

Context free grammars with storage

Filip Mutňanský (xmutna00)
Filip Klembara (xklemb00)

November 6, 2019

1 Abstract

This presentation introduces context-free grammars with storage, also known as context-free S grammars or $CF(S)$ grammars. Basic context-free grammar (CFG) represents the idea of recursion. It is basically nondeterministic program consisting of recursive procedures without parameters. The idea behind context-free S grammars is that each non-terminal of CFG has one parameter of given type (where S is the type of the parameter).

A storage type S is a set of objects (called storage configurations), plus allowed tests and operations on these objects. Since context-free S grammars can be viewed as transducers, it is needed to define set of input elements, with the set of possible encodings to storage configurations.

There are many formalisms corresponding to the context-free S grammars. Depending on the choice of S (e.g., pushdown, counter, etc.), context-free grammars can be used to model different formalisms (e.g. attribute grammars, macro grammars, indexed grammars, top-down tree transducers, etc.)

Context-free S grammars can be represented by pushdown S automata. A pushdown S automaton can be described as classic pushdown automaton where each pushdown cell contains an object of type S . Convenient fact about pushdown S automata is that it corresponds to all formalisms that can be written as context-free S grammars (e.g. attribute grammars, macro grammars, etc.).

At first the presentation defines context free S grammar and its relation to attributed context free grammars. Then we define pushdown S automata as a method to accept all languages generated by context free S grammars. At the end of our presentation we define strength of context free S grammars and pushdown S automata.