Enthymeme as Rhetorical Algorithm

Kevin Brock
University of South Carolina

http://www.presenttensejournal.org | editors@presenttensejournal.org
Enthymeme as Rhetorical Algorithm

Kevin Brock

Rhetoric as a discipline has primarily concerned itself with heuristic models of persuasive action. Indeed, a tremendous amount of scholarship, primarily in the late 1970s to early 1980s, has focused on heuristic practices of invention in particular, from general concerns regarding the use of rhetorical appeals (Boley) to dramatistic ratios as inventive tools (Kneupper) to the employment of topos in the composition curriculum (Adams). Heuristic practices are most effective when applied to a macrocosmic scale, where overarching questions of exigence, audience, and genre can be addressed, due to heuristic’s inherently flexible nature and its connection to rhetorical invention and discovery. In fact, the Greek verb heuriskein literally means “[to] discover” (Crowley and Hawhee 12).

However, this focus on heuristic overlooks algorithmic models of persuasion employed in support of the heuristic process(es) of invention. We already incorporate algorithmic procedure into our rhetorical activities—we simply don’t always recognize it. Most likely, the avoidance of algorithmic exploration as a valid rhetorical tool stems from a definition of “algorithm” as a discrete, repeatable process for achieving a reliable result—or, as Bethany Nowviskie described it, “any effective procedure that reduces the solution of a problem to a predetermined sequence of actions” (1). While this definition reflects a particular notion of procedure, it becomes complicated when considering more contingent circumstances; Nowviskie refined “algorithm” as a concept when applied to ambiguous contexts, including “artistic production and to the interpretation of texts and artifacts” (3). For rhetoric, as we turn from larger-scale heuristic considerations (e.g., “what kind of response do I want to induce from this audience?”) into more specific and familiar territory as part of our problem-solving activities (e.g., “how do I frame this point to resonate with this audience toward my desired response?”), we turn to persuasive procedures, or rhetorical algorithms, for solutions.

In this article, I question rhetoric’s preference for the heuristic by reexamining one of the oldest and most fundamental tools available to a rhetor: the enthymeme. The enthymeme, while serving as the basis for heuristic invention, also works at the local level as a rhetorically oriented algorithmic procedure through which a rhetor determines the most probable success for persuading an audience to action. Procedural rhetoric, demonstrated through the enthymeme, provides a means for understanding persuasive engagements with, through, and by digital technologies.

Understanding Rhetorical Procedure

Algorithmic procedure fundamentally involves a set of operations meant to complete a task. These operations demand the computation of one or more variables for the operations to be successfully examined, undertaken, and completed. Usually, algorithms are considered within the domains of mathematics, computer science, and electrical or computer engineering;
however, algorithms are not always so restrictively defined. As Christopher Steiner has noted, “Algorithms needn’t involve graduate school math or even math at all. The Babylonians employed algorithms in matters of law, ancient teachers of Latin checked grammar using algorithms, … and countless numbers of men from all corners of the Earth have used them in an attempt to predict the future” (54). Algorithms are then not restricted to engineers or mathematicians as tools for describing particular situations and the potential ways those situations might be addressed. It is only through the term’s recent connotation gained from the growth of computer science that gives the general population some pause.

Rhetors have at their disposal a powerful algorithmic procedure that exists at the very heart of rhetorical activity alongside the example; as Aristotle noted, “All [speakers] produce logical persuasion by means of paradigms or enthymemes and by nothing other than these” (Aristotle I.i.i.8). Specifically, I refer to the “enthymeme” as a rhetorical algorithm, an incomplete syllogism whose implicit, unstated completion is realized only when an audience is persuaded to perform that completion—optimally in a manner aligned with a rhetor’s intent. What a rhetor can do with that enthymeme depends on whether the audience, being inclined to accept the basic premises of the argument, has successfully computed the necessary variables involved and their operations of delivery. John T. Gage noted that an enthymeme can only be discovered and written in the mental presence of an audience because it is the audience that is mutually engaged in the definition of the question at issue that the enthymeme addresses. The audience is mutually engaged in the definition of the shared assumption that allows the logic of the enthymeme to function (8).

That is, the enthymeme is constructed and completed by a rhetor and audience working in concert to reach the same conclusion about an issue. Jeffrey Walker described the enthymeme as “a stylistically intensified argumentative turn that serves not only to draw conclusions but also … to foreground stance and motivate identification with that stance” (55). For example, if I assert that my friend Lauren has horrible taste in movies and also that she and her partner, Alex, both love the Adam Sandler movie Jack and Jill, then my ultimate position cannot be clarified until the audience first comes to understand that I have implied that Jack and Jill is a horrible movie and also that Alex, like Lauren, must possess horrible taste in movies. Optimally, the audience will agree with my evaluation both of the movie and of Alex and Lauren’s cinematic taste.

This example is incredibly simple, but it demonstrates a point central to the idea of rhetoric as algorithmic: there is a logical procedure to be constructed that extends from the first moments of invention through the rhetor’s inevitable delivery of his or her message. Further, its execution must be played out in the mind of the audience in order to be considered successful as a component of the overall persuasive effort. That is, the enthymeme exists at both the larger-scale heuristic level and the smaller-scale algorithmic level. The success of the larger-scale effort is not guaranteed by the individual enthymemic procedure of a given set of statements, whose optimal success occurs through an audience’s labor in completing that specific enthymeme (see Ed Dyck on how particular topoi influence the construction and execution of individual
enthymemes); however, given effective constraints upon potential ranges of audience interpretations—constraints employed as part of the larger-scale heuristic practice—the types of outcomes generated may be sufficiently limited to facilitate a generally successful endeavor.

Computer Technologies and Rhetorical Algorithms

It is not necessary to understand how computers use algorithms in order to understand the enthymeme as a rhetorical algorithm; however, it is to our benefit that we explore how rhetoric and the enthymeme in particular play critical roles in the construction and execution of algorithms in seemingly a rhetorical contexts and situations, such as those surrounding the development and implementation of computer systems (comprising hardware and software alike).

Ian Bogost has argued for an approach to rhetorical activity that he calls “procedural rhetoric,” which he defined as a means by which an argument can be composed in order to help an audience understand how to engage with a particular system, world, or line of thought. Specifically, Bogost stated that procedural “arguments are made not through the construction of words or images, but through the authorship of rules of behavior, the construction of dynamic models” (29). These rules of behavior include probabilities as well as prescriptions of action—such as how an audience will (likely) respond to a particular case made by a rhetor. The contingent audience interpretations from a rhetor’s efforts, constrained by the variables offered implicitly and explicitly by that rhetor, will impact how the argument, as well as any action in response to that argument, unfolds.

If This, Then That

Among the simplest algorithmic procedures is a basic conditional statement: something should happen when a particular situation facilitates its happening. This idea can be described more concisely as “If this, then that.” For example: if I have no food in my refrigerator, then—assuming I want to eat—I need to visit the grocery store. While this statement can be formed into an imperative (“Visit the grocery store”), it is flexible enough to deal with contingency and probability—, which is exactly the sort of algorithmic problem solving that informs heuristic processes of rhetorical invention. If a political candidate speaks to Democratic party supporters, then that candidate probably should not argue for limiting government funding for social programs—that is, he or she should not do so if the candidate wants the support of that audience for his or her candidacy. If an academic speaks to a public audience about the impact of highly technical research in biomedical engineering, then he or she probably should use language accessible to a broad community rather than engineering jargon to improve the chances that the audience will understand the academic’s point. One can make an argument that goes against these evaluations—the condition does not prevent the possibility of divergence or rejection. As Maxine C. Hairston noted, “The person who argues from an enthymeme is [usually] not trying to prove a proposition; he or she is only trying to establish high probability” (76). The conditional procedure sets up likelihoods of computational potential connected to the conditions influencing rhetorical activity. As a result, the executed argument is a complex heuristic
structure comprising numerous algorithmic operations that lead one to complete an activity: to persuade an audience to do something.

If This, Then What?

Of course, neither is that something always the intended result, nor is the rhetorical procedure always executed as desired. As Wendy Chun has described it, computer code as a type of algorithmic procedure “does not always or automatically do what it says, but it does so in a crafty, speculative manner in which meaning and action are both created” (24). If a rhetorical algorithm has not been constructed or executed successfully (in code or not), then it might iterate through different, possibly unanticipated, operations and invent novel persuasive encounters or “glitches” (Boyle). The algorithm’s rhetorical contingency, its potential for glitches or unintended actions, overpowers its probability of successful intent. For example, Google’s ReCAPTCHA software is a means for distinguishing spam accounts from humans by asking users to accurately type a pair of distorted words displayed from a vocabulary populated by digitized texts and Google Maps images (“ReCAPTCHA”). Because the display is composed randomly, users often encounter, and read unintended meaning into, profanity, nonsensical phrases, or even communication seemingly directed toward the user—all of which influence how users perceive their relationships with ReCAPTCHA (“Make Sure”; “Are You”; “With a Side”).

Beyond code, there are numerous examples of unintended algorithmic outcomes due to miscalculations of audience or generic expectations, such as the Twitter reading of Susan Boyle’s record release hashtag #susanalbumparty as “Su’s Anal Bum Party” instead of “Susan Album Party” (“The Great”). More recently, singer Robin Thicke’s Twitter Q&A hashtag #AskThicke, intended for comments from fans, was flooded with questions about his potentially misogynistic behavior and song lyrics (Broder Van Dyke). It is in this potential for contingency where resistance to the notion of rhetoric as algorithmic most thrives; however, I argue that it is in this contingent space where the algorithm’s persuasive potential is tested most rigorously.

Digital Literacies and Rhetorical Algorithms

This examination of enthymeme as rhetorical algorithm might be especially fruitful for conversations on technological literacies, considering many of these conversations incorporate algorithmic computation as a key factor in literacy knowledge and practice. While there has been relatively robust discussion in the last two decades on electronic literacies, it is only recently that the algorithm has become a prominent factor of discussion.

Among the most intriguing of the relevant conceptions of algorithmic literacy is “proceduracy,” defined by Annette Vee as “the constellation of abilities to break a complex process down into small procedures and then express—or ‘write’—those procedures in a standard format that may be ‘read’ by a nonhuman entity such as a computer” (3). Vee’s proceduracy provides us with an intriguing perspective through which to engage the complex digital technologies with which most of us surround ourselves and by which we are continuously influenced every day. Similarly, David Berry called for “literacy,” which he defined as a skill related to reading and writing code. These and related literacies have become a topic of serious concern for scholars interested in
digital technologies; this concern has only increased since 2012, dubbed the “Code Year,” when a metaphorical avalanche of tutorials and programs proclaiming to support code-based literacy was unleashed upon the public (“Code Year”).

What do these conversations on computational literacy contribute? If we accept that rhetoric already incorporates significant use of algorithmic procedure as an integral component for invention, then it might become easier to understand how algorithms work in and through computer technologies. In fact, we might be more likely to recognize that seemingly “scientific” or otherwise entirely discrete algorithms are as rhetorical and contingent upon interpretation for recognizable success as any conventional persuasive effort. As the greatest existing obstacle to this recognition is the conventional paradigm of computer technology as objectively logical, a shift toward understanding it as rhetorical could radically transform our engagements with electronic media. Bogost’s notion of “procedural rhetoric” can be employed most skillfully when one has a literate grasp—in whatever form—of the sorts of procedural systems through which to compose a given argument.

Conclusions

From a computational perspective, procedural activity is often viewed as context-free, meaning that what is processed matters less than that a procedure occurs. Of course, such an understanding does not really hold. The information provided from procedural action means something, and each step of a procedure is meaningful in how it shapes, and is shaped by, the user constructing and executing it.

From a rhetorical perspective, procedural (algorithmic) activity is often viewed as irrelevant or non-deliberative, meaning that there is perceived to be little or no persuasion occurring as part of the execution of a given procedure. This perspective is similarly erroneous. Because the choices involved in acting possess meaning, how we determine a particular course of persuasive effort must also be significant. The enthymemes we employ are forms of algorithmic procedures through which we induce audiences to action as part of a larger-scale heuristic.

It is in our hands to understand how we can construct and use algorithmic procedures for persuasive ends to work against the systems that maintain a social structure exploiting unwitting populations—such as through increasing citizen interest in a particular political cause, demystifying bureaucratic obstacles to cancelling cable service, or attending to how certain demographics are (un)represented in video games—to support or defy stereotypes; Bogost has recently written about how the Star Trek: The Next Generation episode “Darmok” champions procedural rhetoric over conventional verbal discourse. Helping students—and the publics they join—learn how to read, interpret, and create rhetorically powerful algorithms for varied purposes is a pedagogical approach that does not aim to reduce rhetoric to an inflexible or even technologically specific mechanism. Instead, it seeks to show how procedural logics influence our rhetorical acts as well as those actions that our persuasive efforts strive to instigate and change.
Works Cited


Kevin Brock is an Assistant Professor in the Department of English Language and Literature at the University of South Carolina, where he teaches courses in composition, rhetoric, and professional communication. He studies digital rhetoric, with a focus on the rhetoric of software and its development, looking at code and the discursive commentary surrounding its creation. His work has appeared in such journals as Technoculture, Enculturation, and Composition Studies.