

発表概要

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

白 楊^{1,a)} 篠埜 功¹

2016年6月10日発表

キーワード補完とは様々な言語の開発環境において提供されている機能であり、入力中の文字列を接頭辞に持つキーワードをポップアップウィンドウ等に表示するものである。キーワード補完機能を用いることによりキーワード入力時間や綴りミスを削減することができる。本発表では、様々な言語に対して仕様が明示されたキーワード補完機能を提供することを目的とする。入力中の構文が不完全なソースコードに対応するため、Yacc による LR 構文解析の誤り回復機能を用いることとし、キーワード補完機能の実装を仕様から系統的に導出する手法を提案する。カーソル位置の（入力途中の）キーワードの字句を識別するため、入力途中であることを示す字句を字句解析および構文解析の仕様記述に追加する。本提案手法においては構文解析を行うため、補完候補計算において構文に関する文脈が考慮される。実装の導出は Byacc という構文解析器生成系のソースコードを用いて行う。Yacc の構文規則記述およびユーザが記述したキーワードを入力とし、構文規則の記述への字句 error の機械的挿入により補完候補計算プログラムの自動生成を行うツールを作成した。例として、C 言語の Yacc の仕様記述ファイルを入力とし、C 言語のキーワードの補完を行うプログラムを生成した。現状では字句の仕様記述にはカーソル位置を表す字句を手動で追加する必要がある。補完対象キーワードは自動生成ツールの使用者が指定できる。

Derivation of Keyword Completion Programs Using the Error Recovery in LR Parsing

YANG BAI^{1,a)} ISAO SASANO¹

Presented: June 10, 2016

The keyword completion, which is a functionality for popping up keywords starting with a string currently being input, is provided in the IDEs for various languages. The keyword completion reduces the time for inputting keywords and spelling errors. In this presentation we aim at providing a keyword completion whose specification is given explicitly. We use the error recovery in LR parsing in order to cope with incomplete program text and systematically derive an implementation of keyword completion from a specification. In order to identify the incomplete keywords in the cursor position, we add a token, indicating it is currently input, to the specification of the lexer and the parser. We parse the program being input so that the syntactic context is taken into account in computing candidates. We systematically derive an implementation of the keyword completion by using a parser generator BYacc. We have implemented a tool that generates programs computing candidates by inserting a special token error to the Yacc specification. The tool takes as its input keywords listed by the users and a Yacc specification. As an example, we generate a program for completing keywords in programs in the C language from a Yacc specification. Currently we need to manually add a token for the cursor position in the specification of the lexer. Keywords to be popped up are specified by the tool user.

¹ 芝浦工業大学大学院理工学研究科
Graduate School of Engineering and Science, Shibaura Institute of Technology, Koto, Tokyo 135-8548, Japan

a) ma14501@shibaura-it.ac.jp