

学位論文題名

PROLOG プログラム変換に関する論理学的研究

学位論文内容の要旨

プログラミング言語をより非手続き的にしようという試みはこれまで主として2種類の言語クラスを生んできた。一つは関数型の言語クラスであり、他の一つは関係型の言語クラスである。これらは、出発点として、何らかの論理系の論理式をプログラムの表現として用い、計算機構として論理の推論機構の一部を採用しているという意味において、総称的に論理型言語という名で呼ばれている。これらの言語によって、プログラマはプログラムの手続き的な詳細や効率面等にあまり関心を払うことなく、むしろ計算対象となるデータとその構造、及びその上のオペレーションの間の論理的関係により焦点を当ててプログラミングを行うことが可能となってきた。その結果、そのような言語は、プログラムとそれが解こうとしている問題の関係を理解するのがより容易で、したがって、信頼性の高いプログラムの作成等に役立ち、主にソフトウェア工学上の諸問題に寄与するところが大きいものと期待されている。他方、その代償として、そのような言語のインタプリタやコンパイラに対しては、言語に固有のより効率的なインプリメント技法や最適化手法等が強く求められることになった。

本論文は、非決定的なPrologプログラムをソースレベルで最適化変換するために、論理学的考察を踏まえて得られた数々の手法と、それらを一つに統合して構成されたPrologオブティマイザを提案し、この課題に対して一つの有力で実地的な解答を与えるものである。

Prologプログラムは、ゴールと呼ばれる手続きの呼出し、及びそれを定義している手続きの集まりから構成されている。したがって、プログラムのソースレベルでの最適化手法としては、最適化の対象となるスコープによって、(1)ゴールの定義体による展開、(2)論理式としての節の変形、(3)述語定義体単位の変形、などが考えられる。

(1)は通常のプログラム言語ではサブルーチン展開とかインライン展開と呼ばれ、複数プロシージャ間での最適化手法の一つであり、計算の部分実行に相当する。(2)は一つの節内に限って、冗長性の除去を行う局所的最適化手法である。(3)は一つの述語定義体内での最適化に関するものである。これらの最適化手法には、いくつかの適用条件が必要になる。中でも述語呼出しが決定的であるか否かを検出することが重要となる。また、Prologはバックトラックに基づく非決定性を言語の大きな特徴としているにも拘らず、それに起因する言語処理系の空間的・時間的負荷はかなり大きい述語の決定性検出は、そのような負荷を軽減する最適化手法において有用になるものである。このような理由から、本論文では述語（呼出し）の決定性の定義とその数学的性質を特に綿密に論じた上で、最適化手法の研究を行った。

本論文のPrologオブティマイザは次の三つの方針の下に構成された：(i)最適化のための情報抽出、(ii)インライン展開、(iii)局所的最適化。まず、Prologプログラムの最適化に当たって、あらかじめプログラムテキストより最適化に必要な情報が抽出される。本論文の目的のためには、それらはプログラムの型と決定性に関する情報である。ユーザプログラムの各述語は次のいずれかの型に分類される：直線型、終端再帰型、一般再帰型。そして本論文で与えた述語の α -決定性と r -決定性なる概念に従って述語が決定的であるか否かが検出される。次に、述語呼出しの決定性情報を用いて、不必要なバックトラックが起り得る場所にカット記号が挿入される。これにより、知的バックトラックの簡単な場合が達成されることになる。その後、サブルーチンリンクのオーバーヘッドを除去すること、および局所的最適化手法に対してより大きなプログラム単位を与え最適化の機会を増すことを目的として、プログラムのインライン展開が行われる。Prologのインライン展開はPrologプログラムの実行メカニズム“最左及び深さ優先”に従って、述語呼出し（ゴール）をその定義節の選言項で置き換えるという方法で行われる本論文で考案された局所的最適化変換手法には次のようなものがある：部分ユニフィケーション、ゴール列の単純化（多重連言・多重選言の除去、冗長述語 'true'、'fail'の除去、不実行ゴール列の除去、共通ゴールのくくり出し）、等式代入による冗長変数およびゴールの除去、等式ゴール列の統合化、節の分解

以上の最適化手法を一つに統合化することによって、Prologオブティマイザを試作し評価を行った。

いくつかの典型的な例による性能評価実験によると、本論文のオブティマイザは平均20%位の速度効率を、明らかに冗長と思われるプログラムに対しては40%～60%の速度効率を達成することを示した。

また、本論文では、様々な論理系を扱うことのできる汎用論証支援システムEUODHILOSの諸機能とその上で行なわれた様々な論理定義、証明例による評価についても述べた。そして、このシステムの論理型プログラムの変換システムとしての利用法について論じた。

本論文は次のように構成されている。第1章ではPrologプログラムの最適化変換に向けての本論文の方針を述べた。第2章では最適化図式で用いられる記法とプログラムの型、決定性の定義を与えた。特に、述語（呼出し）の決定性については、その異なった定義法及びいくつかの性質について詳しく論じた。第3章では各種の最適化図式とその適用条件を与えた。第4章ではオブティマイザのインプリメンテーションを正当化するための有用な結果と議論を述べた。また、決定性の非可解性に関する結果も含めた。第5章ではPrologで書かれたPrologオブティマイザの概要を記述した。第6章ではいくつかの最適化の例とその時間評価を述べ、本論文の方法の有効性を実証した。第7章では汎用論証支援システムEUODHILOSとその論理型プログラムの変換への応用について述べた。第8章はこの分野の最近の研究の現状を簡単に紹介している。ここでは本論文の手法とこれらの研究との相違点を指摘し、我々の方法の特徴利点を明らかにした。第9章において、本論文の主要な寄与をまとめ今後の課題と展望を与えた。

学位論文審査の要旨

主 査	教 授	宮 本 衛 市
副 査	教 授	新 保 勝
副 査	教 授	佐 藤 義 治
副 査	教 授	田 中 讓

学 位 論 文 題 名

PROLOG プログラム変換に関する論理学的研究

Prolog は、新世代のプログラミング言語あるいは知識情報処理を支える言語として注目されている。そして、これまで、Prolog のさまざまな拡張やその効率的な言語処理系の研究が盛んに行われてきた。しかしながら、Pure Prolog と言われる制限された Prolog を除き、実際の Prolog プログラムを効率化のためにソースレベルにおいて最適化変換するという研究課題は依然として残されてきた。

本論文は、こうした状況の下で、非決定的な Prolog プログラムをソースレベルで最適化変換するさいに鍵となる述語の決定性、さまざまな変換手法、それらの有効性を確認するために構成された Prolog オプティマイザ、さらに汎用論証支援システムとそのプログラム変換への応用に関する研究成果をまとめたものである。

Prolog プログラムは、ゴールと呼ばれる手続きの呼出し、およびそれを定義している手続きの集まりから構成されている。したがって、プログラムのソースレベルでの最適化手法としては、最適化の対象となるスコープによって、(1) ゴールの定義体による展開、(2) 論理式としての節の変形、(3) 述語定義体単位の変形に分けて考えることができる。しかしながら、いずれの場合においても、いくつかの適用条件が要求される。中でも述語呼出しが決定的であるか否かを検出することが必要となる。Prolog はバックトラックに基づく非決定性を言語の大きな特徴としているにもかかわらず、それに起因する言語処理系の空間的・時間的負荷はかなり大きく、述語の決定性検出は、そのような負荷を軽減する最適化手法において必須となるものである。このために本論文では、まず述語および述語呼出しの決定性の定義とその数学的性質を特に綿密に述べている。

次に、Prolog オプティマイザを次の3つの方針の下に構成している：(1) 最適化のための情報抽出、(2) インライン展開、(3) 局所的最適化。まず、Prolog プログラムの最適化に当たって、あらかじめプログラムテキストより最適化に必要な情報が抽出される。本研究の目的のためには、それらはプログラムの型と決定性に関する情報である。ユーザプログラムの各述語は次のいずれかの型に分類される：直線型、終端再帰型、一般再帰型。そして本論文で与えた述語の a -決定性と r -決定性なる概念に従って、述語が決定的であるか否

かが検出される。次に、述語呼出しの決定性情報を用いて、不必要なバックトラックが起こり得る場所にカット記号が挿入される。その後、サブルーチンリンクのオーバーヘッドを除去すること、および局所的最適化手法に対して、より大きなプログラム単位を与え、最適化の機会を増すことを目的として、プログラムのインライン展開が行われる。Prolog のインライン展開は、Prolog プログラムの実行メカニズム”最左及び深さ優先”に従って、述語呼出し（ゴール）をその定義節の選言項で置き換えるという方法で行われる。局所的最適化変換手法としては次の手法が考案された：部分ユニフィケーション、ゴール列の単純化（多重連言・多重選言の除去、冗長述語'true' および'fail' の除去、不実行ゴール列の除去、共通ゴールのくくり出し）、等式代入による冗長変数およびゴールの除去、等式ゴール列の統合化、節の分解。

さらに、以上の機能をもつオブティマイザの有効性を検証している。いくつかの典型的なプログラムに対しては、平均 20%位の速度効率を、明らかに冗長と思われるプログラムに対しては 40%～60%の速度効率を達成している。

また、本論文では、様々な論理系を扱うことのできる汎用論証支援システム EUODHILOS の諸機能とその上で行なわれた様々な論理定義、証明例による評価についても述べている。そして、このシステムの論理型プログラムの変換システムとしての利用法を示している。

以上のように本論文は、述語呼び出しの決定性が Prolog プログラムの最適化変換問題にとってキーとなる重要概念であることを明らかにし、現実の Prolog プログラムに対する最適化変換法と変換システムの構築に関して、論理的な見地より有益な知見を与えており、情報工学の進歩に貢献するところ大なるものがある。

よって著者は、北海道大学博士（工学）の学位を授与される資格あるものと認める。