

論文の内容の要旨

論文題目 Automatic Analysis, Verification, and Generation of Functional Programs
(関数型プログラムの自動解析・検証・生成)

氏名 小川 瑞史

関数型プログラムは、もっとも古いプログラミング言語の一つである Lisp にさかのぼるプログラミングパラダイムであり、近年大規模なシステムの実装にも用いられるようになってきた。実装の影響が最小限であるプログラムの宣言的意味、構文的に局所化される束縛変数などの理論的な扱い易さに基づき、本論文では関数型プログラムの自動作成支援として三部構成で自動解析・検証・生成について述べている。

三部構成の前の 1 章ではそれぞれで提案する手法を例に基づき概観し、最後の 7 章で将来の研究の展望について述べる。

第一部では、検証対象としてもっとも難しい性質の一つである停止性検証について、既に知られている自動検証法が失敗した時に、停止性をより証明が簡単な弱停止性に還元するための十分条件を述べる。停止性の自動検証として、書換え系の単純停止性に基づく経路順序の探索、およびその探索を局所化・単純化するための解析・変換手法が知られている。しかし、それらの手法の有効性は主として一階の書換え系の場合に限られ、高階の書換え系の場合にはあまり成功していない。それに対し、関数型プログラムを定式化する高階書換え系は直交なものを考えれば十分なので、直交な高階書換え系において停止性をより簡単な弱停止性に還元するための十分条件を提案した (2 章)。この十分条件により過去に個別に提案されていたさまざまな高階書換え系の停止性を統一的に簡単な系として導くことができる。

第二部では、人間がかいたプログラムのデバッグ・最適化のためのコンパイル時解析・検証を抽象実行の手法に基づき提案した。最初に、抽象実行の構成法について定式化を行い、抽象実行の複数の結合を一つの抽象実行に還元可能とする十分条件を明らかにした。これにより、従来提案されてきたさまざまな解析手法の解析力の比較を行った。さらに、

この定式化に基づき、単項関係の必要条件に基づく抽象化から導かれる解析として計算経路解析 (3 章)、二項関係の十分条件に基づく抽象化から導かれる検証として Le Metayer による自動検証法の拡張を与えた (4 章)。前者の応用としてプログラム内の未定義性や不要性を検出する変則性検出、後者の応用としてソーティングのプログラムの正しさの検証を示した。

第三部では、効率的なプログラムを自動生成する手法について提案した。生成されたプログラムの正しさは自動的に保証されるが、プログラムの自動生成はしばしば計算不能となる。ここでは、組み合わせ理論の成果を応用し、適切な問題領域を切り出すことで線形時間アルゴリズムを生成する。組み合わせ理論に基づく線形時間アルゴリズムには、存在証明 (証明が非構成的なためアルゴリズムが生成できない)、理論的な線形時間アルゴリズム生成 (線形時間でも定数が巨大)、実用的な線形時間アルゴリズム生成の三段階がある。本手法では、構成的証明による理論的な生成手法に関数型プログラムの変換手法 (融合・組化変換) を援用することで実用的な線形時間アルゴリズム生成手法の確立を目指している。その具体的な問題として、Well-Quasi-Order (WQO) に基づく不完全な関係データベースの質問処理の線形時間アルゴリズム生成、およびグラフの木分割に基づく木幅に上限のあるグラフの最大和問題の線形時間アルゴリズム生成を設定した。前者については理論的な線形時間アルゴリズム生成 (5 章) を、後者についてはグラフにおけるプログラム変換手法を確立するための第一段階としてグラフの代数的構成法およびその完全な公理化 (6 章) を与えた。