

氏 名 Zirun Zhu

学位(専攻分野) 博士(情報学)

学位記番号 総研大甲第 2157 号

学位授与の日付 2020 年 3 月 24 日

学位授与の要件 複合科学研究科 情報学専攻
学位規則第6条第1項該当

学位論文題目 Bidirectional Programming for Parsing and Retentive Printing

論文審査委員 主 査 准教授 蓮尾 一郎

教授 龍田 真

助教 加藤 弘之

助教 対馬 かなえ

助教 関山 太郎

教授 胡 振江

北京大学 コンピュータ科学系

(Form 3)

Summary of Doctoral Thesis

Name in full Zirun Zhu

Title Bidirectional Programming for Parsing and Retentive Printing

Language designers usually need to implement parsers and printers for doing the conversion between program text and abstract syntax trees (ASTs), which are commonly found and constantly used in application scenarios such as resugaring, code refactoring, and observing the results of compiler optimisation. It is desirable that when producing a new piece of program text from a slightly modified (e.g. optimised) AST which corresponds to some original piece of program text, the printer can preserve the layouts, comments, and syntactic sugar contained in the original program text as much as possible. In order to do so, the common practice is to enrich ASTs with whatever information that is needed (such as comments), so that the printer can use the enriched information to produce better-printed results. However, the approach violates the design philosophy of ASTs, for ASTs should be abstract and compact and contain only essential information.

This thesis reconciles the compactness of ASTs and printing quality by proposing *retentive printing*, which takes not only an AST but also the piece of original program text, so that there is no need to enrich ASTs with unnecessary information (that can already be found in the program text) to improve printing quality. It is worth noting that retentive printing is a generalisation of the conventional notion of printing, because a retentive printer can accept an AST and an empty piece of program text, in which case it will behave just like a conventional printer, producing a new piece of program text depending on the AST only.

We propose a domain-specific language BiYacc, whose program denotes a consistent (i.e. *well-behaved*) pair of parser and retentive printer for an unambiguous context-free grammar. BiYacc is based on the theory of bidirectional transformations (BX) and in particular (state-based asymmetric) lenses, which helps to guarantee by construction that the generated parser and retentive printer pairs are always well-behaved. We show that BiYacc facilitates many tasks such as resugaring, language evolution, and simple refactoring by demonstrating them in a medium-size case study on the Tiger language.

However, using only unambiguous grammars can be rather inconvenient in practice, as ambiguous grammars (with disambiguation directives) are often considered more natural and human-friendly than their unambiguous versions. Therefore, we make the first move to support ambiguous grammars and tackle grammatical ambiguity in the bidirectional setting by proposing an approach based on generalised parsing and disambiguation filters, which produce all the parse results and (try to) select the only

desired one in the parsing direction; the filters are carefully *bidirectionalised* so that they also work in the printing direction and will not break the well-behavedness between the pair of parser and retentive printer—provided that they satisfy our proposed bidirectionalised filter laws. We extend BiYacc with this functionality and design accessible directives for specifying production rules’ associativity and (relative) priorities, which give rise to compositional and commutative filters; power users can also define their own filters. We demonstrate the extended BiYacc using the same Tiger language defined by an ambiguous grammar and a set of disambiguation rules, which also includes a manually written one for the dangling else problem.

Next, we find that while lenses are designed to retain information, well-behavedness actually says very little about the retention of information. The only law (of well-behavedness) guaranteeing information retention merely requires that the whole source should be unchanged if the whole view is. In other words, if we have a very small change on the view, we are free to create any source we like. This is too ‘global’ in most cases, and it is desirable to have a law that makes such a guarantee more ‘locally’. Thus, to explore the possibility of *local and precise* information retention and formalise the statement ‘the printer can preserve the layouts, comments, and syntactic sugar contained in the original program text as much as possible’, we propose an extension of the (state-based asymmetric) lens framework, called *retentive lenses*, which satisfy a new Retentiveness law guaranteeing that if parts of an AST are unchanged, then the corresponding parts of the original program text are retained in the newly printed program text as well. We verify our idea by introducing a new DSL for writing tree transformations (especially between CSTs and ASTs) and presenting case studies on code refactoring, resugaring, and XML synchronisation. Retentive lenses are not state-based, and its integration into BiYacc is left to our future work.

博士論文審査結果

氏名^{Name in Full} Zirun Zhu

論文題目^{Thesis Title} Bidirectional Programming for Parsing and Retentive Printing

本論文は、双方向変換に基づいてパーサーとプリティプリンターを同時に開発する系統的な手法とその応用に関するものである。パーサー(parser)とプリンター(printer)はコンパイラにおける重要な二つの部品である。パーサーはテキストを計算機で処理しやすい抽象構文木に変換し、プリンターは抽象構文木からテキストに戻す。これまで、パーサーとプリンターはそれぞれ独立して開発されてきたが、共通点が多いため同時に両方を開発することも可能であり、そのための方法がいくつか提案されている。しかし、従来の手法には二つ問題がある。一つは抽象構文木に多くの情報を付け加える必要があり、それにより抽象構文木の処理が不便になってしまうことである。もう一つは開発されたパーサーとプリンターが望ましい roundtrip の性質を持つことを厳密に保証できないことである。本論文は双方向変換の原理に基づいて、パーサーと保持的プリンターを同時に記述できる領域特有言語 BiYacc を提案し、BiYacc の記述からパーサーとプリティプリンターを自動的に生成するシステムを開発した上、その有用性を、コンパイラ教科書によく使われている Tiger 言語のパーサーとプリティプリンターの自動生成を通じて示している。また、曖昧な文法への対応、双方向変換の保持性 (retentiveness) の強化などを行うことにより、プログラムソースと抽象構文木の同期を支援する強力なシステムの実現に成功し、双方向変換の実用化への大きな足がかりになっている。

本論文は英語で記述されており、全 6 章から構成されている。

第1章は序論である。研究の背景、研究目的、主要な貢献など論文全体の構成を述べている。

第2章は基礎知識の紹介である。双方向変換の歴史と基礎知識、パーサーとプリンターを双方向変換と表現する考え方、関連研究について議論している。

第3章では、双方向変換の原理に基づいて、パーサーと保持的プリンターを同時に記述できる領域特有言語 BiYacc の設計、BiYacc の記述からパーサーとプリティプリンターを自動的に生成する手法を示している。

第4章では、第3章で議論している BiYacc を拡張し、曖昧な文法に対応できる手法を示している。生成されたパーサーとプリンターのペアの間で一貫性を維

持しながら、曖昧さを排除するディレクティブをBiYaccに導入し、フィルターの技術に基づいた双方向フィルター (bi-filter) をBiYaccに統合する方法を議論している。

第5章では、双方向変換の保持性を保証する枠組みを与え、第3章のBiYaccをさらに強化させる方法を議論している。抽象構文木の一部に変更がなければ、その部分に対応する具体構文木が変わらないことを保証する双方向言語の設計と実現を示している。

第6章は論文のまとめと今後の課題である。

審査会において、出願者は上述の内容に沿って説明を行い、そのあと審査委員との質疑応答を行い、的確な回答がなされた。なお、本研究の成果は査読付き雑誌論文1本、査読付き国際会議論文1本、査読付き国際会議小論文 (short paper) ・ポスター2本の発表という業績をあげている。また、開発したシステムもウェブで公開し自由にダウンロードできるようになっている。

以上のように、本論文では、コンパイラーのフロントエンドを対象とし、双方向変換の原理に基づいて、一貫性を保持し、パーサーとプリンターを同時に記述できる領域特有言語の設計と実現に成功した。本研究は双方向変換を一層身近なものとなり、再糖化 (resugaring)、リファクタリング、デバッグへの様々な応用が期待できる。以上の理由により、審査委員会は、本論文が学位の授与に値すると判断した。