

Tree-sitter: Parser Generator

Miguel Rodrigues^a and Tomás Melo^a

^aFaculty of Information Technology, Brno University of Technology
Email: {*xboave00*, *xmelot00*}@stud.fit.vutbr.cz

November 1, 2023

Abstract

This talk is about Tree-sitter¹: a parser generator and an incremental parser created by Max Brunsfeld, initially as a side-project and later as part of the Atom editor team at GitHub.

Firstly, we will introduce the rationale behind the conception of Tree-sitter, describing its usage, particularly in IDEs, text editors, and other development-related tools such as code navigation tools or static code analyzers.

Then, it will be time to cover some of the Tree-sitter's internals and the advantages of having a parser generator that is both language-agnostic and dependency-free. Nonetheless, at this point, it will be necessary to revisit some fundamental parsing algorithms behind every parser generator. Here, we will describe the approaches of well-established tools such as GNU Bison and ANTLR [1, 2] and compare them to the Tree-sitter's [3].

In this context, we must also discuss incremental parsing [4], i.e., the novelty introduced by Tree-sitter. We will detail how such an approach unlocks an unmatched way of handling errors while parsing, around an order of magnitude, faster than similar tools in scenarios where real-time parsing is required, e.g., IDEs and text editors.

References

- [1] Donald E Knuth. On the translation of languages from left to right. *Information and control*, 8(6):607–639, 1965.
- [2] Terence J. Parr and Russell W. Quong. Antlr: A predicated-ll (k) parser generator. *Software: Practice and Experience*, 25(7):789–810, 1995.
- [3] Masaru Tomita. *Generalized LR parsing*. Springer Science & Business Media, 1991.
- [4] Tim A Wagner and Susan L Graham. Efficient and flexible incremental parsing. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 20(5):980–1013, 1998.

¹<https://tree-sitter.github.io/tree-sitter/>