Efficiently Inferring Non-hierarchical Structure in Parsing and Computation

Danny McClanahan

2021-06-21

Abstract: This paper proposes to revise the general definition of a "grammar"[1] as well as "parsing"[2]. A meta-language named "Simultaneous Productions" (or "S.P") is proposed for specifying a parseable grammar (similar in intent to EBNF[3]). This meta-language is shown to be able to represent recursively enumerable languages[4], and several favorable properties of this meta-language across various parsing use cases are discussed.

An actual "evaluation method" (or "parsing algorithm") is then introduced for the proposed S.P. grammar model, which shares some similarities with the CYK algorithm [5].This method is shown to terminate in **?** [6] time across **?** [7] inputs. This method is then demonstrated to have the peculiar property of directly parameterizing context-sensitivity [8] **?** [9]. The practical and theoretical ramifications of this result are discussed and speculated on.

This paper will frequently refer to free software provided at [1] which implements all of the theoretical mechanisms discussed.

[1] cite: grammar defn
[2] cite: parsing defn
[3] cite: EBNF
[4] cite: RecEnum defn
[5] cite: CYK algorithm!
[6] TODO: parsing runtime?
[7] TODO: parsing inputs?
[8] cite: context-sensitivity defn
[9] TODO: is this peculiar?

Contents

1 Motivation

2 Meta-Language for Grammar Specification

3 Fully General Parse Method

4 Re-parameterization of the Chomsky Hierarchy

TODO:

- describe the hierarchy as being better described by derivatives (?) of a function corresponding to stack depth (!!!) and link it to approximation difficulty![10]
  - link between the # of derivatives required to approximate the function and the taylor series (???)
  - the content of the turing tape is the S.P. initial state assignment!
  - the action of the turing machine on the tape is one particular evaluation/priority ordering of "adjacency resolution"!**?** [11]
- make S.P. support multiple parallel stacks**?** [12]
  - *then reduce (2) to a k-stack automaton (???) (equivalent to a T.M.!)*[13]
  - the above may not be necessary if we can realize the direct reduction from T.M. in 4!

[10] cite: approximation difficulty/the PCP theorem
[11] TODO: describe "adjacency resolution" in (3)
[12] TODO: stacks with a discrete finite set of "colors"
[13] cite: k-stack automaton definition, equivalence to 2-stack automaton, equivalence to T.M.

References

[1] Danny McClanahan. *Simultaneous Productions git repository on github*. `https://github.com/cosmicexplorer/simultaneous-productions`. Accessed on 2021-06-20. 2019.