Abductive Reasoning : Backward and Forward

Philippe Codognet
INRIA - Rocquencourt
B. P. 105, 78 153 Le Chesnay Cedex, FRANCE
Philippe.Codognet@inria.fr

Abstract

We propose to reconsider abductive reasonning in the context of empirical reasonning, and to investigate its deep relations with constraint-based reasonning, a currently very successfull approach used in the problem-solving community.

1 The Roots of Abduction

Abduction has been proposed in the first years of this century by the logician and philosopher C. S. Peirce as a new inference mechanism for logical system along induction and deduction [16].
To quote Peirce himself (Collected Papers, 6.524) :
" The first starting of a hypothesis and the entertaining of it, whether as a simple interrogation or with any degree of confidence, is an inferential step which I propose to call *abduction* (or *retroduction*)." The alternative denomination of "retroduction", that have been first used by Peirce in 1898 and abandonned circa 1901, shows well the fact that the key issue in abduction is to go back from the facts to the hypotheses. But abduction is more than that and it differs form induction because it accounts for a second forward step from the hypotheses to the facts, as Peirce explained (Collected Papers, 6.525) :
"Long before I first classed abduction as an inference it was recognized by logicians that the operation of adopting an explanatory hypothesis – which is just what abduction is – was subject to certain conditions. Namely, the hypothesis cannot be admitted, even as an hypothesis, unless it be supposed that it would account for the facts or some of them. The form of inference, therefore, is this :

- the surprising fact, C, is observed;
- But if A were true, C would be a matter of course,
- Hence, there is reason to suspect that A is true."

It seems however that the premisses of abductive reasoning can be traced back to the XVIIth century in the logical writings of G. W. Leibniz, who was the first to notice the insufficiency of the traditional (Aristotelian) modes of reasoning, either purely deductive (analytic) or purely inductive (synthetic). Louis Couturat, in his study of Leibniz's logic [6], pointed out that Leibniz was trying to define a new mechanism for hypothesis formation that would take from both induction and deduction. Leibniz was indeed interested in defining a very uniform methodology for a universal science that would apply to both exact and experimental sciences. He considered that induction alone cannot be sufficient for explaining new facts or experiments, and could even lead sometimes to wrong conclusions. Hypotheses have to be generated and then tested for their explanation power versus the facts. As an inventor of infinitesimal calculus and *analysis situ*, Leibniz hoped that most of the time hypotheses could be generated by some continuity argument. He distinguished however between exact sciences like Geometry, where the truth of an hypothesis could be shown by "inverse deduction" from the facts and empirical sciences such as Physics and Astronomy where the validity of hypotheses cannot be asserted and where several sets of hypotheses could compete as admissible explanations. Admissibility of an explanation is of course its ability to have the observed facts deduced from it.

One could try to trace back the roots of abductive reasoning even earlier, and further rewind history back for a few centuries. A. C. Crombie proposed in [7] to place the origin of the experimental method in science in the XIIIth century, with in particular the work of the scientist and philosopher Robert Grosseteste. He considered that a new methodology was then defined by joining the scholastic developments of inductive logic and the observation of facts needed in experimental science. This new methodology was clearly beyond the Greek or Arabic models available at that time. Grosseteste stated that " there is a double path from already existing knowledge to new knowledge, that is, from the simple to the complex and conversely", i.e. going from principles to effects and vice versa. Grosseteste considered that for natural sciences, one has to distinguish a possible cause or explanation from others by a verification or falsification procedure. He explicitely said that a theory obtained by induction should make it possible to derive by deduction facts that should generalize those on which the induction was based. This is but a very early form of abductive reasoning.

Abduction is currently used and successful in various domains such as semiotics [8], linguistics [9], and artificial intelligence, since [17], mainly in the perspective of diagnostic reasoning, see e.g. [11] for a general presentation. Abduction has also been introduced in Logic Programming by R. Kowalski in [13], and the combination of abductive capabilities within the rigorous framework of Logic Programming brings up to an elegant simple and powerful sytem, further developments of which can be found in [12].

2 Constraint-based Reasonning

A constraint is simply a logical relation between several unknowns, those unknowns being variables that should take values in some domain. A constraint thus restrict the degrees of freedom (possible values) the unknowns can take; it represents some partial information relating the objects of interest. The whole idea of constraint solving is to start reasoning and computing with partial information, ensuring the overall consistency and reducing as much as possible the domains of the unknowns. This is in sharp contrast to classical programming (either imperative, object-oriented or functional) where one only computes with complete values and has no support for propagating partial information.

A Constraint Programming system will therefore accumulate constraints during computation and maintain a constraint store, communicating back with the program execution when appropriate, that is, upon inconsistency ("error") or when constraint solving has led to some variable instantiation. One can thus consider a split architecture with a continuous dialogue between a computation engine on one side, in charge of the program execution, and a constraint solver on the other side, in charge of the constraint store.

An instance of this scheme is given by the Constraint Logic Programming (CLP) paradigm which is based both on constraint solvers and on the idea of a declarative logical language based on Horn clauses (cf. Prolog). Constraint Logic Programming has shown to be a very active field over recent years, and a good survey of this research area is given in [10]. Languages such as CLP($\mathcal{R}$), CHIP or PrologIII have investigated several constraints domains : finite domains, reals/rationals, booleans. Most promising among these seem to be finite domains which are most suited for many real-life applications such as combinatorial problems, scheduling, cutting-stock, circuit simulation, diagnosis, decision-making, etc. As exemplified by the last two items, the application field of CLP is quite related to that of abductive reasoning.

3 Constraints as abduction

Interestingly, constraints can be considered as some sort of hypotheses that are stated on the (partially defined) objects of interest, and the task of the constraint solver is therefore to maintain the consistency of the assumptions and to reduce them to simpler forms. Hence constraint reasoning has close links with abductive reasoning.

Indeed both abductive and constraint processing systems are intended to deal with partial information and to avoid floundering when dealing with under-specified problems, both in fact conjecture relations between unknowns instead of getting stuck in a "dead-end" when complete characterization is missing. This correspondence is stressed in that the basic step of an abduction process consist in adding an abducible literal to the current set of hypotheses as long as it is consistent with it, while the basic step of constraint solving consists in adding a new constraint to the current set of constraint and checking consistency.

4 Constraints as a Tool for Abductive Systems

Finite Domains (FD) have been introduced by the CHIP language [18], which proposes arithmetic constraints such as (linear) equations, inequations or disequations (e.g. $X = 3 * Y + Z$, $X < Y$ or $Z \neq 2 * Y - 3$). Constraint solving is done by so-called consistency techniques or local propagation techniques. Such techniques stem from Constraint Satisfaction Problems in Artificial Intelligence [14; 15]. The basic idea is to manage a network of constraints between a set of variables which can take values in some finite domains by ensuring local consistency propagation through the constraints linking the variables.

We have been developing for some years at INRIA a Constraint Logic Programming system with constraints on finite domain variables [2]. This system is based on a novel approach for constraint solving, which could be called RISC by analogy with microprocessors. It consists in translating at compile-time all complex user-constraints (e.g. disequations, linear equations or inequations) into simple, primitive constraints at a lower level which really embeds the propagation mechanism for constraint solving. This gives us a language to express propagation methods and leads to a better understanding of the overall process. In this approach, the user can very easily define new constraints, by simply defining new predicates and their definition in therms of primitive (low-level) constraints. Therefore new abductive theories can be encoded, as constraints, and processed efficiently.

5 Conclusion

We are currently investigating the use of this CLP system for abductive reasoning, continuing researches on the relations between abduction and constraint processing initiated in [5] and [1]. The relations between Constraints and Abduction have also been a key issue discussed at two previously organized workshops [3; 4].

References

[1] P. Codognet and C. Codognet. Abduction and Concurrent Logic Languages. In *proc. ECAI'94, European Conference on Artificial Intelligence*, T. Cohn (Ed.), Amsterdam, Netherlands, 1994.

[2] P. Codognet and D. Diaz. Compiling constraints in `clp(FD)`. *Journal of Logic Programming*, Vol. 27, No. 3, 1996.

[3] P. Codognet, P. Dung, A. Kakas and P. Mancarella (Ed.). *proceedings of the ICLP'93 workshop on Abductive reasoning*, International Conference on Logic Programming, Budapest Hungary, 1993.

[4] P. Codognet, H. Guesgen, W. Hower and M. Meyer (Ed.). *proceedings of the ECAI'94 workshop on Constraint Processing*, European Conference on Artificial Intelligence, Amsterdam, Netherlands, 1994.

[5] P. Codognet and S. Saraswat. Abduction for concurrent constraint languages. In *proc. of the COMPULOG workshop on Logic Programming and Artificial Intelligence*, R. Kowalski (Ed.), London, April 1992.

[6] L. Couturat. *La logique de Leibniz*, Paris, 1903. New edition by Olms, Hildesheim, 1969.

[7] A. C. Crombie. *Robert Grosseteste and the Origins of Experimental Science 1100-1700*, Oxford, Clarendon Press 1953.

[8] Umberto Eco. *Semiotics and philosophy of language.* Indiana University Press 1984.

[9] J. Hobbs, M. Stikel, P. Martin and D. Edwards. Interpretation as abduction. In *proceedings of the 26th Annual Meeting, Association for Computational Linguistics*, Buffalo, USA, 1988.

[10] J. Jaffar and M.J. Maher. Constraint logic programming : a survey. *Journal of Logic Programming*, vol. 19 & 20, pp. 503–581, 1994.

[11] J. R. Josephson and S. G. Josephson. *Abductive inference : computation, philosophy, technology*, Cambridge University Press, 1994.

[12] A. C. Kakas, R. A. Kowalski and F. Toni. Abductive Logic Programming. *Journal of Symbolic Computation*, vol. 2 no. 6, pp. 719-770, 1992

[13] K. Eshghi and R. A. Kowalski. *Abduction through Deduction.* Research report, Imperial college, March 1988.

[14] U. Montanari. Networks of constraints: Fundamental properties and application to picture processing. In *Information Science, 7 (1974).*

[15] A. K. Macworth. Consistency in Networks of Relations. *Artificial Intelligence*, 8 (1977).

[16] Charles S. Peirce. *Collected Papers.* Harvard University Press, 1934-1948.

[17] H. E. Pople Jr. On the mechanization of abductive logic. In *proceedings of IJCAI 73*, 1973.

[18] P. Van Hentenryck. *Consistency techniques in Logic Programming.* The MIT Press, 1989.