

平成24年度 情報工学コース卒業研究報告要旨

酒井 研究室	氏 名	栗 木 隆 太 朗
卒業研究題目	Erlang 言語を用いて実装した完備化手続きにおける並列実行の評価	
<p>関数型プログラムの検証技術の一つとして、項書換え系 (TRS) の定理自動証明がある。定理自動証明では、適切な戦略や順序を与えることで証明を発見できるが、問題ごとに戦略や順序を試行錯誤しなければならない。順序に関する試行錯誤を省くために、停止性証明ツールを用いる方法があるが、証明の序盤には左右どちら向きにも順序が付く可能性があるため、戦略における選択肢の幅が大きくなる。この試行錯誤を自動化する単純な方法は、複数の戦略を並列に実行し、さらに戦略中での選択肢についても証明プロセスを複製し、すべての選択肢を試すという方法である。この方法は人間の試行錯誤のコストを軽減する一方で、計算機における実行コストは非常に大きい。</p> <p>近年、CPU コア数の増加に伴い、Erlang という言語が注目されている。Erlang は並列処理指向の関数型言語であり、大量の並列処理を素早くこなすことを得意としている。</p> <p>本研究では、定理自動証明を Erlang で実装することで、試行錯誤のコスト、すなわち、実行時間を軽減できるかを調査する。まずその準備として、完備化手続きを Erlang で実装し、並列実行を導入することによる効率の改善を評価する。ここで、完備化手続きとは、停止性を保証するための順序と等式集合を与えられ、その等式集合と等価な、停止性と合流性を持つ TRS を求める手続きである。完備化手続きの応用の一つに定理自動証明があり、定理自動証明で用いる推論規則および適応戦略は、完備化手続きのものと同様である。また、完備化手続きは定理自動証明よりも推論規則が複雑でないため、実装が容易で、実装段階での性能の差異が少ない。そのため、並列実行による効率化を評価する観点から、定理自動証明よりも本研究の目的に適している。</p> <p>まず、Standard ML of New Jersey で実装した完備化手続きを Erlang へ単純に移植する。本研究では、並列化の対象として、ブール値を返す関数、特に辞書式経路順序による順序付け判定関数 lpo に注目する。論理積や論理和は一つの論理式の結果で全体の結果が決まる場合がある。しかし、逐次的に計算を行う場合、最初の論理式を評価し、その結果で全体の値が確定すればそれを結果として終了し、そうでなければ次の論理式を評価し始める。lpo の定義では多くの論理積と論理和を用いる。これを並列実行したときの効果を検証するため、lpo の並列実行の割合を変えつつ実行時間の比較を行った。その結果、可能な箇所すべてに並列処理を施すと、実行時にプロセス不足が起こることが分かった。また、並列化後の実行時間はあまり変わらず効率に変化はなかった。これは、コストの小さい計算では、並列処理を管理するコストが並列化による効果を相殺するためと考えられる。</p> <p>次に、定理自動証明での場合分けによる試行錯誤を想定し、完備化手続きの方向付けにおいて選択可能な等式が複数あった場合、すべての場合をリストで保持し、以降の処理をリストの要素全てに適用するように変更する。このように変更した手続きと、リストで保持せずに独立したプロセスで並列実行したものとを比較実験を行った。その結果、推論規則の適用回数はリストで保持する方が少ないが、実行時間は並列実行する方が圧倒的に短いことが分かった。定理自動証明においては、推論規則の適用回数より実行時間の方が重要視される。また、完備化手続きを変更する際、プロセス不足を防ぐためにプロセスの序盤に選択可能な等式が複数あったときのみ全てリストで保持し、以降は元の方法で手続きを進める。定理自動証明では序盤は場合分けが多く、終盤は少ないという場合が多いので、このような並列処理は定理自動証明と相性が良いと言える。以上の結果から、定理自動証明を Erlang で実装することで大きな効率化が期待できると考えられる。</p>		