Available online at www.sciencedirect.com
SCIENCE DIRECT
Artificial Intelligence 162 (2005) 3–5
Artificial Intelligence
www.elsevier.com/locate/artint

Introduction to the Special Volume on Reformulation

The performance of a problem-solving system often depends on the representation of problem instances or background knowledge. A change in representation may therefore improve (or degrade) performance. *Reformulation* is a process of changing the representation of a problem instance or knowledge base in order to improve the performance of a problem-solving system. Two special types of reformulation are abstraction and approximation. *Abstraction* is typically viewed as a change that improves the performance of a representation, while diminishing the scope of problems that can be solved or the precision of the solutions that are generated. *Approximation* is typically viewed as a change that improves the performance of a representation, while diminishing the accuracy of the solutions that are generated.

The study of reformulation in Artificial Intelligence began in the 1960s. A seminar at Carnegie-Mellon University in 1966 was devoted to studying the effect of representation on puzzle problems, such as "Monkeys and Bananas", "Mutilated Checkerboard", "Tic Tac Toe" and "Missionaries and Cannibals", as well as mathematical problems in algebra, geometry and logic. Alan Newell discussed the relation between representation and problem solving using the Mutilated Checkerboard problem as an example [12]. Newell suggested that "a representation of possible representations", if such could be developed, would be a useful addition to the "current stock of ideas about problem-solving". Saul Amarel carried out one of the first detailed studies of reformulation, using the Missionaries and Cannibals problem as an example [1]. Amarel described a series of problem formulations, each slightly different from the previous one, in which an apparently difficult problem is gradually reduced to a trivial one. Korf further developed this early work by attempting to carry out Newell's suggestion [10]. He designed a search space in which states are problem-representations and operators change one representation into another. He also showed how the search space could be used to reformulate a number of well-known puzzle problems.

Reformulation techniques have since been developed and studied in a variety of settings, including classical planning [17]; theorem-proving in first-order logic [15]; relational database query processing [8]; reasoning with propositional databases [18]; axiomatic formal systems [7]; heuristic search of state spaces [14,16]; constraint satisfaction [2,4,5]; pro-

0004-3702/$ – see front matter
doi:10.1016/j.artint.2004.07.001

gram synthesis [11]; simulation of physical systems [6]; and explanation and diagnosis of physical systems [13,19]. Interest in reformulation, abstraction and approximation led to a number of independent workshop series in the late 1980s and 1990s, conducted under the sponsorship of AAAI and IJCAI. Eventually the workshops were merged into the ongoing series of "Symposia on Abstraction, Reformulation, and Approximation" [3,9].

Many reformulation techniques can be understood in terms of the problem-solving framework outlined below. One starts with a problem instance P and a knowledge base K. A direct problem-solving technique would apply a solution algorithm A to (P, K) to obtain a solution S. An indirect technique uses reformulation in the following way: (1) Apply a problem reformulation technique R to (P, K) to obtain (P', K'), where P' and K' are idealized versions of P and K; (2) Apply an idealized solution algorithm A' to (P', K') to obtain an idealized solution S'. (3) Apply a solution adaptation technique D to S' to obtain an actual solution S. The framework can be instantiated in many ways, e.g., by choosing languages to represent (P, K) and (P', K'); solution algorithms A and A'; reformulation technique R; and adaptation technique D. Furthermore, solution strategies may differ according to whether the reformulation process is carried out once for an entire problem class, or must be carried out once (or many times) for each problem instance. Finally, some components of the instantiated framework can be trivial and not appear explicitly.

$$\begin{tikzcd}
(P', K') \arrow[r, "A'"'] & S' \arrow[d, "D"] \\
(P, K) \arrow[u, "R"] \arrow[r, "A"] & S
\end{tikzcd}$$

P	Problem Instance
P'	Idealized Problem Instance
K	Knowledge Base
K'	Idealized Knowledge Base
A	Solution Algorithm
A'	Idealized Solution Algorithm
R	Problem Reformulation Technique
S	Solution Adaptation Technique

The present volume of *Artificial Intelligence* is devoted to recent research on reformulation, approximation and abstraction. In "First Order LUB Approximations: Characterization and Algorithms", del Val extends and generalizes previous work on "knowledge compilation" [18], a technique for approximating propositional theories by upper and lower bounds and using the approximations to answer queries over the original knowledge base. In "Partition-Based Logical Reasoning for First-Order and Propositional Theories", Amir and McIlraith present algorithms for decomposing sets of logical axioms into partitions and subsequently restricting inference to occur within partitions, in order to improve the efficiency of a reasoning process. In "Compiling Problem Specifications into SAT", Cadoli and Schaerf present a compiler that translates declarative specifications of NP-hard problems into propositional satisfiability problems, so they can be solved by any of the state-of-the-art solvers available from the SAT community. In "Task-Dependent Qualitative Domain Abstraction", Sachenbacher and Struss present methods for automatically abstracting behavior models of engineering devices in order to carry out diagnosis or behavior prediction at an appropriate level of generality. In "Towards a Practical Theory of Reformulation for Reasoning about Physical Systems", Choueiry, Iwasaki and McIlraith present a framework for specifying, classifying and evaluating techniques for reformulating models of physical systems.

References

[1] S. Amarel, On representations of problems of reasoning about actions, in: D. Michie (Ed.), Machine Intelligence, vol. 3, American Elsevier, New York, 1968, pp. 131–171.
[2] S. Bistarelli, P. Codognet, F. Rossi, Abstracting soft constraints: framework, properties, examples, Artificial Intelligence 139 (2002) 175–211.
[3] B. Choueiry, T. Walsh (Eds.), Abstraction, Reformulation, and Approximation, 4th International Symposium, Lecture Notes in Artificial Intelligence, vol. 1864, Springer, Berlin, 2000.
[4] T. Ellman, Abstraction via approximate symmetry, in: Proceedings of the 13th International Joint Conference on Artificial Intelligence, Chambéry, France, 1993, pp. 916–921.
[5] E. Freuder, D. Sabin, Interchangeability supports abstraction and reformulation for multi-dimensional constraint satisfaction, in: Proceedings of the 14th National Conference on Artificial Intelligence, Providence, RI, 1997, pp. 191–196.
[6] B. Falkenhainer, K. Forbus, Compositional modeling: finding the right model for the job, Artificial Intelligence 51 (1991) 95–143.
[7] F. Giunchiglia, T. Walsh, A theory of abstraction, Artificial Intelligence 57 (1992) 323–389.
[8] T. Imielinski, Abstraction in query processing, J. ACM 38 (1991) 534–558.
[9] S. Koenig, R. Holte (Eds.), Abstraction, Reformulation and Approximation, 5th International Symposium, Lecture Notes in Artificial Intelligence, vol. 2371, Springer, Berlin, 2002.
[10] R. Korf, Toward a model of representation changes, Artificial Intelligence 14 (1980) 41–78.
[11] M. Lowry, Algorithm synthesis through problem reformulation, in: Proceedings of the Sixth National Conference on Artificial Intelligence, Seattle, WA, 1987, pp. 432–436.
[12] A. Newell, Limitations of the current stock of ideas about problem-solving, in: A. Kent, O. Taulbee (Eds.), Electronic Information Handling, Spartan Books, New York, 1965, pp. 195–208.
[13] P. Nayak, Causal approximations, Artificial Intelligence 70 (1994) 277–334.
[14] J. Pearl, Heuristics: Intelligent Search Strategies for Computer Problem Solving, Addison-Wesley, Reading, MA, 1984.
[15] D. Plaisted, Theorem proving with abstraction, Artificial Intelligence 16 (1981) 47–108.
[16] A. Prieditis, Machine discovery of effective admissible heuristics, Machine Learning 12 (1993) 117–141.
[17] E. Sacerdoti, Planning in a hierarchy of abstraction spaces, Artificial Intelligence 5 (1974) 115–135.
[18] B. Selman, H. Kautz, Knowledge compilation and theory approximation, J. ACM 43 (1996) 193–224.
[19] D. Weld, Exaggeration, Artificial Intelligence 43 (1990) 311–368.

Thomas Ellman *
Department of Computer Science,
124 Raymond Avenue, Vassar College, Poughkeepsie, NY 12604, USA
E-mail address: thellman@vassar.edu (T. Ellman)

Fausto Giunchiglia
Department of Information and Communication Technology,
University of Trento, 38050 Povo di Trento, Italy

Available online 8 December 2004

* Corresponding author.