Expressing Type-0 Languages in Terms of Context-Free Ambiguousness

Stanislav Láznička
ilaznicka@fit.vutbr.cz

October 10, 2015

Abstract

Often in the beginning of formal languages study, after getting to know what a formal language actually is, the Chomsky hierarchy of languages is introduced. We learn that it is possible to classify formal languages into four basic classes - regular, context-free, context-sensitive and recursively enumerable - and that there are also languages that are beyond this classification. These classes form the hierarchy in a way where each of the language classes is a subset of the following class in the respective order, with recursively enumerable languages forming the largest set of formal languages.

Each of the four mentioned language classes has a way to describe the languages that form it. For all the classes, there is a grammar, a mathematical structure, that shows how to create sentences for that certain language in the certain class. A part of these grammars are the rules that describe the way a language should be formed. The shape of the rules is what actually describes the differences among each of the language classes.

The general thought is that the means which describe the respective class and its languages should not be able to describe a higher language class, although they are of course able to describe the lower class. This is obvious - if such a thing happened, it would break the hierarchy.

However, in 1967, Ginsburg, Greibach and Harrison proved that it only takes two deterministic context-free languages to get a type-0 language (recursively enumerable)[1]. Here, we elaborate on that idea and try to take two deterministic context-free grammars to generate ambiguity and then use it to our benefit to express a type-0 language.

[1] GINSBURG, Seymour, Sheila A. GREIBACH, Michael A. HARRISON. One-Way Stack Automata. *Journal of the Association for Computing Machinery.* 1967, Vol. 14, No. 2, pg. 389-418. ISSN 0004-5411.