International Journal of Innovative Computing, Information and Control
Volume **5**, Number **2**, February **2009**

ICIC International ©2009 ISSN 1349-4198
pp. **263–277**

CREATION OF ET RULES FROM LOGICAL FORMULAS REPRESENTING EQUIVALENT RELATIONS

Katsunori Miura[1], Kiyoshi Akama[2] and Hiroshi Mabuchi[3]

[1]Faculty of Computer Science
[2]Information Initiative Center
Hokkaido University
Sapporo, Hokkaido, 060-0811, Japan
kmiura@uva.cims.hokudai.ac.jp; akama@iic.hokudai.ac.jp

[3]Faculty of Software and Information Science
Iwate Prefectural University
Takizawa, Iwate, 020-0193, Japan
mabu@soft.iwate-pu.ac.jp

Received December 2007; revised June 2008

Abstract. *The Equivalent Transformation (ET) computation model can create programs which are correct with respect to their specifications. In the ET computation model, a program is a set of ET rules, and it is guaranteed that a correct ET rule set will result in a correct program. In addition, since each ET rule is completely independent and individually correct, a program can be created by simply accumulating ET rules one by one. As a result, when viewed from the perspective of component-wise program creation, ET rules can confer considerable advantage. Therefore, the creation of correct ET rules is very important. The aim of this paper is to present a theory for creating correct ET rules, and propose a method for creating correct ET rules from a logical equivalence. A logical equivalence is one class of logical formulas and describes an equivalent relationship between atom conjunctions under some specified conditions. This paper examines the degree to which ET rules obtained using the proposed method can cover those created intuitively in actual programming. It also discusses the reasons why this method can create a broader range of ET rules than existing methods.*
Keywords: Equivalent transformation computation model, ET rule, Logical equivalence, Correctness, ET rule creation, Specification

1. **Introduction.** The Equivalent Transformation (ET) computation model [4] can determine the correctness of a program with respect to a specification and can also create correct programs. In the ET computation model, a procedure is described by a set of meaning-preserving rewriting rules, each of which is called an *ET rule*. A program is a set of ET rules. It is guaranteed that a set of correct ET rules is a correct program [2]. The high expressive power of ET rules with regards to procedures has been shown in previous studies. ET rules make it possible to correctly and easily create complex programs, such as programs solving constraint satisfaction problems and parallel programs [3]. In addition to extensive use in the domain of first-order terms, the ET computation model has been applied to knowledge processing systems in other data domains such as RDF [5], UML [18] and XML [22].

Other computation models, such as Logic Programming (LP) [16] and Constraint Logic Programming (CLP) [14,21] also give correct programs. They can create correct programs from correct problem descriptions. In LP and CLP, a problem is described by definite clauses (or extended definite clauses) and is solved by SLD resolution based on the problem