

## 等価変換ルールの生成方法の理論的基礎

赤間 清† 小池 英勝† 宮本 衛市†

本発表では仕様からプログラムを合成する方法を提案し、その正当性を保証する理論を開発する。基礎となるのは「等価変換モデル」と呼ばれる新しい計算モデルである。等価変換モデルでは仕様は(拡張された)確定節の集合で表現され、プログラム合成の主要部分は仕様からの新しい等価変換ルールの生成である。等価変換モデルでは新しいルールを採用することによってプログラムを改善でき、最終的に効率の良いプログラムを得る可能性がある。Prologのプログラム変換では、確定節(仕様)が確定節(プログラム)に変換されるが、ルール生成はそれとは異なる。

ルールを生成するためにメタ記述を用いる。メタ記述は具体化によって得られる(無限の)記述を代表している。メタ記述が等価変換されるとは、それが代表するすべての記述が等価変換されることである。メタ記述の対から等価変換ルールが得られる。

メタ記述はメタルールによって変換する。アンフォールド変換、フォールド変換、その他の変換などに対応するメタルールをさせるので、提案する方法は論理プログラミングのアンフォールド/フォールド変換に劣らない効率化を達成することができる。本方法の最も重要な特徴の1つはモジュラー性である。すなわち、個々のメタルールは他のメタルールとはまったく無関係に正当性を判定できる。メタルールは任意の順序で適用できるが、論理プログラミングのプログラム変換ルールはそうではない。正しいメタルールの繰り返し適用によって等価な2つのメタ記述が得られ、それらから正当な等価変換ルールが得られることを証明する。

## A Theoretical Foundation for Generation of Equivalent Transformation Rules

KIYOSHI AKAMA,<sup>†</sup> HIDEKATU KOIKE<sup>†</sup> and EIICHI MIYAMOTO<sup>†</sup>

In this presentation we develop a theory to guarantee the correctness of a method of generating equivalent transformation rules from specifications. We use a new computation model, called the "Equivalent Transformation model" (ET model). In the ET model, specifications are represented by a set of (extended) definite clauses, and the main part of the synthesis of programs is the generation of new equivalent transformation rules from the specification. In the ET model, programs may be improved by adoption of each new rule, and finally an efficient program may be obtained. The rule generation is different from the program transformation in Prolog, where a set of definite clauses (specification) is transformed into a new set of definite clauses (which is regarded as a program in logic programming).

In order to generate rules, a new concept of meta-descriptions is used, each of which represents (infinite) descriptions by instantiation. A meta-description is said to be transformed equivalently iff all the descriptions that are obtained from it by instantiation are transformed equivalently. An ET rule is obtained from each pair of meta-descriptions that are equivalent.

Meta descriptions are transformed in terms of meta-rules. Since we can use many meta-rules corresponding to unfolding, folding, and other PT rules, the proposed method can increase efficiency in computation at least as much as unfold/fold transformation in logic programming does. One of the most important characteristics of the proposed method is modularity. A meta-rule is modular in the sense that its correctness can be judged independently of other meta-rules. Meta-rules can be applied in any order, while program transformation rules for logic programs can not. It is proven that repeated application of correct meta-rules produces two equivalent meta-descriptions, which are used to generate a correct ET rule.

(平成 11 年 3 月 24 日発表)

<sup>†</sup> 北海道大学大学院システム情報工学専攻  
Department of System and Information Engineering,  
Hokkaido University