

## **The role of heuristics in automated theorem proving. J.A. Robinson's resolution principle**

Roberto Cordeschi

Homer wrote *Odyssey*; given infinite time, with infinite events and changes, it is impossible one doesn't write, at least one time, *Odyssey* (J.L. Borges, *The Aleph*)

**Abstract** The aim of this paper is to show how J.A. Robinson's resolution principle was perceived and discussed in the AI community between the mid sixties and the first seventies. During this time the so called "heuristic search paradigm" was still influential in the AI community, and both resolution principle and certain resolution based, apparently human-like, search strategies were matched with those problem solving heuristic procedures which were representative of the AI heuristic search paradigm.

### **1 Introduction**

In this paper I shall focus on a less well known aspect of the evolution of ideas of AI (Artificial Intelligence). It is well known that the resolution principle was introduced in theorem proving by J. Alan Robinson during the mid sixties. It is also well known that resolution deeply influenced Robert Kowalski's logic programming, which opened up the way to PROLOG during the first seventies. Nevertheless, the AI community reaction to resolution principle in the time interval between the mid sixties and the first seventies is less well known. My aim is to show how the resolution principle was perceived and discussed in the AI community over that period. During this time the so called "heuristic search paradigm" was still influential in the AI community, and both resolution principle and certain resolution based, apparently

human-like, search strategies were matched with those problem solving heuristic procedures which were representative of the AI heuristic search paradigm.

## 2 The butcher's knife

When in the mid fifties Allen Newell, Clifford Shaw and Herbert Simon designed Logic Theorist (LT) and General Problem Solver (GPS), the heuristic programs which were to have such a strong influence on early AI, they claimed that the only alternative problem-solving programs which could compete with the heuristic ones they proposed, but which proved to be failures, were "simple algorithms that carry out exhaustive searches of all possibilities, substituting 'brute force' for the selective search of LT" (Newell, Shaw, Simon, 1958, p. 156).

The authors were referring to the well-known *British Museum algorithm*, as (in honour of the primates who are credited with employing it, given enough time) they called the algorithm they were proposing as an example of exhaustive generation of all the theorems of sentential calculus, once the axioms and the rules of inference of Russell and Whitehead's *Principia mathematica* were given. The British Museum algorithm could now be described as a blind, breadth-first search for a proof through a forward exploration of the proof space. Intuitively this procedure, although *complete* (with respect to the proof) is highly *inefficient*. As Newell, Shaw and Simon concluded, "even with the speeds available to digital computers, the principal algorithm we have devised as an alternative to LT [i.e. the British Museum algorithm] would require times of order of hundreds or even thousands of years to prove theorems that LT proves in a few minutes. LT's success does not depend on the 'brute force' use of computer's speed, but on the use of heuristic processes like those employed by humans" (*ibid.*).

The authors' reference to the quantitative analyses of LT efficiency was thus explicit. The picture of the situation regarding the study of algorithmic methods and the very nature of such methods must have appeared to them as more backward than it actually was. In fact only a short time later, Hao Wang succeeded in identifying much more powerful algorithms from the standpoint of the efficiency of performance than the LT heuristic methods. While acknowledging that the comparison between Newell, Shaw and Simon procedures and his own could be considered "unfair", as the respective underlying premises were "basically different", Wang came to the conclusion that "to argue the superiority of 'heuristic' over algorithmic methods by choosing a particularly inefficient algorithm seems hardly just": the "butcher's knife"

of Newell, Shaw and Simon ultimately proved to be as disproportionate as it was ineffective (Wang, 1960, p. 227).

While it is possible to imagine that the case of the "particularly inefficient" algorithm to which Wang refers without further details (apparently the British Museum algorithm) was for its inventors little more than a provocative expedient, the fact remains that they were proposing LT as an alternative at the efficiency level to algorithmic procedures in general. Furthermore, no clarification was forthcoming from the authors of LT: they continued to represent the problem of algorithmic proofs vis-à-vis LT in terms of brute force (see for example Newell and Simon, 1972). At the origin of this debate may be the *ambiguous* nature of the very word "heuristic" in the pioneering stages of AI, originally used both in the *simulation* of a human theorem prover, as an explanatory tool in the study of intelligent behavior, and with reference to the *performance* of a mechanical theorem prover, as an incomplete, but at least efficient, theorem proving procedure. Wang was later to point out somewhat ironically that "in the earlier stages when the Logic Theorist was in vogue, I designed a much simpler program which did much better. And yet it could be claimed, contrary to the popular opinion, that my program also was a simulation, only of a more sophisticated people" (Wang, 1974, p. 310). For their part, and this time without any irony, Newell and Simon, mentioned among others Wang's and Robinson's approaches to theorem proving as examples of research in AI, albeit not in the psychological simulation of human behavior, and concluded:

The interest of these programs [i.e. Wang's and Robinson's] for psychology, which was considerable, is in showing what kinds of mechanisms may be sufficient to give a program power to match, or exceed, human performance [...]. There is no claim that these programs simulate, in detail, the processes used by humans; but they demonstrate *what kinds of search heuristics provide problem-solving power in these well known task environments*. (Newell and Simon, 1965, p. 115, my Italics)

The attribution by Newell and Simon of the presence of "search heuristics" to Wang's programs (even though the latter originally preferred the "less esoteric" term of "strategy" to that of "heuristics") is in a sense not inappropriate. In fact, Wang's program is not a brute force program, and includes a process of backward search for a proof, a typically heuristic procedure from Newell and Simon's point of view. In order to clarify the meaning and the limits of Newell and Simon's reference to Robinson it is sufficient to briefly recall Robinson's *principle of resolution* (Robinson, 1965).

Resolution is a single rule of inference for a test of unsatisfiability (for further details, see e.g. Nilsson, 1971; Chang and Lee, 1973). Let  $A$ ,  $B$ ,  $C$  be *literals* (atomic formulas or their negations) occurring in the *clauses*  $A \vee \neg B$  and  $B \vee C$  (well formed formulas consisting precisely in disjunctions of literals). From such clauses, the so

called *parent clauses*, we can infer the *resolvent*  $A \vee C$ , i.e. the clause consisting in the disjunction of not complementary literals of the parent clauses. We call *resolution* the rule that yields such an inferred clause. Generally speaking, by progressively applying this rule to a starting set consisting of the premises and the negation of conclusion  $P \cup \{-T\}$ , written in *clause form*, and thus to the elements of this set and the related resolvents, and so on, if  $P \cup \{-T\}$  is unsatisfiable, we will be applying resolution to a pair of *complementary clauses* (of the type  $A$  and  $\neg A$ ), from which we infer the presence of a contradiction. This is customarily indicated by means of a symbol for the *empty clause*. This is the simplest case: that of the clauses which do *not* contain any individual variables. In the case of clauses containing these variables, expressed in first order predicate language, we shall in any case have to take the normal precautions required by the replacement of the variables with terms and, in the second case, it will be necessary to identify in the complementary literals *those* substitutions that make them identical, except for the negation sign. In other words, it is necessary to follow a procedure, known as *unification algorithm*, which makes it possible to identify the pairs of non identical complementary literals, in which substitution can be legitimately carried out by means of a *match* aimed as *eliminating the differences*. In fact the unification algorithm is the heart of resolution: it makes it superfluous to generate all the instances of possible substitutions through a brute force procedure. On the other hand, resolution leads to a *sound* and *complete* formal system.

### 3 Formal rules and heuristic search procedures

Now the fact that resolution apparently does not use brute force in theorem proving may have influenced Newell and Simon's opinion on the value of resolution as a *search heuristics*, even without simulating human problem solving processes. Let us see to what extent this opinion is plausible and what problems it raises as far as the relationship between heuristics and complete procedures is concerned.

At the outset, it is significant that, in his now classic 1965 paper on resolution, Robinson himself pointed out that the basic resolution procedure, essentially consisting in repeatedly generating resolvents, resolvents of resolvents, and so on, until the empty clause occurs, is still inefficient: a lot of irrelevant or redundant clauses are so generated, even in relatively simple examples of theorem proving. He actually acknowledged the need for "search principles" to be added to basic resolution procedure in order to limit the generation of these irrelevant or redundant clauses (Robinson, 1965). One of these search principles was the *unit-preference strategy*, invented by Wos, Carson and G. Robinson (1964) on the basis of a previous

unpublished draft paper by Robinson on resolution. Briefly, unit-preference strategy has the attractive feature that it always produces shorter clauses. In this way the length of generated clauses is decreased, and this is intuitively satisfactory, because the empty clause (of zero length) has to be generated. After all, this is a strategy that could be applied by a human theorem prover: it is a typical heuristic *selective* procedure, in a sense that could be shared by Newell, Shaw and Simon.

Robinson himself made a detailed examination of the relationship between heuristics such as unit-preference and heuristics *à la* Newell-Shaw-Simon in a further paper (Robinson, 1967a). He acknowledged from the outset that both resolution and search strategies could be evaluated on the basis of the same attitude with which early AI evaluated *selective* procedures: both resolution and search strategies had to be considered heuristic *insofar as* they were selective and *therefore* generally incomplete. He apparently considered that resolution itself shared an intuition of this kind. Indeed he remarked:

It is interesting to note [...] that the underlying idea of the resolution principle, and the resolution principle itself, are very much like the sort of stratagem earlier characterised as *heuristics* - to the point of intuitively appearing, at first, to involve the "loss of completeness" now taken [...] to be part of the essence of a heuristic method. In spite of its being heuristic in the sense of being the sort of thing an intelligent human problem-solver would do, the resolution principle turns out *not* to be heuristic in the "loss of completeness" sense. (Robinson, 1967a, p. 118)

Nevertheless, as we saw, basic resolution procedure produces a "demographic explosion" of clauses: the parent clauses add to the community of clauses *all* those resolvents that they are capable of generating:

[Resolution] is still not a practically feasible theorem-proving procedure, even though it is orders of magnitude more efficient than the older theorem-proving procedures which were based on the enumeration of instances. The problem therefore presents itself of introducing some form of "birth control" or "selective breeding" into the fundamental growth process. [...] It is very natural to think of restrictions on the growth process which are *heuristic* in character, i.e. which reflect the principles of selection that an intelligent human seeking to generate [the empty clause] would use. (Robinson, 1967a, p. 121)

In this connection Robinson mentioned the example of the *set of support strategy*, which he describes in a way that Newell and Simon would have perhaps agreed with: set of support "has the attraction of being 'purposive', in the sense of being deliberately *steered* towards the production of the empty clause". As Robinson states, set of support seems to reflect "what most experienced mathematicians would instinctively do" when faced with the problem of seeking the empty clause. This may

be summed up in the suggestion: do not generate the resolvents of those clauses in the starting set of clauses which constitute the assumptions or axioms or premises of the problem, and which, as such, will presumably not give rise to contradictions.

However, this kind of heuristics, such as set of support or unit-preference, which were initially introduced in automated theorem proving without any explicit regard for the requirement of completeness, proved to be not less complete than resolution itself. How was this to be interpreted? In what way can a heuristics be *complete* when it is by definition *selective*? And in what sense is resolution, insofar as it is a *complete* rule of inference, *selective* vis-à-vis the older (i.e. Herbrand style) theorem proving procedures?

These issues arose at a time when AI had its own dominant version of heuristics, which was viewed in any case as something inevitable with regard to the question of combinatorial explosion. It is interesting to note that Robinson's above-mentioned analysis was commented upon in appreciably different ways in AI community. In Saul Amarel's view, resolution, the set of support and the hyper-resolution (Robinson's refinement of resolution) are all on a par in demonstrating how these principles

can benefit from cooperation with heuristic principles in obtaining efficient strategies for proof finding. [Robinson (1967a)] traces the evolution of [these] systematic logic principles and he shows that they were initially conceived as intuitively powerful heuristic principles that subsequently changed status when they were proved to be generally valid. This *movement of knowledge from informal, heuristic, status to systematic status* is extremely important for the gain in power and efficiency of a problem solving procedure. (Amarel, 1970, p. 240, my Italics)

On the other hand, Edward Feigenbaum, had already pointed out:

Robinson's resolution method for theorem proving in the predicate calculus has received much attention in the past few years. Unfortunately this has been accompanied by a sentiment that the resolution method "liberates" one from the gussy, messy, chaotic world of heuristic search. Again a false distinction, based on unclear understanding either of resolution or of the existing heuristic search proof-finding programs or both, sorts the world along the wrong lines. The resolution method does provide a systematic formal mechanism for guaranteeing the completeness of the space of proof candidates, but it does not by itself deal with the well known and inevitable problem of the proliferation of problem states (see Robinson, 1967a). Thus search strategies have been overlaid to bring about effective problem solving using resolution (e.g. "unit preference", "set of support"). (Feigenbaum, 1969, p. 1012)

We have quoted Amarel and Feigenbaum verbatim as this allows us to put forward a set of considerations concerning the above-mentioned issues raised by resolution. Amarel seems to have interpreted Robinson's analysis in his 1967 paper by

imagining a "movement" from the informal to the formal more or less in the following terms: *every* more or less efficiently selective principle *changes status* whenever it is demonstrated to be complete (this is perhaps the sense of "liberation" stressed by Feigenbaum). This interpretation may give rise to a certain ambiguity. Robinson's analysis leads us to conclude that the proof of completeness of a procedure that is *in any case* selective (whether it be an efficient inference rule or a search strategy), is "psychologically later in being unearthed" than the details of that very procedure. Anyway, Amarel does not distinguish between selectivity and completeness *in the sense* of an inference rule and selectivity and completeness *in the sense* of a proof finding strategy. This distinction emerges from Robinson's analysis and is stressed by Feigenbaum, whose remarks are perhaps closer than Amarel's to the basic tenets of Robinson resolution.

Following such a distinction, resolution is an inference rule for a formal system (the first order predicate calculus), not a proof finding procedure (for example, the set of support). Of course (and is this perhaps what causes the confusion Feigenbaum refers to) the completeness proof for resolution uses a proof-finding procedure, and indeed the proof consists of defining this algorithm and then showing that it will always terminate when applied to an unsatisfiable set of clauses. Resolution is complete in this sense, *sense 1*, so to speak, and completeness is a pure logic notion. It pertains to a set of deduction rules, and consists in a positive answer to the following question. Let R be a set of deduction rules, which determines a class P of proofs (all those formed using only rules in R, including among R the rules establishing what are axioms); has R the property: whenever a sentence S is a logical consequence of sentences A, is there a proof of S from A among the proofs in P?

But given a set R of deduction rules we can consider search procedures Q for R - procedures for finding a proof of a sentence S from sentences A in the class of proofs P. We can ask of Q: is Q complete? I.e., has Q the property: whenever a proof exists in P of a sentence S from sentences A, will Q find such a proof? This *sense 2* of completeness pertains to a search procedure and not to a set of deduction rules.

So, for example, resolution is complete in *sense 1* and the set of support strategy is complete in *sense 2* (for resolution as R). Accordingly, the completeness of a heuristic as a search procedure is not the completeness of a formal rule. Resolution establishes "*the fundamental structure of the 'search space'*", in Robinson's words in an other 1967 paper (Robinson, 1967b, p. 17). The exploration of such a search space is made more feasible through heuristic search procedures. From this point of view, Feigenbaum's remark is quite correct: resolution does not per se solve the problem of the growth of irrelevant clauses, which is faced (if not generally solved) through efficient heuristic search procedures.

In a somewhat similar way to this evaluation, resolution has also been described by Simon as an "information accumulator" which works forwards. Resolution does not generate a search tree but a gradually increasing set of derived expressions. What Simon is concerned with is to distinguish a process that proceeds directly towards the solution by exploiting small amounts of input information at a time from a solution *seeking* process that evaluates interesting information stepwise. Simon considered set of support strategy as an example of the latter kind of process. In this case we have a function with which to evaluate the input information (which is "purposive", to quote Robinson), which selects the clauses according to whether or not they belong to the set (Simon, 1972, pp. 271-272).

If resolution is defined as a "fundamental structure of the 'search space'", to use Robinson's words, we may ask something about its admittedly *selective* feature. As we saw, in the above mentioned contraposition between the British Museum algorithm and LT, the initial opposition between procedures that were complete and exhaustive as a result of brute force (algorithms) and incomplete and selective procedures (heuristics) was radicalised. However, in Wang too there was a (natural) need to curb the "demographic explosion" of irrelevant or redundant clauses produced by exhaustive generation. This is the line of research (from Wang, and Herbrand before him, to Davis and Putnam and Dag Prawitz) on which the very philosophy of resolution is based. At that stage, the problem was, in the words of Meltzer, "how to select the crucial contradictory set without going through the 'British Museum' process of turning out vast numbers of irrelevant clauses" (Meltzer, 1969, p. 42). To quote Robinson himself,

The main difficulty with making these [i.e. pre-resolution] procedures efficient is in devising methods of enumerating Herbrand expansions which are on the one hand 'exhaustive' but on the other *in some sense* selective. (Robinson, 1970, p. 4, my Italics)

It is perhaps worth emphasising Robinson's caution in defining resolution as selective "*in some sense*", insofar as it represents an inference rule for a complete system, and is "exhaustive" in this sense. *In what sense*, in fact, does resolution reduce the search space with respect to the preceding automatic proof systems? How can we compare the relative efficiencies of (*non-heuristic*) proof systems? As we have seen, we *intuitively* recognise that resolution leads to a *reduction of complexity* in the search space compared with the earlier pre-resolution procedures, more or less based on Herbrand's work: this space is less "cluttered" with irrelevant or redundant clauses. However, at the time Robinson was writing it was difficult to define more accurately (less intuitively) what was meant by this reduction in complexity. These were the

years in which the problem of computational complexity had just been raised, and the problem was beginning to be investigated.

#### 4 GPS and resolution

In concluding the passage quoted above, Feigenbaum made reference to research on the version of GPS by Ernst and Newell in order to assert that

The net result is that the processes these [i.e. resolution plus search strategies] programs carry out are much the same as (in some cases identical to) those carried out by the heuristic search proof-finding programs that are thought to be so different: e.g. Ernst e Newell (1969). (Feigenbaum, 1969)

This conclusion introduces a further evaluation of resolution in the early AI period which may be considered extremely dubious. Newell and Ernst had in fact equipped GPS with a rule based on the principle of resolution. They concluded that "the matching procedure of LT represents the inner component of Robinson's unification algorithm" (Ernst and Newell, 1969, p. 181, n.1).

In the early AI period the attractiveness of the LT matching procedure in automated theorem proving was widely acknowledged. It consisted of a kind of pattern matching in which the more similar patterns or symbolic structures were selected by means of a *similarity test* and the others eliminated. However, it is difficult to imagine that anything more than a superficial analogy could exist between the unification algorithm and the matching procedure peculiar to LT. The unification method is, on the one hand, a complete procedure and, on the other, is much more general and powerful than the simple matching procedure of LT. To begin with, its most interest application bears on first order predicate calculus, not on sentential calculus, the LT level of performance (in this case, resolution may be viewed as a simple generalisation of Davis and Putnam's *one-literal rule*: see Davis and Putnam, 1960). Indeed, unification refers to the individual variables of predicate calculus. As we have seen, this is where the importance of the procedure lies, although this is not the case compared with LT, but rather with the fundamental inspiration pervading the preceding automated theorem proving: Robinson's method, as Meltzer correctly observes, is "the first practical method of proving theorems on the basis of Herbrand's theorem without actually resorting to the examination of ground clauses - it operates entirely at the free-variable level" (Meltzer, 1969, p. 43).

However, it is the actual implementation of resolution in GPS that is seen to be particularly inefficient. Note that the observation is relevant: Ernst and Newell were not investigating GPS from the point of view of psychological simulation but as an AI

program, albeit one associated with the study of *generality*. In order to reproduce the resolution rule in GPS language, three operators were required:

- (I)  $(B \text{ and } \neg B)$  produces FALSE;
- (II)  $((B \vee \dots C) \text{ and } \neg B)$  produces  $C$ ;
- (III)  $((B \vee \dots C) \text{ and } (\neg B \vee \dots D))$  produces  $(C \vee D)$ .

The authors themselves acknowledged that this aspect, among others, rendered the GPS formulation of the task of proving theorems "somewhat clumsy" (Ernst and Newell, 1969, p.179). On the other hand, the fact that GPS applies these operators successively in the order indicated meant, according to Ernst and Newell, that GPS could use the unit-preference strategy (*ibid.*, p.180). In the case of GPS, the operator (I) produces an empty (i.e. *no* literal) clause, while (II) *always* gives rise to a clause containing fewer literals than (III). It should be noted that the formulation of resolution in terms of the three operators mentioned means that even the well-known "table of connections" of GPS is reduced, in the words of Ernst and Newell (*ibid.*, p. 172), to a "formality" as there is *only one* difference between the object given and the one sought (the empty clause), and all the operators are equally important in eliminating it. In other words, as a resolution-based theorem prover, GPS may be said *not to use means-end analysis* to prove the theorems (indeed it does not use different types of differences to select the different operators)!

This was a very high price to pay for having a GPS resolution-based theorem prover, that had consequences in terms of inefficiency. Indeed, when disguised as a resolution-based theorem prover, GPS, even though it uses unit-preference, generates 59 goals in order to prove the so called "Gilmore theorem" (Gilmore, 1960; Davis and Putnam, 1960). Using resolution together with a couple of simple additional search principle Robinson (1965) managed to prove the same theorem in six stages. It is evident that, while resolution was found to represent a suitable "fundamental structure" of the search space of proofs, its formulation in terms of GPS was of little interest (above and beyond the now known limits of GPS as a *general* AI program).

## 5 What is living in resolution

Although from different points of view, Amarel's and Feigenbaum's comments reflected the sense of the prediction made by Robinson in 1967, namely that the development of complete heuristic search strategies would make a decisive contribution to solving the problem of the selective exploration of the "fundamental structure" of the search space determined by resolution (Robinson, 1967b). Therefore,

future research would consist of the discovery of increasingly selective and efficient strategies without sacrificing the requirement of completeness:

I believe that there is still much to be discovered in the way of controlling the rate and direction of growth *intelligently yet automatically*, without disturbing the basic completeness property. I believe that there is nothing inherently conflicting in the two leading concepts—*heuristic control* of the process and *systematic, combinatorial control* of the process. (Robinson, 1967a, p. 123)

Indeed, until the early seventies, a true demographic explosion of resolution-based heuristics and "refinements" of resolution was in progress. This was accompanied by a proliferation of various different taxonomies of such heuristics and refinements (see e.g. Meltzer, 1971; Nilsson, 1971; Hunt, 1975; Loveland, 1978). These different evaluations and taxonomies of heuristics and refinements are faced with the problem of the actual *extent* to which they reduce the search space determined by resolution "without disturbing the basic completeness property", in Robinson's words.

Starting in the early seventies a more radical theorem proving philosophy seems to embody the feeling of disillusionment that followed the euphoria over heuristic principles which were complete as indicated by Robinson. The requirement of completeness (of heuristics) again becomes a thorny question, but this time, for those who raised it in the terms of Robinson's prediction. Newell, Simon, Minsky and the "antilogicist" AI in general had never attributed any importance to it. In the appendix to his famous frame paper, Minsky liquidated the requirement of completeness as non essential (Minsky, 1975). Simon, in the above mentioned 1972 paper, emphasised that the crucial problem facing heuristics was efficiency, and that it was fruitless to limit AI research to *complete* heuristics as there was no reason to believe that the activity of the creative mathematician and of the intelligent problem-solver in general, was based on complete procedures (even when decidable domains were involved). This implicitly reversed Robinson's 1967 prediction:

There are some signs that the undesirability of this limitation [i.e. the search for complete selection rules only] is now beginning to be recognised, and that the next decade will see a free exploration of heuristic procedures in this [i.e. theorem proving] and other problem domains. (Simon, 1972, p. 263)

In a sense, Simon's analysis proved to be accurate. The predominant feeling in the early seventies was, in fact, that, on the one hand, the refinements of resolution and the complete strategies had attained peak efficiency (as the improvements obtained were often marginal in terms of efficiency when it was a matter of tackling problems of medium difficulty); on the other, it was felt that, although the search space had been

restricted so as to render exploration easier, the price paid for these positive results was often an overall increase in the required space and machine time. Furthermore, the refinements were independent of the nature of the problem and the heuristic evaluation functions were based exclusively on the syntactic properties of the clauses; the effects of these limitations were noticeable when the theorem to be proved was not all that interesting.

Suffice it to read "Artificial Intelligence" and *Machine Intelligence* issues of those years to appreciate this substantial reversal of trend in the seventies. The reduced importance of resolution as the "fundamental structure" of the search space in Robinson's sense is paralleled by the experimentation of alternative principles, more directly based on human intuition and practice, in a "much more pragmatic attitude towards heuristics", as Simon concluded (Simon, 1979, p. 1090). In other cases, resolution was relegated to a secondary role compared with the numerous heuristics, general or specific. For instance, W.W. Bledsoe's theorem proving programs were based on the principle: "[make] much more use of heuristics and minimize, or possibly replace, resolution" (Bledsoe, 1971, p.76). Bledsoe himself was the author of the first review on "non resolution theorem proving" (Bledsoe, 1977), where he pointed out, among others, Douglas Lenat's recent AM "discovery program", that was so much impressive at that time (see Simon, 1979) - but so much distant from the traditional line of research on automatic theorem proving that Wang judged it "thoroughly unwieldy", and could not see "how one might further build on such a baffling foundation" (Wang, 1984, p. 50).

Nevertheless, this re-discovery of more human-like heuristics did not lead to the death of resolution. On the contrary, in the very years in which the latter was considered as dead, it opened up the way to a new style of programming, which was introduced by Robert Kowalski as "logic programming" (see Kowalski,1979; Robinson, 1983). Kowalski used a refinement of resolution which is complete and efficient for a particular and important class of clauses, i.e. Horn clauses. It was this refinement that was embodied in PROLOG, and in 1972 Alain Colmerauer wrote the first PROLOG interpreter (see Cohen, 1988). But this is more recent and known history.

## **Acknowledgements**

I thank Settimo Termini for his observations on a previous version of this paper.

## References

- Amarel S. (1970), On the representation of problems and goal directed procedures for computers, in Baneji R., Mesarovic M. (eds.), *Theoretical approaches to non-numerical problem solving*, Berlin, Springer, 179-244.
- Bledsoe W.W. (1971), Splitting and reduction heuristics in automatic theorem proving, *Artificial Intelligence*, 2, 55-77.
- Bledsoe W.W. (1977), Non-resolution theorem proving, *Artificial Intelligence*, 9, 1-35.
- Chang C.L., Lee R.C.T. (1973), *Symbolic logic and mechanical theorem proving*, New York, Academic Press.
- Cohen J. (1988), A view on the origins and development of PROLOG, in *Communications of the ACM*, 31, 26-36.
- Davis M., Putnam H. (1960), A computing procedure for quantification theory, *Journal of the Association for Computing Machinery*, 7, 201-215.
- Ernst G.W., Newell A. (1969), *GPS: a case study on generality and problem solving*, New York, Academic Press.
- Feigenbaum E.A. (1969), Artificial intelligence: themes in the second decade, in *Information Processing 68*, Amsterdam, North Holland, 1008-1020.
- Gilmore P.C. (1960), A proof method for quantification theory, *IBM Journal for Research and Development*, 4, 28-35.
- Hunt E.B. (1975), *Artificial intelligence*, New York, Academic Press.
- Kowalski R. (1979), *Logic for problem solving*, New York, Elsevier.
- Loveland D. (1978), *Automated theorem proving: a logical basis*, Amsterdam, North Holland.
- Meltzer B. (1969), The use of symbolic logic in proving mathematical theorems by means of a digital computer, in Bulloff J.J., Holyoke T.C., Hahn S.W. (eds), *Foundations of mathematics*, Berlin, Springer, 39-45.
- Meltzer B. (1971), Prolegomena to a theory of efficiency of proof procedures, in Findler N.V., Meltzer B. (eds), *Artificial intelligence and heuristic programming*, Edinburgh, Edinburgh University Press, 15-33.
- Minsky M. (1975), A framework for representing knowledge, in Winston P. (ed.), *Psychology of computer vision*, New York, McGraw-Hill, 211-280.
- Newell A., Shaw J.C., Simon H.A. (1958), Elements of a theory of human problem solving, *Psychological Review*, 65, 151-166.
- Newell A., Simon H.A. (1965), Simulation of human processing of information, *American Mathematical Monthly*, 72, 111-118.
- Newell A., Simon H.A. (1972), *Human problem solving*, Englewood Cliffs, Prentice-Hall.
- Nilsson N.J. (1971), *Problem solving methods in Artificial Intelligence*, New York, McGraw-Hill.
- Robinson J.A. (1965), A machine oriented logic based on the resolution principle, *Journal of the Association for Computing Machinery*, 12, 23-41.
- Robinson J.A. (1967a), Heuristic and complete processes in the mechanization of theorem-proving, in Hart J.T., Takasu S. (eds.), *Systems and computer science*, Toronto, University of Toronto Press, 116-124.

- Robinson J.A. (1967b), A review of automatic theorem-proving, in *Proceedings of a Symposium in Applied Mathematics*, vol. 19 of the American Mathematical Society, 1-18.
- Robinson J.A. (1970), An overview on mechanical theorem proving, in Banerji R., Mesarovic M. (eds.), *Theoretical approaches to non-numerical problem solving*, Berlin, Springer, 2-20.
- Robinson J.A. (1983), Logical reasoning in machines, in Hayes J.E., Michie D. (eds.), *Intelligent systems*, Ellis Horwood, Chichester, 19-36.
- Simon H.A. (1972), The theory of problem solving, in *Information processing 71*, Amsterdam, North Holland, 261-277.
- Simon H.A. (1979), Artificial Intelligence research strategies in the light of AI models of scientific discovery, in *Proceedings of the International Joint Conference on Artificial Intelligence*, 6, 1086-1094.
- Wang H. (1960), Toward mechanical mathematics, *IBM Journal for Research and Development*, 4, 2-22. Reprinted in Wang H., *A survey of mathematical logic*, Peking and Amsterdam, Science Press and North Holland, 1963, 224-268.
- Wang H. (1974), *From mathematics to philosophy*, London, Routledge and Kegan Paul.
- Wang H. (1984), Computer theorem proving and Artificial Intelligence, in Bledsoe W.W., Loveland D.W. (eds.), *Automated theorem proving: after 25 years*, American Mathematical Society, Providence, 49-70.
- Wos L., Carson D., Robinson G. (1964), The unit preference strategy in theorem proving, in *Proceedings of the AFIPS Conference*, 26, 615-621.