International Journal of Innovative Computing, Information and Control
Volume **7**, Number **8**, August **2011**

ICIC International ©2011 ISSN 1349-4198
pp. 4703–4716

ALGORITHMIC DEBUGGING OF EQUIVALENT TRANSFORMATION PROGRAMS USING ORACLE RULES

Shinya Miyajima[1], Kiyoshi Akama[2] and Hiroshi Mabuchi[3]

[1]Faculty of Engineering
Gifu University
1-1, Yanagido, Gifu-shi, Gifu 501-1193, Japan
miyajima@gifu-u.ac.jp

[2]Information Initiative Center
Hokkaido University
Kita 11, Nishi 5, Kita-ku, Sapporo, Hokkaido 060-081, Japan
akama@iic.hokudai.ac.jp

[3]Faculty of Software and Information Science
Iwate Prefectural University
152-52 Sugo, Takizawa, Iwate 020-0193, Japan
mabu@soft.iwate-pu.ac.jp

Received May 2010; revised September 2010

Abstract. *A debugging algorithm in equivalent transformation (ET) model is proposed. The proposed algorithm returns a bug rule in a program by receiving new rules from oracles (oracle rules). We construct theory about correctness of the proposed algorithm, present two strategies with respect to the requests of oracle rules, report examples to show the property of the proposed algorithm, and compare the proposed algorithm with the other debugging algorithms in the ET model.*

Keywords: Equivalent transformation model, Algorithmic debugging, Program correctness

1. **Introduction.** In many cases, bugs in programs are detected by programmers themselves. In this detection, they trace computational states and judge correctness of each computational state one by one. If the state can be traced and the correctness can be judged automatically, then bugs will be detected automatically. However, doing this judgement automatically is impossible in most programming languages, since correctness of each state is not known before execution of the program. Therefore, detecting bugs automatically is a difficult task in most programs.

In this paper, we consider detecting bugs automatically in equivalent transformation (ET) programs, which are the programs in the ET model proposed by Akama et al. [1, 2, 3, 4, 13, 14]. A program in the ET model is a set of rules for meaning-preserving transformation. Execution of programs in the ET model is a successive meaning-preserving transformation for given problems, and a problem solving process consists in successive rule application (see Figure 1). Namely, programming in the ET model is defined as creation of the set of rules. In [15], it is shown that using ET programs in component-based programming, i.e., using ET rules as the components, is effective. In this paper, we assume the situation where incorrect components are created in the component-based programming, and consider the method for detecting these incorrect components.

In the ET model, program correctness is defined as follows: If each rule in a program transforms each computational state, then each transformation preserves meaning of each state. From this correctness, we can conclude that the rule applied to the transformation which does not preserve the meaning is incorrect. From this principle, the theory about