様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

令和　６ 年　 ６ 月 １０ 日現在

機関番号：３２６１９
研究種目：基盤研究(C)（一般）
研究期間：2020～2023
課題番号：２０Ｋ１１７５２

研究課題名（和文）構文解析を用いたテキストベースコード補完

研究課題名（英文）Text-based code completion using syntax analysis

研究代表者
篠埜　功（Sasano, Isao）

芝浦工業大学・工学部・教授

研究者番号：１０３６２０２１

交付決定額（研究期間全体）：（直接経費）　　2,500,000円

研究成果の概要（和文）：本研究課題では、テキストエディタを用いたプログラミング時のコード補完を行う新たな方式を提案し、実装した。この方式はLR構文解析器の内部状態を用いるものであり、プログラムの先頭からカーソル位置までのコードを構文解析し、その時点の構文解析器のスタック上の記号および状態の列をもとにカーソル位置以降に入りうる構文の候補を計算してプログラマに提示するものである。実装はEmacsに対して行い、補完候補計算はHaskell、インターフェースはEmacs Lispで実装し、Haskell、C等の言語で実際に使えることを確認した。また、実装したコード補完システムのソースコードをweb page上で公開した。

研究成果の学術的意義や社会的意義
Visual Studio等の開発環境において構文補完機能が使われ始めているが、通常、仕様が明示されていない。本研究課題は、構文解析を用いて、カーソル位置までに構文エラーがない場合に、カーソル位置以降の構文の候補を提示する方式を考案したものである。学術的には1980年代にSynthesizer GeneratorやMENTOR等、構造エディタの研究が行われ、近年では、構造エディタでの編集の正しさを論じた研究もある。しかし構造エディタはプログラマが自由にプログラムテキストを編集するのを妨げる。本研究は、自由なプログラム編集を妨げることなく構文補完を行うための方式を提案したものである。

研究成果の概要（英文）：In this research project, we proposed and implemented a new method for code completion during programming using a text editor. This method utilizes the internal state of an LR parser, parsing the code from the beginning to the cursor position. Based on the sequence of symbols and states on the parser's stack at the moment, it calculates potential syntax candidates that can follow the cursor position and presents them to the programmer. The implementation was done for Emacs, with the completion candidate calculation implemented in Haskell and the interface in Emacs Lisp. It has been confirmed to work with languages such as Haskell and C. Additionally, the source code of the implemented code completion system has been made available on a web page.

研究分野：プログラミング言語

キーワード：コード補完　構文補完　LR構文解析　文形式　簡約　統合開発環境

様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９（共通）

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 Microsoft の開発環境など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。コード補完は文脈から何らかの構文解析を行うことによって行われるが、コード補完と構文解析の関係を明確に論じた文献は申請者の知る限り存在しない。また、コード補完の仕様を明確にした上でどのように構文解析器と連携して実装するかは自明ではない。既存の Eclipse、Visual Studio 等の開発環境においては、識別子付け替えや識別子補完等の機能が提供されているが､形式的仕様が定められていないか､あるいは公開されていない。熟練プログラマは正確に把握できない機能の使用を避ける場合がある。また学術的には 1980 年台に Synthesizer Generator [1] や MENTOR [2] 等、構造エディタ（structured editor あるいは projectional editor）の研究が多く行われた。構造エディタを用いることによりプログラムの編集の仕様が定めやすく、また構文補完などの機能の実装が容易になった。近年では、構造エディタでの編集の正しさを論じた研究[3]などもある。しかしながらテキストベースのエディタと異なり、構造エディタはプログラマが自由にプログラムテキストを編集するのを妨げる。本申請においては、プログラマが自由にテキストを編集するのを妨げないということをテキストベースと呼ぶ。この不便さにより、多くのプログラマは（プログラムのソースコード以外の構造データを編集するような場合を除き）構造エディタは用いないのが現状である。

２．研究の目的

上記の問題を解消するために、テキストベースでかつ形式的に仕様が定められたコード補完機能、特に構文補完機能を構文解析器と連携しながら実現する方式の考案および実装を行う。またそれに基づいて Standard ML 等の言語に対してコード補完機能を提供する。これにより、プログラマがテキスト編集の邪魔をされることなく、かつ確信を持ってコード補完機能を用いることができるようになり、プログラミングの効率が向上する。また実装技術についても構文解析プログラムとの連携方式を考案する。

３．研究の方法

文脈を考慮したコード補完の実装の系統的導出技法に関し、理論、実装の両面から研究を推進する。この研究は全南大学校の Kwanghoon Choi 氏との共同研究である。

[研究項目 A(理論)]：コード補完の基本方式の確立
国際会議[4]で発表した構文補完方式を参考にしつつ、構文解析器の内部状態を用いたコード補完方式を開発する。また、申請者らが過去に提案した、型を考慮した識別子補完方式[5]、構文誤りを許す識別子補完方式[6]等と組み合わせることも検討する。

[研究項目 B(実装)]：補完機能の仕様からの系統的実装
上記方式に基づいた実装を構文解析器と連携しながら系統的に行う技法を考案し、Emacs 等のエディタの編集操作に応じて動作するコード補完機能を提供する。まず、Choi 氏らによる構文解析器記述方式[7]をもとに構文補完の基本的な実装を行う。この実装について、2019 年 9 月に Choi 氏と議論し、彼と共同で行うことにした。また、申請者が提案した構文誤りを許す識別子補完方式[6]や、近年行われた、構文から誤り回復規則を導出する研究[8]などを参考に、型が明示的に与えられない場合や型誤りを含む場合に対するコード補完方式を確立する。さらに、型誤りが含まれる部分については、型誤りの原因となる箇所を検出する型誤りスライシングという手法[9]や、型誤り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[10]があり、それらを参考にして対処方法を検討する。申請者らの既存研究[5]における識別子補完の実装[11]は、Emacs Lisp により Emacs モードとして直接記述されたものである。本研究では、構文解析器と連携しながらコード補完の候補を計算しようとするものである。申請者らの既存研究[6]は、それを識別子補完について部分的に行ったものである。

４．研究成果

全南大学校の Kwanghoon Choi 氏とともに LR 解析に基づく構文補完方式を整理し、国際会議 PEPM2021 に論文を投稿し、採択され、発表を行った[12]。発表した内容は、エディタのカーソル位置までのテキストを LR 構文解析器で解析し、カーソル位置に達した時点における LR 構文解析器のスタックの情報をもとに、カーソル位置以降に入りうる構文の候補を計算するというものである。核となる考えは、カーソル位置まで解析した時点の構文解析器において、何らかの

終端あるいは非終端の記号列を shift した後で reduce できる場合、その記号列を補うと 1 つの構文の単位が完結するだろうというものである。まず、入力途中の一番内側の構文について候補を計算する方式を考案した。この計算方式は、有限時間で計算が終了し、かつ実用になる時間で補完候補計算が行われる。各補完候補は終端あるいは非終端記号の列であるが、非終端記号については、...などの適当な記号列に置き換えてユーザに提示する。またこれを組み合わせることにより、入力途中の構文が入れ子になっている場合に対応する計算方式を考案した。これは文法が直接あるいは間接左再帰になっている場合に計算が終わらなくなるが、これに対処するためにヒューリスティックな計算方式を考案した。考案した方式に基づき、Choi 氏が 2019 年に開発した、スタック情報にアクセスできる LR 構文解析器生成系[7]を用いて、Emacs 上で動作する構文補完システムを実装した。

その後、国際会議 ACM Partial Evaluation and Program Manipulation (PEPM 2021) で発表を行った LR 構文解析に基づく構文補完方式について、Haskell、C、Small Basic、PolyRPC 等の言語に実際に適用したところ、問題点がいくつかあり、それらに対応するため、simple candidate, nested candidate, extended simple candidate 等、補完候補の仕様をいくつか定め、それらを計算するアルゴリズムを考案し、Haskell 言語で実装した。実装した補完候補計算システムは、Emacs Lisp で記述されたプログラムにより、Emacs 上でプログラム編集中に Tab キーを押すことにより呼び出され、計算結果である補完候補が Emacs に送られ、popup window として表示され、プログラマが候補を選択するとカーソル位置に挿入される。さらに、さまざまなプログラム例に対し補完候補計算にかかる時間や計算される補完候補を確認し、整理した。複雑な仕様における補完候補を求める場合においても、約 0.2 秒以下で半分の補完候補が計算され、9 割程度の補完候補が約 1 秒以内で計算されることを確認した。これらの内容を論文としてまとめ、Science of Computer Programming という Elsevier の国際論文誌へ投稿し、採録され、2023 年 6 月に掲載された[13]。実装したシステムのソースコードは github (https://github.com/kwanghoon/{yapb, arith, smllike, sbparser, polyrpc, c11parser, haskellparser}) 上で公開している。

さらに、この補完方式をもとに、既存のソースコードを事前に構文解析し、構文解析器の各内部状態に対する action の頻度情報を計算しておき、それを用いる補完方式を考案して国際会議 ACM SIGAPP Symposium on Applied Computing (SAC 2024)に投稿し、採録された[14]。

参考文献

[1] Thomas Reps and Tim Teitelbaum. The Synthesizer Generator. *ACM Software Engineering Symposium on Practical Software Development Environments*, 1984.

[2] Veronique Donzeau-Gouge, Gerard Huet, Gilles Kahn, and Bernard Lang. Programming environments based on structured editors: the MENTOR experience. Technical Report RR-0026, INRIA, 1980.

[3] Friedrich Steimann et al. Robust Projectional Editing. *ACM SLE 2017*.

[4] Isao Sasano. An approach to generating text-based IDEs with syntax completion. *ACM PEPM 2018*, short presentation, poster, demo.

[5] Isao Sasano, 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, 2013. Springer.

[6] Isao Sasano. Toward modular implementation of practical identifier completion on incomplete program text. *ICST MPSE 2014*, ACM digital library.

[7] J. Lim, G. Kim, S. Shin, K. Choi, I. Kim, Parser generators sharing LR automaton generators and accepting general purpose programming language-based specifications, *J. KIISE*, Volume 47, No. 1, pp. 52-60, 2020. (in Korean).

[8] Maartje de Jonge, LENNART C. L. KATS, Eelco Visser,and Emma Söderberg, Natural and Flexible Error Recovery for Generated Modular Language Environments. *ACM TOPLAS*, Volume 34, Issue 4, Article No. 15, pp. 1-50, 2012.

[9] C. Haack and J. B. Wells, Type error slicing in implicitly typed higher-order languages. *Science of Computer Programming*, Volume 50, pp. 189-224, 2004.

[10] Benjamin S. Lerner, Matthew Flower, Dan Grossman, and Craig Chambers, Searching for type-error messages. *ACM PLDI 2007*.

[11] Lambda-mode: http://www.cs.ise.shibaura-it.ac.jp/lambda-mode/.

[12] Isao Sasano, Kwanghoon Choi, A text-based syntax completion method using LR parsing, *ACM SIGPLAN 2021 Workshop on Partial Evaluation and Program Manipulation (PEPM 2021)*, pp. 32-43, Virtual meeting, Denmark, January 18-19, 2021.

[13] Isao Sasano, Kwanghoon Choi, A text-based syntax completion method using LR parsing and its evaluation, *Science of Computer Programming*, Volume 228, 102957, 2023.

[14] Kwanghoon Choi, Sooyeon Hwang, Hyeonah Moon, Isao Sasano, Ranked syntax completion with LR parsing, *The 39th ACM SIGAPP Symposium on Applied Computing (SAC 2024)*, pp. 1242-1251, Avila, Spain, April 8-12, 2024.

５．主な発表論文等

〔雑誌論文〕　計1件（うち査読付論文　1件／うち国際共著　1件／うちオープンアクセス　0件）

１．著者名 Isao Sasano and Kwanghoon Choi	４．巻 228
２．論文標題 A text-based syntax completion method using LR parsing and its evaluation	５．発行年 2023年
３．雑誌名 Science of Computer Programming	６．最初と最後の頁 102957
掲載論文のDOI（デジタルオブジェクト識別子） 10.1016/j.scico.2023.102957	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

〔学会発表〕　計1件（うち招待講演　0件／うち国際学会　1件）

１．発表者名 Isao Sasano and Kwanghoon Choi
２．発表標題 A text-based syntax completion method using LR parsing
３．学会等名 ACM SIGPLAN 2021 Workshop on Partial Evaluation and Program Manipulation (PEPM 2021)（国際学会）
４．発表年 2021年

〔図書〕　計0件

〔産業財産権〕

〔その他〕

―

６．研究組織

	氏名 （ローマ字氏名） （研究者番号）	所属研究機関・部局・職 （機関番号）	備考

７．科研費を使用して開催した国際研究集会

〔国際研究集会〕　計0件

８．本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関