

## 発表概要

# 段階的・対話的プロセスによる例文に基づいた構文設計支援手法の提案

澤入 圭佑<sup>1,a)</sup> 佐々木 晃<sup>2</sup>

2019年3月19日発表

言語処理系開発の際、ふつう開発者は構文を設計する前に独自言語の入力例文を想定し、複数の例文から構文の形への一般化を繰り返すことで構文構築を進める。その過程では演算子の順位など構文どうしの優先順位、採用する構文解析法や形式文法のルールを考慮しながら段階的に構文構築を行う必要がある。たとえば、トップダウン構文解析法を採用する場合には解析の途中で無限ループに陥るのを防ぐため、左再帰が構文内に現れないよう注意しなければならない。また、PEG (Parsing Expression Grammar) には曖昧性がないため、これを利用する場合には到達不能な要素が構文内に現れないようにする必要がある。構文どうしの優先順位の定義方法も解析法や形式文法によって異なるため、それらの理解も不可欠である。さらに、解析結果を利用したツールを作成する場合には、抽象構文木のようなデータ構造を出力するようパーサを開発する必要がある。本研究では柔軟な構文設計の支援を目的として、入力例文から構文への一般化作業を段階的・対話的に進める手法を提案する。本手法は以下の2つの特徴を持つ：(1) 段階的：構文設計図から構文構築のボトムアップガイドラインを策定し、それに沿って構文構築を進めさせることで段階的な構文構築を支援する。(2) 対話的：ユーザからの入力例文を受け構文の一般化例を提案し、必要な部分について対話的に編集を行わせることで一般化を支援する。また、得られた構文情報から対応する抽象構文も生成する。さらに、本手法をプロトタイプシステムとして実装し評価を行った。

## Presentation Abstract

### A Supporting Method for Designing Syntax Based on Example Sentence via Incremental and Interactive Process

KEISUKE SAWAIRI<sup>1,a)</sup> AKIRA SASAKI<sup>2</sup>

Presented: March 19, 2019

When developing language processing systems, developers usually assume example sentences of their own language before designing the syntax. They may construct language syntax by repeating the generalization from multiple example sentences to the form of syntax. In the process, it is necessary to construct the syntax step by step while considering the priority of the syntax such as the order of the operators, the syntax analysis method to be adopted and the rule of the formal grammar. After defining syntax, they also need to implement a parser that generates abstract syntax trees. In this research, for the purpose of designing syntax of programming languages, we propose a method that can generalize task of creating syntax from example sentences stepwise and interactively. The method has two strategies. (1) Stepwise: We support incremental syntax construction process by formulating bottom-up guidelines from syntax diagrams and advancing syntax construction accordingly. (2) Interactive: We support generalization using example sentences provided by users. The system proposes candidates of syntax rules, and letting interactive editing on required parts. After the syntax is obtained, corresponding abstract syntax is also generated from its syntax information. Furthermore, we implemented this method in a prototype system and evaluated it.

This is the abstract of an unrefereed presentation, and it should not preclude subsequent publication.

<sup>1</sup> 法政大学大学院情報科学研究科  
Graduate School of Computer and Information Sciences,  
Hosei University, Koganei, Tokyo 184-8584, Japan

<sup>2</sup> 法政大学情報科学部  
Faculty of Computer and Information Sciences, Hosei University, Koganei, Tokyo 184-8584, Japan  
a) keisuke.sawairi.2h@stu.hosei.ac.jp