

研究種目：若手研究（B）

研究期間：平成19年度 ～ 平成22年度

課題番号：19700021

研究課題名（和文） プログラム変換に基づく対話的プログラム開発システム

研究課題名（英文）

研究代表者

篠埜 功 (SASANO ISAO)

芝浦工業大学・工学部情報工学科・助教

研究者番号：10362021

研究分野：総合領域

科研費の分科・細目：情報学・ソフトウェア

キーワード：プログラミング言語、プログラム変換、プログラム開発環境、変数名補完

1. 研究計画の概要

プログラム変換は、仕様からのプログラムの導出、コンパイラにおける最適化などにおいて有用であることが知られている。変換規則としては、関数の展開、畳み込み、具体化、一般化といった基本的なものや、関数融合、組化等を行うための、より複雑な規則までさまざまである。コンパイラに入れるプログラム変換は有用であるが、すべてのプログラムに一律に適用されるため、効率が悪くならないことを保証するために変換機能が限定されてしまうという一面がある。本研究においては、プログラム変換の側面からプログラム開発を支援する、対話的プログラム開発システムの作成を目的とする。本研究はこれまでに行われてきたプログラム変換、型システムの研究成果を用いて対話的プログラム開発システムを構築するものである。既存のプログラム変換の研究においては、プログラム一般に有用な変換については言語処理系に埋め込まれているが、そうでない変換については紙の上で行われるだけにとどまっているのが現状である。そこで従来紙の上で行っていたプログラム変換を実際のプログラム開発に用いることを目指している。

2. 研究の進捗状況

まず既存の代表的プログラム開発システムを調査した。現在広く用いられているプログラム開発環境としては、商用のものでは Microsoft の Visual Studio、フリーな開発環境では Eclipse が広く使われている。これらでサポートされているのは現在広く使われている C, C++, Java 等の言語である。現在、自動インデント、変数名補完等の基本機能に

加え、変数名の付け替え等のリファクタリング機能が実用化され、広く用いられている。本研究はプログラム開発においてプログラム変換の機能を変数名付け替え等のリファクタリングと同様に適用できるようにする。対象言語としては近年徐々に広まりつつある関数型言語を念頭に置いている。関数型言語にもさまざまなものがあるが、純粋な関数型に限定せず、副作用も持つものを対象とする。型については型付きと型無し言語があるが、本研究では静的型付き言語でかつ型の記述をソースプログラム中に必ずしも与えなくてもよい、型推論機構を持つ言語とする。これは型無し言語の柔軟さと型付き言語によるある意味における正当性の保証との両方のメリットを併せ持つ言語であり、現在 Standard ML, Haskell, Objective Caml 等の言語が主に研究者の間で実際に使われている。これらは C 言語で書いた場合と比べ一般に少し効率が落ちるが、現在コンピュータの速度は大幅に上がっており、ごく一部の本当に速度が要求される部分やオペレーティングシステムなどの低レベルなプログラム以外では十分に実用になる。またプログラム量が C 等に比べ一般に格段に少なくプログラム開発効率が高い。これらの利点により、上記の言語は今後一般に広まることが期待されている。これらの言語の開発環境は C や Java 等のメジャーな言語に比べ整備されていないのが現状である。また、これまでのプログラム変換の研究において主に関数型言語を対象として研究されてきたこともあり、本研究ではこれらの言語に対するプログラム開発環境を構築することを目標としている。関数型言語に対する開発環境はこれまで

整備されていなかったこともあり、開発環境の基本的機能も十分には提供されておらず、また、静的型付き言語であることのメリットが十分に生かされてもいない。そこで開発環境の最も基本的な機能の一つである変数名補完について検討を行った。型推論機構を持つ言語という特徴を生かすため、変数名補完時に型推論を行い、その結果を提示する候補の絞り込みに反映させるという問題を設定し、その解法の検討を行った。まず、問題を単純にするため、カーソル位置までのプログラムは完全に与えられていると仮定し、カーソル位置以降のプログラム情報は一切用いないという問題設定とした。この条件下で構文解析を行うため、ダミー要素とカーソル要素という抽象的要素を導入した。これらを用いてカーソル位置をカーソル要素、カーソル位置より後のプログラムをダミー要素を使って補い、あらゆる可能な補い方で構文木を作成するという方針とした。結果として得られる構文木それぞれを抽象構文木に変換し、型推論を行った結果を用いて、カーソル位置において有効な変数から型が合わない変数を除外することにより補完候補となる変数を求める。この方式では一般に構文木の補い方が無限にあり、何らかの基準で有限個に抑える必要があるが、最も単純な基準として、構文木の深さに閾値を設けることにより生成する構文木を有限個とした。この成果を第12回プログラミングおよびプログラミング言語ワークショップにおいて発表した(5. 代表的な研究成果を参照)。

3. 現在までの達成度

③やや遅れている。

現在はプログラム開発システムにおいて基本となる機能の1つである変数名補完について整理し、新しい方式の提案および核言語に対する Emacs 上での実装を与えたところである。当初の計画からは遅れてはいるが、実用的プログラム開発システムとして備えているべき最も基本的機能である変数名補完方式を基本的枠組みから整理し、核言語上での実装を与えており、それ自体が重要な成果となっている。開発した変数名補完機能はこれまでにない方式を考案し、それに基づいて実装したものであり、独創的なものである。

4. 今後の研究の推進方策

今後、変数名補完方式をさらに整備し、実用的な補完方式を構築する予定である。2で述べた補完方式は無限の構文木を有限個に抑えているため、本来候補になるべき変数が候補にならない場合がある。また、生成する構文木の個数が膨大であり計算時間がかかっている。そこでこれらの問題を解決するため、構文木補完と型推論フェーズを連携させ、生

成する構文木の個数を抑え、かつすべての候補を提示する方式を考案する。この方式は特定の核言語においては既に構築されており、Emacs モードとしての実装もできている。そこでこれをさらに発展させ、言語の構文をパラメータとした補完候補作成プログラムの系統的作成手法の構築を目指す。さらに、現状では一文字打つ毎に補完候補計算が最初から行われているが、一度行った計算を再利用する形に改良する。計算の再利用には、構文解析と型推論の大きく分けて2つあり、どのように計算を保存し再利用するか考案し実装する。ここまでできれば実用的な変数名補完機能の基礎が完成である。現実の言語に応用する際にはキーワードの補完も重要であり、これを変数名補完機能に組み入れる。また、現状では型の記述がプログラム中に一切ないことを前提としているが、実際には部分的に型の記述を与えても良く、これに対応させる。また、相互再帰関数などにおいてはカーソル位置より後に変数の束縛が行われることがあり、この場合に対応させる。また、カーソル位置より前のプログラムが完全であるという仮定は現実には厳しい仮定であり、構文解析のエラー回復などの機能を参考に不完全なプログラムに対して意図した補完候補を出されるようにする。変数名補完の他には変数名の付け替えが最も基本的な機能である。変数名の付け替えも人間にとっては自明な変換だが、これを正確かつ効率よく行うアルゴリズムは自明ではない。これについても変数名補完同様、基本的枠組みから整理し、実装を行うことを考えている。これらの基盤の上に基本的なプログラム変換をプログラム開発時に適用することをサポートするシステムを、基本的枠組みから整理しプログラム開発システムに組み入れる。

5. 代表的な研究成果

(研究代表者、研究分担者及び連携研究者には下線)

[学会発表] (計 1 件)

後藤 拓実, 篠埜 功, 多相型言語の変数名補完を行う Emacs モードの開発, 第12回プログラミングおよびプログラミング言語ワークショップ(PPL2010) 論文集, pp. 177-190, 香川県 琴平温泉, 2010年3月3日~5日.

[その他]

Web page

<http://www.cs.ise.shibaura-it.ac.jp/complement/>

において、let 多相の関数型言語の核言語に対しての変数名補完の Emacs モードによる実装を公開している。