



PDF Download
3478432.3499140.pdf
18 February 2026
Total Citations: 1
Total Downloads: 89

 Latest updates: <https://dl.acm.org/doi/10.1145/3478432.3499140>

POSTER

ITT: An Interactive Tutoring Tool to Improve the Learning and Visualization of Compiler Design Theory From Implementation

RAFAEL DEL VADO-VÍRSEDA, Complutense University of Madrid, Madrid, Madrid, Spain

Open Access Support provided by:

Complutense University of Madrid

Published: 03 March 2022

[Citation in BibTeX format](#)

SIGCSE 2022: The 53rd ACM Technical Symposium on Computer Science Education

March 3 - 5, 2022
RI, Providence, USA

Conference Sponsors:
SIGCSE

ITT: An Interactive Tutoring Tool to Improve the Learning and Visualization of Compiler Design Theory From Implementation

Rafael del Vado Vírveda*
 Universidad Complutense de Madrid
 Madrid, Spain
 rdelvado@sip.ucm.es

ABSTRACT

In this poster we present ITT, and Interactive Tutoring Tool that allows teachers of a Compiler Design course to interactively guide the learning from the implementation of the main theoretical concepts on lexical, syntactic analysis, and syntax-directed translation. With the help of ITT, our students interact with the visualization of the automata underlying the implementation obtained by the compiler-writing tools. Preliminary results show the benefits of using ITT to improve their understanding of theoretical concepts from the interactive execution of the implementation, deepening and reinforcing the understanding of theory in relation to the code.

CCS CONCEPTS

• **Theory of computation** → **Grammars and context-free languages**; • **Software and its engineering** → **Compilers**;

KEYWORDS

Compiler design course; compiler writing tools; interactive learning

ACM Reference Format:

Rafael del Vado Vírveda. 2022. ITT: An Interactive Tutoring Tool to Improve the Learning and Visualization of Compiler Design Theory From Implementation. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 2 (SIGCSE 2022)*, March 3–5, 2022, Providence, RI, USA. ACM, New York, NY, USA, 1 page. <https://doi.org/10.1145/3478432.3499140>

1 INTRODUCTION AND MOTIVATION

Compiler Design courses have given rise to a beautiful combination of theory and practice in Computer Science, helping students to mature as computer scientists and engineers. Unfortunately, traditional educational approaches sometimes fail to fill the gap that students find between the understanding of Compiler Design theory, and the required techniques for designing and implementing a modern compiler. In this poster, we propose an Interactive Tutoring Tool (ITT) to improve the learning and visualization of the main theoretical concepts of a Compiler Design course from the implementations provided by automatic code generation tools.

*The author wishes to express his gratitude and appreciation to José Guillermo del Vado Olalla (1939-2021).

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
 SIGCSE 2022, March 3–5, 2022, Providence, RI, USA
 © 2022 Copyright held by the owner/author(s).
 ACM ISBN 978-1-4503-9071-2/22/03.
<https://doi.org/10.1145/3478432.3499140>

2 RELATED WORK AND CONTRIBUTIONS

In the last two decades, three strategies have evolved for a Compiler Design course in the Computer Science curriculum [3]. First, emphasize programming methodology by constructing a compiler as a software project. Second, emphasize the role of theory to enable automation of the compiler tasks. Third, emphasize the broad applicability of compiler technology to implement languages for special purposes. The ITT tool approaches the second of these strategies from an alternative point of view, highlighting the role of the implementation to be interactively visualized closer to each theoretical concept at each step of the compiler tasks. Following our previous works [1, 2], we have employed a new course format in which the theory of the subject is incrementally introduced and visualized, through several implementations of increasingly complexity.

3 THE INTERACTIVE TUTORING TOOL ITT

Students begin their interactive learning with ITT from a lexical and syntactic-semantics specification of a formal language. From these specification files, each student easily obtains its implementation in C/C++ using the *Flex/Bison* tools, together with files in DOT format to generate the graph of the finite automata and the attribute-dependency graph underlying their implementation. When the student executes the ITT.jar file, ITT creates an interactive session on the theoretical tool *JFLAP* to visualize and relate the most significant parts of their code to each of the theoretical concepts.

4 METHODS AND PRELIMINARY RESULTS

In the 2020/21 academic year, ITT was tested for the first time in a teacher-guided and controlled experience to evaluate its application. Students who used the ITT tool achieved better results and motivation than students who followed the traditional method.

5 CONCLUSIONS AND FUTURE WORK

The contribution of ITT is that students can interact with automata visualizations, and that their interactive learning allows them to better understand the main concepts of a Compiler Design course underlying the implementation. Our future work consists of studying the scalability of ITT in all phases of compiler construction.

REFERENCES

- [1] Rafael del Vado Vírveda. 2020. An Interactive Tutoring System for Learning Language Processing and Compiler Design (*ITiCSE '20*). Association for Computing Machinery, New York, NY, USA, 552. <https://doi.org/10.1145/3341525.3393969>
- [2] Rafael del Vado Vírveda. 2021. Learning Compiler Design: From the Implementation to Theory (*ITiCSE '21*). Association for Computing Machinery, New York, NY, USA, 609–610. <https://doi.org/10.1145/3456565.3460041>
- [3] William M. Waite. 2006. The Compiler Course in Today's Curriculum: Three Strategies (*SIGCSE '06*). Association for Computing Machinery, New York, NY, USA, 87–91. <https://doi.org/10.1145/1121341.1121371>