

## 発表概要

# 抽象構文木を自動生成するパーサジェネレータの効率的な実装

平崎 昌志<sup>†1</sup> 木山 真人<sup>†1</sup> 芦原 評<sup>†1</sup>

抽象構文木はコンパイラにおける最適化や目的コード生成に使用される。抽象構文木の生成には、構文規則を表す具象構文とは別に抽象構文を利用する。抽象構文は抽象構文木の形式を決定する構文である。この構文は、プログラムの構造とユーザが用いる表現形式の対応関係を示す。この抽象構文を文法の構文規則内に記述し、抽象構文木を自動的に生成する手法を提案する。提案する手法は、構文規則に抽象構文としての情報を付加し木の生成に利用する。抽象構文木の生成には、一般的に解析木から変換する手法が用いられているが、本実装では LALR 構文解析の特徴を利用する。LALR 構文解析では解析にトークンスタックを用いる。このとき、抽象構文に基づくノードを生成し、トークンスタックに保持することで動的に木を構成する。これにより、解析結果から木への変換を不要とし、かつ必要最小限のノードで木を構成する。構文解析中に抽象構文を読み取り、スタックの動的な処理で抽象構文木を決定する手法である。生成された木は抽象構文に従ったプログラム構造を持っており、中間表現やオブジェクトコード生成に利用できる。さらに提案手法に基づくパーサを作成し、抽象構文を利用した文法を用いて抽象構文木の生成を行う。提案する手法が、抽象構文木が最小限の情報によって効率的に生成できることを示す。

## Efficient Implementation of the Parser Generator Which Generates an Abstract Syntax Tree

MASASHI HIRASAKI,<sup>†1</sup> MASATO KIYAMA<sup>†1</sup>  
and HYO ASHIHARA<sup>†1</sup>

An abstract syntax tree is used for the optimization and the purpose code generation in a compiler. An abstract syntax is used for generation of an abstract syntax tree apart from the concrete syntax showing a syntax rule. An abstract syntax is syntax of determining the form of an abstract syntax tree. This syntax shows the correspondence relation of the expressive form which the

structure and the user of a program use. We describe this abstract syntax in the syntax rule of grammar, and propose the technique of generating an abstract syntax tree automatically. The technique to propose adds the information as an abstract syntax to a syntax rule, and uses it for tree generation. Although the technique generally changed from an analysis tree is used for generation of an abstract syntax tree, the feature of LALR syntax analysis is used in this implementation. A token stack is used for analysis in LALR syntax analysis. At this time, the node based on an abstract syntax is generated and a tree consists of holding to a token stack dynamically. Thereby, conversion to a tree from an analysis result is made unnecessary, and a tree consists of necessary minimum nodes. It is the technique of reading an abstract syntax during syntax analysis and determining an abstract syntax tree by dynamic processing of a stack.

(平成 20 年 8 月 6 日発表)

<sup>†1</sup> 熊本大学大学院自然科学研究科

Graduate School of Science and Technology, Kumamoto University