

## Research Article

### Ambiguity Detection Methods in Context Free Grammar

Sundus Shaukat, Kashif Ayyub, Wasif Nisar, Mudassar Raza and Tariq Umer

Department of Computer Sciences, COMSATS Institute of Information Technology, Wah Cantt, Pakistan

**Abstract:** Problem arising in CFG (Context Free Grammar) due to ambiguity can be trace to 1962. Even now there is no general method or procedure introduced to detect ambiguity in CFG. In parser generation and in language design, ambiguity in context free grammar, is a frequent problem as well as in application where it is used for the representation of physical structure. For creating a language it should be necessary that it is unambiguous. Ambiguity has some advantages as well as disadvantages. The aim of this study is to analyze different methods dealing with the ambiguity detection in Context Free Grammars. In this study, we will observe usefulness of Ambiguity Detection Method (ADM) in CFG with respect to ambiguity detection, assurance of termination of the process and accuracy.

**Keywords:** Ambiguity, CFG, detection, review

#### INTRODUCTION

CFG is one of the classes which are used to define rule of grammar. CFG is primarily used to construct compiler or conform the structure of computer programs. Languages which are constructed by using CFG require a unique formation for each line which the grammar generates. But mostly it is not possible to have only one formation for each line by the definition of CFG. Simmons *et al.* (2011) the possibility of having more than one formation is known as ambiguity which causes serious problems.

Ambiguity cause problem in language translation and interpretation. Because of that unambiguous language is preferred. A procedure should be introduced to distinguish between ambiguous and unambiguous language. There is general algorithm available for this method. The ambiguity detection algorithm discussed in the literature review is used find whether the certain sentence of a grammar is ambiguous or not and if the given set of string is finite then find if it is ambiguous (Jampana, 2005).

Parsing technique let in use of entire CFG class for specifying programming languages. Passos *et al.* (2007) This helps the grammar developer to have choice he want to have in his grammar but this also can have the risk of unnecessary ambiguity. Some grammars are created to be unambiguous while other has ambiguity to some degree. But in both cases the source of ambiguity should be known so it can be removed or distinguished. But there is no general method for detecting ambiguity. There are a few ambiguity detection methods available that can be with their own weakness and strength.

#### LITERATURE REVIEW

Vasudevan and Tratt (2012) discussed that there are more than one way to allow input in programming language. Statically it is undecidable to detect ambiguity. In this study, ambiguity searching techniques are introduced.

In 2001, Flajolet (1987) discussed the analytic method through which it is tested that CFG is inherently ambiguous. To test this different theorems of context free language are used. It is found by testing that some grammar inherently ambiguous and there are some conditions through which that can be identified.

In 2007, Schmid (2004) discussed efficient parsing of highly ambiguous CFG with bit vector method. It represents the efficient context free parsing. It provides analysis for context free Treebank grammar and lays input sentences. Large CFG have big ambiguity tree with cause problem in parsing.

In 2002, Wich (2000) discussed that ambiguous grammar can be found by the degree of ambiguity. Some tools are introduced for examining ambiguity of CFG; in this manner ambiguity function is formed. This function records the number of grammar and their maximum numbers of derivation trees which a grammar can have.

In 2008, Basten (2009) discussed that there are some benefits of having ambiguity. Through ambiguity grammar can be detected. In CFG there is no general method to detect ambiguity but there are some methods through which ambiguity can be detected but they are sometimes not effective. Three methods are being

**Corresponding Author:** Sundus Shaukat, Department of Computer Sciences, COMSATS Institute of Information Technology, Wah Cantt, Pakistan

This work is licensed under a Creative Commons Attribution 4.0 International License (URL: <http://creativecommons.org/licenses/by/4.0/>).

discussed which are AMBER, LR (k) test and the Noncanonical Unambiguity test.

Basten (2007) wrote a thesis on ambiguity detection method on CFG. In which he discussed different methods to detect ambiguity which are Gorn, Cheung and Uzgalis, AMBER, Jampana, LR (k) test, Brabrand, Giegerich and Moller, MTSA and Schmitz. These methods are analyzed and the reason for ambiguity produced is detected. Basten also discussed research future use.

In 2007, Bouwers *et al.* (2008) discussed that different languages have different method of compiling. There are methods and tools to compare different precedence rule to define different grammars and parser generator.

In 2011, Basten (2007) wrote a thesis on ambiguity detection for programming language grammars. In which he discussed all aspects of ambiguity which are usability of ambiguity detection methods for CFG, faster ambiguity detection by grammar filtering, tracking down the origins of ambiguity in CFG, scaling to scannerless, implementing AMBIDEXTER and parse forest diagnostics with Dr. Ambiguity. To discuss these topics different methods are being used like AMBER, LR (k) test and the Noncanonical Unambiguity test.

Pandey (2012) discussed ambiguity and different method for detecting ambiguity. He discussed eighty four different type of grammar for detecting ambiguity. He also tested those grammars with help of MATLAB.

In 2004, Wich (2005) wrote thesis on ambiguity functions of context free grammars and languages. In which different basic definitions are defined relating context free grammar. Different properties, functions and method of ambiguity relating the context free grammar are discussed in this thesis.

Kruse and Pfahler (2008) wrote a thesis on ambiguity detection for context free grammars in Eli. In this thesis, he discussed different ambiguity detection methods, their design and how they should be implemented. He also evaluated these techniques and also proposed work for the future.

## METHODOLOGY

Following are different methods which are used for detecting ambiguity in context free grammar:

- Gorn's method
- Cheung's and Uzgalis's method
- Jampana's method
- Brabrand's, Giegerich's and Moller's method
- Schmitz's method

The most used and current methods for detecting ambiguity in context free grammar are given below.

**Gorn's method:** Gorn (1963) proposed this method in 1963. In this method, it is discussed that Turing machine can create all probable type of strings from a grammar which is known as derivation generator. When a new string is generated, it investigates whether this string has been created before or not. If it is generated before, the string become ambiguous and it has multiple derivatives. The algorithm for this method is simple which is breadth first. It can find every probable derivatives of the defined grammar which begins with a start symbol then produce new production by expanding the non-terminals. Production or string containing terminals are considered as the end statement of the grammar and tried to reach to that level of derivation expanding the non-terminals.

Gorn's method could not expire for the use of unambiguous recursive grammar. This method has infinite number of derivation. As language being finite, the grammar is not very useful. This method is competent for finding ambiguity in a specific grammar. As the grammar has infinite derivation, it is impossible to run this method eternally. It has to be stopped at some point. Even the method find an ambiguity, it is all uncertain about an undiscovered ambiguity.

**Cheung's and Uzgalis's method:** This method was proposed by Cheung (1995). This method consists of breadth first search which help in reducing all the potential derivatives of the grammar. It can also be known as optimized Gorn method. It produces every potential statement of a grammar and search whether a statement has been duplicated. The method also ceases the extension of statements due to specific reasons. The statement is terminated when the result is unable to locate an ambiguous string or same type of statement is found. For finding this, the production has to be compared with pre and post fixed of the desired statements. For example, finding a statement that terminates different pre and post fix terminals and it middle part expand independently. This helps in finding a repetitive pattern. Following is a good example relating the pattern:

$$A \rightarrow X|Y$$

$$X \rightarrow xX|x$$

$$Y \rightarrow yY|y$$

From the above given example, it is shown that when start symbol A expands it results in string X and Y. When string X expands it results in string xX and x. Similarly when string Y expands it results in string yY and y. now the expansion will be terminated. Otherwise the string will be duplicated. The pre fix, post fix and middle part of the terminal have been expanded in the

second and third statement. The ambiguity detection method then terminates and informs non-ambiguity.

After certain derivations, the method stops expanding. It may be due to the reason that it terminates the recursive grammars (Cheung, 1994). But this does not always happen. This method can detect ambiguity accurately just like Gorn's method. This also locates non-ambiguity with infinite language for some grammar. If all the statement is terminated prior to the detection of ambiguous statement then grammar is known as unambiguous.

**Jampana's method:** An ambiguity detection method was introduced by Jampana (2005). He proposed this method for his Master's Degree. In this method, he assumed that ambiguity formed in the Chomsky Normal Form (CNF) can transpire derivation for which all the production is used at least once. The main focus of Jampana's algorithm is to convert the entered grammar into CNF and then explore the derivatives for those duplicate strings just like Gorn's method.

The method assured to terminate all the grammar introduced. Set of derivatives without any duplicate production of CNF grammar is finite. According to his thesis, the assumption of repetition of production has been declared faulty. If this assumption was correct, the set of production has to be verified to make sure ambiguity is finite. Unable to find repeated derivatives will the ambiguities unable to detect. This will show that algorithm is producing false result.

**Brabrand's, Giegerich's and Moller's method:** Brabrand *et al.* (2010) proposed a method in 2007. In this method, ambiguities are explored vertically and horizontally in a grammar. Horizontal ambiguity is investigated by the single production while vertical ambiguity is investigated by the group of production for similar non-terminals. In this method, ambiguity is because of the language and not because of the grammar. Vertical ambiguity is formed when intersection of the two production of the same language is not empty. Horizontal ambiguity is formed when different languages overlap then production can be divided into two. This method actually conform the languages instead of the definite production. The language can estimate the overlapping and intersection of the production. The algorithm used in this method is introduced by Mohri and Nederhof (2001). The

algorithm expands the language that articulates the regular grammar. That is known as conventional approximation. This is due to the reason that string of the original language is part of the regular language. When regular language is constructed, the value of overlapping and intersection is calculated. Horizontal and vertical ambiguity is searched when the result is not empty.

**Schmitz's method:** Schmitz (2007) proposed a method for detecting ambiguity in 2007. This method use estimation for forming a search space which is finite. This method is formed by the use and combination of different estimation technique. The algorithm basically finds various derivatives of the similar string of a grammar. The first step of this method is to convert the entered grammar into bracket form. Two terminals are introduced for all the production which is derivation and reduction. They are denoted by *d* and *r* respectively. Letter *n* will represent the number of production. These two terminals are introduced in every production of the grammar, forming new group of production. All the derivatives of original grammar possess an exclusive string of the bracketed grammar.

## RESULTS AND DISCUSSION

In this study, different ambiguity detection methods have been discussed and all the discussed methods have their strengths and weakness as well. The method proposed by Gorn does not assure the termination of the grammar and neither has it detected non ambiguity. This method cannot be considered best for detecting ambiguity. The Cheung's and Uzgalis's method does not assure termination, that's why it is not considered the top method for ambiguity detection. The Jampana's method does not show accuracy, that's why it is not considered the best method for detecting ambiguity.

## CONCLUSION

The method proving to be more effective for detecting ambiguity and resolving it is the one proposed by Schmitz as this method assures the termination and show the accuracy as well as shown in Table 1. It creates a unique grammar which avoids the language from creating ambiguity.

Table 1: Comparison between techniques

Technique	Assures termination	Detect ambiguity	Detect non-ambiguity	Show accuracy
Gorn's method	No	Yes	No	Yes
Cheung's and Uzgalis's method	No	Yes	Yes	Yes
Jampana's method	Yes	Yes	Yes	No
Brabrand's, Giegerich's and Moller's method	Yes	Yes	Yes	Yes
Schmitz's method	Yes	Yes	Yes	Yes

## REFERENCES

- Basten, H.J.S., 2007. Ambiguity detection methods for context-free grammars. M.A. Thesis, Universiteit van Amsterdam, August.
- Basten, H.J.S., 2009. The usability of ambiguity detection methods for context-free grammars. *Electron. Notes Theor. Comput. Sci.*, 238(5): 35-46.
- Bouwers, E., M. Bravenboer and E. Visser, 2008. Grammar engineering support for precedence rule recovery and compatibility checking. *Electron. Notes Theor. Comput. Sci.*, 203(2): 85-101.
- Brabrand, C., R. Giegerich and A. Møller, 2010. Analyzing ambiguity of context-free grammars. *Sci. Comput. Program.*, 75(3): 176-191.
- Cheung, B.S.N., 1994. A Theory of automatic language acquisition. Ph.D. Thesis, University of Hong Kong.
- Cheung, B.S., 1995. Ambiguity in context-free grammars. *Proceedings of the 1995 ACM Symposium on Applied Computing*, pp: 272-276.
- Flajolet, P., 1987. Analytic models and ambiguity of context-free languages. *Lect. Notes Comput. Sc.*, 49(2): 283-309.
- Gorn, S., 1963. Detection of generative ambiguities in context-free mechanical languages. *J. ACM*, 10(2): 196-208.
- Jampana, S., 2005. Exploring the problem of ambiguity in context-free grammars. M.A. Thesis, Oklahoma State University, Stillwater, OK.
- Kruse, M. and P. Pfahler, 2008. Ambiguity detection for context-free grammars in eli. B.A. Thesis, University of Paderborn, Paderborn, May 7th.
- Mohri, M. and M.J. Nederhof, 2001. Regular Approximation of Context-free Grammars through Transformation. In: Junqua, J.C. and G. Van Noord (Eds.), *Robustness in Language and Speech Technology*. Kluwer Academic Publishers, the Netherlands, pp: 153-163.
- Pandey, H.M., 2012. A case study on ambiguity detection method using some set of grammars. *Pac. J. Sci. Technol.*, 13(1): 277-286.
- Passos, L.T., M.A. da Silva Bigonha and R. da Silva Bigonha, 2007. A methodology for removing LALR(k) conflicts. *J. UCS*, 13(6): 737-752.
- Schmid, H., 2004. Efficient parsing of highly ambiguous context-free grammars with bit vectors. *Proceedings of the 20th International Conference on Computational Linguistics, (COLING, 2004)*. Geneva, Switzerland.
- Schmitz, S., 2007. Conservative ambiguity detection in context-free grammars. *Proceedings of the 34th International Conference on Automata, Languages and Programming (ICALP'07)*, pp: 692-703.
- Simmons, R.J., B. Toninho and F. Pfenning, 2011. Distributed deductive databases, declaratively: The L10 logic programming language. *Proceedings of the X10 Workshop, ACM*.
- Vasudevan, N. and L. Tratt, 2012. Search-based ambiguity detection in context-free grammars. *Proceeding of ICCSW*, pp: 142-148.
- Wich, K., 2000. Exponential ambiguity of context-free grammars. *Proceedings of the 4th International Conference on Developments in Language Theory*, pp: 125-138.
- Wich, K., 2005. Ambiguity functions of context-free grammars and languages. Ph.D. Thesis, University of Stuttgart.