

平成16年度 情報工学コース卒業研究報告要旨

坂部 研究室	氏 名	植 村 太 一
卒業研究題目	プログラム生成系 <i>GeneSys</i> による融合変換のための戦略	
<p>一般に仕様は、プログラムをつくるための十分な情報を含んでいるが、計算機でそのまま実行させることは不可能である。実行を可能にするためには、仕様からプログラムへの変換が必要であるが、この過程を自動的に行う手法は大変難しいと考えられている。近年研究が盛んなプログラム変換においても、実行不可能である仕様を対象とするものはない。</p> <p>仕様からプログラムへの自動変換を目指した新たな理論的枠組みとして、プログラム生成系 <i>GeneSys</i> が提案されている。<i>GeneSys</i> では、既知プログラムを関数型言語のモデルである項書換え系として扱う。一方、生成したいプログラムの仕様は、既知プログラムとの関係を示す一階述語論理式の集合で表現する。一階述語論理式の採用により、一般の等式集合では扱うことの出来ない論理和や存在限量子を用いることができるため、変換途中で必要となる情報を十分記述できる。プログラム生成は、6つの変換規則を用いて、既知プログラムとして与えられた項書換え系を参照しながら仕様を表現した論理式を変換することによって行っている。各々の変換規則について、それを適用した後の論理式が真であるならば、適用する前の論理式も必ず真であることが証明されている。したがって、変換後の論理式と等価な項書き換え系が得られるとき、項書換え系は仕様を満たすプログラムであることが保証される。</p> <p>しかし、<i>GeneSys</i> がどのような入力クラスに対しプログラム生成が成功するのかが完全に判明されていない。そのために、変換規則の適用順序を決めるための戦略が確立されておらず、プログラムの自動生成は行われていなかった。<i>GeneSys</i> の性質（扱う入力クラスや適用戦略など）を解明するには、実験を行い多くのデータを収集し解析する必要がある。よって、機械的にデータが収集出来るように、<i>GeneSys</i> の実装することは有益であると考えられる。</p> <p>本研究は、<i>GeneSys</i> の2つの変換規則 (Deduction, \forall-Expansion) を実装した。さらに規則適用戦略を提案し、これを実装した。結果として、合成関数を効率よく計算するプログラムを生成する Deforestation を <i>GeneSys</i> で自動的に行えるようになった。</p> <p>実装するにあたって一つの変換規則で複数の場所で適用できる場合に、どの場所に適用するのかが問題となる。本システムの入力を Deforestation が対象とするプログラムと仕様のみで制限することにより戦略を定めた。この戦略では Deduction の適用場所は、最左最内戦略で探し、\forall-Expansion で用いる被覆集合は、項書換え系の左辺の第一引数にある項としている。</p> <p>変換規則を用いるための条件は、論理式に単調性があることであるが、入力制限により \wedge と \forall と \approx のみを含む論理式文脈しか扱わないため、常に単調性が成り立つと仮定している。</p> <p>適用戦略を提案するために、Deduction を項書換え系を参照して行うものと、論理式を参照して行うものとに分けて使用することが、必要かつ十分であることがわかった。</p>		