

9-2013

LING 572.01: Generative Syntax

Tully J. Thibeau

University of Montana - Missoula, tully.thibeau@umontana.edu

Let us know how access to this document benefits you.

Follow this and additional works at: <https://scholarworks.umt.edu/syllabi>

Recommended Citation

Thibeau, Tully J., "LING 572.01: Generative Syntax" (2013). *Syllabi*. 492.
<https://scholarworks.umt.edu/syllabi/492>

This Syllabus is brought to you for free and open access by the Course Syllabi at ScholarWorks at University of Montana. It has been accepted for inclusion in Syllabi by an authorized administrator of ScholarWorks at University of Montana. For more information, please contact scholarworks@mso.umt.edu.

Linguistics 572
Generative Syntax
Fall 2013
TR, 12:40 to 2:00, GBB L11

Dr. Tully J. Thibeau
Office: Social Sciences 208 (x2156)
Office Hours: M 1:10-2:30, T 11:10 -12:30
Email: tully.thibeau@umontana.edu

This syllabus for the graduate course in Generative Syntax (LING 572C) serves as a supplement to the syllabus for the co-convening undergraduate course (LING 472) under the identical title; that is, graduate and undergraduate students meet together according to the same course schedule and confront the same lecture and reading material specified in that schedule for class meetings, but the quantity and, more importantly, the quality of the work that is completed for a final grade in the course is of a higher order for graduate students (otherwise called a graduate increment).

In other words, whereas both groups of students who convene during course meetings try to develop skills in methods of linguistic analysis particular to the science of sentence-formation (syntax), graduate students enrolled in LING 572 apply analytical syntactic methods at a caliber that heightens understanding of human language as “an abstraction of utterances in the form of mathematical objects” (see the course description for LING 472 in the university course catalog).

Consider the notion *constituent*, one or more words functioning as a single unit, a notion preceding the inception of generative syntax that is representable using formal *bracket* notation:

[~~{~~ cats ~~}~~ [~~{~~ chase ~~}~~ ~~{~~ mice ~~}~~]]

The outer brackets represent the sentence constituent, and each word also receives its own set of brackets (in ~~strike~~through); however, another set of brackets (in **bold**) represent the notion that *chase mice* functions as a constituent independently of the individual words contained therein. This intuition can be tested for constituency by applying a grammatical operation that is known as clefting (breaking the sentence in two) whereby *chase mice* is displaced from its basic position and relocated at the left-edge position of a new derived sentence that adds several other words:

[chase mice] is what cats do

Conversely, the clefting transformation that changes the basic sentence into a derived one cannot operate on the words *cats chase* because no single set of brackets **exhaustively** contains them:

* [cats chase] is what mice undergo (* means ungrammatical)

While methods of syntactic analysis prior to the advent of generative syntax can conceptualize layers of constituency graphically (e.g., bracketing), no technological counterpart existed based on such formal notation could operationalize what human beings know intuitively about how sentence-formation systems work (grammaticality) why at times they do not (ungrammaticality).

This point, intuitions of what is, respectively, well-formed and ill-formed constituencies, is the lynchpin of a generative syntax, or a sentence-formation system that is sufficiently general, a machine that fabricates every grammatical sentence constituency (an infinite number) and does not fabricate ungrammatical ones (intuitions of ill-formedness). The sentence-fabrication machine is an analogy (either apt or false) for a mental faculty that fundamentally characterizes humans. The first person to crack the code of infinity was a graduate student named Noam Chomsky, who was studying linguistics at MIT and devised a programming language prompting the innovation of a new automaton (i.e., computer) that modeled, to a degree, humans' capability of infinity.

Incarnations of his work attempt sufficient generality yet remain computer models, and the only language computers understand is mathematic. This course covers development of generative syntax from Chomsky's graduate-student years in the early 1950s until the mid 1980s.