

平成 29 年度 情報工学コース卒業研究報告要旨

酒井 研究室	氏 名	西 江 一 志
卒業研究題目	書換え解析ツールを利用した C 言語プログラムの漸近的計算量と実行命令数の解析	
<p>複数の計算機やコアにタスクを割り振り並列処理を行う際、それぞれのタスクの処理に要する時間はあらかじめ決められた要件を満たすように割り振りを決めるための有用な情報である。よって、タスクを記述しているプログラム（やその断片）について漸近的計算量だけでなく実行される命令数をできる限り正確に見積もることができれば、より適切なタスクの割り振りを可能にすることを期待できる。</p> <p>項書換え系の分野では、与えられた項が正規形に書き換えられるまでの書換えのステップ数の漸近的計算量を求める手法が研究されている。また、停止性証明ツールである AProVE に計算量解析の機能が追加されたり、TcT のような計算量解析を目的としたツールが開発されたりしている。さらに、AProVE や TcT は整数上の制約付き書換え系の 1 つとみなせる整数状態遷移系 (ITS) の書換えステップ数の計算量解析も行える。制約付き書換え系は項書換え系の書換え規則に論理式を制約として付随して拡張した計算モデルであり、命令型プログラムを変換して得られた制約付き書換え系を検証することで元のプログラムを検証する手法が研究されている。変換して得られた制約付き書換え系の書換えステップは元のプログラムの命令の実行に対応している。よって、命令型プログラムを変換して得られる制約付き書換え系の書換えステップ数の計算量を AProVE や TcT のような書換え解析ツールによって解析できれば、命令型プログラムの実行命令数の解析を行えることが期待できる。しかし、AProVE や TcT が解析対象とする ITS では並列処理が可能な状態遷移しか記述できず、整数と変数の演算式しか許されていない。そのため、配列や構造体を含むプログラムを ITS に等価変換し解析することはできない。</p> <p>本研究報告では、配列や構造体を含む C 言語プログラムの漸近的計算量や正確な実行命令数を解析するために書換え解析ツールを応用する手法を提案する。ここでは、プログラムの実行命令数はそのプログラムをコンパイルして得られる LLVM コードで実行される命令数とする。</p> <p>まず、配列や構造体を含む C 言語プログラムを漸近的計算量を保存する配列や構造体を含まないプログラムに簡易化し ITS へ変換する手法を提案し、それにより、AProVE や TcT によってベンチマークの漸近的計算量が正しく解析できたことを報告する。一方、プログラムを簡易化したことで元のプログラムと比較して実行命令数が少なくなるため簡易化したプログラムから得られた ITS では元のプログラムの正確な実行命令数を解析できない。これを解決するために元のプログラムの実行命令数と書換えステップ数が一致するように ITS の書換え規則の補完を試みる。この書換え規則の補完方法を 3 種類提案する。1 つ目は変数の値を変化させずに遷移するだけの書換え規則を増やす方法である。2 つ目は補完分の遷移をするための書換え規則を追加し通常の手書換えと補完分の書換えを並列に行う規則を導入する方法である。3 つ目は項の引数を 1 つ増やし制約を用いて引数回分の遷移をさせ補完する方法である。これらの方法で ITS の補完を行い AProVE により解析した実行命令数の精度を検証する。</p> <p>上記の補完方法により、変換して得られる ITS の書換え規則が増え可読性が低下する。補完の目的は書換えステップを実際の実行命令数と一致させることであるので、規則を補完するのではなく規則に重みを付与できる枠組みを提案する。また、ITS の計算量解析では制約による計算量は考慮されていない。そこで、提案する枠組みでは制約に重みを与えられる定式化を行う。</p>		