

The Logic Grammars based Genetic Programming System

Man Leung Wong

Department of Computing and Decision Sciences
Lingnan University, Tuen Mun
Hong Kong
mlwong@ln.edu.hk

Kwong Sak Leung

Department of Computer Science and Engineering
The Chinese University of Hong Kong
Hong Kong
ksleung@cse.cuhk.edu.hk

ABSTRACT

Inductive Logic Programming (ILP) and Genetic Programming (GP) are two approaches in program induction. However, they are restricted in the computer languages in which programs can be induced. Generic Genetic Programming (GGP) is a novel, powerful, and general approach that combines ILP and GP. Using GGP, programs in various programming languages such as Lisp, Prolog, and Fuzzy Prolog can be evolved. The approach is also powerful enough to handle context-sensitive information and domain-dependent knowledge. In this paper, we describe the details of an implementation of GGP called LOGENPRO (The LOGic grammars based GENetic PROgramming system).

1. Introduction

Inductive Logic Programming (Muggleton 1992) and Genetic Programming (Koza 1992, 1994; Kinnear 1994) are two approaches in program induction. However, they are restricted in the computer languages in which programs can be induced. Inductive Logic Programming (ILP) can only learn logic programs. On the other hand, Genetic Programming (GP) represents induced programs as parse trees. In theory, programs in any computer languages such as Prolog, C, and Fortran can be represented as parse trees. Hence, GP should be able to handle them as well. In fact, the process of translating a program in some languages to the corresponding parse tree is not trivial. Since the syntax of Lisp is so simple and uniform that the translation process can be achieved easily, programs evolved by GP are usually expressed in Lisp.

A novel approach called Generic Genetic Programming (GGP) that combines GP and ILP has been proposed (Wong and Leung 1995, 1996). Using GGP, programs in various

programming languages such as Lisp, Prolog, and Fuzzy Prolog can be evolved. The approach is also powerful enough to handle context-sensitive information and domain-dependent knowledge which can be used to accelerate the learning speed and/or improve the quality of the programs induced. This paper describes the details of an implementation of GGP called LOGENPRO (The LOGic grammars based GENetic PROgramming system).

Since LOGENPRO can induce programs in various programming languages, it must be able to accept grammars of different languages and produce programs in them. LOGENPRO is based on logic grammars because they are expressive enough to represent context-sensitive information of some languages and domain-dependent knowledge of the target programs being induced.

Bibliography

- Koza, J. R. (1992). *Genetic Programming: on the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press.
- Koza, J. R. (1994). *Genetic Programming II: Automatic Discovery of Reusable Programs*. Cambridge, MA: MIT Press.
- Kinnear, K. E. Jr., editor (1994). *Advances in Genetic Programming*. Cambridge, MA: MIT Press.
- Muggleton, S., editor (1992). *Inductive Logic Programming*. London: Academic Press.
- Wong, M. L. and Leung, K. S. (1995). Combining Genetic Programming and Inductive Logic Programming using Logic Grammars. In *Proceedings of the 1995 IEEE International Conference on Evolutionary Computing*.
- Wong, M. L. and Leung, K. S. (1996). Evolving recursive functions for the even-parity problem using genetic programming. In P. J. Angeline and K. E. Kinnear, Jr. (Eds.) *Advances in Genetic Programming 2*. MA: MIT Press.