Generation of Concurrency Control Program by Extending Functions in Genetic Programming

Teruhisa Hochin, Kyoto Institute of Technology, Kyoto, Japan
Tatsuya Saigo, Kyoto Institute of Technology, Kyoto, Japan
Shinji Tamura, Kyoto Institute of Technology, Kyoto, Japan
Hiroki Nomiya, Kyoto Institute of Technology, Kyoto, Japan

ABSTRACT

This paper tries to generate an appropriate concurrency control program by using genetic programming (GP). Although two variables have been introduced for generating concurrency control programs, these made program generation difficult because of the explosion of combination. By limiting the usage of variables to one of two variables, an appropriate program could be generated. This method, however, could not create all of concurrency control programs. This paper extends the variable to bring more information than before for creating any concurrency control programs. It is experimentally shown that an appropriate concurrency control program can successfully be generated by extending the variable.

Keywords: Combination, Concurrency Control, Genetic Programming, Program Generation, Variable

INTRODUCTION

Information and network technologies have rapidly and widely spread. Databases have been used in various application areas. Database management systems are used in these areas. It tends to be difficult that a monolithic database management system fits all of database application areas. The database management system that could be adapted to each database application area will be required (Stonebraker & Cetintemel, 2005; Seltzer, 2008).

This paper focuses on the concurrency control mechanism, which is a very important component of the database management system because it guarantees the reliability and influences the performance of database processing (Silberschatz, Korth & Sudarshan, 2002). A series of database operations could be treated as a unit in the database management system. This unit is called a transaction. Features of transactions differ according to the database application areas. Examples of the features are the number of operations, the ratio of write operations to...
read ones, the number of tables manipulated, and the time period of the first operation to the last one. These features influence the concurrent execution of database operations. When all of operations are read ones, all of operations could concurrently be executed. When all operations are the write one updating the value of the same data item of the same table, all of the operations must serially be executed. That is, no operations could concurrently be executed. The concurrency control mechanism controls concurrent execution of the operations. The concurrency control mechanism suited to a database application area may be different from that suited to another one. The database management system, however, supports only one concurrency control mechanism in spite of kinds of database application areas. Overhead of concurrency control, which may not be needed for a specific database application, may have to be paid. This may degrade the performance of database processing. The concurrency control mechanism suited to each database application is preferred.

We have proposed a generation method of concurrency control program suited to a specific database application by using Genetic Programming (GP) (Tamura, Hochin & Nomiya, 2011; Tamura, Hochin & Nomiya, 2012a). The functions and the terminals of program, and the fitness measure function used in GP have been proposed. The functions and the terminals include those for the management of concurrent execution of database operations. This management is captured as procedures using the variables attached to data items and transactions. The experiments of the program generation showed that the popular locking algorithm could be generated under the concurrent environment, while the algorithm better than the popular locking algorithm could be generated under the not-so-concurrent environment (Tamura, Hochin & Nomiya, 2011). Moreover, the generation method has been extended to support semantic concurrency control (Tamura, Hochin & Nomiya, 2012a). It has also experimentally been shown that reducing functions and terminals is effective in generating appropriate concurrency control programs in GP (Tamura, Hochin & Nomiya, 2012b). This reduction is realized by limiting the usage of variables to one of two variables.

Although reducing functions and terminals has been shown to be effective, all of concurrency control programs could not be created. Making it possible to generate any concurrency control program is required.

This paper extends the variable used, which is important in generating concurrency control programs, to bring more information than before for creating any concurrency control programs. It is experimentally shown that an appropriate concurrency control program can successfully be generated by extending the variable.

The remaining of this paper is as follows: Section 2 describes GP and concurrency control mechanism. The generation method of concurrency control mechanism is described in Section 3. Section 4 extends the variable used for creating any concurrency control program. The experiment of the program generation is conducted in Section 5. Section 6 concludes the paper.

PREPARATION

Genetic Programming

Genetic Programming (GP) is a biologically inspired method that is able to create computer programs from a high-level problem statement (Affenzeller, Winkler, Wagner & Beham, 2009; Iba, Paul & Hasegawa, 2010). GP works on populations of solution candidates for a given problem and is based on Darwinian principles of survival of the fittest (selection), recombination (crossover), and mutation. GP generates the program suited to the situation.

In GP, computer programs are represented with rooted, labeled structure trees, which are called program trees in this paper. Nodes of trees are functions or terminals. Terminals are evaluated directly, while functions are evaluated after the evaluation of children’s nodes. Functions and terminals are called symbols in this paper. An example of a program is shown in Figure 1. The leaf nodes of the program tree
Preference Coalition Formation Scheme for Buyer Coalition Services with Bundles of Items


[www.igi-global.com/article/preference-coalition-formation-scheme-for-buyer-coalition-services-with-bundles-of-items/78918?camid=4v1a](www.igi-global.com/article/preference-coalition-formation-scheme-for-buyer-coalition-services-with-bundles-of-items/78918?camid=4v1a)