scopio MAGAZINE
ARCHITECTURE, ART AND IMAGE

UTOPIA VOL.1

CONTEMPORARY ARCHEOLOGY
Editor Isa Neves

Techniques of Discovery: Cryptography and Design

Roberto Bottazzi

Scopio Architecture, Art and Image
Utopia Vol.1 | publication year: 2023
ISSN: 1647-8274 [online]
DOI 10.24840/1647-8274_2023-0001_0001_177

U.PORTO PRESS

scopio EDITIONS

Techniques of Discovery: Cryptography and Design

Roberto Bottazzi

Abstract:

Among the core technologies forming the rich archaeology of computation, cryptography is perhaps a subject that has received little attention in architectural studies thus far. However, there are fruitful considerations to draw from a closer inspection of the vast repertoire of techniques that cryptography has developed over a period of about seven centuries. First of all, a deeper historical perspective will help frame cryptography as a technology for discovery rather than solely protecting military and diplomatic secrets. Secondly, these considerations can be of relevance to design as they offer thoughts for both conceptual reflections and practical applications. At a conceptual level, the symbolic, discrete computation accompanying the evolution of cryptographic methods – such as Alberti's one in 1467 – marked a radical departure from the iconic semiotics of analogue machines. The non-mimetic nature of symbolic computation provided the technological means to significantly widen its range of applications and enhanced speculative thinking. Understood along these lines, cryptography found more general applications beyond concealing diplomatic secrets to provide a rigorous method for inquiry into unknown domains in order to make 'noisy data' intelligible. Finally, symbolic computation also provided more advanced techniques for abstraction that were also instrumental for constructing notational drawings, whose emergence coincided with the introduction of more advanced mathematical instruments in the renaissance.

The essay will discuss the key paradigmatic moments in the history of cryptography such as the polyalphabetic techniques proposed by L. B. Alberti in 1467 and the use of binary cryptography by Francis Bacon in 1605. Despite their distance from the present, these experiments provide a useful segue into a discussion on how the notion of cypher as a conceptual instrument accompanying the introduction of Machine Learning models in architectural and urban design.

Keywords: Cryptography, Methods, Architecture, Symbolic Computation, Machine Learning

Roberto Bottazzi is an architect, researcher, and educator based in London. He studied at University of Florence, Italy and University of British Columbia, Canada before moving to London. His research analyses the impact of digital technologies on architecture and urbanism. Roberto has lectured and exhibited internationally and is the author of Digital Architecture Beyond Computers (Bloomsbury Visual, 2018, 2021). He was Visiting Professor at the Politecnico ofMilan and Visiting tutor at the Innovation Design Engineering [IDE] at theRoyal College of Art. He is Programme Director of the MArch Urban Design at the Bartlett School of Architecture.

This essay surveys the archaeology of the digital to tease out references, techniques, and concepts that resonate with and help us think through the rapid and disruptive introduction of Machine Learning (ML) models in design. Rather than focusing on pragmatic applications or ethical conundrums, the essay will concentrate on design processes – the set of techniques tasked with the translation of ideas into actionable instructions – and how machine learning will impact them. Design techniques simultaneously constrain our imagination ("nothing gets built that isn't transposable onto AutoCad", in Alejadro Zaera-Polo's words[1]) and can be exploited to give rise to original approaches and languages. For instance, mathematical perspective produced Pienza's central square and paintings such as *The Ideal City*, whereas the early seventeenth century saw improvement in the field of optics that played a decisive factor in the formation of Baroque aesthetics.

ML models introduce several new representational techniques to encode space which dislodge established notions opening up a theoretical void. Notions such as latent space, training models, or the organisation of spatial data through statistical tools are examples of such novel techniques which mark a discontinuity with previous technologies of spatial representation. What all these new techniques have in common is that they are computational in nature. We can distinguish between the digital and the computational domain by indicating with the former digital devices in general (microprocessors, screens, etc.), while the latter is concerned with the actual operations of calculus. Computation denotes a method, a particular way to encode/decode information, it is a technology to 'reckon with' according to the original meaning of the Latin expression *computare*.[2] Though a subset of the digital, computation is a more fundamental component of any digital device as it is tasked with the manipulation, transformation, and generation of ideas and knowledge. By placing the challenges posed by ML models on the side of computational problems, we want to foreground that the introduction of ML models in design is primarily an issue concerning the organisation and instrumentalisation of knowledge and, only successively, the invention of novel forms. In short, the issues and opportunities engendered by applying ML models to design are strategic before being formal.

How should designers think of new notions such as statistical distribution and inductive training when designing? At a basic level, ML models are implemented to seek out patterns and correlations in very large and variegated datasets. Clustering algorithms – a particular branch of ML – categorises data to extract functions to then be used for generative purposes. The designer tasks the algorithmic procedures to probe the input datasets, to seek for patterns to

1. Quoted in Bernard Cache. "Towards a Non-Standard Mode of Production". In *Projectiles* (London: Architectural Association, 2011), p.61.

2. *Online Etymology Dictionary*. "Compute". Accessed 6 February, 2024. Available from: https://cse.buffalo.edu/~rapaport/584/computetymology.html#:~:text=The%20word%20'compute'%20comes%20from,arithmetic%2C%20accounting%2C%20reckoning%22.

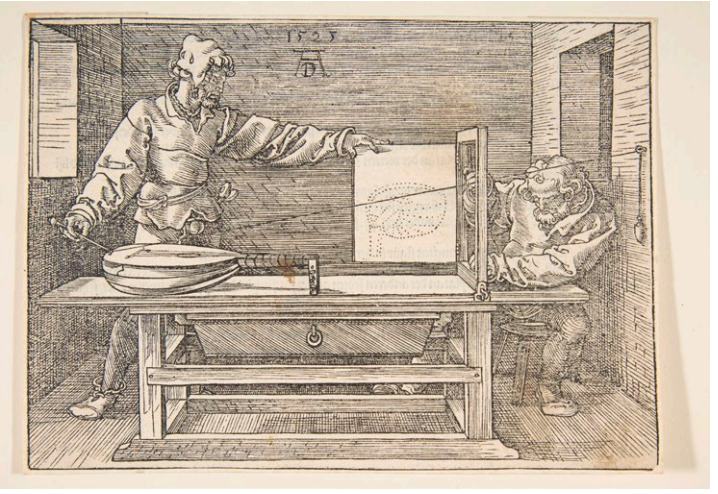
compress it. 'Noisy' input data are ordered to give rise to intelligible and meaningful outputs. By sifting through the potentially endless sequences of trivial or spurious correlations, designers seek for original, counter–intuitive, perhaps overlooked patterns that can be lifted out from the algorithmic process to guide the design investigation. The abstract nature of data provides a technology for quantification that can encompass many aspects of objects, well beyond the ones we can perceive. Out of the archaeology of the digital, cryptography represents a fundamental subject of computation that can offer a useful analogy to conceptualise the use of ML models in design. Cryptography, in fact, offers a vast repertoire of rigorous methods to access and manipulate objects beyond their phenomenal qualities, to negotiate between noise, randomness and intelligibility; all conditions that also characterise the exploration and organisation of input data by ML models. The specific element tasked with establishing a bridge between noisy and intelligible data is the cypher; an artificial code designed to connect cypher and plain texts. In some cases the analogy with cryptography can be quite literal: autoencoders – a class of ML algorithms – in fact learn an optimal function to encrypt and decrypt the input dataset. In other words, they seek to extract a cypher, a 'key' through which noisy and intelligible representations of input data can communicate.[3]

Despite the deep and rich repertoire of techniques that animated the history of cryptography, this core aspect of computation has received little attention from architects. The traditional domain of application of cryptography is not space or spatial representation, but rather that of diplomacy and military secrets. The first move is to disentangle cryptographic methods from such readings to highlight the role that they had in organising knowledge and providing accountable methods for discovery. To better grasp how different computational logics organise information and engender different kinds of creative operations, we will first review how analogue and discrete computation each work.

Analogue and Discrete computation

Different modes of computing engender different types of operations for designers to perform. This is due to the material and semiotic logic underpinning each type of computational approach. On the one hand, analogue computation makes use of materials and contraptions that can only compute continuous quantities: lengths, angles, volumes of liquids are some of the physical properties exploited. Semiotically, the parts of analogue computers establish an iconic relationship with the processes they embody. The construction of an analogue computing machine implies a process of reification of the very theory it will embody. If we take as an example the perspective machines that were so popular in the renaissance (all based on analogue computation), we can see that each material chosen and the arrangement of the different parts was a reification of the theory of vision that mathematical perspective had codified (Fig. 1).

3. 'Autoencoders'. *Wikipedia*. Available from : https://en.wikipedia.org/wiki/Autoencoder.



As iconic signs are in principle subjected to material debasement, perspective machines would perform incorrectly or not at all, if the material properties of their components worn out or parts broke down. More importantly, because of their inherent connection to material logic, analogue computers always have a visual quality that elevate the eye as the most effective sense to appreciate their functioning. In so doing, analogue computing acquires pedagogical qualities: if one is familiar with the theory computed (e.g. mathematical perspective, in the example we have been using), they will be able to 'see' it in action as materialised by the various parts of the machine. Perhaps, it is for this reason that Bill Philips based his MONIAC (Monetary National Income Analogue Computer) on analogue computation.[4] Built in 1949, the purpose of this computer was to visualise the working of a large, complex economy such as that one of a nation. As water was poured in a container placed at the top of the machine, gravity, a system of interconnected pipes and gates would visualise the flow of water/capital according to varying levels of taxation and interest rates. Water not only made MONIAC an analogue computer, but it also gave the whole system visual and pedagogical qualities that made it deliberately direct

4. Wikipedia. "Phillips Machine". January 4, 2024. https://en.wikipedia.org/wiki/Phillips_Machine. Several videos showing the computer in action can be found online.

Fig. 1 – Albrecht Dürer, The Draughtsman of the Lute.

and uncryptic. These considerations directly affect the type of operations analogue computers allow. The analogue computer is so inherently linked to material properties that its design can only be tasked to compute a specific problem first and, only successively, be applied to other cases in order to test its transferability. The process of generalisation of analogue computation is always one that moves from the specific to the general.On the contrary, discrete computation is always first and foremost posed as an universal method that can be applied to specific tasks. Semiotically, in fact, discrete computation establishes arbitrary relations between signs and things.[5] Such arbitrary relation does not initially rely on any specific material property (e.g. the Turing Machine was a thought experiment) as discrete computation is a 'portable' method; that is, it is independent of the task it is applied to and is loosely related to the material logic that embodies it. Contrary to analogue, discrete computation shares with cryptography the search for not mimetic semiotic systems. In cryptography, in fact, one seeks to maximise the gap between signs and things in order to make decryption as difficult as possible.

Anthony Wilden captures well the implications that materiality has on the type of computation performed when he states that: ''The analog computer cannot represent nothing (no-thing) because it is directly or indirectly related to 'things', whereas the 'language' of the digital computer is essentially autonomous and arbitrary in relation to 'things' (except in so far as all the information requires matter-energy in the form of markers for its transmission). The analog computer is an icon or an image of something 'real', whereas the digital computer's relationship to 'reality' is rudimentarily similar to language itself.''[6]

Finally, discrete computation is an extremely economic system as it only requires two signs in order to compute. As we shall see, though binary numeration had been long known in the West, it was within the field of cryptography and, more precisely with the work of Francis Bacon, that a system consisting of only two characters (biliteral cypher) was developed to encode all types of messages. We should note the importance of cryptography in the development of the philosophy of computing as an early example and application of binary coding emerged from this field. As we will also discuss in relation to creativity, the gap between signs and phenomena that is inherent in discrete computation should not be seen as a lack, a deficiency limiting the descriptive potential of system devised. Rather, it is precisely this gap that constitutes the conceptual space in which speculation and creativity can be articulated.

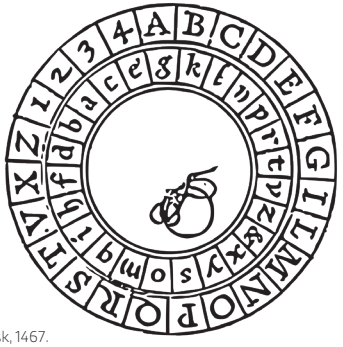
5. The characterisation of code as a disembodied concept is not entirely precise as it only accounts for how code is conceived. At its conception code is a pure abstraction which does not strictly need to exist in reality; however, computers are physical devices which require code to be inscribed onto a material support. The gates of a digital computer are the physical equivalent of the 'disembodied' 0s and 1s of Boolean logic and so are the perforated cards controlling the weaving patterns of a Jacquard loom. As Aden Evens suggests, it follows that at their very core digital computers are still analogue machines: 0s and 1s are arbitrarily assigned to, for instance, fluctuations in voltage whose variation is continuous, not discrete.
6. Anthony Wilden. *System and Structure*. (London: Tavistock Publications, 1972), pp. 162–163.

Finally, we observe a property of discrete computation: the ability to negate. Analogue computation rests on the existence of a physical property (length, voltage, etc.) which is replicated in the analogue computing device. Analogue computation necessarily establishes a positive relation between signs and phenomena. An analogue computer cannot say 'not-A'[7]; however, it can assign a zero value to one of its variables, which marks a conceptual difference between the notion of zero and that of negation. On the contrary, the denotative, arbitrary semiotics of discrete signs has the possibility to negate an object (this is 'not-A') and, therefore, to play with the disjunction between representation (code) and phenomena. As we have seen, this is a fundamental property of symbolic sign systems which are in fact nothing but a system to mark oppositions, or, better, differences from which more structured strings emerge.

From the vast archaeology of cryptographic studies, we will dwell on Leon Battista Alberti's and Francis Bacon's methods. Both experiments mark important moments in the definition of non-mimetic languages which introduced either technical innovation (the extensive use of mathematics in Alberti) or the speculative application of cryptography to investigate domains well beyond that of military secrets (Bacon).

Beyond mimesis, mathematics: Alberti's cipher

The cryptographic method introduced by Leon Battista Alberti in his *De componendi cifris* (1466 ca.) represents an important watershed in non-mimetic thinking in cryptography and computation (Fig. 2). Alberti bases his system on polyalphabetic encryption which relies on multiple cyphers to change numerous times throughout the encoding/decoding process. Two independently-rotating wheels can be aligned in order to match letters in the plain text with corresponding ones in the cypher text. By changing the alignment of the wheels repeatedly, decoding by frequency analysis becomes impossible as characters often repeated in the cypher text no longer correspond to characters frequently used in natural language.



7. *Ibid.*, p. 162.

Fig. 2 – Leon Battista Alberti, Cypher Disk, 1467.

Any possibility to rely on visual clues for decryption is eliminated and so are indexical traces to connect cypher and plain text. Whilst some of these considerations were already understood by the cryptographers of the time, the polyalphabetic nature of Alberti's invention also meant that the size of the space of all possible combinations exponentially grew each time the cypher would change. More abstract and complex methods are needed and mathematics is therefore designated as the instrument able to handle abstraction, navigate domains beyond sensible or even intelligible outcomes (such as the enormous space of all combinations) and act as communication systems tasked with the translation of string of 'meaningless' signs (cypher text) into intelligible ones (plain text). Two elements are important to foreground in this discussion. The introduction of non-mimetic notational systems is an important step towards the notion of 'portability' of discrete computation we mentioned in the previous paragraph. A symbolic, arbitrary system of signs that does not relies on material properties is more suitable to be applied universally. Secondly, Alberti's method elevates mathematics to the only technology able to provide guidance in the face of ever larger quantities of information. Mathematical instruments inform and guide discovery. This point, that will also be discussed in the section on Bacon, is also central in the use of ML models in design processes. Statistical analysis are also at work in the operations of ML models which avail of mathematics to analyse, compress, and categorise input data. In other words, they guide designers through the immense space of data and allow them to speculate on and create with them.

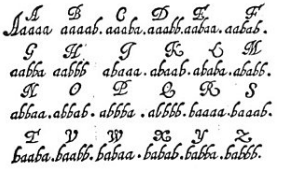
An important development to the polyalphabetic method is provided by Gottfried Leibniz's own cryptography machine. Though Leibniz' machine was never built, the descriptions we have illustrate an automated mechanism constituted by a keyboard and different rotating drums that combined to automatically translate plain text in to cypher text. Beside removing error-prone complicated operations by hand, the machine completely removes the possibility of decryption by frequency analysis as the encrypting cypher can change at varying intervals. Both principles would still underpin cryptographic methods in use in the first part of the 20th century.

Omnia per Omnia: Francis Bacon's biliteral cypher

If Alberti's polyalphabetic encryption mainly concentrates on technical improvements by assigning to mathematical techniques the role of guiding navigation through the massive combinatorial space generated, Francis Bacon's approach to cryptography innovates on both technical and conceptual levels. In his *The Advancement of Learning* (1605), Bacon identifies the perfect cypher as the one that would translate any message into any other message: "Omnia per omnia" in his words.[8] Beside Bacon's ill-considered confidence that such cypher 'is undoubtedly

8. For ciphers, they are commonly in letters or alphabets, but may be in words...But the virtues of them, whereby they are to be preferred, are three; that they be not laborious to write and read; that they be impossible to decipher; and, in some cases, that they be without suspicion. The highest degree whereof is to write omnia per omnia; which is undoubtedly possible, with a proportion quintuple at most of the writing infolding to the writing infolded, and no other restraint whatsoever." Francis. Bacon. *The Advancement of Learning* [1605] (Oxford : at the Clarendon Press, 1869) p.161.

possible'',[9] his biliteral cypher perhaps best approximates the quest for an universal systems of signals for encryption and decryption. Technically a steganographic method operating by substitution, Bacon's cypher most noticeable novelty consists in conflating plain and cypher text; that is, hiding secrets in plain sight. The presence of an encrypted message is disguised through a simple difference in the text: two distinct typefaces are used (e.g. standard and bold typeface). To decode the message, only the characters in one of the two typefaces are singled out and translated by using Bacon's cypher. Such cypher takes each letter and transforms it into a 5-digit binary code formed by the letters 'a' and 'b' only (Fig. 3). Finally, the same process is reversed to turn the binary code back into natural language. The use of two typefaces is the material expression of a more fundamental principle underpinning the logic of this system: that of difference. Bacon maintains that any system ''capable of a twofold difference onely: as by Bells, By Trumpets, by Lights and Torches, by the report of Muskets, and any instrument of like nature''[10] would meet the material requirements of his cryptographic method. Among the range of technologies that could be capable of establishing clear and unequivocal differences between members of their set are also numbers. 0s and 1s could replace the letters 'a' and 'b' and be used as cyphers. At the time of Bacon's writing, binary numeration was already a known technology, but the innovation introduced by his biliteral cypher is to establish difference as a sufficient requirement of a computing system to work. Sound, light, and, of course, numbers all happen to be technologies that can satisfy this requirement.



Previous considerations on the abstract and speculative qualities of discrete computation now acquire a clearer meaning as we can see them applied to a specific method. The gap between coded representation and things is here as wide and abstracted as possible; a condition, we argued, that is essential for speculative thinking. Such intuition did not go unnoticed and plays a central role in later experiments that are considered decisive for the construction of the modern computer. Both the idea that semiotic difference precedes the emergence of complex statements (rather than resulting from them) and that signalling system does not need to be grounded in empirical phenomena or a-priori reality also inform Hegel's logic. The abstraction of this approach is not bound to a specific application. In Hegel's words: ''With this...()

9. *Ibid.*, p.161.

10. Francis Bacon. *The Dignity and Advancement of Learning.* Latin edition, 1623, chapter 1.

Fig. 3 – Francis Bacon, Biliteral Cypher, 1605.

indeterminateness and vacuity of conception, it is indifferent whether this abstraction is called space, pure intuiting, or pure thinking''.[11] Difference (between 0 and 1) is also the foundation of Geoge Boole's logic which infuses clear meaning into each term: ''the symbol 0 represents Nothing,'' whereas the symbol 1 represents '''the Universe' since this is the only class in which are found *all* the individuals that exist in *any* class.[12] Boole's approach had already been prognosticated by Leibniz famous motto ''*Omnibus ex nihilo ducendis sufficit unum*'' (To draw all things out of nothing, one thing is sufficient) which posited that thought could be encoded into two integers only.[13] In 1936, Turing's thought experiment on computation also availed itself of only two numbers: 0 and 1. Finally, Ferdinand Saussure's observation that language is only constituted by ''negative facts''[14] also places differences between signs as the pre-requisite for the emergence of statements. Saussure's reference to the negative qualities of language remind us of the possibility to use discrete computation speculatively because of its ability to articulate negative conditions.

Beside the innovations that emerged as a result of the technologies utilised, Bacon's approach to cryptography also presents conceptual novelties that are useful to discuss in regards to ML models and speculative and creative thinking. Beyond the more practical applications to conceal secrets, Bacon thinks of cryptography as a method to penetrate the secrets of nature. Nature, in his view, does not speak in the same language as God or man: God plays an ontological role to 'write' nature (which thus appears concealed), whereas humans need to develop instruments to 'read' it to gain knowledge (epistemology). The gap between how things are created and how they can be accessed demands the invention of a vicarious language governing communication: in other words, a cypher. Such language can only be non-mimetic: ''no longer metronomically intertwine as part of the divine microcosm, the language of God, man, and thing begin to pull apart''.[15] The core preoccupation of Bacon's ''new science'' is therefore to devise such scientific instruments to gain access to the encrypted ('noisy' in computational parlance) secrets of nature. Cryptography is here understood as an instrument for discovery; its mathematical and combinatorial qualities can penetrate beyond the phenomenal appearance of objects, into their

11. Georg Wilhelm Friedrich Hegel. *Science of Logic [1812–31].* Translated by A. V. Miller. (London, NJ: Allen & Unwin, 1990). Quoted in David Link. *Archaeology of Algorithmic Artefacts.* (Minneapolis: Univocal Publishing, 2016), p. 16.
12. George Boole. *An Investigation of the Laws of Thought, on which are founded the mathematical theories of logic and probabilities.* (London: Walton & Maberly, 1854).
13. Letter to the Duke Rudolph August of Brunswick, December 1696. Quoted in Lloyd Strickland and Harry R. Lewis. Leibniz on Binary: *The Invention of the Computer Arithmetic* (Cambridge, Mass.: The MIT Press, 2022), p. 99.
14. In Saussure's own words: ''...in language there are only differences *without positive terms*''. Ferdinand Saussure. *Course of General Linguistics.* Edited by C. Bally and A. Sechehaye, with the collaboration of A. Riedlinger; translated and annotated by W. Basking. 1st ed. 1916. (London: Duckworth, 1959), p. 120. Quoted in Paolo Virno, *Saggio sulla Negazione: Per una Antropologia Linguistica* (Turin: Bollati Boringhieri,2013) , p. 28.
15. Michael C. Clody. '' Deciphering the Language of Nature: Cryptography, Secrecy, and Alterity in Francis Bacon''. *Configurations*, Volume 19, Number 1 (Winter 2011), p. 127.

''as-yet unheard potential''.[16] It is the speculative quality of the Bacon's cypher that perhaps best resonates with earlier considerations on the introduction of ML models in design processes. The automatic analysis and categorisation of data produced by ML algorithms is accurate and yet impenetrable to human faculties; mathematics plays both a rigorous and experimental role in discovering new untapped connections and providing an '''outside' representation from within''.[17]

Conclusions

By surveying some key examples from the vast archaeology of cryptography, we aimed at foregrounding how computational thinking re-organises our existing knowledge, and provides a rigorous instrument for discovery. In so doing, the aim was to furnish theoretical instruments that may guide design with ML models and move the discussion beyond either functional applications or general ethical concerns. To study the relation between machine learning and design through the lenses of cryptography offers rich ideas to develop. First, it establishes the centrality of non-mimetic notational systems and mathematics as the key technologies for manipulation and creative exploitation of signs. These operations possess speculative and playful qualities that are not only shared with some cryptographic methods, but also remind us that design is a creative (poietic) endeavour rather than an analytical one. The abstraction of phenomena through data and their algorithmic manipulation involves both ontological and epistemological processes. Algorithms write the world in mathematical and statistical language as well as allow us to read it. ML models conflate the two operations in ways that broadly match those that cryptographers have been developing and perfecting for centuries: that is, to develop instruments to venture into as yet-known domains.

Bibliography:

Bacon, Francis. *The Advancement of Learning* [1605]. Oxford : at the Clarendon Press, 1869.

Boole, George. *An Investigation of the Laws of Thought, on which are founded the mathematical theories of logic and probabilities*. London: Walton & Maberly, 1854.

Cache, Bernard. ''Towards a Non-Standard Mode of Production''. In *Projectiles*. London: Architectural Association, 2011.

Clody, Michael C.. '' Deciphering the Language of Nature: Cryptography, Secrecy, and Alterity in Francis Bacon''. *Configurations*, Volume 19, Number 1 (Winter 2011), p. 117-141.

Evens, A. *Logic of the Digital*.London: Bloomsbury Academic, 2015.

Hegel, Georg Wilhelm Friedrich. *Science of Logic [1812-31]*. Translated by A. V. Miller. London, NJ: Allen & Unwin, 1990.

Link, David. *Archaeology of Algorithmic Artefacts*. Minneapolis: Univocal Publishing, 2016.

Saussure, Ferdinand. *Course of General Linguistics*. Edited by C. Bally and A. Sechehaye, with the collaboration of A.

Riedlinger; translated and annotated by W. Basking. 1st ed. 1916. London: Duckworth, 1959.

Strickland Lloyd, Harry R. Lewis. *Leibniz on Binary: The Invention of the Computer Arithmetic*. Cambridge, Mass.: The MIT Press, 2022.

Virno, Paolo. *Saggio sulla Negazione: Per una Antropologia Linguistica*. Turin: Bollati Boringhieri, 2013.

Wilden, Anthony. *System and Structure*. London: Tavistock Publications, 1972.

16. *Ibid*., p. 128.
17. *Ibid*., p. 128.