Non-Terminal Encyclopedia Article

Non-Terminal

The following sections of this BookRags Literature Study Guide is offprint from Gale's For Students Series: Presenting Analysis, Context, and Criticism on Commonly Studied Works: Introduction, Author Biography, Plot Summary, Characters, Themes, Style, Historical Context, Critical Overview, Criticism and Critical Essays, Media Adaptations, Topics for Further Study, Compare & Contrast, What Do I Read Next?, For Further Study, and Sources.

(c)1998-2002; (c)2002 by Gale. Gale is an imprint of The Gale Group, Inc., a division of Thomson Learning, Inc. Gale and Design and Thomson Learning are trademarks used herein under license.

The following sections, if they exist, are offprint from Beacham's Encyclopedia of Popular Fiction: "Social Concerns", "Thematic Overview", "Techniques", "Literary Precedents", "Key Questions", "Related Titles", "Adaptations", "Related Web Sites". (c)1994-2005, by Walton Beacham.

The following sections, if they exist, are offprint from Beacham's Guide to Literature for Young Adults: "About the Author", "Overview", "Setting", "Literary Qualities", "Social Sensitivity", "Topics for Discussion", "Ideas for Reports and Papers". (c)1994-2005, by Walton Beacham.

All other sections in this Literature Study Guide are owned and copyrighted by BookRags, Inc.

Contents

Non-Terminal Encyclopedia Article.....1

Contents.....2

Non-Terminal.....3

Non-Terminal

A non-terminal element operates in context-free grammar. This form of grammar is necessary to enable the translation of source **code** to **object** code in a meaningful way, so as to preserve all the information. Specifically, a non-terminal element, which can also be called a **variable**, is capable of being broken down into other elements, called tokens or **terminal** elements. A terminal element cannot be broken down into other symbols. Terminal elements of source code are typically passed on to the parser, which is a program that dissects the source code in order that the code can be translated into object code.

A second meaning for non-terminal relates to query languages such as the Abundantia Verborum, languages that are hierarchical, or tree-like, in their structure. The hierarchical approach to problem solving in this scheme is also referred to as a semantic **tree**. In this structure, the non-terminal nodes types the AND, OR, and NOT function to combine so-called atomic queries--that address a single issue and are not qualifiable--into more complex queries. The non-terminal **node** types are Boolean operators; that is, they produce an outcome that is true or false. The larger queries, which are qualifiable due to the AND, OR, and NOT nodes, enable a more all-encompassing query to be constructed.