

**科学研究費助成事業 研究成果報告書**

平成 28 年 5 月 30 日現在

機関番号：32619

研究種目：若手研究(B)

研究期間：2013～2015

課題番号：25730047

研究課題名(和文) 暗に型付けられた言語に対する型主導コード補完に関する研究

研究課題名(英文) Type directed code completion for implicitly typed languages

研究代表者

篠埜 功 (Sasano, Isao)

芝浦工業大学・工学部・准教授

研究者番号：10362021

交付決定額(研究期間全体)：(直接経費) 1,600,000円

研究成果の概要(和文)：コード補完はプログラミングにおいて基本的で有用な機能であり、EclipseやVisual Studioなどの開発環境において広く用いられており、コード補完が有効に用いられることはプログラミングの効率に大きな影響を及ぼす。本研究では関数型言語の核言語について型を考慮した識別子補完を問題として定式化し、それを解くアルゴリズムを考案し、そのアルゴリズムが満たす性質を示した。また、入力中の不完全なプログラムに対応するためにLR構文解析の誤り回復機能を用いた方式を考案、実装した。さらにその方式をもとにキーワード補完プログラムを一部機械的に導出する方式を考案し、Yaccのソースコードを用いて実装した。

研究成果の概要(英文)：Code completion is a basic and useful functionality in programming and is widely used in IDEs like Eclipse and Visual Studio. Effectively using code completion makes a substantial influence on efficiency of programming. In this research we specified an identifier completion problem for a core functional language with taking into account types, designed and implemented an algorithm for solving the problem, and showed some properties of our algorithm. We also proposed a method for coping with incomplete program text being currently edited, utilizing error recovery in LR parsing, and implemented the method as an Emacs mode. Based on the method we proposed a method for partially deriving a program for completing keywords from a specification and implemented the method by modifying the source code of Yacc.

研究分野：プログラミング言語

キーワード：コード補完 変数名補完 識別子補完 開発環境 暗に型付けられた言語 関数型言語 LR構文解析  
誤り回復

## 1. 研究開始当初の背景

コード補完はプログラミングにおいて基本的で有用な機能であり、Eclipse や Visual Studio などの開発環境においてよく用いられており、コード補完が有効に用いられることはプログラミングの効率に大きな影響を及ぼす。静的型付き言語においては型の整合性がコンパイル時に検査できることが大きな利点であるが、型情報が開発環境において用いられないことは、プログラムの開発効率に潜在的に大きな影響を与えている。コード補完機能に関しては、もし型情報が考慮されていれば、補完候補の提示の際には型が合わない候補を除外することにより余分な候補を表示しないようにすることができる。静的型付き言語は型をプログラム中に記述する際に型付けられた C や Java 等の言語と、プログラム中に型を必ずしも記述しなくてもよい暗に型付けられた Haskell や Standard ML 等の言語に分けられる。暗に型付けられた言語には型推論という機構が備わっており、これによりプログラム中に型を記述しなくても、プログラムのコンパイル時に型が定まる。このような暗に型付けられた言語に対し、どのようにコード補完を行うと効率よくプログラミングできるかは自明ではない。

既存の Eclipse、Microsoft 社の Visual Studio 等の開発環境においては、変数名付け替えや変数名補完等の基本機能さえも ad-hoc に実装されており、場合によっては機能が間違ったまま提供されている場合がある。このように、現在広く用いられている開発環境はプログラミングの効率向上に十分に役立っているとは言い難い状況にある。

## 2. 研究の目的

上記の問題を解決する第一歩として、暗に型付けられた言語に対し型情報を有効に用いたコード補完機能を実現するための基本方式を提案し、それに基づいて Standard ML 等の言語に対してコード補完機能を実装することを目的とする。

## 3. 研究の方法

本研究課題は、コード補完の基本方式を確立し、実用的なシステムとして実装するというものである。研究の方法としては、以下の項目 A、B で、理論・実装の両面において研究を推進する。

[研究項目 A(理論)]: コード補完の基本方式の確立

我々の既存研究で既に提案した変数名補完の基本的な枠組みを発展させ、式を補完する方式を考案する。その際、計算の再利用およびエラーへの対処に関して検討を行い、効率良く実用的な補完アルゴリズムを開発する。

[研究項目 B(実装)]: 実用的なシステムの実装

上記基本方式に基づき、Emacs 等のエ

ディタにおいて、実際に使えるコード補完機能の実装を行う。また、実装においては対象言語のコンパイラ中のコードの再利用を検討する。

各研究項目について以下のように小項目を設定して研究を推進する。

(1)研究項目 A: コード補完の基本方式の確立  
小項目 A1: 計算の再利用について

暗に型付けられた言語のコード補完において型情報を用いるためには、字句解析、構文解析が必須である。プログラミング時にはキーを一文字打つ毎に状況が変化するため、素朴に最初からすべての計算をやり直すと毎回ほとんど同じ計算を行うことになり無駄である。無駄な計算のうち、字句解析、構文解析に関しては、計算の再利用を一般的に論じた既存研究が多くある。これらを参考に、コード補完に適した字句解析、構文解析に関する計算再利用方式を考案する。型推論の計算の再利用に関する研究はほとんど見られないが、数少ない既存研究として[1,2] が知られている。だた、[1] は型検査の計算の再利用、[2] はプログラムを宣言の単位で型推論するという研究であり、コード補完に関する型推論計算の再利用には直接は適用できない。そこで、コード補完に適した、型推論に関する計算の再利用方式について考案する。

小項目 A2: 不完全なプログラムへの対応について

我々の既存研究[7,8,9]で提案した変数名補完の基本的枠組みは編集時のプログラムにおいてカーソル位置までは構文および型に関して正しいという前提で構築されている。これは現実のプログラミングにおいては厳しい仮定であり、書き間違いがあったり、ある部分を書いている途中で別の部分を編集するということが実際には行われる。構文が不完全な部分については通常の構文解析のエラー回復機能を参考にし、構文を補うか一部削除する等してコード補完を行うことを検討する。また、型エラーが含まれる部分については、型エラーの原因となる箇所を検出する型エラースライシングという手法[3]や、型エラーの原因となる箇所をワイルドカードで置き換えることにより良いエラーメッセージを生成する研究[4]があり、それらを参考に対処方法を検討する。

小項目 A3: コード補完方式について

関数型言語について、与えられた型を持つ式を生成する既存研究がある[5,6]。これらを参考に、型が明示的に与えられない場合やプログラムが不完全な場合に対するコード補完方式を確立する。

小項目 A4: 計算の再利用の属性文法による定式化について

言語の意味の定義法として Donald E. Knuth が考案した属性文法の考え方がコンパイラ構築において広く用いられている。属性文法の一つの重要な適用例として構造工

ディタが挙げられる。構造エディタは構文を編集作業に反映させるエディタであり、一般に、構文に関連した計算の再利用の定式化に属性文法が適していると考えられる。そこで、開発するコード補完機能の属性文法による定式化を試みる。

#### (2) 研究項目 B: 実用的なシステムの実装

##### 小項目 B1: コンパイラとの連携

現在の我々の既存研究での核言語に対する変数名補完システム [9] は Emacs 上で Emacs Lisp を使って直接記述されたものであり、少なくとも構文解析についてはコンパイラの構文解析フェーズのコードを再利用するのが望ましい形であると考えられる。また、型推論については、コード補完においては部分的型推論を行う必要があり、コンパイラのコードを直接用いることはできないが、部分的にコンパイラのコードを使えるかどうかについて検討する。

##### 小項目 B2: 属性文法の枠組みによる実装

小項目 A4 の属性文法による定式化が成功した場合、それをコード補完機能の実装に反映させる。

#### <引用文献>

- [1] W. Miao et al. Incremental type-checking for type-reflective metaprograms. *GPCE 2010*.
- [2] S. Aditya et al. Incremental polymorphism. *ACM FPCA 1991*.
- [3] C. Haack et al. Type error slicing in implicitly typed higher-order languages. *Science of Computer Programming* 50:189--224, 2004.
- [4] B. Lerner et al. Searching for type-error messages. *ACM PLDI 2007*.
- [5] L. Augustsson. Djinn. <http://www.augustsson.net/Darcs/Djinn/>.
- [6] S. Katayama. Systematic search for lambda expressions. *TFP 2005*.
- [7] Takumi Goto and Isao Sasano. An approach to completing variable names for implicitly typed functional languages. *ACM PEPM 2012*.
- [8] 後藤拓実, 篠埜功. 暗に型付けられた関数型言語に対する変数名補完方式の提案. *PPL 2011*.
- [9] Lambda-mode: <http://www.cs.ise.shibaura-it.ac.jp/lambda-mode/>.

#### 4. 研究成果

2013年度は、暗に型付けられた関数型言語の核となる言語について、識別子補完方式を提案する論文を国際論文誌 "Higher-Order and Symbolic Computation" において発表した。これは関数型言語に関するレベルの高い国際論文誌であり、2012年のACMのプログラム変換に関する国際会議である "Workshop on Partial Evaluation and Program

Manipulation" の special issue である。識別子補完はコード補完の最も基本的な場合の一つである。今回この論文で提案した識別子補完の枠組みは編集集中のプログラムにおいてカーソル位置までの部分プログラムは構文および型に関して正しいという前提で構築されている。ここで、カーソル位置以降に何らかのプログラムを補うことによって構文エラーおよび型エラーがないプログラムになる場合、カーソル位置までの部分プログラムは構文および型に関して正しいとしている。国際会議の論文での問題の定式化および補完候補計算アルゴリズムに若干の修正を加え、健全性について証明した。ここで、提案したアルゴリズムによって計算される識別子の集合は問題設定の条件を満たすという性質を健全性と呼んでいる。また、提案したアルゴリズムを簡略化したアルゴリズムについて完全性を証明した。ここで、問題設定の条件を満たす識別子は、アルゴリズムによって計算される変数の集合に必ず含まれるという性質を完全性と呼んでいる。これらの健全性、完全性の証明は、Hindley-Milner 型推論アルゴリズムにおける完全性と健全性の証明のある種の拡張と言える。

2014年度は、関数型言語の核となる、ラムダ計算に let 式等が加わった言語に対し、文脈を考慮した識別子補完を行う方式を提案、実装し、BICT 2014 Special Track on Modularization for Practical Software Engineering という国際会議において発表した。本提案手法において、補完候補時に構文解析を行うが、そのプログラムは Yacc を使って自動生成したものをを用いているところが特徴である。Yacc は誤り回復機能を持っており、error という特別な字句を構文定義中に記述することにより、ソースコード中に構文誤りがあった場合の誤りからの回復方法を制御することができる。一般にプログラミングにおいては一部が書きかけのまま他の部分を記述したりすることにより、誤りを含んだ状態で補完候補の計算を行う必要がある。現在広く使われている開発環境においても誤りを含んだ状態で補完候補が表示されるが、内部でどのような計算が行われているかプログラマには明確に分からない。熟練プログラマにとっては開発環境の機能の挙動が予想できることが望ましく、本研究の提案手法により、LR 構文解析および Yacc の誤り回復を知っているプログラマにとって補完機能に関する挙動が明確に分かり、かつ誤り回復の制御をある程度行うことが可能である。上記提案手法は Emacs モードとして実装し、web page 上で公開した。補完候補の計算は Yacc で自動生成された C 言語プログラムを用いて行い、Emacs Lisp プログラムとの間で通信を行っている。

2015年度は2014年度に発表したLR構文解析の誤り回復を用いた識別子補完方式をも

とに、キーワード補完プログラムを構文定義から一部機械的に得る手法を考案し、試験的実装を行った。2014年度の発表においてはYaccの構文定義中にerrorという特別な字句を適切に挿入することによって入力中の不完全なプログラムにおいて有効範囲を考慮して補完候補を計算する方式を示した。この成果をもとに、errorという字句およびカーソル位置を表す字句をYaccの構文定義中に機械的に挿入し、構文解析によって得られる構文木をもとにキーワードを補完するというプログラムを一部機械的に生成する方式を考案、実装した。この成果について第18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という国内会議においてポスター・デモ発表を行った。この成果は、開発環境においてよく用いられるコード補完について、仕様が明示された補完機能の提供に向けた第一歩となるものである。また、開発環境の機能の一つとしてC言語で書かれたプログラムを対象としてバッファオーバーフローを可視化するシステムをgdbというデバッガの機能を用いて実現する方式の考案および実装を行った。この成果は BICT 2015 Special Track on Modularization for Practical Software Engineering という国際会議において発表した。また、関数型言語のプログラムに対し、関数適用により生じたギャップに着目した新たなコードクローン検出手法を考案、実装した。この成果は第18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という国内会議において発表した。

研究期間全体を通し、関数型言語の核言語について型を考慮した識別子補完を問題として定式化し、それを解く1つのアルゴリズムを考案し、そのアルゴリズムが満たす性質を示したこと、および入力中の不完全なプログラムに対応するためにLR構文解析の誤り回復機能を用いた方式を考案、実装したこと、およびその方式をもとにキーワード補完プログラムを一部機械的に導出する方式を考案、実装したことが主な成果である。国内外において類似した研究は非常に少なく、得られた成果はいずれも独創的である。今後はプログラミング支援機能を仕様がユーザに明示された形で提供できるようにさらに研究を進める予定である。

#### 5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

〔雑誌論文〕(計1件)

Isao Sasano and Takumi Goto, An approach to completing variable names for implicitly typed functional languages, Higher-Order and Symbolic Computation, Volume 25, Issue 1, pp. 127-163, August 2013. Springer. DOI:

10.1007/s10990-013-9095-x

〔学会発表〕(計5件)

松下 翼, 篠埜 功, 関数適用によるギャップを考慮したコードクローンの検出および除去に関する研究, 第18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 (PPL2016) 論文集(インフォーマルな論文集), 岡山県玉野市たまの温泉 ダイヤモンド瀬戸内マリホテル, 2016年3月7日~8日.

白楊, 篠埜 功, LR構文解析のエラー回復機能を用いたキーワード補完機能の系統的導出, 第18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 (PPL2016), ポスター・デモ発表, 岡山県玉野市たまの温泉 ダイヤモンド瀬戸内マリホテル, 2016年3月7日~9日.

松下 翼, 篠埜 功, 関数型言語における関数適用によるギャップを考慮したコードクローン検出および除去に関する研究, 第17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 (PPL2015), ポスター・デモ発表, 愛媛県松山市道後温泉 道後プリンスホテル, 2015年3月4日~6日.

Isao Sasano, A tool for visualizing buffer overflow with detecting return address overwriting, *BICT 2015 Special Track on Modularization for Practical Software Engineering (MPSE)*, New York City, New York, United States, December 3-5, 2015.

Isao Sasano, Toward modular implementation of practical identifier completion on incomplete program text, *BICT 2014 Special Track on Modularization for Practical Software Engineering (MPSE)*, Boston, Massachusetts, United States, December 1-3, 2014.

〔その他〕

ホームページ等

LR構文解析の誤り回復を用いた識別子補完方式に基づいた識別子補完システムを公開している web page:

<http://www.cs.ise.shibaura-it.ac.jp/mpse2014/>

#### 6. 研究組織

(1)研究代表者

篠埜 功 (SASANO ISAO)

芝浦工業大学・工学部情報工学科・准教授  
研究者番号: 10362021