

平成 18 年度 情報工学コース卒業研究報告要旨

酒井 研究室	氏 名	小 林 弘 昌
卒 業 研 究 題 目	高階書換え系における型付けによる非停止性の保存に関する研究	
<p>項書換え系 (Term Rewriting System; TRS) は, 項の書き換えの繰り返しにより計算を表現する関数型プログラムの計算モデルであり, 定理自動証明や代数的仕様記述, プログラム検証などに利用されている. 一方, TRS は関数プログラムで広く利用される高階関数を直接取り扱うことができない. そこで, 型なし項書換え系 (Untyped Term Rewriting System; UTRS), 単純型項書換え系 (Simply-Typed Term Rewriting System; STRS) など, 高階項の書き換えにより計算を表現する項書換え系が提案され, 研究が盛んに行われている.</p> <p>TRS の性質の一つに, 停止性がある. TRS が停止性をもつとは, TRS による項の無限書換え系列が存在しないことをいう. これは, プログラムがその計算過程によらず必ず停止することを意味しており, 非常に重要な性質である. しかし, 停止性は一般には決定不能であるため, TRS の停止性を示す方法についての研究が進められている.</p> <p>近年, STRS の停止性証明法に computational closure および強計算依存対法が提案された. これらは強計算性に基づく非常に強力な手法であるが, 強計算性は型上の再帰で定義されるため, Scheme のような型なし言語で記述された関数の停止性証明には応用できない.</p> <p>一方, 一階の TRS では, あるクラスの下で停止性が persistent であることが知られている. persistent とは, 与えられた TRS の規則が型付け可能であり, 型付けた項上の多ソート付き TRS (Many-Sorted Term Rewriting System; MS-TRS) が停止性を持つならば元の TRS も停止性を持つことをいう.</p> <p>本研究では, この成果を高階に拡張し, UTRS の停止性問題を STRS の停止性問題に帰着させる方法を 2 通りのアプローチで試みる. 一つはカーリー化に基づく高階書換え系の一階書換え系への 1 対 1 対応変換によるものである. 関数適用を表現する特殊な 2 引数関数記号を用いて, 高階項を一階の項に変換し, 高階の停止性問題を一階の停止性問題に帰着させるアプローチである. もう一つは, 一階の停止性問題の persistent の証明を, 高階の書換え系で再構築するアプローチである. 項の構造の複雑さを表す指標として項のランクを考え, 書き換えにより項のランクが増加しないような条件を洗い出す. この条件下で高階の書換え系においても停止性が persistent であることを示す.</p> <p>残念ながら, 上記 2 つの成果は共に map や fold のような代表的な高階関数に適用できない. これらの問題を解決するためには別の手段を試みる必要がある. C_e 停止性と呼ばれる概念は modular と呼ばれる性質を持つことが知られている. また, 一階の persistent の成果は modular 性をほぼそのまま再構築することにより得られているため, C_e 停止性も persistent であることが予想される. さらに, C_e 停止性の modular 性の適用条件の性質上 map や fold のような代表的な高階関数に適用できる成果が得られることが期待できる. これは今後の課題である.</p>		