Realtime Syntax Error Correction For All

ANONYMOUS AUTHOR(S)

In this work, we illustrate how to lower context-free language recognition onto a tensor algebra over finite fields. In addition to its theoretical value, this connection has yielded surprisingly useful applications in incremental parsing, code completion and program repair. For example, we use it to repair syntax errors, perform sketch-based program synthesis, and decide various language induction and membership queries. This line of research provides an elegant unification of context-free program repair, code completion and sketch-based program synthesis. To accelerate code completion, we design and implement a novel incremental parser-synthesizer that transforms CFGs onto a dynamical system over finite field arithmetic, enabling us to suggest syntax repairs in-between keystrokes.

1 INTRODUCTION

Our tool accepts an arbitrary sequence of lexical tokens and suggests a set of possible syntax repairs. We evaluate it using a corpus of syntactically correct single-line statements containing between 10-80 lexical tokens in various programming languages (e.g., Java, Kotlin and Python), synthetically corrupted by erasing up to three uniformly at random. The tool suggests a repair in less than 0.1 seconds on average, and the original statement is ranked within the first ten in P% of the cases.

We also evaluate it on a corpus of natural syntax errors, and find the tool is able to suggest at least one syntactically valid repair within 20 seconds in 100% of the cases presented, and typically several hundred depending on the length of the fragment and grammar, as well as the number of available processor cores. For example, consider the following text that was synthetically corrupted:

```
 Original:  val splitTokens : MutableList < String > = ArrayList ( )
Corrupted:  var splitTokens : MutableList < String > = ArrayList ( )
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
D=1 repair: var splitTokens : MutableList < String > = ArrayList ( )
D=1 repair: val splitTokens : MutableList < String > = ArrayList ( )
D=2 repair: var protected : MutableList < String > = ArrayList ( )
D=2 repair: var splitTokens : MutableList < tailrec > = ArrayList ( )
D=2 repair: var splitTokens : MutableList < String > = inline ( )
D=2 repair: val splitTokens : MutableList < override > = ArrayList ( )
...
Found 426 valid repairs in 13062ms, or roughly ~32.61 repairs per second.
Original string was ranked #1 in repair proposals!
```

In the following paper, we will describe the theory and practice of real-time syntactic error correction, and demonstrate how the same techniques can be used to accelerate code completion, program repair and sketch-based program synthesis. Although we primarily focus on the Kotlin language, the techniques described are applicable to any language whose syntax can be expressed as a context-free language. In practice, this includes a substantial fragment of most programming languages. We will also describe how the techniques described can be extended to support certain context-sensitive languages and thereby used to correct various kinds of semantic errors.

POPL'24, January 17-19, 2024, London, United Kingdom
2023. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn