Abstract

Parsers are used in many applications such as compilers, NLP and other applications. Parsers that are developed by hand are a complex task and require a generator to automatically generate the parser. The generator reads a grammar and generates a fully working parser.

This paper proposes separating the semantic actions’ execution from the parsing phase. The parser generates a queue of semantic actions attached with grammar rules to be visited in case of successful parsing. By this separation, the execution time of the parsing phase can be enhanced. More importantly, this will avoid the incorrect execution of semantic actions when dealing with non-deterministic grammars. Investigating an implementation for the parallelization of the parsing phase for non-deterministic rules is also another contribution of the paper. A previous theoretical work of this paper was made in [1]. The experimental work shows that working with single threaded backtracking with storage of intermediate results, as well as following the Fork/Join parallel execution model without intermediate storage perform in most cases better than working with raw threads execution or by predicting rules as in ANTLR V4.
An Implementation of a Fast Threaded Nondeterministic LL(*) Parser Generator

The generator assumes that the grammar is left-recursive free.

References

An Implementation of a Fast Threaded Nondeterministic LL(*) Parser Generator


Index Terms

Computer Science Applied Sciences

Keywords

Non-deterministic grammar, LL grammar and compilers