http://www.bcs.org/upload/pdf/oray.pdf.

[Sowa, 2006] Sowa, J. F. (2006). The Challenge of Knowledge Soup. In: J. Ramadas, S. Chunawala, (Eds.), Research Trends in Science, Technology and Mathematics Education, Homi Bhabha Centre, Mumbai, 2006, 55-90.

[Thagard & Nowak, 1990] Thagard, P. Nowak, G. (1990). The conceptual structure of the geologic revolution. In J. Schrager & P. Langley (Ed.), Computational Models of Scientific Discovery and Theory Formation. Morgan Kaufman, San Mateo.

[Voisard, 1999] Voisard, A. (1999). Abduction and deduction in geologic hypermaps. In: Güting, R.H., Papadias, D., Lochovsky, F. (Eds.), Advances in spatial databases: SSD'99, Hong Kong, China, July 20-23, 1999, LCNS, 1651, Berlin, Springer, 311-329.

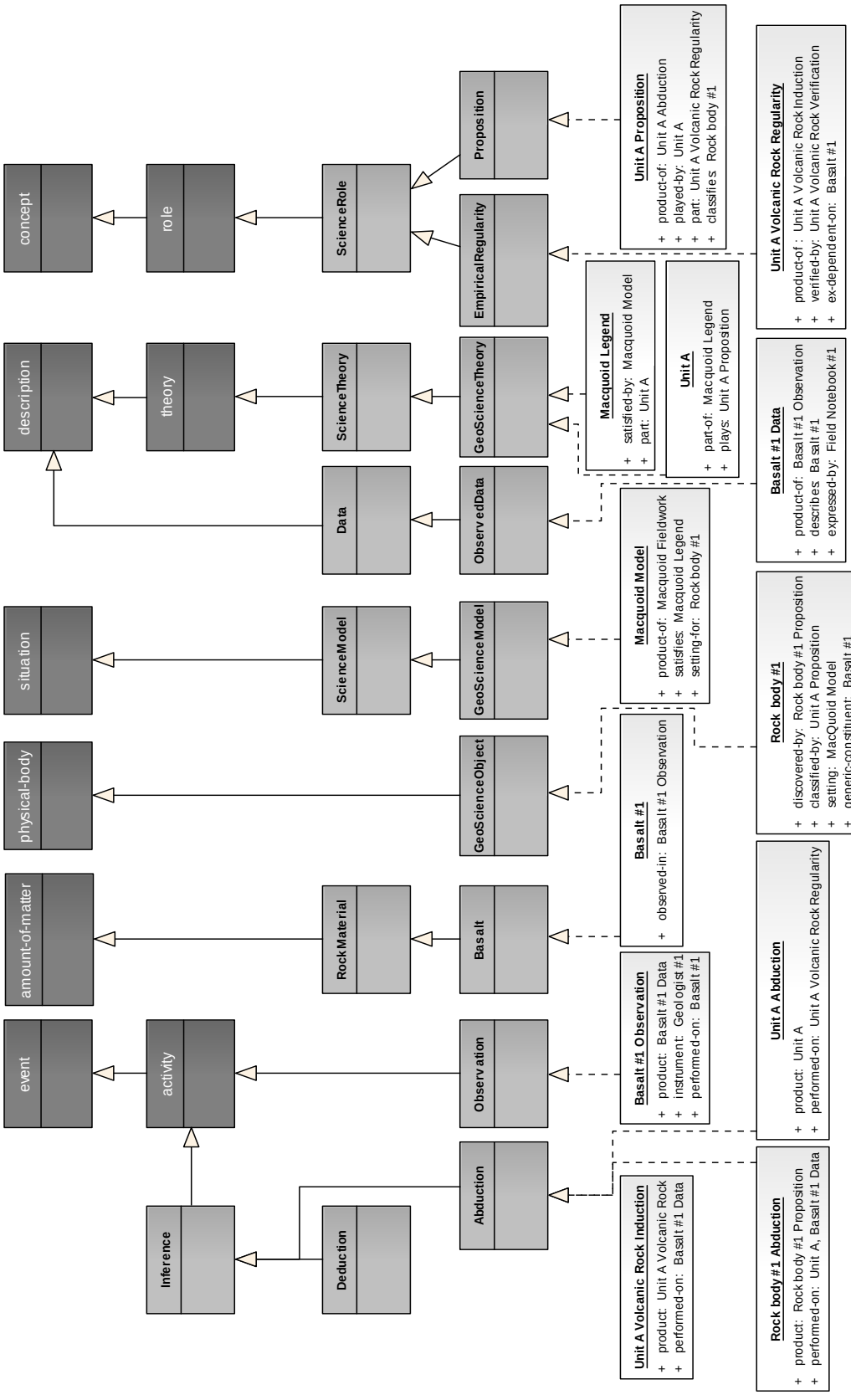

Figure 7: representing the geologic mapping knowledge model in SKIo; DOLCE categories are in dark grey at the top.

The Reasoning Process Underlying Claude Bernard's Scientific Discoveries

Bassel HABIB
LIP6, University of Paris 6
Paris, France
bassel.habib@lip6.fr

Jean-Gabriel GANASCIA
LIP6, University of Paris 6
Paris, France
jean-gabriel.ganascia@lip6.fr

Abstract

This paper presents an attempt to rationally reconstruct the discovery process in medicine. Our aim is to rebuild Claude Bernard's intellectual pathway leading to important discoveries, in particular his understanding of the effects of curare. Achieved in collaboration with epistemologists, we refer to the notebooks where Bernard recorded his experiments. Based on this material, this paper presents a computational model of Bernard's activity. Our study shows that Bernard did not only used induction, but also deduction and abduction. Deduction anticipates the consequences of working hypotheses; experiments attempt to confirm or to infirm those hypotheses; then abduction generates new hypotheses that explaine unexpected observations. We focus on the deductive part of this process, with a virtual laboratory which allows the construction of virtual experiments associated with different working hypotheses. Then, we show how this deductive part is related to the abductive and the inductive steps in the reasoning process about Claude Bernard's scientific discoveries.

1 Introduction

In the past, there have been many attempts to rationally reconstruct scientific discoveries with Artificial Intelligence techniques [Feigenbaum *et al.*, 1971; Langley *et al.*, 1986; Shrager and Langley, 1990; Kulkarani and Simon, 1988]. According to Herbert Simon, creativity, which is involved in the discovery process, is akin to the manner in which we find our pathway in a labyrinth [Simon, 1957; 1983]. From a technical point of view, creative behavior can be seen as a graph search. Even if this view is efficient and fruitful from a practical point of view, it does not tell anything concerning the logical status of the scientific discovery process. Is it mainly an inductive, a deductive on an abductive process? Epistemologists do not agree in this point; but whatever their underlying theories, it appears that many different kind of inferences are involved in scientific discovery. Nevertheless, up to now, most of the simulations of scientific discovery processes that have been achieved in Artificial Intelligence correspond to the simulation of inductive processes [Corruble and Ganascia, 1994]. Moreover, today, Knowledge Discovery from Data Bases corresponds naturally to an inductive process, since it builds general knowledge from pieces of information that describe particular cases. Cybernard project [Ganascia and Debru, 2007] constitutes an attempt to reconstruct some of Bernard's scientific steps that are mainly abductive [Josephson and Josephson, 1996]. It explores with the help of Knowledge Representation and Multi-Agent techniques, some aspects of the discovery process that are not directly related to inductive processes. Our goal is to validate our rational reconstruction with historical knowledge about Bernard's scientific discoveries. But our ultimate goal is to help scientists, especially clinical physicians, to design their experimentations in consideration of the fundamental theory they have in mind. There is also substantial litterature in the field of Science and Technology Studies into which this work could perfectly fit [Kuhn, 1962].

In previous papers, we began the study of the process of scientific discovery [Ganascia and Habib, 2007; Habib and Ganascia, 2008] by implementing a virtual laboratory that is able to anticipate the consequences of an hypothesis. It corresponds to a deductive process. The aim of our present study is to focus on the way this deductive process can be articulated to the abductive process, i.e. to the hypothesis generation.

The paper is organized as follows: Section 2 describes the nature of our epistemological study. In Section 3 we provide a brief overview of Claude Bernard's ontology and how this ontology has been represented. Section 4 is dedicated to the presentation of Bernard's method. We make reference in Section 5 to our virtual laboratory containing core models and meta-operators. Section 6 briefly recalls what is abductive logic programming. Then, we present the model, the agents used in it and how our model has changed to take into account the changes in Claude Bernard's ontology. Next Section describes our results simulating one of Claude Bernard's experiments. Finally the conclusion summarizes our work and what our future directions are.

2 Epistemological Study on Claude Bernard's Manuscripts and Knowledge Represenation

As previously introduced, the focus of our work is on Claude Bernard's discovery about the effects of curare. Our work is based on data gathered from his notebooks and manuscripts

between 1845 and 1875. Since Claude Bernard's manuscripts contain descriptions of experiments in natural language, it was necessary to abstract from these descriptions a number of attributes (experimental criteria), which are rich enough to reflect the complexity of the original descriptions, and sufficiently representative of their variability. An attribute is created if this potential attribute intervenes in a significant proposition of available experiments.

Claude Bernard's manuscripts have been, in a previous study, the subject of an epistemological study, which consists of several steps:

- The transcription of these manuscripts using a text editor. These manuscripts contain experiments using curare or strychnine as a toxic substance;
- The sorting of this work in a chronological order;
- The formalization of an table in which Claude Bernard's experiments are annotated according to several experimental criteria (attributes) such as weight, age, dose, animal, preparation/manipulation, point of insertion, date, ideas of experiments, observations, hypotheses and references.

The identification of the main attributes allowed us to formalize Claude Bernard's experiments. This is a preliminary step to the simulation of these experiments in a virtual laboratory previously built [Habib and Ganascia, 2008]. Prior to the simulation, Claude Bernard's experiments are classified into several sets of experiments. The classification of experiments is done according to one precise criterion; for instance, the set of experiments using dogs as experimental animals, or even the set of experiments including some nerve manipulations, etc. This classification is a methodological problem, because it constitutes an important step in the process of empirical discovery that concerns us, but it is not systematic, and even less, automatic.

Since Claude Bernard does not write down all the details about preparation, observations or even less about the inferred hypotheses, some experiments are not complete comparing to others in the same set of experiments. Hence comes the idea to complete experiments' descriptions by fusing them with descriptions about other experiments from the same set, which are compatible [Laudy *et al.*, 2009].

Fusion allows us, on the one hand, to reduce the number of experiments within a set of experiments and thus, to reduce the number of possible simulations in a particular set of experiments since each experiment may be the object of a simulation. On the other hand, fusion allows to complete descriptions about some experiments with information of a great interest in our reasoning process. After the fusion step, information includes not only the complete set of observations resulting from experiment but also the hypotheses inferred by Claude Bernard.

3 Claude Bernard's Ontology

According to his writings (manuscripts, notebooks, etc.), we suppose that Claude Bernard had in mind an ontology upon which he generated all his experiments. His ontology has changed gradually during more than twenty years of his scientific career. Claude Bernard's ontology is considered as the core of his knowledge base upon which he constructed his reasoning process.

Since Claude Bernard did not always explicitly describe his ontology, many ontologies can be derived by studying his experiments at different points of his career. Then, it is easy to formulate it using an ontology description language similar to those that are nowadays used in life sciences to represent biological and medical knowledge [Smith and Ceusters, 2006].In his ontology, organs, vessels or even the nervous system are seen as classes. These classes are, in their turn, subcategorized into subclasses, sub-subclasses etc. Each class and subclass has its own characteristics, which can be easily formulated according to Claude Bernard's explanations. He considered that internal environment, mainly the blood, the responsible for exchanges between organs via vessels. Blood carries all organ's aliments and poisons. As a consequence, interactions between blood and one of the organs may have different effects on other organs and, as a result, on the whole organism.

Note that most of ontologies used in the biomedical community, for instance the OBO (Open Biological Ontologies) refers mainly to three levels: one for the organs and the anatomy, the second for the cells and the third for molecules. For obvious reasons Claude Bernard's ontology refers mainly to the first. However, it would be possible to extend our model to a three level ontology that is more appropriate in contemporaneous medicine.

The physiological ontology plays a crucial role in the way Claude Bernard erected new hypotheses. It can be considered as a clue for the discovery process. All scientific hypotheses obviously depend on the concepts with which they may be expressed.

4 Claude Bernard's Experimental Method

This Section recalls Peirce's inferential theory (see [Ray, 2005] for further details) and situates Claude Bernard's scientific approach comparing to it.

The modern classification of logical reasoning into abduction, induction and deduction is due to the American Pragmatist C.S. Peirce (1839-1914) [Peirce, 1931]. Around 1900, Peirce was led to his so-called inferential theory, where abduction and induction are seen as complementary processes cooperating with deduction and experiment in a cycle of scientific knowledge discovery.

As shown in Figure 1, the cycle usually begins with an anomaly that is not explicable by one's existing knowledge. Some plausible hypothesis must then be sought to account for this fact. This process of hypothesis is what Peirce now calls abduction. Testable predictions must then be extracted that would follow if the hypothesis were true. This process of prediction is the task of deduction. The predictions must then be compared against the result of experiment. Support for the predictions may justify the acceptance of the hypothesis as part of one's growing knowledge, but insufficient support may rule out one hypothesis in favour of another and may result in the discovery of new anomalies in need of further

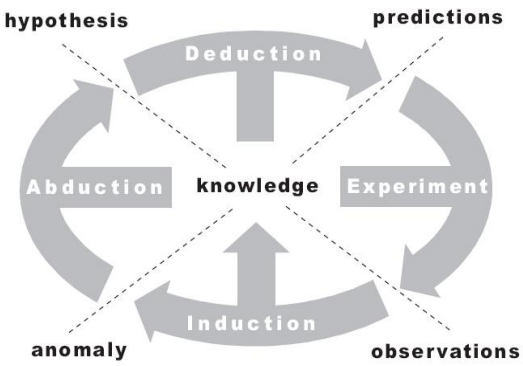

Figure 1: Peirce's 'Inferential Theory'

explanation (thereby invoking a new cycle). This process of evaluation is what Peirce now calls induction.

Claude Bernard's experimental method corresponds, at a grosser or finer level of approximation, to the above described cycle. His experimental method usually begins with an initial hypothesis he had, most of the time, even before some inexplicable anomalies are given. Claude Bernard does not detail all the time the way his initial hypothesis is built. It corresponds to an intuition that has to be validated, refined or adjusted according to empirical results generated by relevant experiments.

Once Claude Bernard has an initial theory, his experimental method begins and it proceeds in three steps, each step involving a specific scientific function:

***Experimentation*:** After considering an initial hypothesis or several initial hypotheses in parallel , Claude Bernard designs an experimental apparatus able to generate observations that can be compared to predictions derived from the current theory. These hypotheses were formulated using the above described *ontology* consisting the core of Claude Beranrd knowledge base. This step corresponds to the experiment step in Peirce's cycle.

***Observation*:** This step consists in collecting observations from the designed experiments that can be compared with predictions derived from his initial hypothesis.

***Analysis*:** The third step is the most crucial and original. it is to confront the predictions extracted from the initial theory to the observations. According to the observational results, the scientist was able to generate, on the one hand, new theories to add to his knowledge base which corresponds to the induction in Peirce's cycle. On the other hand, he generates new working hypotheses from anomalies he might obtain, this step coressponds to the abduction in Peirce's cycle. As a consequence, he reconstructs new experiments validating or invalidating his new working hypotheses.

Claude Bernard's experimental method is an iterative procedure of theory refinement. The role of the induction in his experimental method is limited to the refinement of some thresholds, such as the dose of curare used in paralysis, after repeating the same experiment and adding new thresholds values to his knowledge base. That leaves un with the abduction as the main logical reasoning model used in his scientific approach. We describe by an example how we used Abductive Logic Programming (APL) [Kakas *et al.*, 1992] to find some hypotheses using one of Claude Bernard's experiments.

5 Experiment Design

We have devoted a great part of our work designing our experiment by constructing a virtual laboratory simulating Claude Bernard's experiments which corresponds to the experiment in Peirce's cycle. This virtual laboratory allows us to construct virtual experiments where the input is some attributes formalizing Claude Bernard's experiment and the output is the observations he obtained after doing the experiment.

5.1 The virtual laboratory

In order to construct virtual experiments based on Claude Bernard's notebooks, a virtual laboratory has been built. We will continue to refine it gradually during the project to take into consideration all of Claude Bernard's studied cases. This virtual laboratory has some core models describing the physical architecture of the organism on which the experiments are constructed. It has also many experimental operators, called meta-operators, such as toxic substance injection. The virtual laboratory should contain, as well, models of the configuration of a laboratory, such as instruments for making observations. The simulation of the organ system is done according to Bernard's hypotheses. Observations are the output of each simulation. The choice of both core models and meta-operators depends on Claude Bernard's experiments. Nevertheless, the ontology, on which core models are built, is previously given and evolves very slowly during the Claude Bernard's career.

After building the virtual laboratory, we can choose our own ingredients, from organs to meta-operators, which are needed in our recipe, according to Claude Bernard's scenarios.

Many meta-operators are presented in our virtual laboratory including: toxic substance injection, tourniquet application, interaction with the external medium, substance ingestion and excitation. Here are, in more details, some of these meta-operators used in our simulations showed in the Section devoted to the results:

- ***Toxic substance injection***: In his experiments, Claude Bernard used toxic substances as tools of investigation. He assumed, as underlying principle, that each toxic substance neutralizes the function of one particular organ. He then studied the consequences of organ's dysfunction on other organs and, consequently, on the functionality of the whole organism. Claude Bernard evoked the idea of toxic substances as (chemical scalpel), because they were used to isolate each organ's function. In practice, Claude Bernard took into account the percentage of toxic substance injected and where to inject it. For instance, he devoted an important time of his experiments to the study of *curare*'s effects as one of these toxic substances.

This operator is presented in the virtual laboratory using the the following predicate:

$$injection(V, [ToxSub, Val], T) \quad (1)$$

- *ToxSub*: the toxic substance injected
- *V*: the vessel in which the toxic substance is injected
- *Val*: the dose of the toxic substance
- *T*: time at which the injection is applied

- ***Interaction with the external medium***: As previously seen, Claude Bernard considered internal environment as a medium of exchanges between organs. But his studies were not focused only on the internal medium but also on external medium, which is the air for outside animals. The fact that external medium may carry aliments, poisons, etc, introduced external medium as a way of exchanges for organisms. As a consequence, changing the nature of the gas breathed (e.g. by adding carbon monoxide) or even carrying artificial respiration may affect the state of the whole organism.

  This operator can take different forms. One of these forms is artificial respiration which can be formalized by the following predicate:

$$artificial\_respiration(T1, T2) \quad (2)$$

- *T1*: time at which the artificial respiration is carried
- *T2*: time at which the artificial respiration is stopped

5.2 The virtual laboratory's implementation

In our model, organs, connections between organs and nervous system components are represented by agents. Agents are represented using automata, each agent has its own inputs, outputs, transfer function and states. Blood is represented by a *list* of blood components and their associated values. These values may be changed according to blood circulation through the organism. Time is discrete and after each period of time, the states of different agents belonging to the core model and their outputs are modified.

Claude Bernard's experiments are represented using a number of attributes. Since his manuscripts contain descriptions in natural language, it was necessary to abstract from these descriptions a number of attributes (experimental criteria). An attribute is created if this potential attribute intervenes in a significant proposition of available experiments.

The implementation makes use of object oriented programming (OOP) techniques. Inheritance and instantiation mechanisms of object oriented programming facilitate the implementation of those agents. It helps both to simulate the "*core model*" evolutions and to conduct virtual experimentations on it, which fully validates our first ideas concerning the viability of the concept of "*core model*".

Within this implementation, organs and connections between organs are associated to objects that implement agents. Organs and connections between organs are instantiations of concepts of the initial ontology. However, since our ultimate goal is to simulate the hypothesis generation and especially the abductive reasoning on which relies the discovery process, we have chosen to build "core models" using logic programming techniques on which it is easy to simulate logical inferences, whatever they are, either deductive or abductive.

The agents are implemented in SWI Prolog. It makes use of modules to emulate object oriented programing techniques, mainly the instantiation, inheritance and message sending mechanisms. The choice of logic programming techniques was motivated by our ultimate goal to simulate the abdutive way of reasoning that explores the hypothesis space.

6 Reasoning: Abductive Logic Programming (ALP) Task

ALP is the field of Artificial Intelligence concerned with finding hypotheses Δ to explain a goal *G* with respect to a theory *T* and integrity constraints *IC*. In brief, the goal *G* is a set of literals to be explained, the theory *T* is a normal program expressing some prior knowledge, and the integrity constraints *IC* are a set of formulae that restrict the acceptable hypotheses. Informally, the explanation Δ is a set of ground atoms that, relative to *T*, 'cover' *G* and are 'consistent' with *IC*. Typically, Δ is restricted to the ground instances *A* of some given set of abducible predicates.

Definition 6.1 An abductive context is a tuple $\langle T, G, IC, A\rangle$ where *T* is a normal program, *G* is a set of literals, *IC* is a set of closed first-order formulae, and *A* is a set of ground atoms.

Definition 6.2 Suppose that *X* = $\langle T, G, IC, A\rangle$ is an abductive context. Then an abductive explanation of *X* is a set of ground atoms $\Delta \subseteq A$ for which there exists a stable model *M* of $T \cup \Delta$ and a ground instance $G\theta$ of *G* such that $M \models G\theta$ and $M \models IC$.

Definition 6.3 Let *X* be an abductive context $\langle T, G, IC, A\rangle$, Δ be an abductive explanation of *X*, and *Y* be the abductive context $\langle T, \Delta, IC, A\rangle$. Then Δ is *minimal* iff there is no $\Delta' \subset \Delta$ such as Δ' is an explantion for *X*; and Δ is *basic* iff there is no explanation Δ' of *Y* such that $\Delta \not\subseteq \Delta'$.

These definitions are taken directly from [Ray, 2005]. A detailed example is given in the next Section devoted to results.

7 Case of Study

The first part of the simulation translates one of Claude Bernard's experiments, concerning the intoxication with the curare, into a virtual experiment. The construction of virtual experiments allows, gradually, the complement of organism's entities with different linked functions. These entities is the subject of the reasoning process.

In his writings [Grmek, 1973], Claude Bernard had an initial hypothesis that he tried to improve by constructing an experiment. Then, he validated or rejected his initial hypothesis according to the observational results of the experiment.

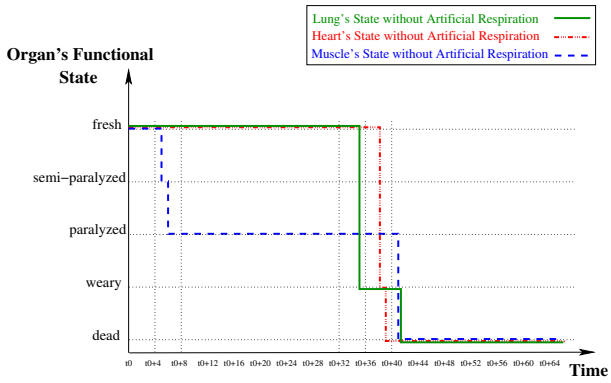

Figure 2: State of different organs in a virtual organism when an important dose of curare is injected and no artificial respiration is carried

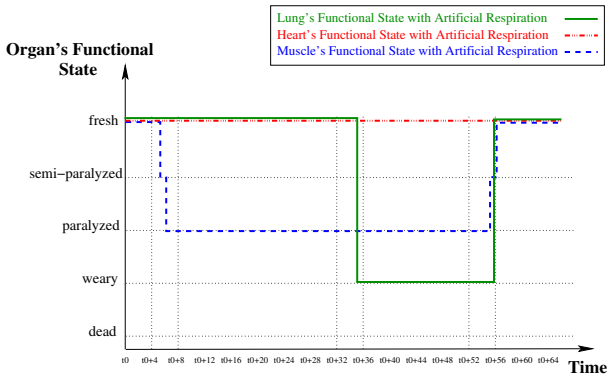

Figure 3: State of different organs in a virtual organism when an important dose of curare is injected and artificial respiration is carried before breathing stops

Here is the first part of an extract of one of his experiments and the corresponding hypothesis taken from his notebooks:

Experiment: *"We introduce under the skin of a frog's thigh a small piece of dry curare. Three minutes later, paralysis occurs. Six minutes later, the nerves, by pinching or by electricity, don't determine any kind of contraction in the muscles. Nine minutes later, the heart stops contracting and the frog died."*

Hypothesis: *"In curare poisoning, voluntary motor nerves are much more quickly extinguished than the nerves of organic life. But when respiratory movements are, themselves, paralyzed, then asphyxia occurs and quickly paralyzes motor nerves of organic life ..."*

The translation of Claude Bernard's experiment into a virtual experiment is illustrated in Figure 2 which shows the state of different organs during the simulation.

According to the observations and knowing that the curare is injected in the animal's blood and its dose is sufficient to paralyze the animal, he gives a hypothesis considering the asphyxia as the main reason about the animal's death. So he repeats the same experimnts using artificial respiration before breathing stops.

The second part of the simulation shows how results change when the scenario proposed by Claude Bernard changes by using artificial respiration.

Here is the complete extract of the experiment previously described and the corresponding hypothesis:

Experiment: *"We introduce under the skin of a frog's thigh a small piece of dry curare. Three minutes later, paralysis occurs. Before breathing stops, we just replace it with artificial respiration. Nine minutes later, the nerves, by pinching or by electricity, don't determine any kind of contraction in the muscles. The heart contracts all alone again after one hour."*

Hypothesis: *"In curare poisoning, voluntary motor nerves are much more quickly extinguished than the nerves of organic life. But when respiratory movements are, themselves, paralyzed, then asphyxia occurs and quickly paralyzes motor nerves of organic life.* ***But if, when breathing stops, we just replace it with artificial respiration, then the nerves of organic life awake while the nerves of animal life will paralyze more.****"*

The translation of Claude Bernard's complete hypothesis into a virtual experiment is illustrated in Figure 3. This figure shows what could happen to organs if artificial respiration is carried before breathing stops and with the same dose of curare used before.

Let us reprensent now the previous example using ALP. Let $X = \langle T, G, IC, A\rangle$ be an abductive context. The theory T contains four clauses describe the domaine. The first rule states that a paralysis occurs when an efficient injcetion takes place and an artificial respiration is carried. The second rule states that for an injection to be efficient, the poison's dose has to be sufficient (comparing to some thresholds) and the injection takes place in the animal's blood. The remaining facts state that the poison's dose is sufficient and is injected in the frog's blood.

$$T = \begin{cases} paralysis(x) \leftarrow effInjection(x), artRes(x) \\ effInjection(x) \leftarrow suffDose(x), inBlood(x) \\ suffDose(frog) \\ inBlood(frog) \end{cases}$$

$$G = \{paralysis(frog)\} \quad (3)$$

$$IC = \{\leftarrow artRes(x), \neg alive(x)\} \quad (4)$$

$$A = \{artRes/1, alive/1\} \quad (5)$$

The abducibles A allow assumptions of the form *artRes(t)* and *alive(t)*, where t is a ground atom; but the integrity constraint *IC* requires that *artRes* can only be carried if the animal is still *alive*. The goal G can be regarded as asking whether there is an explanation of the fact *paralysis(frog)*. With reference to Definitions 6.2 and 6.3, above, it can be shown that

the hypothesis Δ, below, is an abductive explanation of *X* that is both minimal and basic.

$$\Delta = \left\{ \begin{array}{l} artRes(frog) \\ alive(frog) \end{array} \right.$$

8 Conclusion and Future Directions

In this study, a virtual laboratory has been built allowing the construction of virtual experiments associated with different working hypotheses. It was then possible to correlate those virtual experiments to actual experiments done by Claude Bernard. As a consequence, we are able to reconstruct computationally part of Claude Bernard's intellectual pathway. We showed also how we used ALP to reason about Claude Bernard's scientific approach.

To achieve our ultimate goal concerning the rational reconstruction of Claude Bernard's scientific process, we must first complete the construction of our virtual laboratory. To do so, we are working on making a categorization of the experiments according to different core models, and identify some key attributes allowing the formalization of these sets of experiments and then simulating them. This allows the expert, on one hand, to select the virtual organism on which experiments will be conducted, without having to build it again at the beginning of each experiment. On the other hand, he can choose the key attributes necessary for the simulation among theattributes corresponding to a category of experiments.

However, our further research concerns the reasoning process of the formation of Claude Bernard's hypotheses and theories.We want to provide input cases described in Claude Bernard's manuscripts before reaching valid hypotheses. Then, we compare the results of these experiments to validate the algorithm which will complete our model.

We also investigate the possibility of building multi-scale *"core models"* in which physiological behaviors can be studied at different scales, e.g. organ, cell, molecule etc. Today, the effect of new substances is usually studied at the cell or molecule scale, while the organ scale was dominant at Claude Bernard's epoch. A model that could help to simulate the consequences of physiological dysfunctions at different levels would be of great help to determine the effects of new substances by recording different experiments and by ensuring that all the working hypotheses have allready been explored.

References

[Corruble and Ganascia, 1994] V. Corruble and J.-G. Ganascia. *Aid to discovery in medicine using formal induction techniques*. pages 649–659, 1994.

[Feigenbaum *et al.*, 1971] Edward Feigenbaum, Bruce Buchanan, and John Ledeberg. On generality and problem solving : a case study using the dendral program. Edinburgh University Press, Edinburgh, 1971.

[Ganascia and Debru, 2007] Jean-Gabriel Ganascia and Claude Debru. "cybernard: A computational reconstruction of claude bernards scientific discoveries" in *Studies in Computational Intelligence, (SCI), 64*. pages 497–510, 2007.

[Ganascia and Habib, 2007] J-G. Ganascia and B. Habib. *An Attempt to Rebuild C. Bernard's Scientific Steps*. *The Tenth International Conference on Discovery Science (DS-2007)*, 2007.

[Grmek, 1973] Mirko Grmek. *Raisonnement expérimental et recherches toxicologiques chez Claude Bernard*. Droz, Genève, 1973.

[Habib and Ganascia, 2008] B. Habib and J-G. Ganascia. *Using AI to Reconstruct Claude Bernard's Empirical Investigations*. In *The 2008 International Conference on Artificial Intelligence (ICAI'08)*, pages 496–501, Las Vegas, USA, July 2008.

[Josephson and Josephson, 1996] J. Josephson and S. Josephson. *Abductive inference: computation, philosophy, technology*. 1996. Cambridge University Press.

[Kakas *et al.*, 1992] A. C. Kakas, R. A. Kowalski, and F. Toni. *Abductive Logic Programming*. *Journal of Logic and Computation*, pages 719–770, 1992.

[Kuhn, 1962] Thomas Kuhn. *The Structure of Scientific Revolutions*. Chicago: Unversity of Chicago Press, 1962.

[Kulkarani and Simon, 1988] D. Kulkarani and H.-A. Simon. *The processes of scientific discovery: the strategy of experimentation*. pages 139–175, 1988. Cognitive Science.

[Langley *et al.*, 1986] Pat Langley, Jan Zytkow, Herbert Simon, and G. Bradshaw. "the search for regularity : Four aspects of scientific discovery" in *Machine Learning : an Artificial Intelligence Aproach vol II*. 1986.

[Laudy *et al.*, 2009] C. Laudy, B. Habib, and J-G. Ganascia. *Fusion of Claude Bernard's Experiments for Scientific Discovery Reasoning*. In *The 17TH International Conference on Conceptual Structures (ICCS'09)*, Moscow, Russia, July 2009.

[Peirce, 1931] C.S. Peirce. *Collected Papers of Charles Sanders Peirce*. 1931. Harvard University Press.

[Ray, 2005] O. Ray. *Hybrid Abductive Inductive Learning*. PhD thesis, Department of Computing, Imperial College London, 2005.

[Shrager and Langley, 1990] J. Shrager and Pat Langley. *Computational Models of Scientific Discovery and Theory Formation*. Morgan Kaufmann,, San Mateo, California, 1990.

[Simon, 1957] Herbet Simon. *Models of man. Mathematical Essays on Rational Human Behavior in a Social Setting*. John Wiley & Sons, 1957.

[Simon, 1983] Herbert Simon. *Reason in Humans Affairs*. Stanford University Press, Stanford California, 1983.

[Smith and Ceusters, 2006] B. Smith and W. Ceusters. *Ontology as the core Discipline of Biomedical Informatics, Legacies of the Past and Recommendations for the Future Direction of Research*. 2006. Forthcoming in computing, philosophy, and cognitive science.