Universitat Autònoma de Barcelona

Computational intractability, artificial intelligence, and the Cold War

Javier Ramiro Poveda Figueroa

ADVERTIMENT. L'accés als continguts d'aquesta tesi queda condicionat a l'acceptació de les condicions d'ús establertes per la següent llicència Creative Commons: http://cat.creativecommons.org/?page_id=184

ADVERTENCIA. El acceso a los contenidos de esta tesis queda condicionado a la aceptación de las condiciones de uso establecidas por la siguiente licencia Creative Commons: http://es.creativecommons.org/blog/licencias/

WARNING. The access to the contents of this doctoral thesis it is limited to the acceptance of the use conditions set by the following Creative Commons license: https://creativecommons.org/licenses/?lang=en

Universitat Autònoma de Barcelona

Computational intractability, artificial intelligence, and the Cold War

Author: Javier Ramiro Poveda Figueroa
Supervisor: Dr. Thomas Sturm

Programa de Doctorado de Historia de la Ciencia
Institut d'Historia de la Ciencia (IHC-UAB)

September 2022

Universitat Autònoma
de Barcelona

Index

Part I: A prehistory of computational intractability

Part II: Simon, Newell and the emergence of artificial intelligence

Part III: The debate over computational intractability

This thesis is dedicated to my parents Marcial and Nancy.

Also, my doctoral thesis is dedicated to those who have taught me the value of hard work: My calculus professor at Universidad Politecnica Salesiana Julio Cesar Loja Quezada, and the friend of my family Jose Ugarte Noboa.

I want to also dedicate this thesis to the memory of my loved ones. My grandfathers Luis Poveda Orellana, Enrique Figueroa Polo, and my greatgrandparents Luis Alberto Figueroa Espejo and Amelia Virginia Polo Palacio.

Acknowledgments

First, I express my most sincere gratitude to my Ph.D. thesis advisor, professor Thomas Sturm. He helped me to overcome obstacles that appeared in the development of this doctoral thesis. I consider him one of the most influential persons that contributed to my intellectual development during this period of my life.

The help of other researchers and academics was vital for developing this thesis, too. I want to thank many people that contributed in several ways. First, I am very grateful to the historians of science, Stephanie Dick (Simon Fraser University) and Michael Barany (Edinburgh), for helping me think about computation and implementation problems. I also want to thank Tasha Schoenstein, a Ph.D. candidate at Harvard University, for introducing me to the essential historical works of Michael Mahoney. Furthermore, I am thankful to the artificial intelligence and computer science communities, specially to Adolfo Guzman Arenas (Instituto Politécnico Nacional de México), Michael Sipser (MIT), Pablo Noriega and Jordi Levy (IIIA-CSIC), Ulises Cortez and Jordi Fornes de Juan (Universidad Politécnica de Cataluña). I also want to express my sincere gratitude to the emeritus professor Ludmilla Jordanova (Durham), the entrepreneur Daniel Crevier, and the former director of the Max Planck Institute for Human Development, Gerd Gigerenzer. Lastly, I want to add the former master student in history of science at Universitat Autònoma de Barcelona, Xavier Mauri, for sending me his master's thesis, *Fuzzy logic i el seu rol en la superació de la crisi de la Intelligencia Artificial* (2015-2016).

The history of science community in Barcelona was significant for the development of this thesis. I start by thanking professors José Pardo-Tomás, Oliver Hochadel, Jon Arrizabalaga, and Alvaro Girón Sierra (IMF-CSIC), one of the places in my mind heart. To the professors of the master in history of science at the IHC-UAB, Annette Mülberger, Agusti Nieto Galán, Xavier Roqué, Mónica Balltondre, Jaume Sastre, Silvia De Biachi, and Jorge Molero Mesa. Finally, I want to express my gratitude to the professor José Romo Feitó (Universitat de Barcelona), one of the persons who strongly influenced my view of history of science.

A significant mention goes to Stephen Cook, the developer of the P/NP distinction and now emeritus professor at the University of Toronto. He kindly answered my emails asking about tractable and intractable problems. In a way, this thesis honors his work.

Many people help you when times are difficult. First and foremost, I want to thank Jorge Alfredo Uquillas from the Eindhoven Institute of Technology, Dennys Paillacho Chiluisa from Escuela Politécnica del Litoral, Antonio Malo Larrea from Universidad Católica de Cuenca, Melanie Valencia Velasco who is working to improve the world for the better, Vanessa Seifert from the National and Kapodistrian University of Athens, Joan Cabestany Moncusi and Albert Samà Monsonis from Universitat Politècnica de Catalunya. I respect and admire these people, independently of whether we share the same beliefs or not. Furthermore, I want to extend my acknowledgments to the Catalan Association for Artificial Intelligence for trusting me. Special thanks to Antonio Moreno and Aïda Valls from Universitat Rovira I Virgilí, and to Josep Pujol Gruart and Marco Schorlemmer and from IIIA-CSIC.

Finally, I express my sincere gratitude to my parents and grandparents. My father Marcial, my mother Nancy, my grandfather Enrique, and my grandmother Enriqueta. Without their love, support, and patience, I would never be who I am.

Abbreviations and acronyms

ACM: Association for Computing Machinery.

AI: Artificial intelligence.

ARPA: Advanced Research Projects Agency of the United States of America.

Caltech: California Institute of Technology.

CIA: Central Intelligence Agency of the United States of America.

ECA: Economic Cooperation Act.

ENIAC: Electronic Numerical Integrator and Computer.

FBI: Federal Bureau of Investigation.

GE: General Electric.

GPS: General Problem Solver.

GSIA: Graduate School of Industrial Administration at the Carnegie Institute of Technology.

IAS: Institute for Advanced Study.

IBM: International Business Machines Corporation.

ICMA: International City Manager’s Association.

IIT: Illinois Institute of Technology.

LT: Logic Theorist.

MEA: Mean-Ends-Analysis.

MIT: Massachusetts Institute of Technology.

NSF: National Science Foundation.

ONR: Office of Naval Research.

OSRD: Office of Scientific Research and Development.

RAND: Research And Development Corporation.

USA: United States of America.

Abstract

This doctoral thesis's primary goal is to analyze the discovery of computational intractability from an integrated historical, scientific, and sociological point of view. Computational intractability is a concept developed in mathematical logic and computer science. It consist in the claim that a computer algorithm cannot solve a particular problem in a reasonable polynomial amount of time- if it can, experts call this (P), if not, then (NP)-hard. Stephen Cook's classic paper "The complexity of theorem proving procedures" (1971) formalized the concept of computational intractability, inspired by Hao Wang's "Dominoes and the AEA case of the decision problem" (1963). Wang discovered the problem of computational intractability in mathematical logic using a game of dominoes. The results obtained by Program P were the basis for the discovery of the problem of computational intractability. His discovery was similar to the concept of computability published in Alan Turing's famous "On computable numbers with an application to the *Entscheidungsproblem*" (1936). Wang's discovery came about when he focused on studying the decidability problem after the early artificial intelligence community ignored his work on automatic theorem proving.

In the summer of 1956, artificial intelligence became a recognized field of study. During the Summer Research Project on Artificial Intelligence held at Dartmouth College, several researchers discussed the possibility of developing machine intelligence. Unfortunately, most attendees had no proposal for showing the chance of achieving that purpose. Herbert Simon and Allen Newell presented the results of proving theorems of mathematical logic with their "Logic theorist" program (LT), saving the participants' enthusiasm. The LT could prove thirty-eight theorems of mathematical logic found in Bertrand Russell and Alfred North Whitehead's *Principia Mathematica* (1910-1913). After the workshop, the LT became the baseline for developing artificial intelligence algorithms. However, the enthusiasm early artificial intelligence programs produced crumbled during the second half of the 1960s because artificial intelligence algorithms could not solve

real-world problems such as translating Russian texts in English. The early artificial intelligence community ignored critics like Wang, who pointed to the limits of the LT. Why did his criticism not become widely accepted at first?

The reason could be the constraints imposed on research during the Cold War. From 1950 to 1972, the USA and the People's Republic of China were in tension because of their highly different, and competing political ideologies. While the USA was capitalist, China was communist. Moreover, China was an ally of the Soviet Union. The USA and the Soviet Union were in tension because both countries were fighting to be the international hegemone. For that reason, both sides also had scientists and professionals to work in activities linked to national security and military power. Artificial intelligence was part of the sciences financed by U.S. institutions like the RAND Corporation and other think tanks to reach those goals. But mistrust concerning scientists with different national backgrounds than one's own could easily exist. That is why the early artificial intelligence community ignored Wang's work, even if his Program P proved all the theorems of the *Principia*. Evidence suggests that the early artificial intelligence community dismissed Wang because he was suspected to be an agent of the Chinese government.

This thesis aims to analyze the development of computational intractability by considering scientific, historical, and sociological aspects of the fields of computational complexity and artificial intelligence through the use of games for modeling human decision-making and problem-solving such as chess and domino. Games were the basis for developing the theory of human decision-making because they provided the elements for conceptualizing several essential concepts linked to the sciences, such as "bounded rationality". Exploring the conceptual and theoretical aspects of games during the Cold War, related to the ideas and practices of the communities of artificial intelligence and computational complexity can illuminate the connections between both disciplines. Games ground the aspects of decision-making and metamathematical algorithmic tractability. Connecting history, science, and sociology to understand the discovery of the problem of computational intractability can shed important light on the developments of the sciences during the Cold War.

Introduction

This thesis aims to analyze the discovery of the problem of computational intractability from an integrated historical, sociological, and conceptual or theoretical point of view. I claim that studying the problem of computational intractability in the context of the Cold War can explain us why computational intractability was discovered in the field of computational complexity instead of artificial intelligence (AI). While computational complexity focuses on studying the complexity of algorithms, AI centers its attention on developing machine intelligence.

But, what is computational intractability? Computational intractability is a problem of mathematical logic and computer science with many applications and consequences for other disciplines, including the cognitive and social sciences. This concept refers to the question of whether a computer algorithm could solve a problem in a reasonable amount of time (P) or not (NP). For example, consider the so-called traveling salesman problem (TSP), described formally first by mathematician Karl Menger in 1930 in “Das botenproblem”. The person wants visit a high number of cities, say 40, without visiting any city more than once. Depending on the number of cities the traveller has to visit, the algorithm will quickly become incapable of finding an optimal solution: the complexity of the problem increases exponentially with the number of cities, due to the number of possible combinations (Menger, 1930). Until today, no algorithm exists for solveing the TSP optimally in a reasonable amount of time. In that spirit, studying the issue of computational intractability is key to solving scientific and engineering problems by non-optimal methods.

Computational intractability is essential for scientists and engineering professionals trying to solve complex problems. Studying computational intractability from a historical point of view can be helpful for those professionals to understand why different problems cannot be solved using algorithmic techniques developed to this day. Also. I clam that studying computational intractability can help those professionals to find novel ways to solve the problem, such as the proposal shown in section 5 of chapter 9.

Why is it insightful to study the relationship between AI and computational intractability historically? Games used for human decision-making and problem-solving in the context of the Cold War linked AI and computational intractability. Several institutions from public and private sectors were involved in developing “Cold War rationality”, like the Research and Development Corporation (RAND) or IBM (Erickson et al., 2013). Even though computational intractability and AI are intertwined, the discovery of the former happened in outside AI. The enthusiasm AI researchers had from 1956 until the late 1960s blind them to the existence of such problem in the field. Also, the constraints within the Cold War prevented the discovery of the problem of computational intractability within AI. The historical study of artificial intelligence usually treats the discovery of computational intractability as a separate subject. Historians does not know how the discovery of computational intractability in the field of computational complexity instead of AI might have been due to the tense international relations between the USA and China from 1950 to 1972. A main contribution of my thesis is to overcome this blindspot.

The computer vision work at the Artificial Intelligence and Assistive Technologies Research Group at *Universidad Politecnica Salesiana del Ecuador* inspired this doctoral thesis. My bachelor's thesis *Esquemas de votación Borda aplicados a la clasificación de imágenes utilizando los histogramas de color RGB, HSV, y el descriptor de distribución de color MPEG-7* (2013) aimed to classify the images of different databases using Borda voting schemes. One of the main challenges in computer vision is classifying images in a reasonable amount of time. Understanding the importance of reducing the complexity of the images is essential so that the computer can process them efficiently and successfully in P time (Poveda, 2013, p. 51). Technical books and articles such as Poveda (2013) and Poveda & Robles (2012) touched on this issue. Unfortunately, professionals involved in computer vision and other relevant areas of computer science, such as algorithmic theory, ignore the historical aspects of discovery of the problem of computational intractability. Even more, they do not know how social, political, and other factors influenced the discovery of computational intractability outside the field of artificial intelligence.

Three main sections divide this document. The first focuses on describing the prehistory of computational intractability and AI. The second shows the development of AI from the development of the "Logic theorist" (LT) to the institutionalization of AI at Dartmouth Research Project on Artificial Intelligence. Finally, the last section of this thesis explain the reasons for the crisis of AI, and the discovery of computational intractability in the field of computational complexity. Furthermore, the last section proposes a way to overcome the problem of computational intractability by merging AI with computational complexity. A closer chapter outline follows below.

Literature review

Since the development of the P/NP distinction, the problem of computational intractability has been studied extensively in technical books. Michael R. Garey and David S. Johnson's *Computers and intractability: A guide to the theory of NP-completeness* (1979) preface briefly explains the problem of computational intractability:

> "Few technical terms have gained such rapid notoriety as the appellation "NP-complete". In the short time since its introduction in the early 1970s, this term has come to symbolize the abyss of inherent intractability that algorithm designers increasingly face as they seek to solve larger and more complex problems. A wide variety of commonly encountered problems from mathematics, computer science, and operations research are now known as to be NP-complete, and the collection of such problems continues to grow almost daily. Indeed, the NP-complete problems are now so pervasive that it is important for anyone concerned with the computational aspects of these fields to be familiar with the meaning and implications of this concept" (Garey & Johnson, 1979, p. ix).

After this explanation, Garey and Johnson sketch the development of the field of computational complexity. They say that the origins of the intractability problem lie in Turing's "On computable numbers with an application to the *Entscheidungsproblem"* (1936). Their description concludes in 1971, with the development of the P/NP distinction introduced in Stephen Cook's "The complexity of theorem-proving procedures" (Garey & Johnson, 1979, pp. 11-13). While I broadly agree with these points, one of the main problems with Garey and Johnson's narrative is the limited perspective they presented from a historical standpoint. Their narrative aimed to introduce computational complexity to computer science students and practitioners interested in the field instead of developing a proper historical account of computational complexity's genesis and development. Furthermore, Garey and Johnson do not explore the relationship between artificial intelligence and computational complexity.

In contrast to Garey and Johnson, Stuart Russell and Peter Norvig's *Artificial intelligence: A modern approach* (1995) does deal with the connection between computational intractability and artificial intelligence.[1] Russell and Norvig's explanation of the development of artificial intelligence starts with the institutionalization of the field in 1956 at Dartmouth College. They emphasize the early enthusiasm artificial intelligence brought to researchers interested in developing machine intelligence. Also, Russell and Norvig point out that the early artificial intelligence community could not solve some problems. Furthermore, Russell and Norvig explain the problem of intractability briefly:

> "The second kind of difficulty was the intractability of many problems that AI was attempting to solve. Most of the early AI programs solved problems by trying out different combinations of steps until the solution was found. This strategy worked initially because microworlds contained very few objects and hence very few possible actions and very short solution sequences. Before the theory of computational complexity was developed, it was widely thought that "scaling up" to larger problems was simply a matter of faster hardware and larger memories. The optimism that accompanied the development of resolution theorem proving, for example, was soon dampened when researchers failed to prove theorems involving more than a few dozen facts. *The fact that a program can find a solution in*

---

[1] I used the third edition of Russell and Norvig's book.

> *principle does not mean that the program contains any of the mechanisms needed to find it in practice*" (Russell & Norvig, 2010, p. 21).

Unfortunately, Russell and Norvig's explanation of computational intractability is quite limited because their description of the development of artificial intelligence and computational complexity does not include the historical entanglements of both fields. They do not consider the link between computational complexity and AI through games part of "Cold War rationality". Similar to Garey and Johnson, they focus on explaining the connection between artificial intelligence and computational complexity to students and professionals close to the field of artificial intelligence, not to those practitioners in the humanities and social sciences.

Stephen Cook's contribution "An overview of computational complexity" (1983) focuses on presenting the development of computational complexity using scientific articles to explain how different concepts form the field of computational complexity since the publication of Turing's "On computable numbers with an application to the *Entscheidungsproblem*" in 1936. His explanation starts by citing works of mathematicians Michael Oser Rabin, Richard E. Stearns, Alan Cobham, and physicist Juris Hartmanis (Cook, 1983, pp. 401-402).[2] Cook then continues his narrative by explaining each attempt made by different authors to address the problem of computational intractability. Among the authors Cook mentions is mathematician Claude Shannon, one of the early fathers of the field of cybernetics (Cook, 1983, p. 402). Implicitly, Cook is thereby pointing to the connection between the problem of computational intractability and cybernetics. Cook explains in one paragraph the relationship between cybernetic information theory and computational intractability:

> "Another important complexity measure that goes back in some form at least to Shannon [74](1949) is Boolean circuit (or combinational) complexity. Here it is convenient to assume that the function *f* in question takes finite bit strings into finite bit strings, and the complexity C(n) of *f* is the size of the smallest Boolean circuit that

---

[2] See chapter 9 for a closer explanation of the development of the field of computational complexity.

> computes *f* for all inputs of length *n.* This very natural measure is closely related to computation time" (Cook, 1983, p. 402).

Cybernetics had a powerful influence on the development of AI. Because of this, there existed also an implicit relation between AI and computational intractability. Unfortunately, Cook does not explain the connection between AI and computational complexity because each scientific community has treated these fields as separate.[3]

All of the aforementioned authors- Garey and Johnson, Russell and Norvig, and Cook- have not developed an appropriate historical narrative. They wrote for engineers and scientists, not for students and professionals in the humanities and social sciences. Therefore, they focused on explaining computational complexity's technical and conceptual aspects. Writing the history of computational intractability has been difficult for historians of science and technology because just a few people outside the engineering and scientific fields have paid attention to the topic.

The historian of computing Michael Mahoney was the first to explain the field of computational complexity from a historical standpoint. Mahoney contextualized the development of theoretical computer science through the lens of mathematics. His work was intellectually oriented instead of focusing on the institutional relations that might have influenced the development of theoretical computer science (Mahoney, 2011, p. 11). His first attempt at developing an intellectual history of computer science stems from the 1990s. During that decade, he aimed to write a book titled *The structures of computation: Mathematics and theoretical computer science, 1950-1970.* Unfortunately, Mahoney did not carry out this project (Mahoney, 2011, p. 11). He just produced several papers that partially reconstructed the history of theoretical computer science.

Some of those articles also treated the area of computational complexity and its relation to theoretical computer science. Thus, Mahoney's paper "Computer science: the search for a mathematical theory" (1997) explains the importance of

---

[3] I wrote to Cook asking whether Simon's concept of bounded rationality inspired him to develop his P/NP distinction. He responded that he did not use Simon's concept but Wang's mathematical logic and computer-based theorem provers (S. Cook, personal communication, June 22, 2018).

mathematical logic for the development of computer science, if very briefly. One of the topics he touches on is the problem of computing Turing introduced. First, Mahoney describes what a Turing machine is. After that, he explains the use of mathematics by Shannon and Turing (Mahoney, 2011, pp. 131-133). Mahoney wants to illuminate the relationship between cybernetics and the notion of computing. He also shows that Simon and Newell disagreed with the use of mathematical logic for grounding computing as a mathematical science.[4] As Mahoney points out, the position of most computer practitioners in computer science was different from Simon and Newell's:

> "Computer people knew from experience that "finite" does not mean "feasible" and hence that the study of algorithms required its own body of principles and techniques, leading in the mid-1960s to the new field of computational complexity. Talk of costs, traditionally associated with engineering rather than science, involved more than money. The currency was time and space, as practitioners strove to identify and contain the exponential demand on both as even seemingly simple algorithms were applied to ever-larger bodies of data. Yet, central as algorithms were to computer science, the report continued, they did not exhaust the field, 'since there are important organizational, policy, and nondeterministic aspects of computing that do not fit the algorithmic mold'" (Mahoney, 2011, p. 129).

Implicitly, Simon and Newell rejected the notion of computability introduced by Turing because Turing machines relied on mathematical logic for proving which problems could be computable or not. Unfortunately, Mahoney's 1997 paper did not develop a history of computing that adequately explained the relationship between computational complexity with mathematical logic, especially the relationship between Cook and Turing. Mahoney gave a glimpse of the history of computational complexity in a chapter of the book *The space of mathematics* (1992) under the title "Computers and mathematics: The search for a discipline of computer science".

---

[4] In 1967, *Science* published a letter by Simon, Newell, and mathematician Alan Perils. They defended the study of computers as an empirical science instead of engineering. They said that other sciences, like mathematics and physics, could help study computers but insisted that the study of computers was an empirical science (Newell et al., 1967, p. 1374).

Mahoney’s 1992 chapter considers mathematical logic to be the basis for the development of computing. He points out that Turing machines are the core of theoretical computer science, including computational complexity (Mahoney, 2011, pp. 148-149). Thus, he explains:

> “At the other end of the spectrum, during the mid-60's Turing machines of various types became the generally accepted model for measuring the complexity of computations, a question that shifted attention from decidability to tractability and enabled a classification of problems in terms of the computing resources required for their solution. First broached by Michael O. Rabin in 1959 and '60, the subject emerged as a distinct field with the work of Juris Hartmanis and Richard E. Stearns in 1965 and acquired its full form with the work of Stephen Cook and Richard Karp in the early '70s. The field has formed common ground for computer science and operations research, especially in the design and analysis of algorithms” (Mahoney, 2011, p. 149).

Even though Mahoney's history of computing is essential to understanding computers from a historical perspective, he has left out power plays from the picture of when and how scientists and engineers developed their activities. Moreover, he does not consider external social and political conditions for developing the sciences. Finally and more specifically, Mahoney does not consider national and international relations for the developments of computational complexity and AI.

The development of the sciences often depends on how political, economic, and social forces play at the national and transnational level for exchanging knowledge and practices between different types of scientific communities in a certain period. Also, those forces can shape the nature of a scientific field as being opened or closed to certain groups depending on particular circumstances. Many sciences developed during the Cold War, such as artificial intelligence, were closed to certain members of certain social groups or nations. Thereby, these disciplines ignored several significant contributions of those considered to be dangerous to national security interests. “Entangled history” can help to understand how international tensions between different countries limited the development of the sciences during the Cold War (Randeria, 2006).

Constructivism, national history, and the history of computing

Contemporary historians of science very often write their histories using constructivist approaches. Also, they develop narratives that focus on the relationship between the sciences and the nation-state. In this section, I claim that those approaches are limited in the explanation of the discovery of the problem of computational intractability.

Constructivist historiography cannot describe my object of study because I have to focus primarily on concepts not on scientific practices. Instead, I use "entangled history" to explain why the discovery of computational intractability happened outside artificial intelligence (Randeria, 2006). I claim that entangling the histories of China and the USA during most of the twentieth century can explain that discovery. In that spirit, I shall discuss some relevant books to explain how constructivism influenced several histories of science and technology. Later, I will use entangled historiography to show why computational intractability took a different path from artificial intelligence. Moreover, I will justify using entangled historiography instead of the constructivist approach to writing my doctoral thesis.

Jan Golinski's *Making natural knowledge: constructivism and the history of science* (1998) is a compendium of constructivist historiographies. He explains that science is a practice instead of an intellectual pursuit, focusing on the particularities of the culture of a particular place. In line with this, constructivists concentrate on external influences upon science to explain how a scientific discipline comes into existence (Golinski, 1998, p. ix; Golinski, 1998, p. 55).

Simon Schaffer and Steven Shapin's *Leviathan and the air pump: Hobbes, Boyle, and the experimental life* (1985) proposes another exciting case study of constructivism. Schaffer and Shapin's narrative is set up in seventeenth-century England, focusing on the experimental culture during the restoration of King Charles II. In Schaffer and Schapin's narrative, Robert Boyle and Thomas Hobbes debated the existence of vacuum in the air at the Royal Society of London. Boyle built a machine to prove that the vacuum existed. Boyle presented his air pump to the Royal Society of London and, after the presentation of the experiment, the members of the Society would recognize whether the knowledge produced by

Boyle's machine was valid (Schaffer & Shapin, 1985, pp. 55-60; Schaffer & Shapin, 1985, pp. 76-79). Hobbes disagreed with Boyle's experimental method. To him, philosophy should focus its attention on the use of language. In that sense, Hobbes discredited experimentation as a source of getting valid knowledge. Therefore, Boyle's air pump did not prove the existence of vacuum (Schaffer & Shapin, 1985, pp. 92-99; Schaffer & Shapin, 1985, pp. 139-143). Schaffer and Shapin conclude in their book that Boyle's argument in favor of the concept of vacuum won over Hobbes’s because Boyle's proposal was in tune with the culture of the restoration in England. Socialization in the Royal Society of London was a reflection of the culture of the restoration because deliberation about an experiment was key for constructing knowledge, according to Schaffer and Shapin. Knowledge was valid if all the Royal Society members agreed on the experimental results. Hobbes attacked that culture. Therefore, the members of the Royal Society dismissed Hobbes's proposal. Constructivism helps us to see relationships between different actors in a specific scientific culture with various entities, producing a particular type of knowledge.

An approach different than constructivism can be found in the histories of science that focus on the relationship between science with the development of the national-state. Even though state building histories can integrate elements of constructivism, some histories of science focus on elements that the approach cannot reach, such as theoretical computer science and history of mathematical logic.

The historian of science Jon Agar wrote two exciting books about the development of computers during the nineteen and twentieth centuries, considering cultural and institutional elements of both centuries in the United Kingdom. Agar’s *Turing and the universal machine* (2001) is an interesting approach to the division of labor problem and its relationship with the development of the idea of building a machine that could mechanize the calculation of tables needed by industry and government. Those calculations would help optimize resource use (Agar, 2001, pp. 1-20). As Agar shows, Charles Babbage, Konrad Zuse, and Howard Aiken attempted to build the desired machine (Agar, 2001, pp. 21-62). Agar presents the Turing machine as a new type of abstract machine that undermines Hilbert's decidability argument because some of those machines could end up in an infinite loop when trying to solve certain kinds of problems, proving that Gödel was right (Agar, 2001, pp. 85-98).[5] Furthermore, the Turing machine was a model for building a machine during the Second World War to decode encrypted messages generated by the Nazi Enigma machine (Agar, 2001, pp. 101-112). Agar ends his book with a brief explanation of the development of the computer during the Cold War.

He expands his argument in *The government machine: A revolutionary history of the computer* (2003). Here, Agar explains how the nineteenth century established the British state bureaucracy. He focuses on the attempts of the British government to mechanize public service. In this context, Agar mentions the intractability of several problems the bureaucracy could not solve, such as using statistics to control different social groups living in the British Empire (Agar, 2003, p. 87). Agar mentions that a Turing machine could be the perfect analogy to the civil servant for mechanizing bureaucracy's activities, solving the problem of calculations even more efficiently than humans (Agar, 2003, pp. 69- 74).

Agar's narrative of his two books focuses on the context of the United Kingdom. He treats the development of artificial intelligence in the USA only superficially. Paul Edawrds' *The closed world: computers and the politics of discourse in Cold War America* explain in greater detail the development of computing and AI in the context of the Cold War in the USA.

The consideration of local context for describing the development of scientific theories and technologies is a positive aspect of constructivists and those who write nation building histories of science. This thesis criticize the constructivist approach for explaining the discovery of computational intractability in computational complexity because computational intractability lies in the realm of metamathematics, outside scientific practice. Furthermore, computational intractability is found in two separate cultures, not in one. Constructivist historiography does not consider exterior elements to a scientific network. Therefore, constructivism cannot explain why computational intractability was discovered in studies of computational complexity instead of AI research, even if

---

[5] See chapter 1 for a closer explanation.

both fields had the common component of the games appropriate to developing a rationality during the Cold War. Nation building histories of science also cannot explain the discovery of the problem computational intractability because they does not consider the influence of international relations for the development of the sciences in one culture or another.

My approach to explaining the discovery of computational intractability is framed in the “entangled history” perspective (Randeria, 2006). “Entangled history” has treated the topic of international relations to explain historical processes and knowledge creation, considering the political and material conditions shared by more than one culture. Ekaterina Babintseva’s “Overtake and surpass: Soviet algorithmic thinking as a reinvention of Western theories during the Cold War” (2021) is an excellent example of entangling the history of the USA with the Soviet Union for explaining the development of a Soviet version of cybernetics by the time both countries relaxed their political tensions during the 1950s and 1960s.

In the case of artificial intelligence and computational complexity, the use of games close to Cold War rationality connected two separate cultures apparently different from one another.

The international relations between China and the USA are crucial to understanding the discovery of computational intractability during the 1970s. The philosopher Jin Yuelin studied at Columbia University in the 1920s during China's modernization project after the end of the Qing Dynasty in 1912. Yuelin became Wang’s professor during the time the latter was studying at Southwest Associated University, and also his master thesis advisor at Tsinghua University. One year after Wang finished his master’s studies in 1945, he went to study in the USA, supported by a scholarship from the U.S. Department of State.

The USA and the Republic of China had a good relationship before Mao Zedong came to power in 1949. The situation changed one year later after the involvement of China in the Korean War. Furthermore, China was an ally of the Soviet Union since Mao came to power. The U.S. government’s mistrust of Chinese citizens living in the U.S. was based on the belief that Chinese citizens were often collaborating with the Beijing regime. Early artificial intelligence community ignored Wang’s work on automatic theorem provers, even if his work got better results than the LT because he was considered a menace to U.S national security. Faced with this neglect, Wang decided to turn to the problem of decidability in mathematical logic, discovering the problem of computational intractability. His discoveries then later influenced Cook's "The complexity of theorem proving procedures" (1971). In that vein, my thesis will focus on explaining the discovery of the problem of computational intractability outside the field of AI as strongly influenced by the tense international relations between the People's Republic of China and the USA from 1950 to 1972.

This thesis is framed within the history and philosophy of science (HPS). Much of the work nowadays carried out in HPS devotes its attention to the development of science from a predominantly intellectual point of view. Even though my thesis follows certain elements of that tradition, in particular its tendency to study scientific developments in themselves, I also make use of political and social history to explain how scientific concepts came into existence at particular places and times. I combine HPS with the approach of "entangled history" to describe and explain the discovery of metamathematical concepts during the context of war.

Scope and sources

The time frame chosen for my work begins in the second half of the nineteenth century and ends in 1971. That period encompasses the developments of mathematical logic and its influence up on Cook's conceptualization of the P/NP distinction. Also, this thesis explores the relations between the USA and the People's Republic of China from 1950 to 1971. In this latter period, the relation between these two countries influenced the discovery of the problem of computational intractability outside of AI. Finally, the period chosen helps to explain the failure of the high ambitions of AI, starting in the second half of the 1960s.

The period under study coincides with the political tensions between the USA and the People's Republic of China from 1950 to 1972. The USA and Western countries were defenders of capitalism, while China and the Soviet Union were defenders of communism. Not rarely, secrecy was a critical factor in the development of the sciences during that period in both political blocks. In that spirit, a limited number of people developed the Cold War sciences in certain institutions. Hence, researchers of a particular field just cited those in their circles, leaving out those who did not belong to certain institutions (De Solla, 1986, p. 74). Based on the China-USA relations during the Cold War, I decided to focus on the sources published in academic journals to trace the connections between institutions and researchers that developed the fields of artificial intelligence and computational complexity. Although I wanted to include government and diplomatic documents in my research, I could not do so because some documents located in archives in the USA are hard or impossible to access. Also, the Covid19 situation limited international mobility.

I found most of my sources in four digital archives and in one book: the Herbert Simon collection, the Allen Newell collection, the ACM digital library, IEEE Xplore, and Wang (1990). My starting point for unearthing information related to the discovery of the idea of computational intractability was Cook's paper "An overview of computational complexity" (1983). Cook's work explained the development of computational complexity by using scientific articles from the beginning of computational complexity until the state of the field in 1983. Furthermore, I contacted Cook to ask him how he developed the P/NP distinction. Cook guided me, explaining Wang was his primary source of inspiration. In the case of databases, they had important works of seminal scientists and philosophers such as Simon, Allen Newell, Wang, and Cook. The information found in those databases was key for framing the scientific discoveries during the Cold War, such as the concept of bounded rationality.

The reconstruction of chapters six and seven required the research developed in different fields of artificial intelligence during the 1960s and 1970s, such as computer vision. For that reason, I visited the web repository of MIT to find several works and articles related to computer vision, natural language processing, and other areas in the field of artificial intelligence developed in the 1960s and

1970s. Those works helped me understand the early enthusiasm for artificial intelligence in the second half of the 1950s until the end of the 1960s.

Chapter outline

Three blocks make up this thesis. Each block connects with the following by a linear narrative, but the reader can focus on a specific block to focus the attention on a particular theme. This thesis also follows a linear narrative because no significant break in AI development appeared from 1956 until the second half of the 1960s.

The first block explains the foundations of computational complexity and AI, and consists of three chapters. Chapter 1 deals with the development of mathematical logic from the nineteenth century until 1936. The chapter starts introducing some influential ideas of Gottlob Frege, continuing with other developments such as David Hilbert's axiomatic method. The chapter ends by introducing Alan Turing's computability theory. Chapter 1 is essential for understanding the problem of computational intractability discovered by Wang in chapter 8 and Cook's P/NP distinction in chapter 9.

Chapter 2 briefly reconstructs the history of computer science. This chapter explains U.S. scientific policy during the twentieth century. The importance of understanding the context of the scientific policy in the USA is seminal to understand why institutionalization of computer science happened after a decade after AI. This chapter also considers early computing as a U.S. government project for automating mathematical calculations and as an instrument for guaranteeing U.S. hegemony worldwide. The final part of this chapter explains the change of identity of the field during the 1960s, switching from military proposes to civilian ones. This chapter connects with chapter 9 because the formalization of computational intractability happened in computational complexity, part of the computer science.

Chapter 3 reconstructs briefly the history of cybernetics. The chapter starts by considering the beginnings of cybernetics linked to the development of the radar. The chapter continues by describing the influence of different academic areas, such as statistics and physiology, for cybernetics development. The chapter ends with the connection between cybernetics and Herbert Simon's theory of human decision-making. Chapter 3 is linked with chapter 4 because Simon used a servomotor to mature his theory of human decision-making.

The second block of this thesis deals with the emergence of AI. This block has two chapters. Chapter 4 explains the development of the first intelligent computer program at RAND, exploring Simon and Newell's influence over the development of the LT. The chapter starts with a brief reconstruction of Simon's biography to explain how he theorized human decision-making. This chapter also follows the life of Newell to show the connection of his conception of human decision making linked with RAND. This chapter also shows the synergy between Simon, Newell, and John C. Shaw to conceive and program the LT. Chapter 4 is connected with chapter 5 because the LT was the only computer program that showed promising results for developing machine intelligence at the Dartmouth Summer Research Project on Artificial Intelligence in 1956.

Chapter 5 reconstructs briefly the history of the institutionalization of AI at the Dartmouth Summer Research Project. This chapter explains briefly the relationship between private sector with early artificial intelligence development because the Rockefeller Foundation funded the workshop. This chapter also explains how the interaction between different actors provided core for the development of AI in the years to come. Chapter 5 is connected with chapter 6 because that chapter explains how artificial intelligence blossomed after the workshop.

The last block focuses on the discovery of the problem of computational intractability and a possible solution to the problem. This block consists of four chapters. Chapter 6 focuses on showing the confidence AI researchers had at the beginning. Natural language processing and computer vision showed outstanding results. People involved in developing intelligent algorithms believed the field was going to emulate human decision-making efficiently. Researchers thought that “microworlds” were the key for theorizing the laws of human decision-making. This chapter is connected with chapter 7 because that chapter explains the crisis AI suffered.

Chapter 7 shows how artificial intelligence lost its highly ambitious initial impulse. This chapter describes the limits of the most promising areas in AI. Thus, natural language processing and human decision-making suffered the consequences of overconfidence. One of the primary explanations for the failure of AI could lie in the constraints imposed by the Cold War explained in chapter 8.

Chapter 8 deals with the problem of AI as a closed field during most of the Cold War. This chapter aims to explain why the early artificial intelligence community ignored Wang. Starting with Quine’s biography, and its connection with Wang as his Ph.D. supervisor, the chapter presents Wang in the light of the development of mathematical logic, AI, and the dispute with Simon concerning automatic theorem proving programs. Ignoring Wang’s work is the leading cause for the discovery of the problem of computational intractability outside AI as chapter 9 shows. Also, this chapter is connected to chapters 5 and 6 and 7 because they explain the enthusiasm and crisis of artificial intelligence because Cold War constraints led to the crisis of AI.

Chapter 9 deals with the formalization of computational intractability and its direct influence on the development of the field of computational complexity. This chapter touches on the link between the discovery of the problem of computational intractability with games. Finally, the chapter deals with the formalization of the problem of computational intractability in Cook’s P/NP distinction.

Chapter 9 also deals with a proposal for solving the problem of computational intractability by merging artificial intelligence with computational intractability. As will be explained in chapter 9, games shaped “Cold War rationality”, influencing in the developments of AI and computational complexity. AI provides methods for data acquisition and problem-solving, and computational complexity explains whether achieving a solution to a problem in a reasonable amount of time is feasible. The marriage between AI and computational complexity could be the answer for transcending the problem of computational intractability.

Part I

A prehistory of computational intractability

Chapter 1

The development of mathematical logic in the nineteenth and the first half of the twentieth century

Before developing a historical narrative of the discovery of the problem of computational intractability during the Cold War, it is crucial to understand the previous formulations of mathematical logic from the second half of the nineteenth to the first half of the twentieth century. The most essential is here the link between mathematical logic and the rules of correct thinking. Computational intractability, grounded in mathematical logic, explains whether a particular problem could be solved in a reasonable amount of time or not. The rules of correct thinking, insofar as they are expressed in logical terms, require that problems be solved in a reasonable amount of time. In that sense, reviewing the formulations of mathematical logic are helpful to understand the discovery of computational intractability. The very first glimpse into the problem of computational intractability was provided by the conceptual development of the Turing machine. This chapter, accordingly, aims to describe the formulations of mathematical logic from the second half of the nineteenth century until the development of the Turing machine in 1936.

The history of logic from the nineteenth to the twentieth century has been studied extensively. Books such as Jean van Heijenoort's *From Frege to Gödel: A sourcebook in mathematical logic, 1879-1931* (1967) introduces different works of the best logicians from the late nineteenth century to the first half of the twentieth. Among these were, famously, Gottlob Frege (1848-1925) and Bertrand Russell (1872-1970). Even though van Heijenoort does not present a historical narrative about mathematical logic from an externalist historical standpoint, *From Frege to Gödel* introduces each of the works edited in the volume so that the reader can understand the intellectual context of each piece. Unfortunately, essential oeuvres of several logicians, such as Giuseppe Peano's *Principles of mathematical logic*

(1891), are not included. Hubert C. Kennedy's *Selected works of Giuseppe Peano* (1973) complement van Heijenoort's collection.

Selected works of Giuseppe Peano follows the same style as van Heijenoort's. *The geometrical calculus according to the Ausdehnungslehre of H. Grassmann, preceded by the operations of deductive logic* (1888), and his famous *The principles of arithmetic, presented by a new method* (1889), are part of Kennedy's collection. Kennedy briefly introduces the work of Peano in each chapter. Even though *Selected works of Giuseppe Peano* and *From Frege to Gödel* do an essential job of compiling several important works, the main problem with both books is the lack of connection with important mathematical logic written after 1931. But Leila Haaparanta's *The development of modern logic* (2009) is a well-researched and written book about the history of mathematical logic that connects different logical and philosophical works from the early-modern period until our days, thus complementing the volumes edited by Kennedy and van Heijenoort.

Haaparanta's narrative starts with the development of mathematical logic from the early modern period to the twentieth century. Unfortunately, she does not consider the connection between mathematical logic and the P/NP distinction developed by mathematician Stephen Cook. Furthermore, Haaparanta does not analyze the relationship between the Cold War and the development of mathematical logic during the second half of the twentieth century. Likewise, the link between mathematical logic and computational intractability through the works of Wang are not discussed by Haaparanta, nor by other authors in the history of mathematical logic. The main problem with most of the narratives written in the field of history of logic is the lack of connection with external factors such as economy or culture. For the case of my thesis, studying the external elements that influenced certain discoveries and developments in mathematical logic are crucial to understanding the separation between computational intractability from artificial intelligence. Did the Cold War really impose constraints on mathematical logic and scientific developments? Even though this chapter does not yet answer that question, this chapter provides a framework for understanding the connection between Turing and Wang in chapter 8 and 9. Chapters 8 and 9 will answer many important issues related to the connection between the development of science and mathematical logic in the context of the Cold War.

The narrative of this chapter connects with the discovery of the problem of computational intractability in chapters 8 and 9. Those chapters are linked with this chapter because the concept of computability presented in Alan Turing's "On computable numbers with an application to the *Entscheidungsproblem*" (1936) connects with the tractable/intractable distinction developed in Hao Wang's "Proving theorems by pattern recognition II" (1961) and "Dominoes and the AEA case of the decision problem" (1963). Six sections make up this chapter. The first treats the field of mathematical logic in Frege's thought. The second explains mathematician David Hilbert's attempts to save the field of mathematical logic. The third section presents Peano's school as an alternative to Hilbert's axiomatics. The fourth describes Russell's project to overcome the inconsistencies in mathematical logic through his theory of types, while the fifth section shows how logician Kurt Gödel proved the limits of the attempts to axiomatize mathematical logic. Finally, the sixth section of this chapter focuses on the concept of the Universal Computing Machine developed by Turing. Together, all these parts explain how fundamental mathematical logic was to Turing's conceptualization of computability.

1.1 Gottlob Frege's new logic

During the nineteenth century, logic underwent significant changes primarily because logic was mathematized (Haaparanta, 2009, p. 159). Previously, logic centered primarily (though by no means exclusively) on studying the Aristotelian syllogisms. In the mid-nineteenth century, a revolution began when mathematicians became interested in the study of this discipline (Feferman, 1993, p. 376).[6] The pioneers in that respect were George Boole (1815-1864), Augustus

---

[6] During the nineteenth century, the transformation of different disciplines related to mathematics happened because mathematics and its philosophical understanding changed from the intuitive to the intellectual or abstract (Grant & Kleiner, 2015, p. 85).

DeMorgan (1806-1871) and William Stanley Jevons (1835-1882) (van Heijenoort, 2000, p. vi).[7] The most significant change, however, came in 1879 with the publication of Frege's *A formula language, modeled upon that of arithmetic, for pure thought* (*Begriffsschrift*; Haaparanta, 2009, p. 197). *Begriffsschrift* was a revolutionary book because Frege created a notation that replaced the traditional logical analysis of subject and predicate with a mathematical one (Haaparanta, 2009, p. 197).[8] Some of the major contributions Frege made in his *Begriffsschrift* were: a truth-propositional calculus, the theory of quantification, a system of logic in which derivations were carried out exclusively according to the form of the expressions, and a logical definition of the notion of mathematical sequence (Frege, 2000, p. 1; cf. van Heijenoort (2000); Haaparanta, 2009, p. 197). Frege's mathematical logic is framed in his project known as *logicism*.[9]

---

[7] DeMorgan was interested in reforming syllogistic logic. In his *Formal logic* (1847), he proposed to change the entire system of logic by introducing algebraic notation without considering any operational calculus (DeMorgan, 1847, pp. 46-54; Haaparanta, 2009, p. 169). That attitude differed entirely from that of other two mathematicians, Boole and Jevons. They developed methods for doing calculations on logical tasks. Boole's *An investigation of the laws of thought* (1854) tried to emulate the laws of thought that humans perform by developing a symbolic language for the algebraic treatment of logic (Boole, 1854, p. 1; Merrill, 2012, p. vii). Following Boole's work, Jevons's *Pure logic* (1864) presented a system of symbolic logic (Jevons, 1864, pp. 1-3; Haaparanta, 2009, p. 171). See Burris (2018) for Boole; for DeMorgan (Rodriguez, 2017a), and Stanley Jevons (Mosselmans, 2015).

[8] During the nineteenth century, the evaluation of inferences employed Aristotelian and propositional logic (Weiner, 1999, p. 26). Even though both techniques were considered helpful for understanding inferences derived from arguments, no unified method for achieving that goal existed (Weiner, 1999, pp. 26-27). Boole considered the possibility of developing a language that could solve that problem. Unfortunately, Boole's project failed because his notation could not evaluate arguments that were a combination of propositional and non-propositional logic (Weiner, 1999, pp. 27-28). Frege understood that some valid arguments are neither Aristotelian nor propositional (Weiner, 1999, pp. 28). In that spirit, he proposed a new language that could overcome the limits of Boole's system in his *Begriffsschrift.* In his book, Frege introduced the analysis of logical propositions by function and argument (Weiner, 1999, pp. 28-29; Haaparanta, 2009, p. 197, Frege, 2000, p. 1-8).

[9] *Logicism* is the project which suggests that mathematics is reducible to logic.

Unfortunately, Frege's project shook after Russell's paradox appeared. Mathematicians felt during the nineteenth and early twentieth century that the same foundation of mathematics started to crumble. That was why several mathematicians and philosophers, including Hilbert, sought ways to guarantee the reduction of mathematics to logic (Agar, 2001, p. 80).

1.2 David Hilbert's axiomatic method

In 1899, Hilbert (1862-1943) delivered a speech at Göttingen on the foundations of geometry. The speech was included in his *Grundlagen der Geometrie,* published in the same year (Blanchette, 2007). Hilbert presented a technique for deducing theorems of geometry called the "axiomatic method". This method consisted in finding the relationships between axioms of Euclidean geometry and some of the fundamental theorems of geometry (Hilbert, 1968, pp. 1-2).[10] Moreover, Hilbert's technique was useful for metatheoretical reasoning in the process of demonstrating consistency and independence via reinterpretation (Blanchette, 2007).[11] This development made him very optimistic about the future. He shared that view during the Second International Congress of Mathematicians held in Paris in 1900. Here, Hilbert presented his famous list of problems in a special lecture (Haaparanta, 2009, p. 319; Hilbert, 1902, p. 437).[12] He was completely conscious of the fundamental problems many mathematicians claimed their discipline was suffering from. But instead of being pessimistic, Hilbert thought positively about the future possibilities of that area of study:

---

[10] Even though geometry used the axiomatic method, this technique had a powerful influence in other areas of mathematics later (Blanchette, 2007).

[11] Hilbert believed in the simplicity of the axiomatization of geometry (Haaparanta, 2009, p. 324). He thought that the process must reduce the number of axioms to a minimum and show their independence (Haaparanta, 2009, p. 324).

[12] In the lecture given by Hilbert previous to the congress of 1900, he presented 23 unsolved mathematical problems (Agar, 2001, pp. 81-82). Some of them became central in the discipline of mathematical logic, like the decision problem for Diophantine equations (Haaparanta, 2009, p. 319).

> “History teaches the continuity of the development of science. We know that every age has its own problems, which the following age either solves or casts aside as profitless and replaces by new ones. If we would obtain an idea of the probable development of mathematical knowledge in the immediate future, we must let the unsettled questions pass before our minds and look over the problems which the science of today sets and whose solution we expect from the future. To such a review of problems, the present day, lying at the meeting of the centuries, seems to me well adapted. For the close of a great epoch not only invites us to look back into the past but also directs our thoughts to the unknown future” (Hilbert, 1902, p. 437).

That optimism crumbled when Russell presented his famous paradox, giving a blow to the very foundations of mathematics (Zach, 2003).[13] In 1901, Bertrand Russell (1872-1970) found some inconsistencies in Frege’s *Begriffsschrift*. One was related to set theory (van Heijenoort, 2000, pp. 124-125). Russell considered that a predicate cannot be predicated of itself. One can surely think of such a predicate. But can such a predicate be predicated of itself? Both a negative and a positive answer lead into a contradiction; and then such a predicate isn’t really a predicate after all. The same, Russell claimed, was true of the set of all classes that do not include themselves (Russell, 2000, pp. 124-125). In section 1.4 below, we will return to his paradox. Here it needs to be noted that Hilbert became more meticulous in his work after learning about the paradox became known to him. He presented a method to provide a secure foundation for mathematics that avoided any paradox during his speech at the Third International Congress of Mathematicians held at Heidelberg in 1904. The speech referred to his first attempt to provide consistency in arithmetic (Hilbert, 2000, p. 129; Peckhaus & Kahle, 2002, pp. 162-163). After criticizing other authors, Hilbert declared himself as follows:

> “It is my opinion that all the difficulties touched upon can be overcome and that we can provide a rigorous and completely satisfying foundation for the notion of number, and in fact by a method that I would call *axiomatic* and whose

---

[13] Even though Göttingen mathematicians knew about the paradoxes that mathematics had, the discovery of the Russell paradox was problematic for the development of Hilbert’s program because he had to prove his system was consistent enough to avoid paradoxes, specially arithmetic (Peckhaus & Kahle, 2002, pp. 158-163).

> fundamental idea I wish to develop briefly in what follows. Arithmetic is often considered to be part of logic, and the traditional fundamental logical notions are usually presupposed when it is a question of establishing a foundation of arithmetic. If we observe attentively, however, we realize that in the traditional exposition of the laws of logic certain fundamental arithmetic notions are already used, for example, the notion of set and, to some extent, also that of number. Thus we find ourselves turning in a circle, and that is why a partly simultaneous development of the laws of logic and of arithmetic is required if paradoxes are to be avoided" (Hilbert, 2000, p. 131).

Hilbert's treatment of the foundations of mathematics stagnated until 1917 after giving an address before the Swiss Mathematical Society (Hilbert, 2000, p. 129; Hilbert, 1996, p. 1105).[14] Aside from talking about independence and consistency in mathematics, he explained the necessity of axiomatizing logic in order to prove that set theory and number theory are parts of logic (Hilbert, 1996, pp. 1107-1113).[15] Moreover, Hilbert also recognized another problem that would become highly important: the decidability in mathematics:

> "When we consider the matter more closely we soon recognize that the question of the consistency of the integers and sets is not one that stands alone, but that it belongs to a vast domain of difficult epistemological questions which have a specifically mathematical tint: for example (to characterize this domain of questions briefly) the problem of solvability in principle of every mathematical question, the problem of the subsequent checkability of the results of a mathematical investigation, the question of a criterion of simplicity for mathematical proofs, the question of the relationship between content and formalism in mathematics and logic, and finally the problem of decidability of mathematical question in a finite number of operations. Among the mentioned questions, the last- namely, the one concerning decidability in a finite number of operations- is the best known and the most discussed; for it goes to the essence of mathematical thought." (Hilbert, 1996, p. 1113).

With the idea of decidability in mind, Hilbert and the mathematician Wilhelm Ackermann used the concept *Entscheidungsproblem* (decidability problem) in their *Grundzüge der theoretischen Logik (*1928).[16] Their proposal was quite

---

[14] In his speech, Hilbert extended his axiomatic method to other areas of mathematics beyond arithmetic (Hilbert, 1996, pp. 1105-1107).

[15] Hilbert referred to the *Principia Mathematica,* where Bertrand Russell and Alfred North Whitehead (1861-1947) attempted to axiomatize logic (Hilbert, 1996, p. 1113; Zach, 2003)

[16] The mathematician Heinrich Behmann (1891-1970) introduced the concept of *Entscheidungsproblem* during a talk in 1921 at the Mathematical Society in Göttingen

simple. A given logical expression was generally valid or satisfiable if an algorithm could solve a logical problem in a finite number of steps (Hilbert & Ackermann, 1975, pp. 138-139; cf. Mahoney, 2003, p. 620). This concept can be framed in the project called *formalism*. *Formalism* is a school born in the twentieth century that treated mathematics as a purely axiomatic system and that denied that mathematics as such, has to have a content of its own. Instead -and in contrast to, say, Frege's *logicism* and several other positions in the philosophy of mathematics -formalists held that mathematics is merely a formal system, more akin to a play, rather than representing a set of truths refering to abstract or platonic objects. In other words, mathematics was treated axiomatically, i.e, one uses the axiomatic method to deduce proofs (Grant & Kleiner, 2015, pp. 86-87; Grant & Kleiner, 2015, p. 1). Hilbert was confident that his formulations could produce a breakthrough in mathematics, believing every mathematical statement could be proved. However, this optimism was destroyed when mathematician Kurt Gödel proved that Hilbert was mistaken, arguing that mathematics have undecidable propositions (Agar, 2003, pp. 71-72).

1.3 Bertrand Russell's logic, the paradoxes of set theory, and the theory of types

This section focuses on explaining the development of Russell's theory of types. I claim that understanding theory of types is important because Willard van Orman Quine criticized Russell's. The connection between Quine and Russell is important because Quine's mathematical logic influenced Hao Wang. Wang formalized computational intractability in mathematical logic. Chapter 8 deals with the connection in greater detail detail.

Russell was a logician-philosopher who, much like Frege, believed that logic could be the answer to solving the crisis of mathematics during the late nineteenth and early twentieth century. The idea came into his mind when he got in touch

---

(Haaparanta, 2009, p. 382). To see more about Behmann see (Feferman et al., 2013, p. 13).

with the school of Peano during his early career (Dick, 2014, pp. 22-24; Kennedy, 1980, pp. 91-92). He met Peano during the First International Congress of Philosophy in Paris in 1900.

Russell enjoyed studying mathematics since he was young, especially geometry (Russell, 2010, p. 12; Russell, 2010, p. 16). In the beginning, he had problems understanding the foundations of mathematics, but after he discovered the school of Peano, Russell could understand the elements of mathematics he had problems with. Russell asked Peano to provide him with his works. Peano agreed with Russell's request. After studying closely Peano's works, Russell abandoned Boolean algebra (Russell, 2010, p. 16; Kennedy, 1980, pp. 91-92, Haaparanta, 2009, pp. 330-331). In a sense, Peano opened up a whole new world to young Russell. Even though he studied philosophy during his last year at Cambridge, Russell did not consider the existence of a relation between logic and mathematics (Russell, 2010, p. 30; Haaparanta, 2009, p. 331).[17] But after learning about Peano's work, Russell changed his mind. Russell's views about the issue were expressed in greater detail in his *The principles of mathematics* (1903) (Russell, 1996, p. v; Russell, 1996, p. xv).[18]

Most of *The principles of mathematics* was written during the 1900s. In 1901, Russell focused on studying Frege. Russell understood that Frege had reached the same conclusions as him (Russell, 2010, p. 16; Russell, 1996, p. xviii). Moreover, Frege's influence over the young Russell made him see the limitations of Peano's notation, causing him to introduce Frege's notation in his treatment of mathematical logic, i.e., symbols of union and disjunction of propositions (Russell, 1996, p. xviii; Haaparanta, 2009, p. 331). However, as already indicated above (section 1.2), Russell soon came to criticize Frege's work. Russell wrote to Frege in 1902 about his finding (Russell, 2010, p. 16; Haaparanta, 2009, p. 331). Russell, in contrast, began developing his own so-called theory of types to overcome this situation.

Russell's theory of types was presented first in *Principles of mathematics* (1903), and then again in "Mathematical logic as based on the theory of types" (1908), and

---

[17] During his first three years at Cambridge, Russell studied mathematics (Russell, 2010, p. 14; Russell, 2010, p. 30).

[18] Another goal of the book was to show the precision of mathematics (Russell, 1996, p. xv).

finally in his masterpiece written with Alfred North Whitehead, *Principia Mathematica* (1910-1913).[19] In *Principles of mathematics*, Russell's theory was called “simple” theory of types, while in the other two texts was called “ramified” theory of types (Haaparanta, 2009, pp. 332-333). The simple theory of types was Russell's first attempt to overcome the paradoxes of mathematical logic. He ranked classes in the following way. The lowest of all types was related to the type of all individuals. The next one was the class that contained all classes of individuals. The next was the class that contained the class of all classes of type 1. This process continued ad infinitum (Haarapanta, 2009, p. 333).

This point of view changed in Russell’s subsequent publications. Russell’s “On some difficulties of transfinite numbers and order types” (1905) introduced the distinction between predicative and non-predicative norms. For both cases, Russell considered norms of one and two variables. In the first case, predicative norms were those which define classes, while the non-predictive did not. For the case of two variables, a norm that defined a relationship was called predicative, while the other was non-predicative (Russell, 1907, p. 34). Russell differentiated them by proposing a solution to the paradoxes in the first years of the 1900s, i.e., the Burali-Forti paradox.[20] That was the reason why he mentioned three ways of overcoming the paradoxes: the zig-zag, the limitation of size, and no-classes theories (Haaparanta, 2009, p. 334).

The main idea of zig-zag theory was that a propositional function could determine classes if they were pretty simple. Otherwise, they would not. Moreover, depending on how many variables the proposition had, this method could determine a class or a relation. The proposition’s negation was always predicative (Russell, 1907, p. 38). The second idea Russell proposed was the theory of limitation of size. Like the former, it defined a class or a relationship depending on the number of variables of the proposition. Moreover, the proposition was capable

---

[19] Whitehead was a mathematician and logician at Harvard University from 1924 until 1937. After graduating in 1884 from the University of Cambridge, he became a fellow of Trinity College the same year. Before working with Bertrand Russell, Whitehead had been Russell’s professor at Cambridge (Desmet, 2018).

[20] See (Copi, 1958, p. 281) for a detailed explanation of the Burali-Forti contradiction.

of ordering the elements of a class in a well-ordered series (Russell, 1907, p. 38; Russell, 1907, p. 43). The last banned classes and relations (Russell, 1907, p. 45).

Henri Poincaré's "Les paradoxes de la logique" (1906) criticized Russell's proposal.[21] In his work, Poincaré was a strong critic of mathematical logic. He questioned the infallibility of the discipline, saying that mathematical logic did not have consistency. Poincaré contrasted mathematical logic with mathematics. He claimed that the rules of mathematics were infallible, and finding any error produced in the process of doing mathematics was impossible. Poincaré used that argument to say that mathematical logic was fallible (Poincaré, 1906, pp. 295-296). He welcomed Russell's efforts to eliminate mathematical logic's contradictions, but he was also skeptical about the potential of these efforts (Poncaré, 1906, p. 296). Poincaré analyzed the three methods Russell had developed and rejected them, pointing out that it was not clear how they would solve the contradictions. Moreover, Poincaré added that non-predicative propositions contained vicious cycles (Poincaré, 1906, pp. 305-308; Haarapanta, 2009, p. 334).[22]

Russell replied to Poincaré's criticism in his "Les paradoxes de la logique" (1906). Russell defended his no-class theory, pointing out that his idea would help eliminate vicious cycles (Russell, 1906, p. 627). Contrary to Poincaré, Russell mentioned that mathematics and logic are intertwined. His purpose was not to change logic but to understand the relationship between the principles of mathematics and those of logic (Russell, 1906, pp. 628-632). Russell continued by arguing that the problem of the vicious cycle, attributed to William of Ockham (1287-1347), was partially understood by Poincaré because he did not get to the point about the importance of reviewing the principles that guided logic. Russell believed that his theory of classes was closer to Ockham's goal than Poincaré's

---

[21] Poincaré was a mathematician who worked in several areas like mathematics and philosophy of science. Please check Heinzmann (2017) for more information about Poincaré.

[22] The vicious cycle, according to Russell and Whitehead, obeys the following principle: "The vicious cycle principle in question arises from supposing that a collection of objects may contain members which can only be defined by means of the collection as a whole" (Russell, 1927, p. 37).

(Russell, 1906, pp. 632-633).[23] Even though Russell did not make any concrete proposal to solve the vicious cycle, it was the beginning of his ramified theory of types (Russell, 1906, p. 633-634; Haaparanta, 2009, p. 335).

Russell proposed to solve the paradoxes of mathematical logic, reviewing his work to do some corrections in "Mathematical logic as based on the theory of types" (1908). He started by explaining of some contradictions he considered relevant, like the Burali-Forti paradox, pointing out the contradictory property in mathematical logic with the concept of totality (Russell, 1908, pp. 222-225). After that, Russell considered the problem of a variable in a proposition. He pointed out that the attempts to restrict the variable were ineffective in eliminating the contradictions. One possible solution to this problem was the limitation of the size of the set of values it can take. Russell took that idea as a starting point to explain the logical homogeneity of the different elements the set had (Russell, 1908, pp. 231-236).

After that, Russell explained his idea of the hierarchy of types. In his approach, a type is defined as the range of values a function could take. Russell justified the necessity of this formulation to avoid the vicious cycle principle that paradoxes produced. Moreover, Russell explained that an apparent variable could not be related to any other variable. It must be from a higher type. The content of what was in the variable determined its type (Russell, 1908, pp. 236-237). Later, Russell classified two types of propositions: generalized and elementary. The first contained an apparent variable, while the other did not. The former type was created from the latter by a process called generalization.[24] From the concepts of elementary propositions could be distinguished in one or more terms. Russell named them "individuals" and viewed them as part of the lowest type of classes. He pointed out that the lowest type were the most simple entities and were not

---

[23] Ockham worked in several areas of knowledge like logic, natural philosophy, and ethics. His most known idea is Occam's razor, which states that from two competing theories, the most simple must be chosen (Duignan, 2015; Spade, 2019).

[24] According to Russell (1908), the process of generalization is: "The substitution of a variable for one of the terms of a proposition, and the assertion of the resulting function for all possible values of the variable" (Russell, 1908, p. 237). In other words, this process helped to obtain new propositions (Russell, 1927, p. 162).

propositions (Russell, 1908, pp. 237- 238). The propositions that contained only individuals as apparent variables were called first-order propositions, the second logical type (Russell, 1908, p. 238). The first-order propositions that had apparent variables formed the third logical type. The process continued ad infinitum (Russell, 1908, p. 238). Even though Russell defended his idea of a hierarchy of propositions, he said it was better to have a hierarchy based on functions. The order of the functions was obtained by the method of substitution (Russell, 1908, p. 238).

Russell classified functions in the following way. The hierarchy of functions was very similar to the hierarchy of propositions. It obeyed the following logic:

> "A function whose argument is an individual and whose value is always a first-order proposition will be called a first-order function. A function involving a first-order function or proposition as apparent variable will be called a second-order function, and so on. A function of one variable which is of the order next above that of its argument will be called a predicative function; the same name will be given to a function of several variables if there is one among these variables in respect of which the function becomes predicative when values are assigned to all the other variables. Then the type of a function is determined by the type of its values and the number and type of its arguments.
>
> The hierarchy of functions may be further explained as follows. A first order of an individual function x will be denoted as $\phi!x$ (the letters $\psi, X, \theta, F, g, F, G$ will be also used for functions). No first order function contains a function as apparent variable; hence such functions form a well-defined totality, and the ϕ in $\phi!x$ can be turned into an apparent variable. Any proposition in which ϕ appear as apparent variable, and there is no apparent variable of higher type than ϕ, is a second order proposition. If such a proposition contains an individual x, it is not a predicative function of x; but if it contains a first-order function ϕ, and will be written $f!(\psi!z)$. Then f is a second-order predicative function; the possible values of f again form a well-defined totality, and we can turn f into an apparent variable. We can thus define third-order predicative functions, which will be such as have third order propositions for their arguments. And in this way we can proceed indefinitely. A precisely similar development applies to functions of several variables." (Russell, 1908, p. 239).

Russell and Whitehead developed all those ideas in much greater detail in volumes I and II of their *Principia Mathematica* (1910-1913) (Russell, 1927, pp. 11-167;

Russell, 1927, pp. vii-xiii).[25] The Russellian idea of the theory of types was criticized later by the young logician Willard van Orman Quine.[26]

1.4 Kurt Gödel's incompleteness theorem

In 1930, Kurt Gödel (1906-1978) presented his doctoral dissertation at the University of Vienna under the title *Über die Vollständigkeit des Logikkalküls* (*The completeness of the axioms of the functional calculus of logic*). Gödel's thesis focused on proving the completeness of first-order predicate calculus (Gödel, 2000a, p. 582). He started with the achievements of Russell and Whitehead in their *Principia Mathematica*, pointing out their interest in the foundations of mathematics and logic (Gödel, 2000a, p. 583). Then, Gödel focused on an idea: can we know whether the postulated system of axioms and principles of inference incomplete? (Gödel, 2000a, p. 583). He declared:

> "Whitehead and Russell, as is well known, constructed logic and mathematics by initially taking certain evident propositions as axioms and deriving the theorems of logic and mathematics from these by means of some precisely formulated principles of inference in a purely formal way (that is, without making further use of the meaning of the symbols). Of course, when such a procedure is followed the question at once arises whether the initially postulated system of axioms and principles of inference is complete, that is, whether it actually suffices for the derivation of every true logico-mathematical proposition, or whether, perhaps, it is conceivable that there are true propositions (which may even be provable by means of other principles) that cannot be derived in the system under consideration. For the formulas of the propositional calculus the question has been settled affirmatively; that is, it has been shown that every true formula of the propositional calculus does indeed follow from the axioms given in *Principia Mathematica.* The same will be done here for a wider realm of formulas, namely, those of the "restricted functional calculus"; that is, we shall prove" (Gödel, 2000a, p. 583).

Gödel started the enterprise to prove the completeness of the formulas related to propositional calculus (Gödel, 2000b, p. 583). However, his idea was put under

---

[25] This doctoral thesis used the second volume of the book *Principia Mathematica.*

[26] See section 8.1 below.

scrutiny. In *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I* ("On formally undecidable propositions of *Principia Mathematica* and related systems I") (1931), Gödel proved that mathematics has undecidable propositions (Feferman, 1993, p. 378). He showed that some problems in the theory of integers cannot be proved by axioms (Gödel, 2000b, p. 597).[27] Moreover, Gödel showed that mathematics has undecidable propositions because the construction of mathematical statements could not show whether they are true or false (Agar, 2003, p. 72).[28] This development shocked the very foundations of mathematics, producing a crisis in the discipline. One way to save the authority of mathematics, it was hoped, was to find out which statements could indeed be true or false. It is here where Turing entered the picture: he provided the answer to that challenge by developing the concept of a universal computing machine (Agar, 2003, p. 72).[29]

---

[27] As it was explained previously, Hilbert used the ideas of Euclidean geometry for developing his axiomatic method. This development was later applied in other areas of knowledge, including mathematical logic. It was known that geometry used deduction of proofs for reaching conclusions. Gödel proved that Hilbert was wrong because there must be a proposition **p** in a set **S** that cannot be proved in that set using the deductive method, showing that a formal system in mathematical logic was undecidable (Wang, 1991, p. 370; Russell and Norvig, 2010, p. 8).

[28] Gödel also showed that a predicate **p** was a primitive recursive predicate, meaning it was computable. Moreover, he showed that this type of predicates was decidable. For example, if we have the predicate **B(x,y,z)**, it can be proved that it can be true or false for all the values of **x, y,** and **z**. However, there was the possibility of a value **b** that was not computable, meaning that the system had undecidable propositions (Wang, 1991, pp. 370-373).

See also Raatikainen (2013) and Gödel (2000b).

[29] The Turing Machine is considered an abstract prototype of today's digital computer (Mahooney, 2003, 621).

1.5 Turing’s universal computing machine

In 1936, Turing published his famous paper “On computable numbers with an application to the *Entscheidungsproblem*”. Turing's work explained why the problem proposed by Hilbert did not have any solution (Turing, 1936, pp. 230-231). Inspired by proof of the Gödel’s incompleteness of mathematics, Turing’s interest showed which kind of functions could be computable (Russell & Norvig, 2010, p. 8). Turing started as follows:

> “The “computable” numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means. Although the subject of this paper is ostensibly the computable numbers, it is almost equally easy to define and investigate computable functions of an integral variable or a real or computable variable, computable predicates and so forth. The fundamental problems involved are, however, the same in each case, and I have chosen the computable numbers for explicit treatment as involving the least cumbrous technique. I hope shortly to give an account of the relations of the computable numbers, functions, and so forth to one another. This will include a development of the theory of functions of a real variable expressed in terms of computable numbers. According to my definition, a number is computable if its decimal can be written down by a machine.” (Turing, 1936, p. 230).

After this introduction, Turing claimed that the results he obtained with his method were quite similar to those of Gödel’s incompleteness theorem (Turing, 1936, p. 230). Later, Turing developed his idea of computable number (Turing, 1936, pp. 231-232). Moreover, he made an association between the limits of human memory with a machine's finite number of configurations (Turing, 1936, pp. 231-232; Agar 2001, pp. 89-90).[30] The machine Turing proposed consists of a tape divided into sections. Each one of those sections received a scanned symbol (Turing, 1936, p. 231). Depending on the configuration q and the symbol that was going to be scanned at time *t* determined the behavior of the machine at that moment (Turing, 1936, p. 231). [31] Famously, Turing applied his idea in the Second World War, designing an electromechanical machine that could break the secret codes of the Enigma machine created by the Nazis in their war planning. Turing's creation helped the allies to win the war (Watson, 2012). Important for us here, however, is

---

[30] The finite conditions were defined as q1, q2, q3, …., qm (Turing, 1936, p. 231).

[31] This is the concept of a universal computing machine (De Mol, 2018).

the fact that Turing’s computability formulation would serve as the basis for developing the tractable/intractable distinction introduced by Hao Wang during the 1960s.

Conclusions

This chapter showed how mathematical logic influenced the development of Turing's famous computability problem. Knowing the history of mathematical logic is thus essential for understanding the conceptualization introduced by Turing.[32] As I will argue later on in chapters 8 and 9, Turing's concept is similar to the tractable/intractable distinction introduced by Wang.

The first important change in logic came from Frege. He was one of the first logicians to introduce mathematics into logic. Frege's project became highly criticized. One of the main arguments was provided by a paradox discovered by Russell.

Hilbert's axiomatic method sought to give consistency to the field of mathematical logic, but his project crumbled when Gódel discovered that mathematics has undecidable propositions. He concluded that some propositions in an axiomatic system were undecidable, demonstrating the incompleteness of mathematics. Later, Turing came to the same results as Gödel.

Turing conceptualized his Universal Computing Machine in 1936. His machine proved the existence of computable and non-computable problems. Turing found that his machine stayed in a loop while trying to solve certain problems. How are Turing machines linked to the formalization of the problem of computational intractability? Chapters 8 and 9 will answer that question.

---

[32] See chapters 8 and 9 for a detailed explanation of the tractable/intractable distinction influenced by Turing in Wang.

Chapter 2

A new discipline is born

For full historical account of the P/NP distinction in Stephen Cook's "The complexity of theorem-proving procedures" (1971), and of computational intractability more generally to become possible, it is crucial to understand the development and institutionalization of computer science from the 1950s to the 1960s. Cook's distinction was born in the field of computer science, not in artificial intelligence (AI). Why was that so? One indication can be inferred from the dates of institutionalization of both fields because: while computer science was institutionalized during the 1960s, the institutionalization of artificial intelligence happened already in 1956. Chapters 8 and 9 will focus on explaining this issue in greater detail. Computer science was developed primarily during the 1960s because computing changed its disciplinary identity. Before that decade, computing aimed to support activities of the U.S. military to guarantee the hegemony of the USA across the globe. In the 1960s, the field began to focus on civilian activities, such as academic research or the economy. In this chapter, I aim to connect the history of computing before the 1960s with its academic institutionalization during the 1960s.

Very few scholars have treated the institutionalization of computer science during that time. Why is it challenging to develop a narrative that connects both decades? One of the reasons could lie in the two different disciplinary identities of the 1950s and 1960s. The historian of computing Michael Mahoney (1939-2008) has been a pioneer in the study of the history of computer science. Before him, the field had very few works that explained how computer science came into existence.[33] Furthermore, most of the works linked to the history of computer science study the beginning of the discipline, limiting their scope to the 1950s (Mahoney, 2011, pp. 2-3). Mahoney changed that situation because he wrote histories of several areas of computer science during and after the 1950s, such as software engineering and

---

[33] Mahoney was a professor of history of science at Princeton University. He is one of the twentieth century's most influential computing historians (Campbell-Kelly, 2013).

theoretical computer science. Works such as “Computing and mathematics at Princeton” (1999) and “Computers and mathematics: The search for a discipline of computer science” (1992) are essential to explain how theoretical computer science was born from a mathematical point of view. One of the main problems with Mahoney’s historiography is the lack of a description of the institutional and social contexts in which theoretical computer science was born.

Paul Edwards’s *The closed world: Computers and the politics of discourse in Cold War America* (1996) analyses the relationship between the U.S. military with the development of computing and AI during the Cold War.[34] He emphasizes the role of computers as tools for guaranteeing the hegemony of the USA during the postwar years. Furthermore, Edwards explains that computers were helpful for global surveillance to stop the spread of communism during the Cold War. Edward’s narrative is significant for understanding the relationship between computers and the military. Unfortunately, Edward’s narrative does not explain the development of computer science as an academic discipline during the 1960s. Gopal Gupta’s “Computer science curriculum developments in the 1960s” (2007) complements Edwards and Mahoney’s view in that Gupta focuses on computer science development during the 1960s.

This chapter focuses on constructing a brief narrative that links the development of computing for military proposes in the 1950s with the academic institutionalization of computer science in the 1960s. The contribution of this chapter is constructing a brief narrative of the development of computing from the 1950s until the 1960s, describing the change of disciplinary identities of computing. Three sections make up this chapter. The first aims to explain the U.S. scientific policy during the postwar years. The second describes the development of computer technology to aid the establishment of the U.S. hegemony during the Cold War. Furthermore, this part focuses on practical applications of computing technology in military matters. Finally, the last part focuses on explaining the

---

[34] Edwards is a professor of information and history at the University of Michigan. Furthermore, he is William J. Perry fellow in international security at the Center for International Security and Cooperation at Stanford University. See https://cisac.fsi.stanford.edu/people/paul-n-edwards for more information.

change of attitude towards the development of computing, switching its militaristic identity to an academic one.

2.1 The funding of post Second World War science

Two forms of conceiving science were proposed and discussed. The first was proposed by West Virginia's senator Harley Kilgore (1893-1956), while the chairman of the Office of Scientific Research and Development (OSRD) Vannevar Bush (1890-1974) presented a second one.[35] Kilgore and his staff enacted a bill that proposed the creation of the National Science Foundation (NSF) to sustain the development of a fairer and more democratic society (Kevles, 1977, p. 15). The principles of the NSF would give freedom of thought to scientists to exercise their creativity in their scientific work with the only condition the U.S. Government owned the property rights of the resulting work that received public funding. Furthermore, the NSF would fund not only government laboratories but other institutions for supporting scholarships and awarding research contracts (Kevles, 1977, p. 15). Kilgore's proposal was not well received by the advisors of President Franklin Delano Roosevelt, like Oscar Cox (1905-1966). He proposed to convince President Roosevelt to write a letter to Bush. The letter would ask Bush about the future role of science (Kevles, 1977, pp. 15-16).[36]

---

[35] Bush did a bachelor’s and master's degree at Tufts University, graduating in 1913 in electrical engineering. After that, he obtained his Ph.D. in electrical engineering at two universities in 1916: the Massachusetts Institute of Technology and Harvard University (Dennis, 2020). One of the most important contributions in his career was the invention of the so-called “Differential Analyzer” (Dennis, 2020). Bush organized scientific research as part of the machinery of war during the Second World War (Dennis, 2020). His most important position was being the director of the Manhattan Project (Rodriguez, 2022a).

Kilgore was a small-town lawyer who believed in defending the causes of people deprived by the power of big businesses. He became elected senator of West Virginia in 1940 (Kevles, 1977, pp. 7-8).

[36] Cox was born in Portland. He obtained his bachelor’s degree in philosophy at MIT and his bachelor’s in law at Yale University. He became part of the group that advised the Roosevelt

Bush connected science with U.S. government policy during the Second World War, acquiring a great reputation.[37] His work helped the U.S government to carry out projects that were impossible before, like the Manhattan Project (Dennis, 2020). That is why President Roosevelt considered Bush to give his advice (Dennis, 2020). Bush was an anti-New Deal conservative, but he was conscious of the need of U.S. government involvement of scientific production during the years when academia and industry could not finance research by themselves (Kevles, 1977, pp. 13; Kevles, 1977, pp. 16-17). Cox discussed with Bush his plan of blocking Kilgore's bill. Bush and Cox's plan was a success. On November 17, 1944, President Roosevelt sent a letter, drafted by Cox, to Bush asking questions about the role of science (National Academies of Sciences, Engineering, and Medicine, 2020, p. xv). The answer became the report *Science: The endless frontier* (1945) (Agar, 2012, p. 304). The report was received by new President Harry Truman since President Roosevelt had meanwhile died.

Bush's report defended the idea the U.S. government must support basic scientific research to fulfill U.S. needs after the Second World War. The document focused on three fields: health, national security, and public welfare. Bush considered the government's investment in those three areas for bringing more prosperity to the USA. At the same time, research in health, national security, and public welfare were costly for non-governmental institutions. The government, however, could cover those expenses. As a result, government funded science that aimed at creating a more fair society (National Academies of Sciences, Engineering, and Medicine, 2020, pp. 1-5). Although Bush and Kilgore's proposals both sought government funding for scientific research, they differed that Bush proposed federal agencies must not dictate the road science had to take in the future: he defended an ideal of value-freedom of science (Kevles, 1977, p. 19). Unfortunately for both Bush and Kilgore, their proposals had only limited influence when the newly formed NSF was set up in 1950, since the institutions

---

administration from 1938 until Roosevelt's death. See http://www.fdrlibrary.marist.edu/archives/collections/franklin/index.phpp=collections/findingaid&id=456 for more information.

[37] Bush's initiative of merging government and science reduced the influence of the private sector's role on funding science (Dennis, 2020).

created just after the Second World War obtained greater control over scientific policy (Agar, 2012, p. 307). In any case, these governmental institutions became also responsible for providing major funds that helped the development of computing technology. U.S. military agencies, like the U.S. Air Force, were most significant patrons of science during the Cold War (Agar, 2012, p. 307). One of the institutions that had strong relevance during the conflict was the Research and Development Corporation (RAND).

2.2 Early computing

During the Second World War, the U.S. Army used computers to assist in activities related to the conflict. In 1944, Harvard University set up the first computer built in the USA with the support of IBM. The Harvard Mark I was the creation of Howard Hathaway Aiken (1900-1973).[38] In 1936, during his doctoral research, Aiken found that desktop calculators had problems solving certain types of mathematical problems such as calculations for atmospheric research (Aiken et al., 1964, p. 64; Cohen, 2003, p. 1078). For that reason, Aiken wrote a letter between 1936-1937 asking for support to build a machine that could solve those problems. Aiken sent his proposal to the Monroe Calculating Machine Company, but the firm declined Aiken's idea.[39] However, Aiken stubbornly pursued his project, sending his proposal to IBM. The corporation accepted to build the computer, finishing the machine the Christmas of 1943 (Cohen, 2003, p. 1078).[40] Harvard University set up the computer in 1944 (Cohen, 2003, pp. 1078-1079). Harvard Mark I was used from 1944 until the end of the Second World War to calculate dangerous magnetic

---

[38] Aiken was born in Hoboken, New Jersey, in 1900. He studied engineering at the University of Wisconsin. Later, in 1939 Aiken obtained his Ph.D. at Harvard University. (Lotha, 2019).

[39] Founded by Jay Randolph Monroe in 1912, the company produced a calculator that the firm's owner created. Today, the company is known as Monroe Systems for Business. See https://monroe-systems.com/about/ for more information.

[40] The U.S. Navy was one of the main contributors for building the Harvard Mark I when Aiken was commander of the U.S. Naval Reserve (Boden, 2006, p. 825).

fields that U.S. Army ships had to navigate (Cohen, 2003, p. 1079). After the war, Harvard Mark I was no longer a good candidate for carrying out new plans of the U.S. government because it was too slow a machine (Cohen, 2003, p. 1079). Instead, the U.S. government supported other computing projects to achieve its role as the new global superpower. For example, during 1948 the Soviet Union blocked the entrances to West Berlin. In reaction, the USA proposed providing supplies to the British, French, and American-controlled sectors in Berlin. Initially, USA and British airplanes would give supplies by air. Even though the provision of supplies for Western Berlin was successful, the cost of operation was very high (Erickson et al., 2013, pp. 51-57). Computing was essential for solving the cost problem. The machines were responsible for implementing linear programming methods to deliver supplies efficiently (Erickson et al., 2013, p. 67).

The government saw computers as tools for global surveillance and control to prevent the influence of socialism from spreading through the world.[41] The computer became the perfect technology for embracing the new Truman doctrine of helping “free peoples” from becoming subjugated by foreign powers, such as the USSR (Edwards, 1996, p. 1).[42] More specifically, the governmental channels used for funding computer research were mainly military institutions. Computer technology served to build defense systems before 1960 (National Research Council, 1999, pp. 87-88).[43] The Office of Naval Research (ONR) was the most important branch of the U.S. Army for funding early computing projects such as numerical analysis. That project propelled computer design development and processing (National Research Council, 1999, p. 88).

Other military bodies also funded computing research, like the U.S. Air Force (National Research Council, 1999, p. 89). The results seemed to have a positive

---

[41] The role of the U.S. government in funding and purchasing computing technology through the military was critical for positioning the computer industry as an essential productive activity in the USA after the Second World War (Edwards, 1996, p. 62).

[42] See Merrill (2006).

[43] The influence of the U.S. Army in matters of the state grew after the Second World War. Previous to the conflict, the U.S. Army had 185.000 effectives with an annual budget below 500 million Dollars. After the war, the U.S. Army had 12 million men and a yearly budget 100 times greater than in the prewar time (Edwards, 1996, p. 54).

impact in that, for instance, computing technology development helped to centralize the management of the U.S. Army (Edwards, 1996, pp. 5-7).[44]

Furthermore, computers could make calculations that assisted military decisions. The military used simulation techniques to support its decisions. The danger of a nuclear war paved the way for computers to simulate scenarios to reduce catastrophic possibilities derived from a possible confrontation between countries (Edwards, 1996, pp. 14-15). Moreover, computers could help the U.S. military make decisions in a more complex war environment. Furthermore, the machine would reduce the complexity and speed humans could not handle, like using antiaircraft guns and missiles (Edwards, 1996, p. 65). Mathematician John von Neumann (1903-1957) and engineers John Mauchly (1907-1980) and John Presper Eckert (1919-1995) gave life to the project (Mahoney, 2011, pp. 123-124).[45] The result of their collaboration was the creation of the first computer that became name ENIAC (Electronic Numerical Integrator and Calculator). ENIAC was born at the Moore School of Electrical Engineering at the University of Pennsylvania.

ENIAC was the most important project funded by the U.S. government before the Cold War. The machine would automate ballistic weapons calculations. During the Second World War, human calculation of ballistic tables was prone to error, and often also wasted crucial time. Computers seemed to be the solution to that problem. Even though interwar computers were fifty times faster than humans when making calculations, their speed was yet not enough because firing tables

---

[44] The military used also simulation techniques to support its decisions. The danger of a nuclear war paved the way for computers to simulate scenarios to reduce catastrophic possibilities derived from a possible confrontation between countries (Edwards, 1996, pp. 14-15). Moreover, computers could help the U.S. military make decisions in a more complex war environment. Furthermore, the machine would reduce the complexity and speed humans could not handle, like using antiaircraft guns and missiles (Edwards, 1996, p. 65).

[45] Mauchly was born in the city of Cincinnati in the USA. He was a physicist and engineer at the University of Pennsylvania. The U.S. Army invited him to build a technology for automating the calculations of artillery firing tables (Lotha, 2021a).

Eckert was born in Philadelphia in the USA. He went to study at the University of Pennsylvania, graduating as electrical engineer in 1941 and a master of electrical engineering in 1943. During his time as a student, he met Mauchly (Rodriguez, 2022b).

required at least a calculation of 2000 to 4000 trajectories (Edwards, 1996, p. 49). The ENIAC project could solve the problem (Edwards, 1996, p. 50). The machine was presented in December 1945, some months after the Second World War had ended. The computer, built from 1943 to 1945, was a huge 18000 vacuum tube computer with 1500 relays, 70000 resistors, and 10000 capacitors. ENIAC did not help calculate ballistic tables anymore, but ENIAC came to serve for doing hydrogen bomb calculations in 1946 (Edwards, 1996, pp. 49-52; Platzman, 1979, p. 305).[46] After ENIAC came to light, the U.S. Military supported other projects related to the development of computing technology, such as IAS computer. Collaboration between Princeton University and IAS in 1952 was key to building the IAS computer (Mahoney, 2011, p. 122; Edwards, 1996, p. 61).[47]

Supported by the U.S. Navy and later by the Atomic Energy Commission, von Neumann started to build a general-purpose computer to help calculate different subjects. Even though the machine aimed to aid the U.S. military in numerical meteorology, von Neumann's computer would be helpful in other fields such as statistics, traffic simulation, and numerical methods (Mahoney, 2011, p. 122).

Von Neumann saw computers as tools for solving intractable mathematical problems, such as calculating error of inverting matrices of high order. Problems appeared for solving mathematical equations during the twentieth-century. Methods developed for solving equations, such as inner product forms of elimination, produced poor results (Grcar, 2011, p. 621). Concerned about this issue, von Neumann and Herman Heine Goldstine's "Numerical inverting matrices of high order" (1947) focused on calculating error estimate for increasing the accuracy of new methods to perform operations in matrices of high order (von Neumann & Goldstine, 1947, p. 1022). Their paper was written before any computer different than ENIAC was built. During the 1950s, Von Neumann and Goldstine's proposal was programmed first on an IBM 701 obtaining decent

---

[46] Later, Mauchly, and Eckert developed the BINAC and the EDVAC. The first could store programs electronically, while the other was the first electronic com&mercial computer (Edwards, 1996, p. 60).

[47] During the second half of the 20th century, Princeton changed its aim as a university. In 1950, Princeton became a research university. Previously, the institution was a liberal arts college (Mahoney, 2011, p. 121).

results because some precision was sacrificed in order to adapt to machine's capabilities.[48]. Von Neumann's idea became true when the IAS computer came to light in 1952. The IAS computer obtained the best result because the machine had a precise 40-digit arithmetic module (Goldstine et al, 1954, pp. 1-4).

Furthermore, the machine would help to discover new research fields (Mahoney, 2011, p. 125).[49] One of the persons who would come to work with the computer for doing calculations for research purposes was physicist Eugene Wigner (1902-1995).[50] Wigner used the machine to calculate wave function statistics of quantum mechanical systems (Mahoney, 2011, p. 122). Although the machine stopped functioning at the end of the 1950s, the IAS computer inspired other computer projects (Mahoney, 2011, p. 122-123).

One of those was the SAGE project (Semi-Automatic Ground Environment). Originally developed in 1944 at the MIT Servomechanism Laboratory, Project Whirlwind's goal was the development of an airplane stability and control analyzer. Two years later, the project changed of plan focusing on constructing a digital computer to centralize control systems. Project Whirlwind would support projects of logistics, planning, and air traffic control, among others. Still, the project was going to be canceled because authorities thought it had no justification. However, in 1949 the Soviet Union made its first test with an atomic bomb. The USA, fearing war with the Soviet Union, decided to fund the project to protect itself from foreign intervention. Project SAGE was born from Project Whirlwind, having as a goal the development of a continental air defense system that could prevent any catastrophic event (Edwards, 1996, pp. 75-76; Agar, 2012, p. 375).

---

[48] IBM 701 model could perform 16000 addition or subtraction operations, per second. For more information about 701 model, please visit: https://www.ibm.com/ibm/history/exhibits/701/701_intro3.html

[49] The development of the von Neumann architecture opened the possibility of theorizing about the mechanisms of the human brain (Mahoney, 2011, p. 125).

[50] Wigner was born in Budapest in 1902. He studied chemical engineering, earning his Ph.D. in 1925 at the Technische Hochschule Berlin. Wigner taught at the University of Göttingen, the University of Wisconsin, and Princeton University. He won the Nobel Prize in physics in 1963 (Lotha, 2022a).

In 1943-44, the director of the Navy's Special Devices Division, Luis de Florez (1889-1962), suggested developing a general simulator.[51] During the 1940s, servo-operated flight simulators served to train the U.S. Air Force pilots. Those simulators offered a secure and relatively inexpensive environment for simulating the aircraft controls. De Florez proposed developing a simulator to reduce the time and cost of aircraft development and pilot training (Edwards, 1996, p. 76). At the MIT Servo Lab, the idea of building a digital computer that could simulate the flight simulator was approved, giving birth to Project Whirlwind (Edwards, 1996, pp. 76-78). Unfortunately, interest in funding this project decayed in 1949 because ONR concluded that the digital computer was no more than a general-purpose machine (Edwards, 1996, pp. 78 A-79). Jay Forrester (1918-2016), one of the engineers working on Project Whirlwind, gave a justification to continue with the project (Edwards, 1996, p. 79).[52] Before 1949, Forrester and his group discussed the possibilities of making Project Whirlwind something more than an Air Force simulator. His group considered the most critical aspect of Project Whirlwind to build a digital computer serving as a system of defense against ballistic missiles.
Moreover, they thought of computers as tools for controlling and coordinating the entire military power of the USA (Edwards, 1996, pp. 79-81; Agar, 2012, p. 375). (Edwards, 1996, pp. 79-81; Agar, 2012, p. 375). Forrester and his group prepared several grant proposals to get funding from the military for carrying out the changes they wanted. The U.S. Air Force was skeptical about the proposal, but the opinion changed when the Korean War started and the atomic test by the Societ

---

[51] De Florez was the son of Rafael, a Spanish immigrant. His mother, Marie Stephanie Bernard, was the daughter of French immigrants. The De Florez family returned to Spain during the economic depression that hit the USA during the 1890s (Dawson, 2005, pp. 30-33). A Jesuit chaplain close to the De Florez family delivered education to Luis. His tutor inculcated the Jesuit tradition's love for science (Dawson, 2005, p. 33). He returned to the USA, majoring in mechanical engineering at the Massachusetts Institute of Technology in 1911 (Dawson, 2005, pp. 34-35). De Flores became the U.S. Air Force captain in 1943 (Dawson, 2005, p. 118).

[52] Forrester was an MIT professor who developed an area of knowledge for helping corporations measure management policies' impact. That area was called system dynamics modeling (Hafner, 2016).

Union. Fearing a missile attack, the U.S. Air Force approved Project Whirlwind (Edwards, 1996, pp. 81-83).

Project Whirlwind again almost lost funding in 1950. The ONR just delivered 250.000 Dollars in 1950, compared with the 5.8 million Dollars the project received in previous years. The only way to preserve the project was by merging it with another one. SAGE was a project derived from Whirlwind to save the idea of building a digital computer (Edwards, 1996, pp. 90-91; Agar, 2012, p. 375).

Physicist George E. Valley (1913-1999) was appointed as director of SAGE in 1949. This project was born with the idea of creating an air defense system for North America (Edwards, 1996, p. 90; Agar, 2012, pp. 375-376).[53] The project aimed to centralize the U.S. Air Force command in a digital system.[54] In 1949, Valley heard about Project Whirlwind and its advances in digital computers development. Valley approached Forrester to include Project Whirlwind into SAGE, proposing therby to save Forrester and his team's work at MIT (Edwards, 1996, pp. 90-92; Agar, 2012, p. 375).[55] Even though at the beginning, some doubts about early warning system effectivity appeared, the fear of a nuclear holocaust pushed forward the SAGE project.[56] The project became fully operational in 1961. 1 billion Dollars was the construction cost of SAGE's control centers (Edwards, 1996, pp. 96-97; Agar, 2012, p. 375). When SAGE came to light, President Truman's plan of a computer that centralized command, control, and communications was realized (Boden, 2006, p. 829).

---

[53] Valley was part of the U.S. Air Force Scientific Advisory Board from 1946 until 1964. He graduated from MIT in 1935 and obtained his Ph.D. in nuclear physics at the University of Rochester in 1939. Valley was appointed as a professor of physics at MIT in 1946. One of Valley's contributions was measuring iron isotopic abundance in terrestrial and meteoritic materials from 1939 to 1941, when he was a research associate at Harvard University (Massachusetts Institute of Technology, 1999).

[54] In 1951, the U.S. Air Force and the National Security Resources Board concluded that computers would help to improve air defense systems (Edwards, 1996, pp. 93-94).

[55] Forrester gave Valley the L-1 AND L-2 reports on digital computers related to the control of the U.S. Navy (Edwards, 1996, p. 92).

[56] The U.S. Air Force tradition was more active than defensive, so the U.S. Air Force pilots saw as naive the proposal for developing a defense airforce system for protecting the U.S. territory (Edwards, 1996, p. 96).

2.3 Struggles over institutionalization of computer science

The newly created NSF was key for providing funding for colleges to buy computers for their research.[57] Already before 1959, some universities asked the NSF for grants for purchasing new computers that could help them to advance their investigations. Thus, in 1954, the University of California at Los Angeles (UCLA) sent a proposal to the NSF to get funding for a computer that could help the university's research in computer-aided meteorology models (Aspray, 1990, p. 61). However, it was not until 1959 that the NSF had a clear role in aiding universities in buying computers, but then the foundation allocated a budget for helping universities in buying computers (Aspray, 1990, pp. 61-62).

Several schools and universities received larger or smaller grants for buying computers in the years to come. In 1959, the NSF gave 1.5 million Dollars to the University of Chicago and Yale University. Even though the NSF was enthusiastic about aiding universities in buying computers, an unexpected problem appeared: researchers had great problems putting the new computers to use.[58] Louis Fein, a private consultant in California, emphasized the importance of educating college students in the early field of computing.[59] Appointed as chairman of the Association for Computing Machinery (ACM) in 1960, Fein wrote some reports about his view on boosting computing on U.S. campuses.[60] Fein defended the idea

---

[57] Computing industry also provided computers for advancing research at universities independently from any government agency. In 1959, IBM donated its computer Model 650 to more than 50 universities. IBM provided the computers through its educational program (Aspray, 1990, p. 61).

[58] In 1960, a report from the Mathematical, Physical, and Engineering Sciences Division concluded that researchers did not have the access needed at their regional computing centers (Aspray, 1990, p. 62).

[59] Fein studied physics at Long Island University, graduating in 1938. The same year, he entered the University of Colorado at Boulder, obtaining his master's in 1939. Later, he worked at the Submarine Signal Company. While he worked at the firm, Fein could study at Brown University, obtaining his Ph.D. in 1945. From 1952 to 1953, he was a lecturer on digital computer systems at Wayne University. Fein taught at Stanford University three years later (Grosch, 1977).

[60] The ACM was founded in 1947 to support the development of computing. Some members of the ONR were founders of the association (National Research Council, 1999, p. 88).

of supporting computing education in a paper published in 1959. Fein's article explained his view related to the development of computing.[61] Importantly, Fein introduced the term "computer science" when explaining the necessity for creating a Graduate School of Computer Science (Gupta, 2007, p. 40). Yet, Fein saw some difficulties in the early phases of setting up the field of computer science.

After reviewing some practices of college teachers, Fein concluded that university professors were not doing computing research. Professors used computers for programming but refused to leave their teaching hours (Gupta, 2007, p. 41). After Fein's diagnosis, two conferences took place in 1960. The first focused on the role of computers in the education of engineers, while the second emphasized the importance of computer centers (Gupta, 2007, p. 41). After these conferences, other activities discussed the nature of computer science as a discipline.

Universities continued to receive grants to buy computers (Aspray, 1990, pp. 62-63). In 1961, the U.S. Congress authorized a specific budget for building computing facilities at universities, oceanographic research vessels, and the Hawaii Institute of Geophysics (Aspray, 1990, p. 63). During the early 1960s, there was an impetus to formalize the study of how computers work at a software level (Gupta, 2007, p. 40).[62]

In 1963, ACM held a national conference to discuss the nature of computer science. Thomas Keenan, professor at the University of Rochester, explained the differences between computer science and other disciplines, such as physics and mathematics.[63] To Keenan, what made computer science distinct from other fields was the focus of the discipline on how systems work and the process of interaction with different elements of that system through software (Gupta, 2007, p. 43).

---

[61] Unfortunately, I had no access to Feyn's "The role of computer in universities" (1959).

[62] The early years of computing focused on the development of hardware and its maintenance rather than software (Gupta, 2007, p. 40).

[63] Keenan obtained his Ph.D. in physics at Purdue University in 1955. Later, he became a professor at the University of Rochester. While he was teaching at the institution, the university was interested in buying a computer. When the university purchased the machine, Keenan was appointed as the university's computer center director. After using the computer in the computer center for a time, Keenan thought of the need for the intellectual part of computer science. Unfortunately, he found that abstract mathematicians disagreed that computers could help develop their theories. (Aspray, 1990, pp. 3-7).

Moreover, Keenan pointed out the rate of production of computers exceeded the number of people that could learn how to use a computer. Keenan supported his statement by pointing out that 500 computers per month were built (Gupta, 2007, p. 43). At the conference, some college professors gave their points of view on the courses that computer science curricula should have. One proposal came from mathematics professor at Stanford University, George Forsythe (1917-1972)[64].

Forsythe suggested two courses on numerical analysis. The first was aimed at first- and second-year students who knew some programming, while the second course was for graduate students. The second course required a little programming knowledge and a background in mathematics (Gupta, 2007, p. 44). Forsythe believed that numerical analysis was an essential subject in computer science because the course would significantly impact computer science, especially error analysis (Forsythe, 1959, pp. 654-655). Unfortunately, Forsythe's proposal was unrealistic because the quantity of material he suggested was more significant than what could be covered (Gupta, 2007, p. 44). In other conferences related to computer science development, the discipline started to take shape.

The ACM National Conference in 1964 brought some interesting insights into computer science development. Academics attend the event saw that two universities incorporated an undergraduate program in computer science: Purdue University, and the University of Maryland (Gupta, 2007, p. 44). Purdue's undergraduate program included 16 hours of computer science courses like programming and numerical analysis. Furthermore, the program also had 24 hours of mathematics (Gupta, 2007, p. 44). In contrast, the University of Maryland centered on computer science, making compulsory for undergraduates to take at least 30 hours of computer science (Gupta, 2007, pp. 44-45). By the first half of the 1960s, many universities were giving computer science courses. The academic community began to recognize computer science as an academic discipline (Gupta, 2007, p. 45).

---

[64] Forsythe was a pioneer in the development of early computer science. He began his interest in computers while working in numerical analysis in 1948 at the National Bureau of Standards Institute for Numerical Analysis in Los Angeles. During that time, Forsythe programmed the SWAC computer. He became a professor of mathematics at Stanford University in 1957 (Knuth, 1972, pp. 721-722).

In 1965, the ACM presented a preliminary report concerning recommendations for computer science. The society delivered its recommendations based on the conferences of 1963 and 1964. It stressed that computer science curricula should have four compulsory courses and seven elective courses.[65] In 1967, a new forum about computer science nature was taking place at the State University of New York (SUNY) at Stony Brook.

By 1967, the interest in computer science had grown to such a point that curriculum development for computer science graduate program attracted more people than expected at the conference held at SUNY. Initially, the event was going to have 30 participants. However, the conference attendees were nearly 70 participants from different parts of the world, including Australia (Gupta, 2007, p. 45). All the participants went to the event to discuss the future of computer science. The discipline was not yet considered an intellectual discipline, just a tool to be used in other areas of knowledge such as economics. Indeed, some academics said that computer science would disappear in a few years (Gupta, 2007, p. 45).

So, some academics agreed that computer science had a valuable intellectual substance and autonomy, while others disagreed. Mathematicians and other scholars working in other scientific fields such as physics, did not recognize computer science as a science. Engineers thought the other way. One engineer who defended the importance of computer science was Frank Beckman (1921-2009) from IBM (Gupta, 2007, p. 46).[66] Beckman saw computer science as essential to be taught at universities and should have a proper organizational structure. Beckman's view was defended by the mathematician Lotfi Zadeh (1921-2017)

---

[65] The compulsory and elective courses suggested by the ACM in its 1965 report can be seen in table 1 of Gupta (2007).

[66] Beckman obtained his Ph.D. at Columbia University in 1965. By the time he was pursuing his degree, Beckman was working at IBM. In 1971, Beckman left IBM to create the department of computer science at Brooklyn College. Until 1985, he was the chairman of the department. Beckman was also the executive director of the doctoral program of computer science at the City University of New York (CUNY) Graduate Center until 1993. See http://www.ams.org/publicoutreach/in-memory/inmemory-2008-2009 for more information.

(Gupta, 2007, pp. 46-47).[67] On the other hand, professors like Vladimir Slamecka (1928-2006) differed from Zahde's view of science.[68] To Slamecka and others, science discovers principles that explain phenomena, and therefore it was not possible to not include the study of computers in the category of science because computing does not lead to the discovery of laws as physics or chemistry do (Gupta, 2007, p. 47). One year after this controversy, however, computer science gegan to become institutionalized as an academic discipline (Gupta, 2007, p. 49). Moreover, computer science was recognized as part of mathematics (Mahoney, 2011, p. 186). Most importantly for the present inquiry, however, was that the institutionalization of the field helped the development of different areas of the discipline. One of these would be computational complexity, as will be seen in chapter 9.

---

[67] Zahde was born in Azerbaijan, moving to Iran with his parents when he was 10. He came from a very privileged background, which let him use his leisure time in other activities to cultivate himself. Zahde believed in pure research; he never thought to apply his knowledge to industry because Zahde had the money to sustain himself (Perry, 1995, p. 32). Zahde graduated in electrical engineering at the University of Teheran in 1942. After that, he emigrated to the USA to pursue his dream of doing research in science. Zahde obtained his Ph.D. at Columbia University in 1949. Zahde became a professor at Columbia one year later (Perry, 1995, pp. 32-34). The most important contribution Zahde made to science was his theory of fuzzy logic (Perry, 1995, pp. 34-35).

[68] Slamecka was born in Brno, Czech Republic, in 1928. He graduated in chemical engineering in 1949 at the Brno University of Technology. Later, from 1952 to 1954, Slamecka studied physical sciences at the University of Sydney and sociology in 1955 at the University of Munich. Finally, he earned his Ph.D. in library science at Columbia University in 1962. Slamecka was the director of the School of Information and Computer Science at the Georgia Institute of Technology. See cc.gatech.edu/fac/Vladimir.Slamecka/cv.html for more information.

Conclusions

This chapter has considered three important milestones in the emerging field of computer science: Post Second World War science policy in the USA, computers as tools for surveillance and control to guarantee the U.S. postwar hegemony, and the beginnings of computing as an academic discipline during the 1960s. Combining these three elements broadens our understanding of the origins of computer science. The description of the change of the identity of computing during each decade is important in order to grasp why computers first focused on supporting military endeavors in the 1950s, and changed its focus to academic research on the following decade.

The U.S. government played an essential role in promoting the development of computer science for two reasons. First, computing technology would guarantee the hegemony of the USA during the Cold War. Second, the U.S. government saw computers as tools for protecting the USA from foreign military attacks; in particular preventing a military confrontation with other nuclear powers such as the Soviet Union and the People's Republic of China.

The U.S. government also supported academia by giving funding to universities for buying computers. The greatest beneficiary of the help the government gave to the universities was private sector. U.S. Universities bought computers from IBM to support their activities, such as modeling climate. The lack of computing knowledge of academics and researchers was the primary motivator for creating a computer science curriculum. Despite difficulties and debates, the final result was the complete institutionalization of computer science in the late 1960s. Did the slow institutionalization of computer science in comparison with AI bring about the formalization of computational intractability in computer science? Please read chapter 9 for a detailed answer.

Chapter 3
Cybernetics

Before, we can, in chapter 4 provide a historical account of the academic careers of Herbert Simon and Allen Newell in relation to their development of the first intelligent program: “the Logic theorist” (LT), it is necessary to understand the relationship between cybernetics and another of Simon’s theory of human-decision making during the time when he was working at Graduate School of Industrial Administration at Carnegie Institute of Technology (GSIA). This is what this chapter is about. Did cybernetics influence the development of the first intelligent program, built by Herbert Simon, Allen Newell, and J. C. Shaw, known as “the Logic theorist” (LT)? Indeed, such a relationship exists. Why is it important to consider this relationship? Because scholarship on the history of cybernetics does not touch the events of the First World War and their connection with later development of human decision-making during the Cold War. This chapter will focus on explaining cybernetics evolution from its beginnings, starting with the problem of aircraft detection during the First World War, and ending with cybernetics influence in Simon’s theory of human decision-making.

Most authors studying the link between AI and cybernetics focus their attention on the context of the Cold War. However, the connection between developing technology for detecting enemy aircraft during the First World War with human decision-making is barely studied.

Ronald Kline’s *The cybernetics moment or why we call our age the information age* (2015) focuses on the development of cybernetics from information theory until today. Kline takes the Second World War as starting point for explaining the influence of information theory on cybernetics, and then studies the cybernetics development during the Cold War and beyond. Furthermore, several chapters of Kline’s present the influence of on other sciences. What is missing in Kline’s narrative is the relationship between the events of the First World War and the posterior developments in human decision-making. Again, Jon Agar’s *Science in* *the twentieth century and beyond* (2012) touch on some elements not found in Kline's narrative, because Agar considers the relationship between radar development and the influence of that technology on information theory. Agar's narrative uses the concept of working worlds. Instead of using context or metaphors, Agar explains that science solves certain types of problems contained in working worlds. They generate applied technology or a scientific theory for explaining certain phenomena (Agar, 2012, pp. 3-4). In that spirit, the radar is part of the working world of the Second World War. Agar explains the development of the radar briefly from the problem of aircraft detection in the context of the First and Second World Wars. After Agar explains the connection between the working world of the radar with two World Wars, he mentions that two other fields derive from the radar. One of those is cybernetics (Agar, 2012, pp. 268-273). However, Agar's book does not consider the connection between radar development and Simon's theory of human decision-making. Building upon the accounts of Agar and Kline, this chapter aims to contribute to understanding the link between the problem of aircraft detection derived from the First World War to the problem of information handling and decision-making in the Second World War and the Cold War.

Four parts divide this chapter. The first is a brief introduction to the problem of the radar and the problem of decision-making for detecting enemy aircraft. The second section presents the development of information theory by Norbert Wiener and Claude Shannon. The third shows different academic and professional events related to discussions about the emerging field of cybernetics. Finally, the last part describes the relationship between cybernetics and Simon's theory of human decision-making.

3.1 Beginnings: the invention of the radar

During the Second World War, Great Britain and other governments saw the importance of developing reliable technology to transmit information for decision-making to prevent attacks from any foreign power. The experience of developing a system that reported information during the bombardment of London by Zeppelin airships during the First World War served as inspiration for creating a tool that could counteract any external intervention before any assault. The result of such a need was the creation of the radar between 1936-1937 (Agar, 2012, pp. 268-269).[69]
Before the invention of the radar, mechanisms for transmitting information to prevent an enemy attack used other types of signals as evidence. A good example of an alarm system was using sound waves generated by parabolic mirrors located on the cliffs of the southern coast of England to inform whether enemy aircraft was approaching (Agar, 2012, p. 268). Unfortunately, the accuracy of early warning systems techniques was not as precise as expected. During a military exercise in the summer of 1934, the British Royal Air Force bombers could attack several targets without being detected. For that reason, governments invested resources in developing science and technology to provide accurate information for defending their territory. Radio wave technology detection seemed to be the solution for detecting enemy forces (Agar, 2012, pp. 268-269).

Between 1934-1935, the physicist Robert Alexander Watson-Watt (1892-1973) carried out some experiments for detecting radio waves. In February 1935, Watson-Watt supervised an aircraft test to find whether technology could detect radio waves. The test was successful because the aircraft flying above the BBC radio transmitter at Daventry could detect flying machines in a radius of eight miles (Agar, 2012, p. 269). Watson-Watt found a technology to detect radio waves efficiently to prevent an attack (Agar 2012, p. 269).[70] Moreover, his invention

---

[69] The original name of the radar was “radio direction finding”. The term “radar” was coined by the U.S Army in 1940 (Agar, 2012, p. 268).

[70] In 1934, the Air Ministry of the United Kingdom offered 1000 pounds to the person who could create the technology supporting a death ray for killing a sheep at 200 yards. Watson-Watt explained to the Air Ministry's scientist H. E. Wimperis the impossibility of building a weapon with the characteristics the government wanted. However, he explained that

centralized information for supporting decision-making in war. The best example of the efficiency of Watson-Watt's invention was using radar technology for winning the Battle of Britain in 1940 against Nazi Germany (Agar, 2012, pp. 269-270). Watson-Watt was knighted in 1942 because King George VI considered the radar Watson-Watt had invented to be crucial for defeating the German air force in 1940 (Lotha, 2021b). The new possibility to receive and transmit information efficiently inspired the development of a new field of knowledge, namely the discipline that soon became known as “cybernetics”.

3.2 Norbert Wiener and Claude Shannon's information theory

Cybernetics, or the theory of communication and control, thus is the child of the radar. Antiaircraft shooting had a problem with the sequence fire control systems should follow. Moreover, most of those phases were controlled entirely by humans. Mathematicians saw the importance of synchronizing feedback systems and human action to reduce errors (Agar, 2012, p. 275; Thomas et al., 2008, p. 265). Communication theory was born from that need. Several institutions collaborated in developing communication theory, such as AT&T's Bell System (Agar, 2012, p. 275).

Furthermore, a newly created U.S. agency, the National Defense Research Committee, supported scientific research during the Second World War. The result was the support for developing communication theory by the U.S. Government (Thomas et al., 2008, p. 262). Both public and private sectors worked separately in theories of information derived from communication theory. In that context, two researchers developed a new concept that explains the amount of information computers could process from communications systems (Kline, 2015, p. 10). Those researchers were the mathematicians Norbert Wiener and Claude Shannon. Wiener was born in Columbia, Missouri, in 1894. His father, Leo, a professor of slavonic languages and literature at Harvard University, focused on developing his

---

building radio wave detectors for detecting aircraft was possible (Agar, 2012, p. 269).

child's intellectual talent from a very early age. The effort of his father became visible already when Wiener graduated from Tufts University at the age of 15. He continued studies at Harvard University, obtaining a Ph.D. in philosophy in 1913. Wiener's doctoral thesis focused on mathematical logic (Masani, 1990, pp. 29-35; Masani, 1990, p. 39; Masani, 1990, pp. 43-44). Wiener's intelligence helped him to advance in the research world, receiving a grant from Harvard University to study mathematical logic at Cambridge University under the supervision of Bertrand Russell. Also, during the time Winner stayed in Europe, he went to study at Göttingen (Masani, 1990, pp. 44-56).[71] Wiener used the knowledge he acquired during the two world wars to develop the theory of cybernetics because Postwar mathematical logic was fundamental in the development of cybernetics.[72]

For Wiener, statistics bridges machine and natural world. Wiener’s *Extrapolation, interpolation, and smoothing of stationary times* (1949) explained the philosophy behind the idea he had presented in his *Cybernetics* (1948) (Wiener, 1970, p. v). Wiener's 1949 work described his beginning in communication theory, mentioning the connection he found between control and communication theory in machines and animals using statistics (Wiener, 1970, p. v). Wiener reached that conclusion after working with the mathematician Julian H. Bigelow (1913-2003) on a war project related to the research of maneuvers aircraft must do in the context of war (Wiener, 1948, pp. 12-16).[73] Moreover, with the help of the neurophysiologist Arturo Rosenblueth (1900-1970), Bigelow and Wiener compared information transmission in mechanical and electric systems with human nervous system (Wiener, 1948, p. 16).[74] Statistics proved to be helpful for predicting messages (Wiener, 1948, pp. 16-17). It was the cooperation between Wiener and

---

[71] Wiener chose to study with Bertrand Russell because Wiener had the certainty that he would learn the latest mathematical logic (Kennedy, 1980, pp. 134-135; Wiener, 1948, p. 20-21).

[72] Another researcher that contributed to the development of cybernetics was Walter Pitts. The logician Rudolph Carnap influenced Pitts.

[73] Bigelow was born in 1913 in Nutley, New Jersey. He went to study at the Massachusetts Institute of Technology when he was 17 years old, graduating in 1936 with a master’s degree in electrical engineering. Bigelow was of essential importance in the development of computing technology because he was responsible for implementing von Neumann's ideas for building a computer, being a chief engineer in the IAS Electronic Computer Project in 1946 (Dyson, 2013).

Rosenblueth that, in 1947, led to naming “cybernetics” as the entire fields of communication and control theory (Wiener, 1948, p. 19). Cybernetic information theory defined information as negative entropy. In other words, his concept of information sought to explain the amount of order in a message because less information is transmitted through a medium, such as a telegraphic cable, when the selection of messages is more disordered. Wiener's concept sought to explain how much information was ordered in a message (Wiener, 1948, p. 18; Kline, 2015, p. 13-15). His view of information theory was utterly different from Shannon’s.

Shannon was born in 1916 in Petoskey, Michigan. He obtained a bachelor's degree in mathematics and electrical engineering at the University of Michigan in 1936. The same year, MIT hired Shannon as a research assistant (Markowsky, 2021). Later, in 1940 he earned a master’s degree in electrical engineering and a Ph.D. in mathematics. One year later, Shannon joined Bell Labs and remained a member until 1972 (Markowski, 2021). Shannon's work at Bell was significant to the military, focusing his attention on developing antiaircraft missile control systems.

An essential contribution Shannon made during his time at Bell Labs was creating a theory of information published as “A mathematical theory of communication” (1948) (Markowski, 2021). Shannon's paper began by explaining the importance of developing a communication theory. He mentioned that a communication theory served for reconstructing messages close to the originals sent through a medium like a telegraphic network. In that spirit, Shannon said that a monotonic logarithmic function could measure the information sent (Shannon, 2001, p. 3). Shannon's formula measured the entropy of a message (Kline, 2015, p. 10). His view of entropy differed from Wiener's because Shannon's entropy was positive, meaning that his concept of information sought to explain the amount of disorder

---

[74] Rosenblueth was born in 1900 in Guerrero, Mexico. He graduated in 1927 as a medical doctor at the Sorbonne (Quintanilla, 2002, pp. 304-305). After a time, Rosenblueth received a scholarship from the Guggenheim Foundation to study at Harvard. He was supervised by neurologist Walter Bradford Cannon, a pioneer in the use of X-rays for studying soft tissue (Quintanilla, 2002, pp. 304-305). During his time at Harvard, Rosenblueth met Wiener. Unfortunately, Rosenblueth had to return to Mexico because the Second World War made the USA government limit the participation of foreigners in science (Quintanilla, 2002, pp. 304-305). Still, Wiener and Rosenblueth remained contact.

of information in messages (Kline, 2015, p. 15). Moreover, Shannon's proposal was superior to Wiener's because Shannon's formula considered the capacity of a channel in terms of *bits* to transmit a message (Kline, 2015, pp. 15-17). Wiener and Shannon's ideas were a byproduct of a set of conferences sponsored by the Josiah Macy Jr. Foundation.[75]

3.3 Meetings and conferences

Researchers in the emerging field of cybernetics soon found institutions that provided a space for discussing ideas from different areas that contributed to institutionalizing the discipline (Kline, 2015, p. 39). The Macy Foundation was one of those institutions. Before cybernetics existed, the Macy Foundation organized meetings linked to protocybernetics.[76] In the first meeting of 1942, scientists such as neurophysiologist Warren McCulloch attended the conference to discuss a possible connection between control theory and neurophysiology (Wiener, 1948, p. 19). Later, between 1943 and 1944, the University of Princeton held a joint meeting that brought together researchers interested in different disciplines that were going to be part of the emerging field of cybernetics (Wiener, 1948, p. 23). Engineers, physiologists, and mathematicians were part of the gathering to discuss how those fields of knowledge could integrate. Researchers such as mathematician Herman Heine Goldstine (1913-2004) attended the

---

[75] The Foundation was founded in 1930 by Kate Macy Ladd (1863-1945). The institution was named after the memory of Macy's father, Josiah Macy Jr. The main goal of the Foundation now-a-days is the promotion of health. See https://macyfoundation.org/about/who-we-are for more information.

[76] Protocybernetics is a term linked to concepts developed before the institutionalization of “cybernetics” in 1946. In kuhnian terms, the preparadigmatic phase of “cybernetics”.

workshop.[77] The meeting was successful for all attendees because they concluded that their fields could integrate (Wiener, 1948, p. 23).

In 1945, Wiener accepted the invitation of the Mexican Mathematical Institute to participate in a conference. He traveled to Mexico in June and stayed there for ten weeks (Wiener, 1948, p. 25). During his time in Mexico, Wiener worked with Rosenblueth at the Instituto Nacional de Cardiología. He also attended the conference sponsored by the Mexican Mathematical Society (Wiener, 1948, pp. 25-26). At the Instituto, Wiener and Rosenblueth worked on neurophysiology, finding some interesting results like phasic contractions in epilepsy and heart fibrillation. Those results served to study conductivity and latency in uniform conducting media of two or more dimensions and the statistical study of conducting properties of random nets of conducting fibers (Wiener, 1948, p. 25). Some of those results would become presented at the conference of the Mexican Mathematical Society held in Guadalajara. One year later, the Macy Foundation started organizing a group of conferences that gave birth to cybernetics.[78] Cybernetics influenced the development of other fields, such as AI.

3.4 Cybernetics and artificial intelligence

During the Cold War, social scientists saw cybernetics as a field promised to explain many different phenomena. Cybernetics strongly impacted different fields, including economics (Kline, 2015, p. 136). Influential economists such as Kenneth Arrow (1921-2017) and Oskar Morgenstern (1902-1977) claimed that

---

[77] Goldstine was born in Chicago, Illinois, in 1913. He obtained his Ph.D. in mathematics at the University of Chicago in 1936, and then became an assistant professor at the University of Michigan in 1941. Furthermore, in the same year he entered the U.S. Army. He remained in the military until 1945. During that time, he participated in developing the first electronic computer to automate ballistic tables calculation for the U.S. Army, the ENIAC (Electronic Numerical Integrator and Computer).
See https://history.computer.org/pioneers/goldstine.html for more about Goldstine.

[78] Unfortunately, there is too little information for reconstructing the history of the conferences supported by the Macy Foundation in the development of cybernetics.

mathematical models could explain economic behavior. The perhaps most outstanding figure of all was, however, Herbert Simon. He proposed using mathematical models to connect different social sciences (Kline, 2015, pp. 136-137). Cybernetics is linked to Simon's early formulation of human decision-making.

During the time Simon was part of GSIA, Carnegie Institute of Technology was responsible for developing a mathematical model for allocating resources efficiently (Erickson et al., 2013, p. 72). In 1948, the Soviet Union blocked the entrance of supplies to West Berlin.[79] The Western Superpowers found a solution to tha problem, deciding to provide supplies through air to West Berlin (Erickson et al., 2013, pp. 53-54). Operation Vittles was born in that context. One of the projects part of the project was the Scientific Computation of Optimum Programs (Project SCOOP). The goal of Project SCOOP was to create a mathematical model for mechanizing the planning processes through algorithms. In 1949, Project SCOOP and the Bureau of the Budget's Division of Statistical Standards gave GSIA the responsibility of developing a mathematical model for finding the best strategy to deliver the resources needed in a certain number of days to West Berlin (Erickson et al., 2013, p. 72).

In 1947, George Dantzig (1914-2005) developed a technique that would be the core of Project SCOOP.[80] He created the so-called “simplex algorithm” for solving linear programming problems. Danzig algorithm's goal was to maximize the output of a linear objective function by making sense of the relations between resource items and production activities (Erickson et al., 2013, p. 61). Unfortunately, the simplex algorithm could not work very well with computing capabilities of the 1940s. The algorithm could not adjust to computers of the time

---

[79] The USA, Great Britain, and France controlled the western part of Berlin during the Cold War (Erickson et al., 2013, p. 53).

[80] Dantzig was born in the city of Portland in the USA. He did his undergraduate studies at the University of Maryland, earning his bachelor’s degree in physics and mathematics in 1936. One year later, he obtained a master in mathematics at the University of Michigan. In 1946, Dantzig graduated with a Ph.D. in mathematics at the University of California, Berkeley. The same year, he went to Washington (D.C.) to work at the U.S. Department of Defence (Hosch, 2020d).

because the matrix size could not be computed in a reasonable amount of time. The matrix was too big, so it took significant time to do calculations (Erickson et al., 2013, p. 64). The problem of the simplex algorithm can be seen as a contingent form of computational intractability. It is here where Simon entered the picture. Together, with the economist Charles Holt (1921-2010), he developed an alternative model.[81] Holt and Simon's method adapted to the circumstances of a particular environment. Servomechanisms inspired their model (Erickson et al., 2013, p. 73).[82]

Inspired by the work of Wiener in cybernetics, and his experience working in the theory of decision-making such as the International City Manager's Association (ICMA), Simon used a servomechanism for minimizing manufacturing costs of production. Furthermore, Simon's model helped to find the optimal amount of products in an inventory based on customer orders (Kline, 2015, p. 146). Servomechanisms inspired Simon to develop his theory of human decision-making because servomechanisms were useful for making sense of the relationship between decision-making and control systems by adapting decisions to the conditions of the environment (Kline, 2015, pp. 145-147).

Using a servomotor, Simon explained how humans make decisions. Chess is the basis to Simon's LT because the game explains how humans make decisions by adapting to the circumstances of the environment.[83] In other words, cybernetics provided a link between games used for modeling human rationality and Simon's theory of human decision making (Kline, 2015, pp. 145-147). His theory of human decision-making would become the core of the first working artificial intelligence program: the "Logic Theorist" (LT) (Newell et al., 1959b, pp. 5-6). The influence of cybernetics on the first working artificial intelligence program can be traced in Simon's theory of human decision-making. Chapter 4 will deal

---

[81] Holt was a professor in the faculty of business at the University of Texas at Austin. He made important contributions to data analysis, especially in exponential smoothing. See https://academic.microsoft.com/author/2136581708/publication/search?q=Charles%20%20Holt&qe=Composite(AA.AuId%253D2136581708)&f=&orderBy=0&paperId=2085866051 for more information.

[82] According to Young (2013) "[A servomechanism is an] automatic device used to correct the performance of a mechanism by means of an error-sensing feedback".

[83] See Ensmenger (2011).

with the connection between Simon’s theory of human-decision making and the LT.

Conclusions

This chapter had explained the influence of the First and Second World Wars on the development of the radar and, moreover, how this ultimately led to Simon's theory of human decision-making. Even if Simon's theory of human decision-making was really only conceptualized during the Cold War, the problem of human decision-making can ultimately be traced back to aircraft detection during two World Wars. A child of the First and Second World Wars, cybernetics was going to shape the development of information theory and AI.

The leading figure of cybernetics, Wiener, saw the possibility of creating a new field of knowledge after analyzing the analogy between control and communication theory with machine and animal communication. With his idea, Wiener elaborated a metaphor between the human nervous system with mechanical and electric systems.

As I have also shown, cybernetics was also massively promoted at conferences organized by the Macy Foundation. The field resulted from a dialogue between academics and professionals working in different areas of knowledge. Engineers, physiologists, and mathematicians joined together to develop cybernetics. Breaking the disciplinary boundary was necessary for bringing cybernetics to this world.

Cybernetics influenced several areas of science during the Cold War, such as economics and, thereby, Herbert Simon. His theory of human decion-making is a mixture between cybernetics and his previous experience in other institutions. Which is the institutional connection that helped to link cybernetics with Simon's theory of human decision-making? Chapter 4 has the answer to that question. This chapter connects with chapter 4 because cybernetics inspired Simon in developing his theory of human decision-making using a servomotor.

Part II

Simon, Newell, and the emergence of artificial intelligence

Chapter 4

The Logic theory program

This chapter aims to explain the development of the famous "Logic theorist" (LT) from a historical point of view. The LT, programmed by Herbert Simon, John C. Shaw, and Allen Newell, was the first intelligent computer program; it could prove thirty-eight theorems of mathematical logic from Bertrand Russell and Alfred North Whitehead's fundamental contribution to modern logic, the *Principia mathematica* (1910-1913). How is the LT related to the problem of computational intractability? Chapters 8 and 9 will explain why studies on AI (AI) and computational complexity were developed in different periods of time. Moreover, the problem of computational intractability is part of computational complexity studies. The LT is essential for understanding why computational intractability was formalized outside AI.

Reconstructing Simon and Newell's biographies is important for connecting the LT development with their experiences because they shaped LT heuristics. Simon's life has been treated extensively by the literature. Simon's *Models of my life* (1991) or Augier and March's *Models of man: Essays in memory of Herbert Simon* (2004) are works that treat the LT development through the lens of Simon's life. Unfortunately, both books do not treat the dispute between Simon with philosopher Hao Wang. Why is it essential to understand the disagreement between both researchers? Because the dispute has relevance for understanding why the discovery of computational intractability occurred in mathematical logic and computer science rather than in artificial intelligence research. This chapter is linked with chapter 3 because section 3.4 of that chapter explains how cybernetics is related to Simon's theory of human decision-making.

Six sections divide chapter 4. The first will treat Simon's early life, university studies, and the development of his doctoral thesis in political science. After finishing his Ph.D., Simon returned to Chicago to work at Illinois Institute of Technology (IIT), as the second section of this chapter will describe. The third will show a Simon with higher academic and professional reputation, up to being a member of the group of people that founded the Graduate School of Industrial Administration at the Carnegie Institute of Technology (GSIA). Furthermore, this section will show Simon's critique to traditional economic theory. Simon met Newell when he was a contractor at RAND as section four of this chapter will describe. The fifth will show Simon and Newell's plan to develop an intelligence computer program. Finally, the section of this chapter will explain how Newell, Shaw, and Simon programmed the LT.

4.1 Early life (1916-1942)

Herbert Alexander Simon was born on June 15, 1916, in the city of Milwaukee in the USA. His parents were Arthur Simon, a German electrical engineer, and the American Edna Margarite Merkel (Simon, 1996, pp. 3-4; Augier & March, 2004, p. 6). Simon studied at the University of Chicago from 1933 to 1936, becoming interested in studying mathematics, logic, and economics with an emphasis on the applications to the social sciences, especially political science (Simon, 1996, pp. 36-64). After Simon finished his bachelor's degree, he won a scholarship in 1938 to work at the International City Manager's Association (ICMA) as co-editor of a bulletin and statistical *Municipal year book*. He was responsible for writing the formation manuals for the municipal employees of the institution (Simon, 2006, p. XXIII).[84]

During his time at ICMA, Simon studied different theories of administration, concluding that the methodological tools that could help him advance own his research were simply nonexistent (Augier & March, 2004, p. 9). Simon's work caught the attention of the director of the Bureau of Public Administration at the University of California in Berkeley, Samuel May. He invited Simon to join a

---

[84] Simon met his future boss at ICMA, Clarence Ridley, while doing his bachelor's degree. Ridley was Simon's professor in 1936 a course on "Measuring municipal governments". On that subject, Simon got an invitation to join Ridley's research team, ending in the publication *Measuring municipal activities* written in 1937 (Simon, 1996, p. 64).

project to study the local government (Simon, 1996, pp. 75-76).[85] Simon accepted May's proposal, moving to Berkeley in 1939 to work for three years as director of Administrative Measurement Studies (Simon, 1996, pp. 78-79; Augier & March, 2004, p. 9).

Simon and his colleagues were very conscious of the limits for doing their research because the local government used flaky data from previous studies to take decisions. This problem was Simon's starting point to develop his research. During that time, Simon wrote several journal articles and books. Till, the most important of them were *Determining work loads for professional staff in a public welfare agency* (1941), *Fire losses and fire risks* (1943), and *Fiscal aspects of metropolitan consolidation* (1943). The first work tried to determine the value of the effectiveness of social workers' caseloads for an agency's operation.[86] The second work involved maps of San Francisco Bay Area, and its relation between construction building and fire losses. The last study was a theoretical analysis that focused on the relationship between urban property taxes with the pattern of municipal revenues and services in the San Francisco Metropolitan Area (Simon, 1996, pp. 82-83). Simon published his "The incidence of a tax on urban real property" (1943) during the last year of his studies. Importantly, his work considered the limits of neoclassical economics (Simon, 1996, p. 83). Simon found that neoclassical economics does not consider externalities human action produced. For example, humans make decisions when the rate of taxes changes. This idea would play an important role in relating human bounded rationality and decision-making (Simon, 1996, p. 83).

The experience was beneficial for completing his Ph.D. in political science in 1942. Simon wrote his entire thesis during his time at Berkeley. His thesis was presented at the University of Chicago and published as *Administrative Behavior*

---

[85] Simon received a grant from the Rockefeller Foundation in 1938 to work with May in Berkeley on a proposal to the Foundation to get support for continuing the study of local government (Simon, 1996, p. 75). The answer to their proposal came in 1939: the Foundation gave a three-year grant to the project. Simon was appointed director of the studies (Simon, 1996, p. 76).

[86] During the time of his first work, Simon became acquainted with using data manipulators (Simon, 1996, p. 82). The interaction with those machines helped him later to develop his ideas about human information processing.

(1947) (Simon, 2006, p. XXIII). Simon's Ph.D. was the basis for his future theory of bounded rationality (Augier & March., 2004, pp. 9-10). After completing his thesis, Simon accepted a job offer from IIT to apply the theoretical explanations he had developed during his academic training (Simon, 1996, p. 93; Augier & March, 2004, p. 11).

4.2 Life in Chicago (1942-1946)

Simon was very enthusiastic about the new position. The environment of the engineering school was better than any other type of university for a political scientist interested in mathematics. He could experiment with new ideas, such as teaching engineers instead of political scientists (Simon, 1996, p. 93). For example, in 1942, Simon taught the course of constitutional law to engineering students, obtaining excellent results (Simon, 1996, p. 95).[87] Working on diverse subjects helped him to understand the relations between different areas of knowledge.[88] Simon and his family decided to buy a home close to the campus of the University of Chicago, where some of his friends worked. One of them was Bill Cooper (1914-2012). Cooper suggested that Simon organize with him a once-a-week seminar related to the latest developments in economics. Simon accepted Cooper's proposal, and they began to study economics. Later, their interest in economics connected them with the Cowles Commission, an important institution for economics research located at the University of Chicago from 1939 to 1955 (Cooper, 2004, p. 69; Christ, 1994, p. 30).

When Simon arrived at Cowles, he found himself intellectually stimulated by different researchers there, passionate about their subjects.[89] Moreover, Simon found a place for discussing topics he explored in economics. Thus, he could

---

[87] Simon gave a speech at the engineer’s senior banquet. The dean and the president of the faculty attended the event (Simon, 1996, p. 96).

[88] Apart from lecturing on constitutional law, Simon taught architects. Moreover, Simon and other colleges gave a seminar on the philosophy of science to IIT students (Simon, 1996, pp. 97-101).

express his opinion about comparative statistics and dynamics in Paul Samuelson's "The stability of equilibrium: Comparative statics and dynamics" (1941), or the identification problem, which was also discussed in the team.[90] As a result of exchanging ideas with his new collegues, Simon found that a formal concept of causal ordering could be constructed among variables of a system, and the concept could be defined just when the system was fully identified in propositional logic, such as causal relations between logical propositions (Simon, 1996, p. 102; Augier & March, 2004, p. 11; Copper, 2004, p. 70, Simon, 1952, pp. 520-521). Simon's conversations focused on shaping economic aspects of nuclear power (Augier & March, 2004, p. 12).[91] The experience he obtained by working between IIT and the Cowles Commission helped Simon to better understand human decision-making processes by combining the engineering culture of the first institution with economic culture of the second (Simon, 1996, pp. 103-107; Augier & March, 2004, p. 12). It was because Simon contact with experts in new fields of knowledge like operations research and management science, cybernetics, and statistical decision theory, he understood that the core of those disciplines was the theory of decision making (Simon, 1996, pp. 109-110). After all the great intellectual and practical experience at the Cowles Commission, Simon returned to

---

[89] The economists Jacob Marshak (1898-1977), Tjallin Koopmans (1910-1985), and Nobel prize winner Kenneth Arrow (1921-2017) were part of the Cowles Commission at the time (Simon, 1996, p. 101). The Nobel prize winner in economics, Milton Friedmann, gave some advice to the Commission, but he was an outsider in the group (Simon, 1996, p. 102).

[90] The identification problem was related to one of the issues that appeared with statistical data: the statistical ambiguities when trying to estimate supply and demand (Simon, 1996, p. 102).

[91] Simon entered the Cowles Commission before the end of the war (Simon, 1996, p. 101). During that time, the development of computer technology supported mathematical calculations of weapons of the U.S. Army (Edwards, 1996, pp. 49-51). This technological development also helped commission members who were actively developing their economic models. One of them was the mathematician George Dantzig (1914-2005), who developed the simplex method for allocating resources with the help of computer technology (Simon, 2006, pp. 102-103). The only role Simon had with the group was the use of linear programming techniques for investigating the impact of technological change on the economy (Simon, 1996, p. 103).

study administrative science, obtaining a chair at IIT in the political and social science department in 1946 (Simon, 1996, p. 109).

4.3 A new life (1948-1951)

After the Second World War, the USA planned a recovery project, well-known as the “Marshall Plan”, to prevent an economic and social catastrophe in Europe. Simon became involved in this project. In 1948, the Economic Cooperation Act (ECA) was created to implement the Marshall Plan. Herbert Simon became part of this program by invitation of Don Stone. Simon worked as a consultant and then as director of the Engineering Management Branch of ECA (Simon, 1996, p. 117). When Simon arrived at ECA, he found some problems in the institution’s program, but later ECA overcame them. Furthermore, the agency that Simon and his colleagues found when ECA began had a limited infrastructure because it just had a few desks, telephones, and a telephone diary. However, this limitation did not restrict the growth of this program: by July 26, 1948, there were 741 names linked to the organization (Simon, 1996, p. 117). Another obstacle was the lack of time for developing a proper organizational chart for ECA. That issue was solved by writing a document that described essential functions of the organization.[92] The document became named *Basic principles of ECA organization* (Simon, 1996, p. 118). Apart from collaborating with the commission, Simon got involved at the Carnegie Institute of Technology, first as a seminarist and later as part of the staff (Simon, 1996, pp. 135-136).

When Simon gave a seminar for economists at the Carnegie Institute of Technology in 1948, the institution received a gift of 5 million U.S. Dollars in endowment, plus 1 million for building GSIA. The money Carnegie received had the purpose of creating a school for educating a new type of executives to manage modern high-tech firms. Moreover, students were going to take classes related to

---

[92] The document included the mission of ECA (Simon, 1996, p. 118).

managerial skills and knowledge in science and technology.[93] In that spirit, GSIA needed a program that could adjust to new exigencies of the Carnegie Institute of Technology. Simon discussed his plan of developing that program with the provost Elliot Dunlap Smith (1891-1976) and the chairman of the Economics Department, George Leeland Bach (1915-1994). In 1949, Simon was invited to join the university as a professor of administration in the faculty of economics, and chairman of the Department of Industrial Management (Simon, 1996, pp. 135-136; Augier & March, 2004, p. 14; Cooper, 2004, pp. 70-71).[94]

During the first years of GSIA, four founding members of the institution(George Leeland Bach, Elliot Dunlap Smith, Bill Cooper, and Herbert Simon), developed GSIA's program.[95] Bach, Cooper, and Smith decided to make some changes in the undergraduate industrial program curriculum to adapt GSIA's necessities. Moreover, Bach considered creating a doctoral program in economics.[96] Even though their idea was viewed as excellent, it suffered several problems (Simon, 1996, p. 138). The first concerned professional qualifications. Except for Dunlap Smith, non had any expertise of management or business education. Bach, Cooper, and Simon came from the social sciences. Fortunately, they were living in a time when disciplines related to solving management problems with the use of quantitative tools for problem-solving and decision-making were in their infant state (Simon, 1996, p. 139).[97] The fields of knowledge related to behavioral sciences were flourishing because; the literature on problem-solving and decision-making in management was booming (Simon, 1996, p. 139).[98] The combination of

---

[93] In this school, Simon matured his decision-making theory using a servomotor. GSIA received funding for researching optimization methods. Please refer to section four of chapter three for more information.

[94] Simon left the IIT in 1949 (Simon, 1996, p. 112).

[95] Bach was appointed as dean of the GSIA. Simon and Cooper helped to develop the faculty and curriculum of the program. Dunlap Smith supported group members (Simon, 1996, p. 138).

[96] Bach was dean of the GSIA from 1949 to 1961 (Simon, 1996, p. 148).

[97] One of those tools is the simplex algorithm for solving linear programming models (Erickson et al., 2013, p. 61).

[98] Apart from Simon's *Administrative behavior*, Chester Barnard's *The functions of the executive* (1938) studied people's behavior inside organizations (Simon, 1996, p. 129;

management and behavioral science helped to develop a new science GSIA required to manage many challenges of postwar USA (Simon, 1996, p. 139).

The group's second problem during the first years of GSIA was their relationship with economists they hired. Simon believed that economists would like to work with accounting data from factory managers to study the process of decision-making (Simon, 1996, p. 143).[99] At the same time, he found that the economics he had developed during his work was not compatible with mainstream economics (Simon, 1996, p. 144; Augier & March, 2004, p. 17). Simon's view was not purely economic as the neoclassical school but included organizational psychology and sociology for understanding decision-making. Simon's approach to economics caused tension with economists hired by GSIA, who among other things still followed the idea of perfect information (Simon, 1996, p. 144; Augier & March, 2004, pp. 17-18). According to neoclassical economics, perfect rationality exists when economic agents make decisions. In other words, the information the agent uses for decision-making maximizes their utility function, helping to achieve the desired goal. Simon’s human decision-making theory rejects the neoclassical school of thought because the agent has achieved its purpose when it reaches a satisfying solution. Simon reached that conclusion after seeing that an agent can’t make all the necessary calculations for attaining what the agent wants because they are too complex. (Simon, 1996, p. 144; Russell & Norvig, 2010, p. 1049). Economists complained about Simon's ideas and accused him of trying to stop the development of the discipline in the faculty. To calm the nerves of the economists, Simon wrote a memorandum explaining that the consolidation of GSIA as an institution caused some stress on the staff, explaining he would rebuild the morale of economists with different viewpoints (Simon, 1996, pp. 144-146). Once the storm passed, the institution became a referent in business education, with Simon having convinced its members to adopt his new ideas in economics (Simon, 1996, p. 154; Augier & March, 2004, p. 18).

Barnard, 1968, pp. 3-7).

[99] The economists working at GSIA believed that their research would focus on business instead of economics (Simon, 1996, p. 143).

4.4 Simon meets Newell

Allen Newell was born in San Francisco in 1927. His father, Robert R. Newell, was a professor of Radiology at Stanford University, and his mother was Jeanette Le Valley Newell (Simon, 1997b, p. 143). His scientific interest came relatively early. After high school, he enlisted in the U.S. Navy to serve on a ship that carried out two goals: scientific observation of Bikini nuclear tests and making maps of radiation distribution over the atolls from those tests. Working for the U.S. military impacted him severely to the point he decided to pursue a career in science. After his work in the U.S. Navy, Newell enrolled at Stanford University, earning a degree in physics in 1949 (Simon, 1997b, p. 144).

The same year he finished his undergraduate education, Newell started his graduate studies in mathematics at Princeton University. He retired one year later because he found Princeton's method of teaching mathematics unattractive. After studying game theory, he had realized he wanted to do mathematics by combining theory and practice. Newell dropped his master's studies, and obtained a position at RAND Corporation in 1950 (Simon, 1997b, p. 144; Dick, 2015, p. 625).[100]

At RAND, Newell's conception of mathematics began to change. His first paper was written with mathematician Joseph B. Kruskal (1928-2010) in 1950 under the title "A model for organization theory" (Simon, 1997b, pp. 144-145).[101] Some time after, Newell's work was published, he changed his mind because he saw that applying formal methods to understand complex phenomena did not really explain how the world works. Newell reached that conclusion during a six-week field visit to the Munitions Board in the city of Washington. He developed a different view after his visit to the institution (Simon, 1997b, 145). Disappointed with how

---

[100] The RAND Corporation was founded in 1946 as an independent non-profit organization to promote science and education for the security and well-being of the population living in the USA. This institution, based in Santa Monica, California, was established with the support of the U.S. Air Force, the patronage of the firm Douglas Aircraft, and several universities. The institution invited numerous persons to collaborate. One of them was also Simon (Simon, 2006, p. XXXV; Erickson et al., 2013, p. 14).

[101] Kruskal and Newell's paper explored a new way of studying organizations and their properties. Their work tried to use the axiomatic method for understanding how organizations work (Kruskal & Newell, 1950, pp. 1-4).

complex phenomena were treated, Newell thought the best approach was completely different from axiomatization. In that spirit, he continued his work in the laboratory to test his ideas. Newell developed this new phase of his career by designing and doing experiments with small groups in the area of decision-making (Simon, 1997b, p. 145).

Newell searched for a team to learn new techniques to fulfill his interest in formulating mathematical theory. In 1952, Newell joined the team of John L. Kennedy (1913-1984), Bob Chapman, and William Biel to develop a full-scale simulation of an Air Force early warning station for studying organizational processes. Their work resulted in creating the *Systems Research Laboratory* in the same year. This newly created station was an essential element in Newell's development of his own decision-making theory because he saw how the crew interacted with radar screens, interception of aircraft, and between the crew. With the data of his work in hand, Newell could figure out how the crew treated radar information and how they actually made decisions (Simon, 1997b, p. 145). It was in that department that Newell began to work with Simon while the latter was also consultant for RAND (Augier & March, 2004, pp. 18-19; Cooper, 2004, p. 73).[102]

Simon and Newell soon found common ground. They knew each other’s works referring to organizational processes. Their discussion related to the idea the human mind could be a system of symbol-manipulation, or it could process information, showed their capacity to relate to each other in a good way. Unfortunately for them, the work they were doing to understand how air controllers and radar operators make decisions failed (Augier & March, 2004, p. 19).[103] Their expectations were not met during the time they did their experiments because intellectual tools for analyzing the phenomena they were studying were

---

[102] Apart from his contacts with the Cowles Commission, tied to RAND, the founders of the Air Force Early Warning Station came to visit Simon for advice because of his experience in managing different organizations (Simon, 1996, p. 168). RAND hired Simon as a consultant after the meeting in 1952 (Simon, 1996, p. 164).

[103] During that time, Newell was responsible for finding a new technique to simulate a radar display for air traffic. Unfortunately, the technology of that time was minimal for simulating the patterns of blips that move over radar screens (Simon, 1997b, p. 146).

insufficient (Simon, 1996, p. 168). AI developed against the background of that crisis.

4.5 The road to artificial intelligence

With the development of computer technology after the Second World War, calculations of different kinds became more manageable and precise and had began to outperform humans. One of the implementations in computers was programming a mathematical model for the hydrogen bomb (Edwards, 1996, p. 51). In this context, RAND decided to build a computer to support the staff in doing their work. This idea came into reality in 1954, when the institution's computer, the JOHNNIAC, was built (Edwards, 1996, pp. 121-122).[104] Simon and Newell had the experience of using computing machinery to assist the activities of the bureaucracy. For that reason, it was easy for Simon and Newell to use RAND's computer.[105]

Simon and Newell thought about simulating human problem-solving (Simon, 1997b, p. 147; Augier & March, 2004, pp. 18-19). In 1954, Newell became inspired by the possibility of creating adaptive systems, following a talk he had attended by mathematician Oliver Selfridge (1926-2008). Selfridge shown how a

---

[104] The JOHNNIAC was named after mathematician John von Neumann. This computer also was inspired by von Neumann's own model, developed when he was working at IAS. He was invited by the RAND staffer Fred Gruenberger (1918-1988) in late 1950 because von Neumann could help to build a machine that could provide new ways of computation; RAND's six IBM 604 punch-card calculators were not enough for the work RAND staff were carrying out (Edwards, 1996, pp. 121-122; Gruenberger, 1968, p. 2). See Burks et al. (1946) for more information.

[105] While Newell came in contact with computer culture by working at RAND, Simon already acquired some experience before entering the corporation. During his work at ICMA, Simon had the idea of mechanizing the statistical work he was doing for ICMA's *Municipal year book* of the institution. After Simon found an IBM punch card equipment at the University of Chicago bookstore, he taught himself how to use this device to simplify statistical calculations (Simon, 1996, p. 70).

computer program could learn to recognize characters and other patterns (Simon, 1997b, p. 147). The computer program Selfridge was developing at Lincoln laboratories related to pattern recognition and became called the "Pandemonium". That system could recognize letter forms and simple shapes. Even though the innovation was part of cybernetics, Newell took from Selfridge's work the concept of symbolic processing (Edwards, 1996, pp. 250-251). Newell's first try in this new world was to propose an imaginative design of a computer program that could emulate human beings capacity for playing chess. The proposal was presented in 1955 at the Western Joint Computer Conference under "The chess machine: an example of dealing with a complex task by adaptation" (Simon, 1997b, p. 147). In his work, Newell equated the problem of programming a chess player to translate texts or abstracting scientific articles by a computer because these types of tasks were ultra-complicated to solve (Newell, 1955, p. 101).[106] Newell's proposal was considered the first idea related to AI (Simon, 1996, p. 204).

After Newell published his work, Simon and Newell put all their effort into studying chess problems and their relation to human problem-solving. Newell invited the mathematician John Clifford Shaw (1922-1991) to help them (Simon, 1996, p. 202).[107] Once this ambitious project began, Newell was thinking of

---

[106] Newell proposed a solution to the capacity of computing problems, the speed of the computer, and the ability of humans to program a computing machine in order to deal with the difficulties computers had in making sense of the relevance of information, the number of potential solutions to a problem, the processing power, and to finding a good solution to the problem in an acceptable time (Newell, 1955, p. 101).

Several authors inspired Newell in developing his work. Building upon two academic articles by Selfridge and mathematician Gerald Dinneen (1924-2012), Newell showed in 1955 how computers could be non-numerical processors (Simon, 1996, p. 202). Selfridge's "Pattern recognition and modern computers" (1955) try to simulate the process of feature extraction humans do in a computer to do pattern recognition (Selfridge, 1955, p. 91). Dinneen's "Programming pattern recognition" (1955) had the same idea (Dinneen, 1955, p. 94).

[107] Shaw worked with Newell at Systems Research Laboratory. They were very ingenious in developing a program that could simulate radar maps for air defense. This endeavor showed that computers could do symbolic processing (Simon, 1996, p. 201). Shaw also played an important role when the JOHNNIAC was built because he constructed the programming system for this computer (Simon, 1996, p. 202).

writing a Ph.D., but he was not sure about pursuing that degree because he did not want to renounce his work at RAND (Simon, 1996, pp. 202-203).[108] Newell and Simon began to work in the implementation of programs related to problem-solving.

4.6 The logic theorist is born

Conversations between Simon and Newell focused on the problem of chess and human problem-solving. However, they were also interested in other topics of study linked to the human problem-solving, such as proving Euclidean geometry theorems, Katona-type matchstick, and symbolic logic. While Newell was interested in implementing mathematical problems on a computer, Simon was interested in the theory of human problem-solving. The concept of heuristics seemed to be the best candidate for linking Newell and Simon's interests (Simon, 1996, p. 203). One day, in October 1955, before a meeting at the Institute of Management Science, Simon was walking through the campus of the University of Columbia, thinking about how humans solve geometry problems.[109] He concluded it was possible to program a computer for solving those type of problems. That night, Newell and Simon discussed the idea of implementing geometry theorems on a computer with mathematician Merill Flood. Flood accepted the idea and agreed the program would run on the computer before Christmas of 1955 (Simon, 1996, pp. 203-204). That idea changed with time, and they decided to emulate heuristics to solve problems of mathematical logic (Simon, 1997b, p. 148). The original idea was to emulate the capacity of humans to solve geometry. Unfortunately, this goal was not possible, for technical reasons and other problems, like how to make computers to do the diagrams. For that reason, Newell

---

[108] Simon convinced his colleagues at GSIA to accept Newell as a doctoral student. His argument was that Newell's research was important to the business school (Simon, 1996, p. 202).

[109] The idea of implementing a chess player had to wait some years because of the difficulty of simulating human mental processes related to playing chess (Simon, 1996, p. 206).

and Simon could not develop this idea, so they decided to go to the area of mathematical logic (Simon, 1997b, p. 148; Simon, 1996, p. 205).

Simon started the task of implementing heuristics for solving problems of mathematical logic in a computer by studying theorem 2.15 of Rusell and Whitehead's *Principia mathematica*. Most of the work was carried out in November of 1955. Then, in December, Simon had some ideas about implementing heuristics (Simon, 1996, p. 205).[110] Simon was responsible for developing a pen and paper strategy to write the program under the name LT. Newell and Shaw were responsible for implementing the program on the JOHNNIAC (Simon, 1996, pp. 205-206; Augier & March, 2004, p. 20).[111] The first run of the LT took place in the same month. The unpublished RAND Report-850 of May 1, 1956, presented the LT. Still, the presentation of the real power of the LT had to wait until August 9, 1956, producing its first proof of the Theorem 2.01 (Simon, 1996, p. 207).[112] The use of heuristics for solving problems of mathematical logic mimicked how humans solve those type of problems (Simon; Newell & Simon, 1956, p. 78; Newell et al., 1959b, pp. 5-6). Using heuristic methods for mathematical theorem proving, the LT was an impressive success. The computer program would become presented at the Dartmouth Summer Research Project on Artificial Intelligence, setting up the scene of research in AI for the years to come.

---

[110] When the work by Simon and Newell was programmed, they considered two options. The first was programming all the rules of mathematical logic and the inference rules for reaching a particular conclusion from a proposition of the *Principia mathematica* that users input into the computer. Unfortunately, this method was not the most reliable because it exhausted the resources from the computer when it was programmed. Using heuristics was the second option, but also the most uncertain. They used the idea from George Polya's *How to Solve it* (1945) (Dick, 2015, pp. 626-627; Newell et al., 1957, pp. 218-222).

[111] To Simon, the LT showed intelligence because the computer program could do symbol processing. As Simon said, Turing had developed a concept of an intelligent machine because his machine could do symbol processing (Gigerenzer & Sturm, 2007, pp. 325-326; Simon, 1996, p. 193).

[112] Before implementing this program on the JOHNNIAC, Simon simulated the LT with his family and several graduate students giving each one of them a card representing the program's components. Each card gave a subroutine to its owner, so they had to execute it when the moment came (Simon, 1996, p. 207).

Conclusions

This chapter has explained the life of Simon and Newell leading to the development of the first intelligent program: the LT. To do so, I have shown the experiences Simon and Newell had in different places, because the practices and concepts learned helped them to conceive the limits of human decision-making. While Simon set up the basis for his theory of human decision-making after working at ICMA and the University of California in Berkeley, Newell came to the concept of heuristics while being student at Stanford University. This chapter also showed how the contact between different ways of knowing helps to create something new. Breaking disciplinary barriers can lead to new areas of knowledge. Perhaps Simon understood the limits of decision-making not just by working in the field of economics but by getting in touch with other disciplinary areas, such as cybernetics. The contact between Newell's heuristics and Simon's theory of human decision making was important for developing the logic theorist.

As shown, Simon's experience in different institutions such as ICMA and IIT helped him to develop his theory of human decision-making. The main problem he had was the resistance of economists against his view about the limits of human decision-making. Fortunately, Simon found at RAND and GSIA the conditions for maturing and testing his theory of human decision-making. The final result of Simon and Newell's work at both institutions was the first intelligent program that could prove theorems of mathematical logic. The LT was, therefore, a byproduct of economic and military interests and research projects. How Simon and Newell's LT influenced machine intelligence research since 1956? The next chapter 5 will provide the answer.

Chapter 5

The Dartmouth Research Project on Artificial Intelligence

In this chapter I aim to explain the institutionalization of AI from a historical point of view**.** The Dartmouth Summer Research Project on Artificial Intelligence took place in 1956 at Dartmouth College. Several researchers attended this workshop. Two of them were Simon and Newell, presenting their Logic theorist (LT) at this event. The LT was the only functioning project among those presented at the workshop. This made the LT the central paradigm for developing artificial intelligence (AI) in the future. But how is the Dartmouth Summer Research Project important for understanding the discovery of computational intractability outside the field? Because AI was began to become institutionalized in this event, constraining who can do research in AI. This event set up the research constraints in the field for the years to come. Which were the limits imposed on AI after its institutionalization in 1956? As I will explain in chapter 8, researchers who developed an alternative to the LT but had no connection to the workshop attendees were ignored. Philosopher Hao Wang was one of them. He formalized the concept of computational intractability outside AI research. The crisis of AI during the second half of the 1960s is linked to the constraints imposed in this field. Chapters 7 and 8 will deal with the topic.

The Dartmouth Summer Research Project has been presented as something more anecdotal instead of the event where the imposition of several constraints on AI research happened. Pamela McCorduck's *Machines who think* (1979) and Jon Agar's *Science in the twentieth century and beyond* (2012) briefly present constraints imposed on AI, but their explanation forgets to describe how the dominant paradigm of early artificial intelligence depended on power plays – in this case, on the association between government interests with private ones. This chapter explains the link between U.S. interests and private enterprises, such as the Rockefeller Foundation for financing the Dartmouth Summer Research Project. That relationship set up the constraints in the years to come for the development of early artificial intelligence. In this vein, the contribution of this chapter is vital for understanding why computational intractability was discovered outside of AI research.

This chapter has only one part, dealing with the institutionalization of AI in 1956.

5.1 Institutionalization of a discipline

During the mid-twentieth century, scientists were speculating how the brain and mind worked. Two works were very influential in that respect. Warren McCulloch (1898-1969) and Walter Pitts's (1923-1969) "A logical calculus of the ideas immanent in nervous activity" (1943) was of great importance because they described how propositional logic explained the activity inside the nervous system (Agar, 2012, p. 381; McCulloch & Pitts, 2012, p. 99). The second influential work was Donald Hebb's (1904-1985) *The organization of behavior* (1949). Hebb's work explained how neural networks function. His book was considered a bridge between psychology and neurophysiology (Hebb, 1949, pp. xii-xix). Moreover, Hebb developed in his book a rule for explaining the process of synapses between two neurons. This rule is known as Hebbian rule (Hebb, 1949, p. 62).[113] Both papers were highly influential for Harvard students Marvin Minsky (1927-2016) and Dean Edwards, and for psychologist George Miller (1920-2012). Using Hebb's proposal, they built the first neural network computer in 1950 called SNARC.[114]

SNARC simulated the behavior of a rat, learning how to find a route out of a maze. The processing power necessary to simulate the rat simulation used a computer with 3000 vacuum tubes and a surplus automatic pilot mechanism from a B-24 bomber to simulate 40 neurons (Agar, 2012, p. 381; Russell & Norvig,

---

[113] According to Hebb (1949): "When an axon of **cell A** is near enough to excite a **cell B** and repeatedly or persistently takes part in firing **u**, some growth process or metabolic change takes place in one or both cells such that **A**'s efficiency, as one of the cells firing **B**, is increased" (Hebb, 1949, p. 62).

[114] ONR financed SNARC (Agar, 2012, p. 381).

2010, pp. 16-17). When the project ended, Minsky started his Ph.D. research at Princeton University, focusing on universal computation in neural networks. The result was his Ph.D. thesis *Neural nets and the brain problem* (1954) (Russell & Norvig, 2010, pp. 16-17; Minsky, 1961, p. 29). Minsky's work was beneficial to him because it helped him to create social connections, being part of a group of researchers interested in organizing a meeting to discuss the possibility of simulating human intelligence. That symposium was the Dartmouth Research Project on Artificial Intelligence.

McCarthy, the electrical engineer Nathaniel Rochester (1919-2001), Shannon, and Minsky, were organizing in 1955 a meeting for the persons interested in studying the possibility of machine intelligence to discuss their research. The main statement of the proposal for getting funds was the following:

> "We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, from abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer" (McCarthy et al., 2006, p. 12; McCorduck, 1979, p. 93).

The proposal convinced the Rockefeller Foundation. The Rockefeller Foundation gave them approximately 7500 Dollars to finance the workshop (Agar, 2012, p. 382; McCorduck, 1979, p. 94). With that economic guarantee, the next step was to call different professionals to discuss the possibility of creating machine intelligence.

Some researchers were interested in participating in the workshop. Apart from the four organizers of the summer research program, six other researchers assisted: mathematician Trenchard More, electrical engineer Arthur Samuel (1901-1990), Oliver Selfridge, physicist Ray Solomonoff (1926-2009), and Newell and Simon. Moreover, other researchers like the mathematician Alex Bernstein and physicist Herbert Gelernter (1929-2015) went to the event as visitors to talk about their work. Several attendees disliked the term AI. McCarthy, who had the idea to coin this emerging discipline as AI, suggested distinguishing the area from automata theory. Among those who did not like the name were Newell and Simon. They published their work in information processing instead of AI (McCurdock, 1979, p. 97). Finally, they agreed with the name AI to formalize the field (McCurdock, 1979, pp. 94-97).[115]

During the workshop, researchers presented their projects and ideas related to the possibility of developing intelligence in computers. Minsky presented a draft of a paper, published later on as "Steps toward artificial intelligence" (1961) (McCurdock, 1979, p. 97).[116] Solomonoff was certain of the impossibility of developing a concept of what intelligence is without knowing whether a computer could solve problems like humans. McCarthy thought of the possibility of using a Turing machine to represent any intellectual problem (McCurdock, 1979, p. 97). The best proposal of this workshop, however, was Newell and Simon's LT (McCurdock, 1979, pp. 103-104).

The LT was intelligent enough to prove theorems of mathematical logic, bringing enthusiasm to the attendees of the event. Moreover, Simon and Newell also showed their recently developed language for information processing: the Information Processing Language (IPL) or list-processing language (McCurdock, 1979, pp. 104; Edwards, 1996, p. 253).[117] Apart from the LT, this event did not meet everyone's expectations. The enthusiasm generated by the LT made this

---

[115] Several attendees disliked the term artificial intelligence. McCarthy, who had the idea to coin this emerging discipline as artificial intelligence, suggested distinguishing the area from automata theory. Among those who did not like the name were Newell and Simon. They published their work in information processing instead of artificial intelligence (McCurdock, 1979, p. 97).

[116] Minsky's article summarized some of the main issues the emerging discipline of artificial intelligence was suffering. Minsky divided his work into five categories: search, pattern-recognition, learning, planning, and induction (Minsky, 1961, p. 8).

[117] While Simon was thinking about simulating the human cognitive process for problem-solving, he was also concerned about how to program structures related to those processes because no programming language existed for handling elements such as flexibility, flexible memory, and dynamic storage (Simon, 1996, p. 204). IPL was used by Newell and Simon for programming the LT.

The LT version presented at Dartmouth was programmed on an early version of IPL (Stefferud, 1963, p. v; Newell & Simon, 1956, pp. 61-62; Augier & March., 2004, p. 20).

intelligent computer program the inspiration for artificial intelligence researchers for the years to come (Gardner, 1987, pp. 139-140).

Conclusions

This chapter explained the first and most important step towards the institutionalization of the new field of artificial intelligence research. The context for AI to be recognized as a field of study came during the Cold War. The development of machine intelligence was linked to the goal of guaranteeing national security and military might of the USA. A very brief explanation of the relationship between private interests and the state is shown in this chapter.

The involvement of the Rockefeller Foundation in financing the Dartmouth Summer Research Project on Artificial Intelligence held at Dartmouth College was not neutral. The visible relationship between U.S. interests and private ones was the attendance of Simon and Newell, consultants at the RAND Corporation.

After the workshop, researchers who attended the Dartmouth Summer Research Project had become enthusiasts about the promising future of developing machine intelligence. The heuristics of LT became the inspiration of research for the years to come, limiting access to AI to those close to the workshop attendants, as the following chapters will show. Chapter 6 will deal with the enthusiasm of early artificial intelligence research, while chapter 7 will describe the crisis of AI. Chapter 8 will contextualize the debate between Simon and philosopher Hao Wang to understand why AI community did not consider Wang’s work during its early years. Those three chapters are vital to understand the formalization of computational intractability outside ai.

Part III

The debate over computational intractability

Chapter 6

Success of artificial intelligence in the 1950s and 1960s

This chapter aims to explain the advancement of artificial intelligence (AI) before its crisis. Enthusiasm AI generated in attendees to the Dartmouth Summer Research Project on Artificial Intelligence produced interesting results for more than ten years. Why were researchers so enthusiastic? The Logic Theorist (LT) had presented to them the possibility of developing machine intelligence, inspiring the development of new intelligent computer programs, such as the General Problem Solver (GPS). Researcher's enthusiasm shaped AI development from the second half of the 1950s until the second half of the 1960s.

Some historians of computing have treated AI development after the Dartmouth Summer Research Project on Artificial Intelligence. Nils Nilsson's *The quest for artificial intelligence* (2010) explains the developments of different artificial intelligence areas, such as natural language processing, computer vision, and mobile robots.[118] His contribution is essential to understanding how artificial intelligence was being developed after 1956. Unfortunately, Nilsson's narrative has limitations because he does not consider some institutional factors for developing different AI areas, such as natural language processing. Daniel Crevier's *AI: the tumultuous history of the search for artificial intelligence* (1993) provides information needed for understanding how different artificial intelligence areas came into existence from an institutional and scientific point of view.

Building upon Nilsson and Crevier's contributions, this chapter aims to reconstruct one part of AI development after the Dartmouth Research Project on Artificial Intelligence before the field's crisis.

Three sections make up this chapter. The first describes some preliminary elements to consider before the other two, such as institutions that shaped AI after the Dartmouth Summer Research Project on Artificial Intelligence. Researching new types of heuristics different from the LT and seeking new paths for

---

[118] Nilsson was an emeritus professor in the computer science department at Stanford University. See http://robotics.stanford.edu/~nilsson/ for more information.

developing machine intelligence are touched on in the second section of this chapter. Finally, the last part of this chapter deals with the development of new areas in early artificial intelligence, such as computer vision. All parts of this chapter illustrate the enthusiasm researchers had in AI.

6.1 The 1958 conference in Teddington, England

After the Dartmouth Summer Research Project, researchers and other professionals interested in simulating intelligent behavior began to explore the possibilities of developing intelligent computer programs. In that spirit, a conference was organized at the National Physical Laboratory in Teddington, England, under the title “Mechanization of thought processes” in 1958. Many attendees from the Dartmouth event went to the summit, like John McCarthy, Oliver Selfridge, and Marvin Minsky (Nilsson, 2010, p. 56). Even though several institutions contributed to the field’s advancement, scientists that produced the best results in AI came from Carnegie Mellon, Massachussetts Institute of Technology (MIT), Stanford University, and the Stanford Research Institute. The best ideas in the field of AI came from doctoral theses written at those institutions (Nilsson, 2010, p. 123). International Business Machine Corporation (IBM) was the leading representative of the private sector (Crevier, 1993, p. 51; McCorduck, 1979, p. 110).

6.2 Early progress

One year after the Dartmouth Summer Research Project on Artificial Intelligence, Simon and Newell concluded that the LT was not a good candidate to explain human decision-making. Simon and Newell decided to change their research direction after analyzing human decision-making data, concluding the LT did not behave like humans (Crevier, 1993, pp. 52-53). In that spirit, Simon and Newell began to seek other possible options for emulating intelligent behavior.[119] In October 1957, Simon and Shaw proposed the GPS project, published in RAND report No. P-1584 in 1959 (Simon, 1996, pp. 219-220).

The LT inspired the development of GPS, but with important changes. The LT used a heuristic that mimicked the process of mathematical logic theorem proving based on George Polya's work. Even though results were good, the LT could not explain how humans make certain decisions for solving problems. However, GPS aimed to explain how people made choices (Augier & March, 2004, p. 22; Newell et al., 1959a, pp. 2-3). Departing from the use of Polya's heuristics, the program used means-ends analysis (MEA) and planning as heuristics for problem-solving. MEA could break a problem into subproblems so that the program could solve a task easier, while planning could predict the next step of the problem to find the best possible solution based on future states (Crevier, 1993, pp. 53-54; Newell et al., 1959a, p. 18; Nilsson, 2010, p. 88). MEA was used over examples of symbolic logic and trigonometry, while symbolic logic used planning. The research with GPS continued until 1968 (Crevier, 1993, p. 54). When GPS became the baseline for human decision-making research, researchers at the Carnegie Institute of Technology focused on understanding human cognitive processes. Carnegie's work on AI differed from other institutions, such as IBM. IBM's goal was to develop only machine intelligence (Crevier, 1993, p. 55).

Thus, researchers from several institutions began their journey to develop intelligent computer programs. When Nathaniel Rochester returned from the Dartmouth Summer Research Project to IBM, he was enthusiastic about simulating intelligent behavior on a computer. Simon, Newell, and Shaw's LT also inspired Marvin Minsky to write a program that could prove high school geometry theorems. Minsky's work in turn inspired Rochester to reproduce Minsky's idea in an IBM 704 model (Crevier, 1993, p. 55; Nilsson, 2010, p. 85). Rochester chose Herbert Gelernter -who had also attended the Dartmouth Summer Research Project- to carry out the project of developing a Geometry Theorem Prover.

---

[119] In 1955, Newell's priorities changed. He retired from RAND, and began his Ph.D. in industrial management under the supervision of Simon at Carnegie Tech. Newell found a position as professor at Carnegie after he finished his doctoral work (Nilsson, 2010, p. 115).

Gelernter worked on his Geometry Theorem Prover for three years. His program took time to develop because he had to write and debug twenty thousand lines of code (Crevier, 1993, p. 55). Gelernter's technique was unique because he put attention on a set of coordinates in the punched cards that represented a figure. His program functioned by working backward, beginning at the same time when one person introduced a theorem in the program. The program automatically produced a set of intermediate results to prove theorems or axioms (Crevier, 1993, p. 55; Nilsson, 2010, pp. 85-86).[120] Other researchers, like Arthur Samuel and Alex Bernstein, were developing further projects, such as a computer chess player or a program that could play checkers (Crevier, 1993, pp. 57-58) Unfortunately, IBM had to change its priorities after the company investors saw no profits coming from development such research (Crevier, 1993, p. 58). McCarthy said that IBM decided not to support AI research because the company did not want to present the image that it was investing in data processing. IBM was more specialized in building computers (Nilsson, 2010, p. 118).

Of course, researchers outside of IBM kept being interested in advancing AI. In 1958, one year after Minsky worked with Selfridge at the Lincoln Laboratory, Misky decided to continue his research in AI at MIT with McCarthy.[121] McCarthy and Minsky founded the MIT Artificial Intelligence Group. They collaborated until 1962, when McCarthy left MIT for Stanford University to lead own artificial intelligence team (Crevier, 1993, pp. 64-65).[122] Minsky continued the projects of artificial intelligence at MIT as director of the Artificial Intelligence Group. In 1963, MIT received a grant of 2.220.000 Dollars from the Advanced Research Projects Agency (ARPA).[123] MIT Artificial Intelligence Group secured one-third of the funding (Crevier, 1993, p. 65)

---

[120] Gelernter's program showed the link between IPL-V programming languages, developed by Simon and Newell with McCarthy's LISP (McCorduck, 1979, p. 98).

[121] Misky and McCarthy were part of team of organizers of the Dartmouth Summer Research Project in 1956. McCarthy left Dartmouth College to join MIT in 1958 (Nilsson, 2010, p. 116).

[122] Minsky and McCarthy had a dispute concerning the direction artificial intelligence should take. While McCarthy defended artificial intelligence as entirely based on formal logic, Minsky disagreed. Minsky and McCarthy's differences could be why McCarthy left MIT (Crevier, 1993, p. 64).

Results obtained by the research group were astonishing for that time. The Symbolic Automatic INTegrator (SAINT) was one most promising works developed at MIT Artificial Intelligence Group (Crevier, 1993, p. 65). SAINT was programmed in 1961 by James Slagle, one of Minsky's students.[124] SAINT was inspired by Newell and Shaw and Simon's LT, with the difference that SAINT solved problems of symbolic integration. SAINT showed its capacity for problem-solving by resolving eighty-four out of eighty-six problems, most of which were freshman examinations. With SAINT's results, LT's resolution method proved to be successful with logic as well as with algebra (Crevier, 1993, p. 66). As SAINT, many projects came from MIT Artificial Intelligence Group under Minsky's direction during the 1960s, inspiring the next generation of researchers in AI.

6.3 Continued confidence

During the 1960s, the MIT Artificial Intelligence Group made some exciting advances. Researchers followed Minsky's view. He argued that humans did not use a systematic search technique to find solutions to problems. Instead, past experience was the explanation for their decision-making (Crevier, 1993, p. 74).
One of artificial intelligence programs that applied Minsky's idea was called "Analogy". It was programmed in 1963 by Minsky's student Tom Evans.

---

[123] ARPA was one several governmental institutions that financed projects in artificial intelligence. It was a newly created agency of the U.S. Department of State. ARPA aimed to support science projects to guarantee technological lead (Crevier, 1993, p. 65). The institution was set up after the launch of the Soviet satellite *Sputnik* (Nilsson, 2010, p. 119). ARPA was essential in early computer science, funding computing research projects at MIT, Carnegie Mellon University, Stanford University, and the Stanford Research Institute (Nilson, 2010, pp. 119-120).

[124] Slagle received his Ph.D. in mathematics at MIT in 1961. He is a professor at the University of California, Berkeley. Furthermore, Slagle had appointments at John Hopkins University, MIT, and the University of Minnesota. See https://aiws.net/the-history-of-ai/this-week-in-the-history-of-ai-at-aiws-net-computer-scientist-james-robert-slagle-developed-saint/ for more information.

Analogy's method centered its attention on matching experiences. The program compared a previous experience with a recent one in order to make a decision. For evaluating Analogy's performance in problem-solving, Evans used intellectual coefficient tests based on geometry. Analogy achieved good results, but testing the program took place in a controlled environment (Crevier, 1993, pp. 74-76). Analogy was the beginning of continuing research in supervised conditions of artificial intelligence programs.[125]

In 1963, a research assistant of the famous Swiss psychologist Jean Piaget went to MIT to work with Minsky.[126] His name was Seymour Papert (1928-2016). Papert decided to leave Geneva to go to the USA by recommendation of Warren McCulloch (Crevier, 1993, pp. 84-86).[127] With Papert, artificial intelligence algorithms were developed under controlled conditions, obtaining good results.

One of those projects developed a computer programming language for children called "Logo" (Crevier, 1993, pp. 86-87). Logo was developed in 1967 to teach essential elements of geometry. This interactive program helped children

---

[125] The theory of "microworlds" emerged in these environments. The idea was expanded in Papert's *Mindstorms* (1980) (Crevier, 1993, p. 85; Edwards, 1998, p. 55). Mainly, developing the theory of "microworlds" aimed to understand the nature of learning. "Microworlds" represented rules a person could manipulate by interacting with the environment by manipulating particular objects and operations (Edwards, 1998, p. 64).

[126] Piaget (1896-1980) was a psychologist interested in studying child development. He graduated from the University of Neuchâtel with a Ph.D. in philosophy in 1918. Piaget was a multifaceted figure, being a professor of child psychology at the University of Geneva from 1929 until his death. In 1921, Piaget was appointed as director of the Jean Jacques Rousseau Institute in Geneva. Furthermore, Piaget founded the International Centre of Genetic Epistemology in 1955 in Geneva, becoming its first director (Rodriguez, 2022c).

[127] Papert was a mathematician interested in studying the philosophy and mechanism of thinking. Born in South Africa, Papert obtained his Ph.D. in mathematics at the University of Witwatersrand in 1952. Later, he went to study at Cambridge University to earn his second doctorate in mathematics in 1958. Even though Papert had great success in mathematics, his philosophical questions were unanswered. However, his future was going to change the year he graduated from Cambridge. In 1958, Papert met Piaget at a conference, and Piaget invited him to join his International Centre of Genetic Epistemology. Papert worked in the institution until 1963, focusing his research on concept building. The same year, Papert left Geneva to work in the USA at MIT. Papert was the co-director of the MIT Media Lab (Crevier, 1993, pp. 84-86; Nilsson, 2010, p. 116).

understand concepts like square, circle, or triangle by instructing them to command a turtle. The animal moved in one way or another to draw a figure (Crevier, 1993, pp. 86-87). The program seemed useful just for teaching mathematics to children, but Papert did not think the same because he comprehended that Logo promoted learning by experimentation (Crevier, 1993, p. 87).[128]

After the success of Logo, Minsky, Papert, and other researchers focused on developing algorithms that could emulate the visual capacity of humans for imitating intelligent behavior.[129] Several students, and researchers, made progress in developing algorithms that could emulate the human visual system. In 1963, Ph.D. candidate in electrical engineering, Lawrence Gilman Roberts (1937-2018), presented his doctoral thesis *Machine perception of three dimensional solids* (1963).[130] Roberts developed an algorithm that recognized three-dimensional objects. Furthermore, the method he presented in his dissertation could determine the position and orientation in the space of the objects (Roberts, 1963, pp. 8-9; Nilsson, 2010, pp. 128-129).[131]

---

[128] Logo was a great success that inspired the development of similar "microworlds" like Dynaturtle. The program aimed to teach the fundamental laws of physics (Edwards, 1998, p. 56).

[129] Minsky was not the first researcher to work in the field of computer vision, but his team started to develop algorithms in the field later (Nilsson, 2010, p. 130). Minsky and Papert aimed to create an artificial intelligence system that could successfully imitate human problem-solving. They explored the behavior of the human visual system for achieving their purpose (Crevier, 1993, p. 87). Papert's goal was to analyze a scene to detect and name objects by matching the findings of visual algorithms to a vocabulary (Nilsson, 2010, p. 130).

[130] Roberts was an electrical engineer who graduated from MIT in 1959. Four years later, he obtained his Ph.D. at MIT. Robert's doctoral advisor was Peter Elias. In 1965, he became a contractor of ARPA to develop a computer network. Later, in 1966, Roberts became ARPANET's director (the internet's precursor; Gregersen, 2020; Nilsson, 2010, p. 128).

[131] One of the limits of Roberts's algorithm was setting up the color scale because the algorithm could only identify objects in black and white photographs (Nilsson, 2010, p. 128).

Another student interested in computer vision was Adolfo Guzman Arenas.[132] Guzman Arenas joined Minsky's team in 1964. His contribution to computer vision was an algorithm to detect objects of any shape, with the only condition the objects have planar surfaces and do not contain any holes (Crevier, 1993, p. 91; Nilsson, 2010, pp. 134-135).[133]

Research developed by several students and researchers in computer vision created new possibilities to experiment with the concept of “microworlds” in the real world (Papert, 1980). The learner was a robotic arm. Papert and Minsky developed a computer vision program that could build structures made of blocks. Several attempts happened before a robot could interact with the elements of the environment. The first try was a robotic arm that could set its geometric coordinates. Unfortunately, the results were not as good as Papert and Minsky thought because the coordinates of the robotic arm had to be compared with those set manually in the system. For that reason, they added sensors in the fingers for adjusting the robotic arm. However, developing more computer vision programs was necessary to help the system to avoid non-desired elements from the environment. In both cases, the system used complex algorithms for planning the order the blocks were going to be picked up (Crevier, 1993, pp. 92-94). Papert and

---

[132] Electrical engineer Guzman Arenas (1943-) teaches artificial intelligence at Instituto Politécnico Nacional in the city of Mexico. He graduated in electrical engineering at Instituto Politécnico Nacional in 1964. Later, he graduated with a master’s in electrical engineering and a Ph.D. in computer science at MIT, obtaining his doctorate in 1968. Minsky was the Ph.D. advisor of Guzman Arenas. After he finished his Ph.D., Guzman Arenas taught at the department of electrical engineering at MIT from 1969 to 1970. Arenas is a member of the New York Academy of Sciences (Guzman Arenas, 2005). He was the director of the IBM Scientific Center in Latin America.

Guzman Arenas was contacted for this doctoral thesis, asking him for some information about computational intractability (A. Guzman-Arenas, personal communication, November 22, 2017).

[133] Guzman Arenas’ *Computer recognition of three dimensional objects* (1958). presented methods for “1) to partition or decompose a visual scene into the bodies forming it; 2) to position these bodies in three-dimensional space, by combining two scenes that make a stereoscopic pair; 3) to find the regions or zones of a visual scene that belong to its background; 4) to carry out the isolation of objects in (1) when the input has inaccuracies” (Guzman Arenas, 1958, p. 2).

Minsky were so excited that they were certain the Apollo moon landing mission would include their system (Crevier, 1993, p. 94).

Computer vision was not the only area of interest of AI researchers because they also focused on natural language processing. The program “Shrdlu” was developed from 1968 until 1969 by one of Papert's students Terry Winograd.[134] Shrdlu was a system that could answer questions, execute commands, and accept information in natural language. Moreover, it used contextual and semantic information for evaluating sentences (Crevier, 1993, p. 96; Winograd, 1971, p. 3). Shrdlu showed its potential to manipulate natural language in a program that simulated a robotic arm by receiving orders for manipulating objects that appeared on a computer screen. Depending on user actions, the program provided information or executed a command over an item shown on the screen (Crevier, 1993, p. 97). Unfortunately, Shrdlu showed its limits in that it could not be used in the real world just as well as in “microworlds” (Crevier, 1993, p. 102). Researchers in AI were enthusiastic about the advances in the area until the second half of the 1960s, but then their hopes disappeared. Chapter 7 will explain the failure of AI to imitate intelligent behavior in close detail.

---

[134] Winograd was born in 1946. He grew up in the city of Colorado in the USA. In the same city, he studied mathematics at Colorado College, graduating in 1966. Later, in 1967 he studied linguistics at University College London. Finally, he obtained his Ph.D. in applied mathematics at MIT in 1970 (Norberg, 1991, pp. 3-5). See https://hci.stanford.edu/winograd/cv.html for more information.

Conclusions

This chapter illustrated the confidence generated after the Dartmouth Summer Research Project on Artificial Intelligence. It also showed that most research in AI was produced at the University of Carnegie-Mellon, MIT, Stanford University, and the Stanford Research Institute. AI's significance for national security and the military was the most important factor for investing in this research.

GPS was born in the Cold War context. It became the baseline for developing programs that could simulate human decision-making. This computer program replaced LT's heuristics for mean-ends analysis and planning for the process of decision-making and problem-solving, shaping AI for years to come.

Minsky worked with several students, producing remarkable results in several areas of AI, such as computer vision and natural language processing. Two of those students were Slagle and Guzman Arenas. Minsky's laboratory also produced remarkable progress in AI thanks to the collaboration with Papert. They developed computer programs with students that could solve problems in a controlled environment. Later, those problems were the basis for developing the concept of "microworlds" (Papert, 1980).

One exception of the institutions that developed AI independently from the U.S. Military was IBM. Rochester and Gelernter simulated human decision-making and problem-solving by programming a Geometry Theorem Prover, while Samuel and Bernstein programmed a game of chess. Unfortunately, IBM saw no rentability with those programs. For that reason, funding for AI research was cut off. The field was promising, but later it showed limitations. As will be explained in chapter 7 in more detail, those institutions shaped AI using funding from the U.S. military and other institutions linked to the U.S. government.

Chapter 7

The crisis of artificial intelligence

This chapter aims to explain the crisis of artificial intelligence (AI) in the later 1960s. Why did the earlier optimism concerning early artificial intelligence during the 1950s and the first half of the 1960s crumble? And how is the problem of computational intractability related to the crisis of AI? AI computational intractability appeared after specific tasks, such as the traveling salesman problem, could not be dealt with by computer algorithms in a reasonable amount of time.

Historiography has not considered the constraints imposed at the Dartmouth Summer Research Project on Artificial Intelligence as the main reason for the AI crisis. The importance of understanding why complexity of algorithms is vital for getting to know whether algorithms could solve a particular problem in a reasonable amount of time or not. Daniel Crevier's *AI: The tumultuous history of the search for artificial intelligence* (1993) is an attempt to explain the crisis of artificial intelligence historically.[135] Unfortunately, Crevier does not consider AI development against the background of the Cold War.

The contribution of this chapter is an explanation of the link between computational intractability and the crisis of AI. Three sections make up this chapter. The first describes the end of the early enthusiasm in the field of AI produced by the researchers interested in developing machine intelligence. The second section shows that the problem of translating texts from one language to another was an important issue for the failure of AI. Finally, the third and last section will focus on explaining why "microworlds" could not help intelligent algorithms to learn to cope with the complexities of real-world problems.

---

[135] See https://www.brainpreservation.org/team/daniel-crevier/ for more information about Crevier.

7.1 Preliminary considerations

In 1957, Simon forecasted three developments in AI for the next ten years: a computer would win a chess championship, proving an important mathematical theorem with a computer program, and computer programs could prove most psychological theories (Crevier, 1993, p. 108). Simon kept his optimism for years to come. As shown in chapter 6, he infected other AI researchers with it. In the late 1960s, Marvin Minsky's *Computation: finite and infinite machines* (1967) still described that within a generation, most human intelligence would be simulated by computers, promising that many elements of intellect could and would still be understood by AI (Minsky, 1972, p. 2).

AI researchers' naïve optimism fell when first problems appeared. Problems such as the incapacity for extending "microworlds" to other domains during the 1970s, and the failures of natural language processing to translate human languages in the late 1960s, provoked anger among AI researchers. Furthermore, those who were financing the discipline felt disappointed because AI could not fulfill its promise of imitating human behavior (Crevier, 1993, pp. 109-110).[136] Many believed with certainty that faster hardware would reduce problem complexity.[137] Unfortunately, things were more complex as expected.

---

[136] The U.S. Defense Department was considering cutting research funding in AI because artificial intelligence programs were performing poorly in real-world situations (Crevier, 1993, p. 110). One of the disappointments AI produced to the U.S. military was the poor performance of natural language processing programs developed by the University of Carnegie-Mellon in the early 1970s. Those programs could not understand correctly certain types of commands. Therefore, the Advanced Research Projects Agency (ARPA) cut an important part of the funding for AI in 1974 (Crevier, 1993, pp. 115-117).

[137] "Microworlds" researchers were using had few objects to manipulate. For that reason, the number of possible solutions to a problem was minor. Those involved in early AI not consider environments with higher complexity (Stuart et al., 2010, p. 21).

7.2 The limits of natural language processing

In 1966, most research in automatic translation in the USA had no funding. The National Research Council decided to cut all support for automatic translation because the institution considered the area was not useful at all. After the Second World War, the U.S. government concluded that text translation was easy.[138] Unfortunately, when the CIA attempted to translate Russian texts, this view changed. Results were disastrous. Furthermore, most efforts to program a system that followed grammatical rules for language translation failed. The U.S. government decided to cut aid to automatic translation after its failure (Crevier, 1993, p. 110). Fortunately, MIT, Stanford University, and Carnegie Mellon were working in other natural language processing problems not affected by the decision of the U.S. government (Crevier, 1993, p. 110).[139] Unfortunately, problems also appeared in research carried out at those three universities.

Simon and Allen Newell realized their General Problem Solver (GPS) had limitations by the moment of having to introduce information into its knowledge base. The user that interacted with GPS had to introduce information in a highly stylized manner because the program could not understand orders introduced in a sequencial form. For that reason, GPS results could not solve some problems (Crevier, 1993, pp. 110-111).[140]

---

[138] Inspired by Turing's breaking of the Enigma codes, researchers assumed that automatic translation was similar to cipher decoding (Crevier, 1993, p. 110).

[139] One of the most significant problems early AI had was researcher's difficulty for working at institutions different from those present at the Dartmouth Research Project on Artificial Intelligence to enter the field. As already indicated, Carnegie Mellon University, MIT, Stanford University, and the Stanford Research Institute developed most of early AI (McCordock, 1979, pp. 109-110). Moreover, former AI students from Carnegie Mellon, MIT, and Stanford University entered as professors at those institutions, leaving other graduates in the field of computer science from other universities out (Nilsson, 2010, p. 117).

[140] One of the problems GPS could not solve satisfactorily was the banana problem (Crevier, 1993, p. 111). Crevier (1993) stated that the banana problem consisted of "a monkey-faced with the perennial too-high banana. The animal, alone in a room containing a single chair, tried to grab a banana dangling out of its reach. Was the monkey (or GPS) clever enough to

7.3 The failure of microworlds

Microworlds seemed to be reliable environments for helping intelligent algorithms to learn how to manipulate its elements. Unfortunately, the "microworlds" approach had its limitations. Inspired by physics, Minsky and Papert were convinced that using toy examples algorithms could grasp how to extrapolate the basic principles from the environment the agent is interacting with. Nevertheless, a difficulty appeared in 1972 in a Ph.D. project carried out by Eugene Charniak (Crevier, 1993, p. 112).[141] Charniak's *Toward a model of children's story comprehension* was related to developing an intelligent algorithm that could answer questions from children's books.[142] For achieving that purpose, Charniak developed a model that described real-world knowledge that helped the system to guess the story. Unfortunately, the algorithm could not reach its goal because normalizing common sense knowledge was complex. Even more, the system just accepted language represented internally in the model, not plain English (Crevier, 1993, pp. 112-113; Charniak, 1972, p. 2).

The promise of "microworlds" as the solution to all problems in AI failed. Finding funding started to shake because bad results obtained during the 1970s and made AI a less promising field than it seemed to be at its beginnings. Moreover, some persons who reviewed researcher's proposals were AI graduates, so convincing them was difficult based on that antecedent (Crevier, 1993, p. 115).[143] Computational intractability would have helped to understand the limits of "microworlds". Unfortunately, AI was not in touch with the discipline of

---

move the chair to the banana and climb up?" (Crevier, 1993, p. 53). The person had to insert manually into the GPS the coordinates of the monkey, the chair, and the banana to make GPS calculate the distances between objects. After that, the programmer instructed GPS to reduce the distance between those three objects. Unfortunately, GPS could not solve this problem if programmer did not inform the system of the initial conditions (Crevier, 1993, p. 53).

[141] Charniak is a professor of computer science at Brown University. Charniak received a degree in physics from the University of Chicago and a Ph.D. in computer science from MIT. See http://cs.brown.edu/people/echarnia/ for more information.

[142] Charniak's thesis advisor was Minsky.

[143] Some ARPA project reviewers graduated in artificial intelligence (Crever, 1993, p. 115).

computational complexity during the 1970s. Chapters 8 will explain why ignoring Hao Wang led to the discovery of computational intractability outside of AI.

Conclusions

This chapter explained how AI research entered in a crisis after years of optimism. In the beginning, researchers were enthusiastic because the results obtained by intelligent algorithms were promising. Unfortunately, their optimism crumbled when those algorithms could not solve certain types of problems, such as the traveling salesman problem. “Microworlds” served for training early AI algorithms. Controlled conditions for teaching algorithms how to solve real-world problems did not let researchers see the complexity of problem-solving in the real world. Researchers thought more hardware capacity was the solution for intelligent algorithms to solve any real world problem. Unfortunately, their beliefs were wrong.

Another problem for AI algorithms was their incapacity to translate foreign languages. Natural language processing algorithms could not accurately translate spoken language. One of the U.S. needs during the Cold War was the translation of Russian to English for national security and military purposes. Thus, the U.S. government decided to cut funding after natural language processing algorithms failed to deliver on their promises. Some universities, such as Carnegie Mellon, continued the task of developing natural language processing algorithms independently of military funding, but their efforts to develop accurate translations using intelligent algorithms failed. Some people knew the limits of AI before the crisis, but they were ignored. Why did this happen? Did constraints on research by the AI community produce the crisis? Chapter 8 will deal with that question.

Chapter 8

Hao Wang and Herbert Simon's automatic theorem provers

This chapter aims to analyze the controversy between Wang and Simon related to automatic theorem provers during the Cold War. Wang's "Toward mechanical mathematics" (1960) presented a method that proved all mathematical logic theorems found in Bertrand Russell and Alfred North Whitehead's *Principia mathematica* (1910-1913). Why was Wang's journal article controversial? Wang obtained better results than Simon and Allen Newell's Logic theorist (LT) using Herbrand-Skolem's theorem instead of Polya's mathematical heuristics. While the LT could prove 38 theorems of the *Principia,* Wang's Program P proved all of them. Even though Wang obtained better results, he was not taken seriously by the early artificial intelligence community. How Wang and Simon's dispute connects with the problem of computational intractability? Wang focused his energy on the problem of decidability in mathematical logic after he was ignored by the AI community, finding the impossibility of solving problems of predicate calculus in a finite series of steps. Wang's result was similar to Alan Turing's computability theory.

The literature which explores the controversy between Simon and Wang is very scarce. Charles Parsons and Montgomery Link's *Hao Wang: logician and Philosopher* (2011) include a chapter written by philosopher Eckehart Köhler about the controversy.[144] Köhler's "Collaborating with Hao Wang on Gödel's philosophy" (2011) points out the differences between Simon and Wang's work. Köhler said that while Simon focused on the logic of discovery, Wang's computer program just proved theorems of mathematical logic (Köhler, 2011, pp. 62-65). Even though Köhler's chapter explains the difference between Simon and Wang's theorem provers, he does not consider the context of the Cold War to explain why

---

[144] Parsons is emeritus Edgar Pierce professor of philosophy at Harvard University, and Link is an associate professor in philosophy at Suffolk University. See https://philosophy.fas.harvard.edu/people/charles-parsons and https://www.suffolk.edu/academics/faculty/l/i/montgomery-link for more information.

Wang was not taken seriously by the early AI community. Did Cold War constraints on scientific development stop Wang from entering AI? Yes, Wang could not do his work in AI because People's Republic of China and the USA were in conflict from 1950 to 1972. AI was financed for military purposes such as stopping a potential nuclear war. Therefore, this field was closed to any person considered a threat to U.S. interests.

The work of philosopher Gary Mar is a starting point for realizing the separation of Wang from AI.[145] Mar's "Hao Wang's logical journey" (2017) considers racism in U.S. society as a category of analysis to explain why certain circles in U.S. academia ignored Wang. Using the racial category of Oriental, Mar analyzes the discrimination faced by Chinese and Japanese academics in U.S. universities (Mar, 2015, pp. 544-545). For instance, Mar explains the impossibility of Wang to obtain a position as a professor of philosophy at Harvard University because of his nationality. Mar mentioned that Wang solved the entire mathematical logic system developed by his doctoral advisor Quine, producing a vital contribution to mathematical logic. However, his colleagues did not consider him a philosopher. Instead, he continued his career at Harvard University as a lecturer of applied mathematics (Mar, 2015, pp. 541-542). The main problem with Mar's argument is not considering philosophy and other areas of science as funded by the U.S. government for military application. For example, Quine was consultant at Research and Development Corporation (RAND; Quine, 1985, p. 217). Furthermore, Mar does not explain the tense relations between the USA with People's Republic of China from 1950 to 1972.

Stephanie Dick's *Aftermath: (re)configuring minds, proof, and computing in the postwar United States* (2015) explains the dispute between Simon and Wang in a broader context. Dick's thesis focuses on three automatic theorem provers. Two of them are Simon's LT and Wang's Program P. She mentions that Wang's theorem prover obtained better results than Simon's. The difference between Dick's narrative with Köhler and Mar is the use of context for explaining the possible discrimination against Wang concerning political tension between the USA and

---

[145] Mar is a professor at Stony Brook University.
See https://www.stonybrook.edu/commcms/philosophy/people/_faculty/mar.php for more information.

China. She says that Wang was considered a threat to U.S. National Security because he had sympathy for Marxism (Dick, 2015, pp. 110-112). Unfortunately, Dick does not consider the formation of the Chinese state after the fall of the *Qing* dynasty in the first half of the twentieth century and the later rise to power of Mao Zedong. International relations between the USA and China in both periods are entirely different because during the pre-Maoist years, the USA and China had a good relationship in terms of academic exchange for modernizing China. Did the Korean War influence the political shift of the USA with the People's Republic of China? Yes, because those Chinese professionals and academics living in the USA by the time China was involved in the war were seen as spies of China.

This chapter will explain the dispute between Simon and Wang in the context of the Cold War by considering Cold War constraints imposed on AI research focusing on the change of attitude the U.S. government had towards China after Mao came to power and the involvement of China in the Korean War. This chapter is linked with chapter 1 because the computability concept of Turing connects with Wang's 1961 discovery. This chapter is also linked to chapters 5 and 6 because they show artificial intelligence institutionalization and early enthusiasm generated by the artificial intelligence community before finding intractable problems that any intelligent algorithm could not solve. Chapter 7 is also linked to this chapter because the crisis of AI is related to constraints imposed on those who could enter in AI during the Cold War.

Two parts make up this chapter. The first describes the life and trajectory of Wang's doctoral advisor Quine concerning the development of his mathematical logic. The second part covers Wang's life, his relationship with Quine, and the controversy between him and Simon. Both parts explain the link between Wang and Quine, and why early artificial intelligence community ignored Wang. As will be explained, Wang changed of research priorities after his Program P was not taken seriously, focusing on mathematical logic instead of AI.

8.1 Quine’s mathematical logic

Quine was born in the city of Akron, Ohio, in 1908. His father, Robert Stanford Quine, was a machinist, while his mother, Harriet Ellis van Orman, was a teacher at one of Akron’s public schools (Quine, 1985, pp. 1-3). In 1922, Quine began his high school studies at West High School. During those years, he became interested in philosophy. Quine's reading of Edgar Allan Poe’s *Eureka* (1848) stimulated his curiosity to study the universe (Quine, 1985, pp. 36-38).[146]

Quine graduated from high school in 1926 and the same year he enrolled at Oberlin College (Quine, 1985, pp. 46-49). During his first year, Quine chose the academic path. He was thinking about following a fulfilling academic career for the rest of his days. Creative writing was his first choice, but later Quine found that English did not meet his expectations. Later, he thought of mathematics, philosophy, and philology. After thinking for some time, his intention to study philology disappeared because he was more interested in pursuing a career in philosophy of mathematics. Quine chose to review philosophy of mathematics after talking with Bill Bennett, a college senior.[147] Quine considered pure mathematics an arid subject. Still, combining that topic with philosophy attracted him significantly after reading Russell and Whitehead’s *Principia*. Quine decided in his sophomore year he was going to major in mathematics (Quine, 1985, pp. 51-52; Quine, 1985, p. 58). During his junior year, Quine’s decision to study mathematical logic was not going to be easy because nobody knew modern logic at Oberlin College. However, Quine found support from the chairman of the mathematics department William D. Cairns (1871-1955).[148] Cairns helped Quine to

---

[146] The novelist Poe (1809-1849) wrote several short stories during his lifetime. Poe’s *The murders in the Rue Morgue* (1841) was one of his most famous writings. See Quinn (1988) for more information.

[147] Even though not enough information about Bennett exists, Quine mentioned that Bennett became a broker and eventually a donor of books to the Stanford philosophy department (Quine, 1985, p. 52).

[148] Cairns was a professor of mathematics and physics at Oberlin College. He got his A.B degree in 1892 from Ohio Wesleyan University. Some years later, in 1897, he received his A.B degree from Harvard University. In 1907, he received his Ph.D. from Göttingen University. Later, he returned to the United States, serving as head of the department at Oberlin College from 1920 until 1939 (Carver, 1956, p. 204).

get the books he needed, such as Russell's *Principles of mathematics* (Quine, 1985, p. 59).

After Quine graduated from Oberlin College in 1930, he decided to pursue graduate studies at Harvard University. His desire to study with Whitehead influenced his decision to go to Harvard. Quine obtained a scholarship for his postgraduate education. After that, he went to Boston to start his new journey (Quine, 1985, pp. 73-74).[149] Unfortunately for Quine, Whitehead was not lecturing logic because Whitehead was delivering classes of ancient and modern cosmology and in science and the modern world. Instead, Quine took classes in logic with Henry Maurice Sheffer (1883-1964) (Quine, 1985, pp. 82-83).[150] Sheffer talked about several known authors in his lectures. Oswald Veblen (1880-1960), E.V. Huntington (1874-1952), David Hilbert, and Russell, to name a few.[151] Quine found Sheffer's lecture boring, so he decided to write a research paper to learn something that interested him. Quine's starting point was his honor's thesis (Quine, 1985, pp. 82-83).[152] He considered that Russell and Whitehead's *Principia mathematica* formulas could prove symmetric functions. Quine wrote his paper for two months.

Quine's paper was one of the requirements he had to deliver to obtain his Ph.D. Quine sought an advisor to complete his doctoral studies. Whitehead promised Quine he was going to sponsor Quine's dissertation. Whitehead was enthusiastic about studying again mathematical logic. Quine wrote a mathematical thesis with a philosophical conception. Like Russell and Whitehead's *Principia mathematica,*

---

[149] Quine greatly admired Whitehead since he read his book *Principia Mathematica* (Quine, 1985, pp. 73-74).

[150] Sheffer was a logician who taught at Harvard University from 1917 until 1952 (Scanlan, 2000, pp. 193-197).

[151] Veblen was a mathematician interested in several areas of knowledge like differential geometry and topology. In the subject of mathematical logic, he was interested in the axiomatization of Euclidean geometry (Rodriguez, 2017c).

Huntington was a professor at Harvard University from 1905 to 1941. He was interested in studying axioms for different mathematical systems, such as axioms for Boolean algebra (O'Connor & Robertson, 2000).

[152] Quine bachelor's thesis was related to proving a formula of Louis Couturat within the system of *Principia Mathematica* (Quine, 1985, p. 72).

Quine's thesis aimed to understand the foundations of logic and mathematics. The dissertation of his work took place on April 1, 1932, at Whitehead's home (Quine, 1985, pp. 84-86).

Once Quine gave his doctoral dissertation, Quine was ready for his academic journey in Europe. After talking with philosopher Herbert Feigl (1902-1988) and his fellow graduate student John Cooley, Quine decided to start his journey in Vienna.[153] He arrived on September 11, 1932. After a short trip, Quine began his academic journey (Quine, 1985, pp. 86-92). He did not take any logic courses at the university. However, after talking with Moritz Schlick (1882-1936), Schlick invited Quine to Vienna Circle's discussions (Quine, 1985, pp. 93-94).[154] When Quine attended to the discussions, he met Freddie Ayer (1910-1989) and Hans Reichenbach (1891-1953).[155]

Quine's dream to receive lectures from Rudolf Carnap was not possible during the time he was in Vienna, but he met Carnap in the hospital in late 1932 (Quine, 1985, pp. 94-96; Gibson, 2004, p. 4). The next year, in late January, he could take classes with Carnap in Prague. Quine went to see Carnap's lecture at the Physics Institute after he arrived to Prague (Quine, 1985, pp. 96-97, Gibson, 2004, p. 4). Quine attended the classes of Carnap regularly. Furthermore, when Quine was not doing his teaching duties, he visited Carnap to discuss Carnap's work. After talking with Carnap, Quine concluded that he was speaking with his next greatest intellectual hero (Quine, 1985, p. 98).

---

[153] Feigl was a philosopher who emigrated to the USA in 1931. He made some significant contributions to the field of philosophy, like the debate on scientific realism. Moreover, he was part of the Vienna Circle since its beginnings (Neuber, 2018).

[154] Schlick was considered the nominal leader of the Vienna Circle of logical positivists. He obtained tenure at the University of Vienna in 1922. Unfortunately, his brilliant career stopped in 1936 when he was murdered (Oberdan, 2017).

[155] Ayer was a philosopher who defended the idea that knowledge of the world came from sense experience, eliminating the possibility of a deity or any metaphysical entity (Rogers, 2020).

Reichenbach was considered the greatest empiricist of the 20th century. He created a group similar to the Vienna Circle called the Berlin Group. He was a professor at the Humboldt-Universität zu Berlin from 1926 until 1933, when the Nazis came to power (Glymour & Eberhardt, 2016).

After a brief trip through Italy, Carnap and his wife went to Warsaw (Quine, 1985, pp. 99-101). In May 1933, Quine and his wife traveled to Warsaw. On the first day in the city, Quine talked with Jan Lukasiewicz (1878-1956) and Stanislaw Leśniewski (1886-1939) about logic.[156] Even though their conversation was interesting, Quine and his wife were having a problem because they could not find an affordable room in the hotels in the city. Lukasiewicz helped them to find a room at Hotel Victoria. This place would be Naomi and Quine's home when they stayed in Warsaw (Quine, 1985, pp. 102-103). The days Quine spent in Warsaw, he got inspired by the work of several logicians he met during the time spent in the city. Quine usually went to see Alfred Tarski's lectures.[157] Furthermore, Quine used to go to Leśniewski and Lukasiewicz's lectures. One day, Quine presented a paper in a seminar Tarski organized. Unfortunately, the attendees heavily criticized Quine's work because it was outdated. Criticism Quine received from mathematical logic experts was a positive impulse for building a stronger foundation in his work (Quine, 1985, p. 104).

Once Quine finished his stay in Warsaw, he traveled around Europe with his wife Naomi (Quine, 1985, pp. 104-108). Quine got back to the USA in June 1933. One year after his travel around Europe, Quine's book *A system of logistic* was published.[158] Quine's book criticized Russell and Whitehead's *Principia*. According to Church (1935):

---

[156] Lukasiewickz was a logician who introduced mathematical logic in Poland. Moreover, he was one of the founders of the Warsaw School of Logic. He was minister of education in Paderewski's administration from January to December 1919. See Simons (2014) for more information.

Leśniewski was a logician that was one of the founders of the Warsaw School of Logic. Born in Russia, Leśniewski left his home country sometime after the October Revolution in 1917. He worked as a codebreaker for the Polish General Staff's Cipher Bureau during the Polish-Russian war from 1919 to 1921. See (Simons, 2015) for more information.

[157] Tarski (1901-1983) was a logician with an excellent reputation worldwide because he was considered the second greatest logician in the twentieth century after Kurt Gödel. Tarski worked on several topics, such as the completeness and decidability of elementary algebra and geometry. See Tarski see Gómez-Torrente (2019) for more information.

[158] Unfortunately, I could not review the book physically. This doctoral dissertation used a review of the book by Church (1935).

“In this book is presented a system of symbolic logic based on that of Whitehead and Russell’s *Principia Mathematica*, but involving a number of fundamental changes. The most important of these changes are: (1) the representation of functions of two or more variables as functions of one variable through the introduction, as an undefined term, of the operation of ordination, that is, the operation of combining two elements *a* and *b* into the ordered pair *a, b*; (2) the use of this same notion of predication, the proposition *ϕa*, obtained by predicating the propositional function *ϕ* of the argument *a*, being identified with the ordered pair *ϕ*, *a*; (3) the introduction in connection with the operation of abstraction, ^, of a rule of inference, the rule of concretion, which takes place of that tacit rule of *Principia* which, to speak somewhat inexactly, allows the substitution for *ϕx* in any proved expression in which *ϕ* is a free variable, of any appropriate expression containing *x*; (4) a liberalization of the theory of types, by which the axiom of reducibility is rendered unnecessary; (5) the use of the notion of classical referent, introduced by an actual nominal definition, to replace almost entirely the clumsy descriptions introduced in *Principia* as incomplete symbols; (6) the introduction, under the name of cogeneration, of the relation of implication between propositional functions, as an undefined term, out of which both the relation of implication between propositions and the universal and existential quantifiers are obtained by definition” (Church, 1935, p. 598).

Church continued:

“Nowhere in Quine’s book is there a definition of the word *significant* which is used in his statement of the rule of substitution, but from scattered remarks about types and by observation of how the rule of substitution is actually used, it is possible to surmise what probably is meant by the word. Since this is a matter of some importance especially in view of the fact that it is only through this world (or the related term *propositional expression*) that the theory of types enters the formal system at all, an explicit definition of *significant* is attempted here” (Church, 1935, p. 600).

The main difference between Russell’s theory of types with Quine's formulation is the following:

“The italicized clause in the foregoing definition marks a sharp divergence of Quine’s theory of types from that of *Principia Mathematica*; for if the analogy with the theory of types of *Principia* were preserved, *xM* could not be of lower type tan M. It is true, of course, in *Principia*, that if α is a class then a proposition of the form xεα must be of type of x, but it is to be remembered that this situation is brought about only with the aid of the axiom of reducibility, and that, in any case, the classes of Principia are incomplete symbols defined only contextually. Since Quine’s xM is not an incomplete symbol, a truer comparison of the two

> systems appears to be obtained if we compare expressions in Quine of the form xM with the propositional functions of *Principia* rather than with the classes of *Principia.* And from this point of view it is seen that, without claiming to do so, Quine has really made an important modification in the theory of types, in a direction which seems to have been first suggested by F. P. Ramsey" (Church, 1935, p. 601).[159]

One year after Quine's book was published, he reviewed concepts in mathematical logic. He began his journey by checking the concept of variable. Quine differentiated bounded and free variables. In his autobiography, Quine said that "the difference from one type of variable to the other is that bounded variables are pronouns of internal cross-references which links positions within a formula" (Quine, 1985, p. 117). Inspired by the works of Moses Schönfinkel, Quine showed the links between set theory and elementary logic.[160]

Later, in 1936, Quine published his paper "Toward a calculus of concepts" (Quine, 1985, p. 117). In his paper, Quine made his first attempt to develop a new language to eliminate most of the complications metamathematical language had, such as the restrictions bound variables imposed at the moment of doing logical calculations (Quine, 1936b, pp. 2-3). Moreover, Quine's mathematical logic language focused on seeking an alternative to the theory of types because each type excluded other types without any characteristic of being part of each type. (Quine, 1936b, p. 3; Quine, 1936b, pp. 24-25). Unfortunately, Quine's paper was a sketch of the language he wanted to develop. The same year, Quine published another paper called "Set-theoretic foundations for logic". Quine mentioned Ernst Zermelo's (1871-1953) method for avoiding logical paradoxes. Quine wanted to solve the paradoxes of set theory without using Russell's theory of types. Quine saw in Zermelo's method a way to overcome paradoxes because Zermelo devised axioms that help to classify different elements of each class more efficiently than

---

[159] Frank Plumpton Ramsey (1903-1930) was a mathematician who made several important contributions to several fields of knowledge like philosophy, mathematics, and economics. One of those contributions was a review of Russell and Whitehead's *Principia Mathematica* (MacBride et al., 2019). Ramsey believed that the only problem the *Principia* had was that certain predicative functions are indefinable (Linsky, 2019; Ramsey, 1926, pp. 372-374).

[160] Unfortunately, I had not found enough information about Schönfinkel to explain his influence on Quine's works between 1935 and 1936.

Russell's hierarchy of types (Quine, 1985, pp. 123-124; Quine, 1936a, p. 45).[161] However, Quine saw in Zermelo's system a weakness: it had limitations in deducing formulas. Quine mentioned that Thoralf Skolem's *Einige Bemerkungen zu der Abhandlung von E. Zermelo: "Über die Definitheit in der Axiomatik"* (1930) was a starting point that could help to understand how an effective deductive system worked (Quine, 1936a, pp. 45-46).[162] However, he pointed out that Tarski's work was the best option for building his system (Quine, 1936a, pp. 48-49).[163]

Quine continued his work in mathematical logic for some years after the publication of his set-theoretic foundations for logic. From 1936 to 1939, Quine was faculty instructor at Harvard University. He began his first term instructorship as lecturer in mathematical logic at graduate level in the mathematics department.

---

[161] The mathematician Zermelo influenced the development of the foundations of mathematics (O'Connor and Robertson, 1999). Zermelo's ""Untersuchungen über die Grundlagen der Mengenlehre I" (1908). In his work, Zermelo proposed the axiomatization of set theory to eliminate mathematical logic paradoxes such as the Burali-Forti paradox. Zermelo's proposal considered sets as a group of objects with certain axiomatic conditions (Zermelo, 2000, p. 199). In that spirit, Zermelo reduced set theory to seven axioms (Zermelo, 2000, pp. 200-201). While Zermelo's method does not consider the metamathematical aspects of mathematical logic more significantly, Russell's theory of types does.

[162] The mathematician Skolem (1987-1963) was interested in many academic topics like physics, chemistry, zoology, and botany. Skolem did some work in mathematical logic, especially in the field of metalogic (O'Connor and Robertson, 2005). In his paper *Einige Bemerkungen zu der Abhandlung von E. Zermelo: "Über die Definitheit in der Axiomatik"* published in 1930, Skolem wrote a critique to Zermelo's system. In 1929, Zermelo published his paper *Über den Begriff der Definitheit in der Axiomatik.* His work was a more elaborate version of the concept of definiteness that appeared in his 1908 work *Untersuchungen über die Grundlagen der Mengenlehre I* (Ebbinghaus, 2004, p. 78). Skolem, in his article, said that he reached to the same conclusion as Zermelo in a previous work (Skolem, 1930, pp. 337-338; Ebbinghaus, 2004, p. 78). The paper Skolem was published in 1923 with the name *Einige Bemerkungen zur axiomatischen Begründung der Mengenlehre* (Ebbinghaus, 2004, p. 78).

[163] Tarski's *Einige Betrachtungen über die Begriffe der ω-Vollständigkeit* (1933) presented simple and powerful a symbolic language (Tarski, 1983, pp. 279-280). Tarski was inspired by Gödel's "Über formal unentscheidbare Sätxe der Principia Mathematica und verwandter Systeme I" (Tarski, 1983, pp. 279).

Moreover, Quine delivered a seminar on philosophy of mathematics in the philosophy department (Quine, 1985, p. 125). The same year, the Mathematical Association of America, through Quine's old teacher William DeWeese Cairns, invited Quine to deliver an address (Quine, 1985, p. 126). Quine delivered his speech by the end of 1936 in North Carolina. His address served as a base for his paper “New foundations for mathematical logic” (1937) (Quine, 1985, p. 126).

Quine started his paper by reviewing Russell and Whitehead’s book *Principia mathematica*. Quine mentioned their attempt to make logic the foundation of mathematics. Quine pointed out three terms that supported Russell and Whitehead's project: translation, mathematics, and logic (Quine, 1937, p. 70). Quine said the term translation was the most basic unit to represent a statement. Moreover, the union of statements formed expressions, and expressions could be translated into context (Quine, 1937, p. 70).[164] The next concept Quine explained was mathematics. Quine said the term mathematics had to be understood in his paper as pure mathematics. Quine said the notions of logic were the basis for Russell and Whitehead’s *Principia* to construct essential concepts of set theory, arithmetic, algebra, and analysis. Moreover, Quine pointed out that Russell and Whitehead used logical relations to derive abstract algebras. Finally, Quine referred to the logic of the *Principia* as a different type of logic than Aristotle’s. Quine emphasized the notions Russell and Whitehead used in their book differently from those of traditional logic (Quine, 1937, p. 71). After Quine explained Russell and Whitehead’s work, Quine emphasized the core idea of Russell and Whitehead's project. On page 72, Quine wrote:

> “If all mathematics is translatable into the logic of the *Principia*, and this logic is to be translatable into the present rudimentary language, then every statement and statement form constructed wholly of mathematical and logical devices must be translatable ultimately into a formula in the sense just now defined. I will make the translatability of the Principia apparent, by showing how a series of cardinal notions of that logic can be constructed from the present primitives. The construction of the mathematical notions, in turn, may then be left to the Principia” (Quine, 1937, p. 72).

---

[164] Quine’s concept of translation was framed in mathematics to logic (Quine, 1937, pp. 70-71).

In page 76, Quine criticized the system developed by Russell and Whitehead:

> “The procedure in a formal system of mathematical logic is to specify certain formulas which are to stand as initial theorems, and to specify also certain inferential connections whereby a further formula is determined as a theorem given certain properly related formulas (finite in number) as theorems. The initial formulas may either be listed singly, as postulates, or characterized wholesale; but this characterization must turn solely upon directly observable notational features. Also, the inferential connections must turn solely upon such features. Derivation of theorems then proceeds by steps of notational comparison of formulas.
>
> The formulas which are wanted as theorems are of course just those which are valid under the intended interpretations of the primitive signs: valid in the sense of being either true statements or else statements forms which are true for all values of the free variables. In as much as all logic and mathematics is expressible in this primitive language, the valid formulas embrace in translation all valid statements and statement forms of logic and mathematics. Gödel has shown, however, that this totality of principles cannot be comprehended among the theorems of any formal system, in the sense of “formal system” just now described, unless that system be inconsistent. Adequacy of our systematization must then be measured by some standard short of the totality of valid formulas. A fair standard is afforded by the Principia: for the basis of the Principia is presumably adequate to the derivation of all codified mathematical theory, except for a fringe requiring the axiom of infinity and the axiom of choice as additional assumptions” (Quine, 1937, p. 76).[165]

After Quine finished his criticism of Russell and Whitehead's book, he explained theory of types developed by Russell. Quine mentioned the hierarchy of types and the variables that admit one type of class. Once he finished his explanation, Quine brought out two different types of formulas: the stratified and unstratified ones (Quine, 1937, pp. 77-78).[166] Quine saw a problem in Russell’s theory of types because it expurgated all the unstratified formulas, causing a class to be of a uniform type causing contradictions (Quine, 1937, pp. 78-79). Quine proposed a method for avoiding contradictions by eliminating the hierarchy of types and having the variables unrestricted in range. Quine limited the stratified formulas as follows: “If x does not occur in ϕ , $(\exists x)(y)((y\in x)\equiv\phi)$ ” (Quine, 1937, pp. 78-

---

[165] The axiom of infinity is an axiom that guarantees the existence of at least one infinite set (Zermelo, 1908, pp. 266-267). For the case of axiom of choice please refer to Bell (2015).

[166] An example of an unstratified formula is $x\in x$, while a stratified one is $x\in y$ (Quine, 1937, p. 78).

79). Quine ended his paper saying that his system was better than Russell's theory of types (Quine, 1937, p. 80).

Quine's "New foundations for mathematical logic" was his inspiration to write his M*athematical logic* (1940) (Quine, 1985, pp. 142-143).[167] The second edition of Quine's M*athematical logic* started with a brief prologue to the revised version of the book. Quine mentioned that his work would guide the reader step by step in helping the understanding of mathematical logic. Moreover, Quine said his book was part of the material he taught at Harvard University in course nineteen (Quine, 1972, p. 11).[168] Later, Quine explained each one of the contents of his book.[169] After the contexts and prologue, Quine introduced some aspects of mathematical logic to help the reader to understand the very basics of the subject. Quine explained the difference between formal logic and mathematical logic in its formal and mathematical form. Quine also mentioned that logic was quite similar when using logical vocabulary for constructing logical statements (Quine, 1972, pp. 19-21). Moreover, in the introduction he explained how logical language explained the meaning of truth. Quine said that truth in logic could be understood in terms of notational features of the complexity of logical statements (Quine, 1972, pp. 21-23).

After the introduction, Quine explained what statements are in the first chapter. The first part of chapter one treats connectors used on statements to form new ones. Statements are the most basic units combined by logical connectors like conjunction and disjunction (Quine, 1972, p. 27). Quine used a truth table to explain different values each connector produces.[170] Quine also emphasized the importance of distinguishing an object and its name. He mentioned the importance of making that distinction when referencing the name of the object (Quine, 1972,

---

[167] For this doctoral thesis, the second edition of Quine's *Mathematical logic* (1951) was used.

[168] In 1941, one year after Quine wrote the first edition of his *Mathematical logic,* he became a tenured associate professor at Harvard University (Quine, 1985, p. 146).

[169] For the case of this thesis, the first for chapters of Quine's *Mathematical logic* were reviewed.

[170] Quine used truth tables to explain how each connector of statements worked. See Quine's *Mathematical logic* from pages 27 to 39 for more information.

p. 39).[171] Quine suggested the use of quotation marks for distinguishing objects and its names (Quine, 1972, p. 42). Finally, Quine finished the first chapter of his book, emphasizing the formation of statements.

The next chapter of Quine's M*athematical logic* was related to quantification. The second chapter started with the problem of translating natural language that mathematical logic vocabulary used for evaluating statements. Quine pointed out the importance of using simpler notation for a more straightforward evaluation of the statements (Quine, 1972, pp. 79-81). Moreover, Quine mentioned the type of symbolism that represented an object used in mathematics for generalization (Quine, 1972, p. 82). Quine also said that prefixes that contained variables were known as quantifiers, while their use in the formation of statements was known as quantification (Quine, 1972, p. 84).[172] Later, Quine said that statements used in mathematical logic are not as simple as in formal logic, but rather a matrix of statements that depended on the structure of the statement, and the values the variable of the statement had. The process resulted in true and false outputs (Quine, 1972, p. 85). The next point Quine emphasized was finding the truth of the statements based on their structure (Quine, 1972, pp. 94-99). Later, Quine explained what theorems are in mathematical logic. Quine mentioned that just using axioms of quantification did not consider some logical truths. The combination of the axioms of quantification with their likely results in what is called theorems (Quine, 1972, pp. 99-100).

Moreover, Quine distinguished between a statement, an axiom of quantification, and a theorem (Quine, 1972, pp. 100-101).[173] Another element Quine considered in the chapter was the existential quantifier. He explained the existential quantifier is used for showing the existence of at least one feature that satisfies a particular

---

[171] Quine put as an example naming a city as the problem of pointing an object to its lexical representation (Quine, 1972, pp. 39-41).

[172] The variables used in mathematical logic are the same for mathematical algebra and analysis like $w, x, y,$ and z (Quine, 1972, p. 84).

[173] In pages 100 and 101 of his M*athematical logic*, Quine said: "The tautological statements form a class of logical truths, the axioms of quantification form a wider class, and theorems for a class which is the widest." (Quine, 1972, pp. 100-101). Unfortunately, Quine said that the difference between theorems and the other two elements was that theorems could not prove that statements were part of a class or not (Quine, 1972, p. 101).

condition in the statement. Quine also contrasted the universal quantifier with the existential one (Quine, 1972, pp. 115-116).

The third chapter was related to terms used in mathematical logic. Quine started the chapter with the concept of class and membership. Quine explained that symbol ε represents the connection between an object and its membership in a class.[174] Quine also added an essential consideration about members and classes. Quine said that classes are similar if all their objects are the same. However, there is no agreement if objects have the same properties (Quine, 1972, pp. 129-131). After introducing classes and objects, Quine explained what logical formulas are. Quine said that logical formulas are atomic formulas (Quine, 1972, p. 134). Quine also compared logical formulas with logical statements, saying that logical statements could not be atomic because they do not have free variables (Quine, 1972, p. 135). Finally, Quine explained what abstraction is in the context of mathematical logic. Quine separated two terms: abstraction and abstract. The first is related to the formation of the names of the classes represented by ŷ, while the other is the result of a class (Quine, 1972, p. 142).

Quine treated the theory of classes in next chapter of his book. Quine started chapter four by mentioning symbol ε. Quine said that introducing a new feature needed a new set of axioms, like the axiom of belonging. Moreover, Quine extended the concept of theorem to include potentials of axioms (Quine, 1972, pp. 163-164). Quine continued by saying that new axioms are essential to help to understand the reasons why a class could become an element (Quine, 1972, p. 164). Quine continued:

"On the one hand, we should decide whether the notion of a theorem which enclosure a description that has to be understood in restricted sense or a wider sense: potentials of axioms of quantification, potentials of axioms of belonging, etc. In a wider sense, it might vitiate the suggested explanation of the axioms of belonging for its circularity because the term 'theorem' of the explanation would presuppose what the axioms of belonging would be or its condition as a member. We must choose, consequently, the restricted sense" (Quine, 1972, p. 164). After

---

[174] A good example of Quine's point is the following. "Socrates is mortal" had the elements Socrates and mortal. Socrates was the object and mortal the class. The use of ε in the example of Socrates was the following "Socrates ε mortal", meaning that Socrates belongs to the class of mortal.

that, Quine explained how the formulas were stratified (Quine 1972, pp. 166-167).[175]

In 1949, Wang challenged Quine's notion of stratification (Quine, 1985, p. 146; Quine, 1972, pp. 167-168). Wang's criticism to his doctoral advisor happened a few years after he obtained his Ph.D.

8.2 The controversy between Herbert Simon and Hao Wang

Hao Wang was born in 1921 in Jinan, China. Both of his parents received the influence of the revolutionary ideas of Sun Yat-Sen (Mar, 2015, p. 540).[176] Wang

---

[175] In pages 166-167, Quine discussed his idea of class stratification.

[176] During the nineteenth century, China suffered several invasions from Western powers, Japan, and Russia (Buckley, 2010, pp. 252-257). As a result of the aggression of foreign nations, some persons envisioned the idea of a united China to protect themselves from external aggression (Buckley, 2010, p. 262). The Qing dynasty, weakened by the events of the second part of the nineteenth century, was overthrown in 1911. The result of the end of the Chinese monarchy was the creation of the Republic of China. Yat-Sen was the founding president of the new state (Buckley, 2010, pp. 265-266; Lee et al., 2013, p. 358).

Yat-Sen was born in 1866 in the county of Xiangshan to a family of peasants. Yat-Sen went to study medicine at Canton Hospital Medical School, considered the first modern hospital in China. A year after, Yat-Sen was transferred to College of Chinese medicine in Hong Kong, getting in touch with modern medicine. He graduated in 1892 (Lee et al., 2013, pp. 356). Because he could not exercise medicine in Hong Kong, Yat-Sen went to Macau to pursue his career in medicine. Unfortunately, his qualification did not allow him to exercise the profession in Macau, but he found a job at the Kian Wu Hospital. In that institution, he could exercise Chinese traditional medicine. In 1893, Yan-Sen moved to the city of Guangzhou. There, he practiced medicine as a mixture of modern medicine with Chinese traditional medicine. After a time, Yan-Sen was frustrated because people preferred traditional Chinese medicine instead of modern medicine. His frustration made Yan-Sen see the importance of modernizing China (Lee et al., 2013, pp. 356-357). After a time, Yan-Sen left medicine, getting interested in politics.

During his time in Guangzhou, Yan-Sen reconnected with friends and colleagues that shared the idea of modernizing China. The group's ideas influenced Yan-Sen's political reformist ideas, concluding in the presidency of China in 1911. Unfortunately, he was in power for a went to study at Southwest Associated University in Kunming in 1939, graduating in mathematics. Later, he earned his master's in philosophy at Tsinghua University in 1945 (Parsons, 2011, p. 7; Wang, 2011a, p. 27; Mar, 2015, pp. 540-541).[177]

During his time at Southwest, Wang's mentor was one of most important logicians of twentieth-century China Jin Yuelin (Wang, 2011a, p. 31).[178] Wang was interested profoundly in studying philosophy during his bachelor years at Southwest Associated University. During his third year, Wang considered mathematics less important than philosophy, focusing his energy on studying David Hume's problem of induction. Wang wrote a long paper one year after he read Hume. He sent his work to professor Yuelin, who said Wang plagiarized it. Wang defended himself, saying that the paper was his work. Yuelin suggested to Wang his work should be published. However, Wang did not follow Yuelin's advice (Wang, 2011a, p. 31).[179] When Wang went to study for his master's at Tsinghua University, his thesis advisor was Yuelin (Wang, 2011a, p. 33).

After Wang finished his studies in 1945, he was ready to travel to the USA to pursue his Ph.D. in philosophy at Harvard University under the supervision of Quine.[180] Wang arrived in the USA in 1946 to pursue his doctoral studies in logic

---

few days because Qing general Yuan Shikai became first formal President of China after negotiating with revolutionary forces of the Republic. The decision was taken after the menace of foreign intervention (Buckley, 2010, pp. 265-266).

[177] During the time Hao Wang was studying to get a bachelor's degree, China was at war with Imperial Japan (Buckley, 2010, pp. 282-286). The war of resistance against Imperial Japan started in 1937 and ended in 1945.

[178] Yuelin (1895-1984) is considered one of the most important and influential philosophers of China. Yuelin studied political science at Columbia University in 1920 after writing his thesis, *The political theory of Thomas Hill Green* (Thompson, 2013, pp. 40-41). He returned to China in 1927 to teach at Tsinghua University (Wang, 2011a, p. 28). Yuelin was part of a group of intellectuals who wanted a change in the intellectual activities in China (Buckley, 2010, pp. 267-272). See Thompson (2013) and Wang (2011a) for more information about Yuelin.

[179] During his studies at Columbia University, Yuelin entered into contact with the ideas of David Hume. He got fascinated with Hume's *A treatise of human nature* (1738-1740) (Thompson, 2013, p. 41). Hume's work influenced Yuelin's work during the rest of his academic career.

[180] During the time Wang was a student, the USA and the Chinese government reached an agreement. The USA was going to receive students from China to help the country to

and philosophy (Parsons, 2011, p. 7). He finished his Ph.D. in one year and eighteen months (Zhaowu, 2011, pp. 49-50). Wang's *An economical ontology for classical arithmetic* proposed a method for reducing classical arithmetic to a logistic system. The works of Russell, Whitehead, and other logicians interested in making logic the foundation of mathematics inspired Wang (Dick, 2014, 113; Mar, 2015, p. 541).[181] After finishing his doctoral work, Wang became junior fellow of the Society of Fellows at Harvard University. He remained fellow of the society until 1951. During that time, he found an inconsistency in Quine's *Mathematical logic*. Wang proposed a solution to the mistake (Mar, 2015, pp. 541-542; Parsons, 2011, p. 7).[182]

In 1950, Wang left the USA to work with the mathematician Paul Bernays (1888-1977) in Zurich (Mar, 2015, p. 542).[183] From 1950 to 1951, Wang worked with Bernays. During that time, Wang became interested in research related to the relative strengths of axiomatic set theories. Moreover, Wang revived Hermann Weyl's work on predictive mathematics (Parsons, 2011, p.8; Mar, 2015, p. 542).[184]

---

modernize (Wang, 2010, p. 369). Wang went to study with a scholarship from the U.S. State Department (Mar, 2015, p. 541).

[181] Unfortunately, I could not access Wang's doctoral dissertation. I refer to Dick (2014) for mentioning Wang's Ph.D. thesis.

[182] According to Quine (1985), Wang reviewed the first edition of his *Mathematical logic* (1940). Wang said that the solution to the mathematical logic problem Quine had was the limitation of the bound variables because they should be limited to their values (Quine, 1985, p. 146). Unfortunately, for writing this Ph.D. thesis, I could not access the Quine archive located in the Houghton Library at Harvard University. The only way to find Wang's critique of Quine's *Mathematical logic* was inside Quine's autobiography.

[183] Bernays was a mathematician interested in researching mathematical logic. Bernays was working in proof theory and axiomatic set theory. He worked at the University of Göttingen with mathematician David Hilbert from 1917 until 1933. During his last year at Göttingen, Nazis took power. For that reason, Bernays escaped to Switzerland. During his time in Zurich, Bernays worked in set theory, focusing his attention on Zermelo-Fraenkel axioms (Rodriguez, 2017b).

[184] Weyl (1885-1955) was one of the best mathematicians of the twentieth century. One of his most important ideas, according to Parsons (2011), was that "mathematics might be developed in a way that avoids impredicative set existence assumptions" (Parsons, 2011, p. 8).

After his stay in Zurich, Wang returned to the USA to be assistant professor in the philosophy department at Harvard University until 1956. The same year, Wang became a reader in philosophy of mathematics at University of Oxford (Dick, 2014, p. 129). In autumn of 1954, Wang went to University of Oxford, and in spring of 1955, Wang gave six lectures about formalizing mathematical concepts (Mar, 2015, p. 542; Wang, 1990, p. xvi). The same year, Wang was invited to deliver the John Locke lectures. Moreover, he remained a reader in the department of mathematics at Oxford from 1956 to 1961 (Dick, 2014, p. 129; Mar, 2015, p. 542).

Everything seemed well on the surface, but Wang was having some trouble during that time because political relations between the USA and China were tense.[185]Because of the Korean War, Chinese intellectuals and scientists who went

---

[185] During the government of Chian Kaishek (1888-1975) in last part of first half of the twentieth century, the Chinese nationalist government had a problem with legitimacy. Government corruption, spiraling inflation, poverty, and other social issues produced intense anger in Chinese population. Those issues are linked to the rise of the Communist Party in China on October 1, 1949, after a civil war. Mao Zedong took power, giving birth to the People's Republic of China (Buckley, 2010, pp. 286-294).

The relationship between the USA and the Chinese Communist party was very taut from the beginning. Even though Chinese socialism differed from Soviet socialism, the USA did not differentiate one from the other. While the Soviet Union defended the 'dictatorship of the proletariat', the 'people's democratic dictatorship' of China was more plural than the Soviet Union. Rich peasants, national bourgeoisie, and others that shared the ideal of the Chinese Communist Party were part of people in power (Buckley, 2010, p. 295).

The USA did not consider that difference since the end of World War II. Before World War II started, Chinese nationalists and socialists were at war. After the purges Chinese nationalists did from 1927 to 1928, Chinese socialists were in a disadvantaged position. Few surviving members of the socialists saw in Mao Zedong a leader that could help their political idea to become a reality (Buckley, 2010, pp. 286-287). Over time, Mao started to convince the poor and peasants to support socialists' ideals. When World War II ended, and with the surrender of Imperial Japan after the devastating effects of two atomic bombs thrown over Hiroshima and Nagasaki, the final confrontation between Chinese nationalists and socialists was going to happen (Buckley 2010, pp. 288-290). Mao tried to convince President Roosevelt that Chinese Communist Party was not a threat to U.S. interests. However, President Roosevelt did not want to receive Mao, showing the U.S. sided with Chinese nationalists (Yufan & Zhihai, 1990, p. 95). After Roosevelt's death, the next to the USA to study at U.S. universities were suspicious of supporting the Chinese regime. Before the Korean War started, the USA attracted several Chinese students to study in its universities. Nearly 3914 students were enrolled at U.S. universities and private institutions from 1948 to 1949. Moreover, students from China were the second largest group of foreigners studying at U.S. universities after Canadians (4166) (Han, 1993, pp. 78-79). Even when Chinese students had financial problems during the second half of 1948, the U.S. Government took care of them through the U.S. State Department[186] When the People's Republic of China was founded, Chinese nationalists cut all the aid to Chinese students living abroad. The USA considered Chinese students living in the U.S. territory as

---

president of the USA, Harry Truman, continued with the same policy the USA followed: helping Chinese nationalists and keeping distance from Chinese socialists (Yufan & Zhihai 1990, p. 96).

The USA and China fought in the Korean War. When World War II ended, Korea gained independence from the Japanese Empire. Moreover, Korea was divided into two parts. The first was named Republic of Korea, today known as South Korea, occupied by the USA. In contrast, the second part was baptized as Popular Democratic Republic of Korea, today known as North Korea, occupied by the Soviet Union (Cumings, 2010, p. xii; Stueck, 1995, pp. 3-4). The Korean War started on June 25, 1950, when North Korea attacked South Korea (Buckley, 2010, p. 297; Yufan & Zhihai; 1990, p. 99). The USA intervened in this conflict, supporting South Korea since the beginning. China saw USA intervention as problematic because the USA could attack directly to China. The reason China saw a threat of USA intervention in the Korean War was a political one because China was an ally of the Soviet Union, so that the USA could attack China. Moreover, the USA supported Chinese nationalists that retreated to Taiwan after the victory of Chinese communists in mainland China (Buckley, 2010, p. 326; Yufan & Zhihai,1990, p. 100). Mao saw the support of the USA to Chinese nationalists as a threat because Chian Kaishek and Chinese nationalists wanted to retake mainland China (Buckley, 2010, p. 326; Yufan & Zhihai, 1990, p. 104). For those reasons, Mao decided to support North Korea. The Korean War ended in 1953, with an intense hostility of the USA to China. A growing sentiment toward anticommunism and an economic embargo as the U.S. foreign policy toward China were the results of the conflict (Buckley, 2010, p. 297; Cumings, 2010, p. 34; Wang, 2010, p. 370).

[186] The financial situation in China was deteriorating. Because of China's civil war, Chinese students living in the USA had financial problems because the Chinese government and its financial institutions cut off their funding for them (Han, 1993, pp. 79-80).

influence over the Chinese government. For that reason, the U.S. Department of State continued to fund them (Han, 1993, p. 80).[187]

The situation changed dramatically when the Korean War started because the U.S. Government banned the entrance of Chinese students into U.S. territory. Moreover, the Truman Administration enacted three laws to prevent Chinese students living in U.S. territory from returning to China (Han, 1993, pp. 81-82). The first was the Public Law of 1950. Section two of the act provided six million Dollars to help students alleviate their problems with the only condition they must return to China when they finish their studies. However, the act was flexible because it permitted students to stay in the USA for three years if they failed to maintain full-time student status or finish their studies even if they could not find a job (Han, 1993, p. 82).

The other law passed was more substantial than the Public Law of 1950. On June 27, 1952, two years after the Public Law of 1950 passed, the immigration and nationality act, also known as the McCarran-Walter act, was enacted. Section 215(a) explained that a citizen from another country could not leave the USA if the person was considered a menace to U.S. interests. Moreover, section 215(c) complemented section 215(a) by mentioning that disobedience of the law would result in a conviction and/or a fine of five thousand Dollars (Han, 1993, p. 83).

Because the McCarran-Walter act was very strict with foreigners, on August 7, 1953, a new law passed that relaxed the situation of Chinese students a little bit. The U.S. congress approved the refugee relief act in 1953. This law helped to relieve some problems Chinese faced since 1949 after Chinese Communist Party took power in China. The act permitted Chinese students to apply for permanent residency in the USA for those who arrived before July 1, 1953 (Han, 1993, pp. 86-87). Unfortunately, those Chinese students who did not want to apply for permanent residency in the USA had the risk of being deported to China. The problem was that the USA and China did not have any political relations,

---

[187] The financial situation in China became complicated on September 22, 1948. The Chinese embassy in Washington asked the Chinese students to leave the USA with government aid. Unfortunately, a strike in the shipping yards of the west coast made it almost impossible to the Chinese students to return to mainland China. The U.S. Congress passed several acts to help students with financial aid (Han, 1993, pp. 79-80).

producing a problem for those Chinese students who wanted to return to China (Han, 1993, p. 87).

Even though the law regularized Chinese students, some faced problems with the law because the U.S. government believed that Chinese students were communists. Some Chinese graduates, like physicist Xie Jialin (1920-2016), were subjects of distrust by the U.S. Government.[188] When Jialin and other eight students were leaving the USA, U.S. authorities stopped the ship with the students going back to China. Later, U.S. immigration officials and FBI agents interrogated them. Chinese students were forced to get back to the USA. They were detained until 1955 (Han, 1993, p. 92).

Wang suffered this kind of problem when he was going to travel to the UK to join Oxford University as faculty member. On September 12, 1956, Wang was departing to the UK on Queen Mary sail after attending the summer research school at Burroughs Corporation in New York City.[189] Before the ship departed, the FBI stopped Wang from boarding the ship. The FBI hypothesized Wang would meet with Chinese government officials to give them nuclear elements he supposedly was carrying in his luggage. After checking Wang's luggage, the FBI

---

[188] Xie Jialin was born in Heilongjiang, China, in 1920. Jialin obtained his master's degree in physics at Caltech in 1948. After that, he earned his Ph.D. in physics at Stanford University. See Zhang (2012) for more information.

[189] Wang was interested in how computers could solve problems of mathematical logic. Wang was dissatisfied with how mathematical logic was taught at Harvard. Wang believed that computers were closer to his training as a philosopher. Moreover, Wang knew that computers were conceptually elegant (Wang, 1990, p. 63). For that reason, Wang got in contact with Burroughs Corporation so that he could develop his mathematical logic ideas. Burroughs appointed Wang as a research engineer in 1953. Unfortunately, He was not allowed to use local computers. Moreover, he was discouraged from taking electronic technician courses (Wang, 1990, pp. 63-64). The only work Wang was allowed to do at Burroughs was speculating on how to program Turing machines and had some thoughts about theorem-proving machines (Wang, 1990, p. 64). His work as a research engineer ended in 1954 (Dick, 2014, p. 114).

Burroughs Corporation was a computer industry producer based in the USA. Like IBM, Burroughs started as a producer of office machines. Burroughs' first computer was completed in 1950 and constructed in the company's research center in Philadelphia in 1951 (Yost, 2013, p. 7).

just found computer components. Wang told authorities that those computer elements were going to be delivered to his friend, physicist Wen-Yu Chang (Dick, 2014, pp. 111-112).[190]

Wang's precedent was significant to understanding why the artificial intelligence community did not consider Wang's work in AI until the 1980s when he received recognition for his work related to theorem provers.[191] From 1956 until 1961, Wang was a reader in philosophy of mathematics at Oxford University. During that time, Wang did not just work at the university but also as visiting researcher and consultant at International Business Machines Corporation (IBM) and Bell Laboratories.[192] In both corporations, Wang came in contact with computers, having a real possibility to implement his ideas related to logic and mathematics.

The result of such work in both corporations was developing a computer program that could prove mathematical theorems (Dick, 2014, pp. 115-116). During the

---

[190] Chang was a physicist considered one of the leading figures of high-energy physics community. Graduated from Yenching University, he spent three years at Cambridge University studying with physicists Ernest Rutherford (1871-1937), Charles Ellis (1895-1980), and John D. Cockcroft (1897-1967). Chang also worked at Princeton University from 1943 to 1950. He joined the Palmer Lab at Princeton, discovering the first muonic atom (Hu, 2016, p. 296).

[191] There was an interesting point in the development of theorem provers. Philosophers, mathematicians, and computer scientists interested in proving theorems were trying to understand the connection between human intelligence and mathematical proofs (Davis, 2011, p. 75).

[192] IBM is one of the largest computer corporations based in the USA. The company was born in the second half of the nineteenth century, thanks to the invention of punched cards developed by Herman Hollerith (Cortada, 2019, p. 15). The punched card invention was a huge success in the beginnings of IBM because several of them helped to process the data of nearly 63 million people living in the USA during the U.S. national census in 1890 (Cortada, 2019, p. 17). But the most significant economic take-off of IBM happened after World War II when the government of the USA gave financial support for the development of the computer industry. An example of the policy that promoted the computer industry in the U.S. was building computers for being used at university campuses across the USA (Aspray & Williams, 1994, p. 60).

Bell Laboratories, also known as Bell Labs, is a corporation based in the USA. The expertise of Bell is the development of electronic technology. The origins of Bell Labs can be traced back to 1907. Bell Laboratories produced several essential contributions to science and technology, like the transistor in 1947 (Lotha, 2022b).

1960s, Wang published three journal articles on theorem provers linked to his work at IBM and Bell Laboratories. The first was “Toward mechanical mathematics” (1960), published in the *IBM research journal*. Wang's work was his first attempt to develop a computer program that could solve theorems of mathematical logic. Wang started his paper by mentioning that his program was a better problem solving than Herbert Simon and Allen Newell’s LT. Wang said his program could find proofs of nearly the 400 theorems of Russell and Whitehead's *Principia Mathematica.* In comparison, the LT could only find proofs for 38 theorems (Wang, 1960a, pp. 2-3).[193] Wang also compared the time consumed by the computer when processing the theorems. Simon and Newell's LT was programmed on the Johnniac, while Wang's System P was developed on an IBM 704 (Wang, 1960a, p. 3). Three parts had Wang’s System P. The first could prove the whole 200 theorems of the first five chapters of the *Principia* in less than three minutes, while the LT tried to prove 52 theorems chosen by Simon in 5 minutes (Wang, 1960a, p. 3).[194] The second part of System P focused on forming propositions from propositional calculus. The program used basic symbols and non-trivial theorems to form propositions. The result obtained by the second part of System P was the formation of 14000 propositions in one hour (Wang,1960a, p. 3). The last part of System P worked over the last five chapters of the *Principia.* The results were astonishing because System P could prove more than 150 theorems of the book (Wang,1960a, p. 3). After showing promising results, Wang wrote that his technique was superior to the LT.[195] Wang said that his technique derived from cut-free formalisms of predicate calculus developed by Gerhard Gentzen (1909-1945) and Jacques Herbrand (1908-1931) (Wang, 1960a, p. 4).[196]

---

[193] Wang's program was System P (Dick, 2014, p. 35).

[194] The 52 theorems that Simon and Newell chose to be solved by the LT were among the easiest ones (Wang, 1960a, p. 3).

[195] Wang said the technique used by Simon and Newell was not useful for theorem proving because heuristics did not guarantee solving successfully a given problem (Wang, 1960a, p. 4).

[196] Gentzen was a logician who studied in different German universities during the first half of the twentieth century. Gentzen began his journey in 1928 at the University of Greifswald to study mathematics. Gentzen finished his bachelor's studies in Berlin in 1931, returning to the University of Göttingen to start his doctoral studies under the supervision of Hermann

Moreover, Wang used the standard decision methods for the subdomains of predicate calculus proposed in Quine's *Methods of logic* (1950).[197] The second paper Wang wrote about automatic theorem proving was "Proving theorems by pattern recognition I" (1960). Wang's paper showed a new method for proving or disproving theorems. Wang achieved his proposal using pattern recognition techniques (Wang, 1960b, p. 220). Wang presented his new method as Program P.[198] The potential of Program P was proved the mathematical logic theorems found in the *Principia.* The results were astonishing because Program P could solve the entire 350 theorems that belonged just to the domain of mathematical logic in 8.4 minutes (Wang, 1960b, p. 220). Wang used the method Herbrand developed for pattern recognition (Wang, 1960b, p. 223).[199] He mentioned that Herbrand created a system for solving decision problems of mathematical logic. Wang also said pattern recognition could extend to quasi-decision procedures related to predicate calculus (Wang, 1060b, pp. 223-224).

Wang developed his idea in a much more detailed manner in his third paper, "Proving theorems by pattern recognition II" (1961). In his third paper, Wang extended his pattern recognition method to the decision problem. Wang started his

---

Weyl. He obtained his doctorate in 1933 (O'Connor, 2001).Gentzen's main contribution to the realm of mathematical logic was his work called natural deduction (Gentzen, 1935, p. 176). Gentzen criticized Frege, Hilbert, and Russell's works because they mostly eliminate deduction (Gentzen, 1935, p. 176). Gentzen developed his natural deduction on the second section of his "Untersuchungen über das logische Schließen I" (1935).

Herbrand was a logician born in Paris in 1908. He entered the Ecole Normale Supérieure to study at the age of seventeen. During his bachelor's studies, Herbrand became interested in studying mathematics, paying particular attention to *Principia mathematica*. Herbrand received his *Agrégation* in 1928. One year later, Herbrand obtained his doctorate in mathematical logic. Herbrand's thesis supervisor was Ernest Vessiot. Unfortunately, Hebrand died while he was young, producing a significant loss in mathematics (O'Connor, 2006). The legacy he left was his doctoral dissertation published in 1930 under the name *Reserches sur la théorie de la démonstration.* Herbrand's thesis focused on reducing first-order logic to propositional logic (Haaparanta, 2009, pp. 398-400; Herbrand, 1930, pp. 1-7).

[197] Gentzen's natural deduction inspired Quine (Quine, 1974, p. 207).

[198] Program P was developed in an IBM 704, same as System P (Wang, 1960b, p. 220).

[199] In Hebrand's "S*ur le problème fondamental de la logique mathématique*" (1931) a method for finding truth propositions based on recognizing some elements propositions had was developed (Herbrand, 1968, pp. 169-170).

paper by distinguishing between problems of provability and satisfiability. The first was whether the program dealt with a theorem. In contrast, the second part was related to knowing whether the negation of the predicate calculus had any model at all (Wang, 1961, p. 1). After that, Wang explained why focusing on the decision problem was important. Wang understood the impossibility of developing a procedure that could apply predicate calculus to solve problems in a series of steps, but he proved the possibility of satisfying specific problems (Wang, 1961, pp. 1-2).Wang focused on the reduction problem after explaining the importance of the decision problem (Wang, 1960, p. 2).

In his paper, Wang showed theoretically how a program could solve the problem of satisfiability. Even though Wang's method was superior to LT’s heuristics, his work was not recognized by the artificial intelligence community. In the 1990s, Wang's friend Eckehart Köhler talked with Wang about mathematical discovery and theory formation. Köhler said to Wang that Simon was getting excellent results regarding scientific discovery using artificial intelligence methods. Wang exploded, saying that Simon was a terrible logician. Köhler decided to change of topic of conversation after seeing Wang’s reaction (Köhler, 2011, p. 63). After the discussion with Wang, Köler considered that Wang was just interested in narrow topics. Simon had several interests, while Wang did not (Köhler, 2011, pp. 64-65). Unfortunately for Köhler, Wang did not say something important to him about his frustration of not being considered by the artificial intelligence community. In Wang’s “Computer theorem proving and artificial intelligence” (1984), an explanation of the possible reason why the artificial intelligence community ignored Wang is written as follows:

> “[...] Critics of AI quote the less interesting part of my work only in order to berate the unprofessional job of Newell-Simon-Shaw. (e.g., Dreyfus, first edition, p. 8 and Weizenbaum, p. 166). (By the way, Dreyfus reduces all the 220 theorems to only the 53 selected by Newell-Shaw-Simon). They miss the central point that although the predicate calculus is undecidable, yet all the theorems of *PM* in this area were so easily proved. Enthusiasts for AI simply leave me alone. The professional writer McCorduck at least makes a bow to theorem proving: 'and I rationalize such neglect by telling myself that theorems would only scare away the nonspecialist reader this book is intended for'. This incidentally leads to some other factors which render the field of AI controversial. It incites popular interest yet, unlike physics, it is a more mixed field in which natural scientists and social

> scientists (not to mention philosophers) with quite different standards meet. More, since it is near big technology, it is close to industrial and government money. As a result of all these factors, public relations tend to play a larger role than in more mature disciplines. And that tends to put some people off even though they find the intellectual core of many problems in the area appealing and challenging. Exaggerated and irresponsible claims and predictions, instead of being chastised, appear to be a central ingredient of the glory of many of the 'giants' in this field” (Wang, 1990, p. 69).

Wang was right to emphasize the U.S. government’s funding of AI. Advanced Research Projects Agency (ARPA) was the primary source of funding early AI (McCorduk, 1979, p. 110). The suspicion of Chinese citizens that lived in the USA during most of the Cold War of being informants to the Chinese regime produced suspicion by the U.S. government. The USA wanted to protect its military secrets from countries part of the socialist sphere.[200] After some years, the tension between the USA and China reduced. Wang’s reception of the milestone award for automatic theorem proving in 1983 is the best example of the reduced political tension between China and the USA.[201] Unfortunately, Wang’s work took a different route than AI. He formalized the problem of computational intractability developed by Alan Turing in mathematical logic.[202] Wang's work influenced the development of the computational complexity field, being Wang's Ph.D. student, the computer scientist Stephen Cook, the formalizer of the tractable/intracatble problem in computational complexity. The formalization of computational intractability happened in computer science, not in AI, for political reasons.

---

[200] Artificial intelligence was a project funded mainly by the U.S. government for military aims (McCorduck, 1979, p. 110).

[201] During Chinese cultural revolution, China feared an invasion from the Soviet Union. In trying to stop a potential war between China and the Soviet Union, China sought to improve its international relations with the USA. China's strategy worked. In 1972, the president of the USA at that time, Richard Nixon, visited China (Buckley, 2010, pp. 319-320).

[202] Please check part 9.1 of this thesis.

Conclusions

This chapter presented the dispute held by Simon and Wang related to automatic theorem provers. The relation between politics and early AI development during the Cold War was associated with military and national interests. In that spirit, Wang's work was ignored by the early artificial intelligence community. As a result, Wang changed his priorities, focusing on the decidability problem found in mathematical logic using his System P.

This chapter began with a brief reconstruction of Quine's life. As explained, Quine's interest in mathematical logic began after getting in touch with Russell and Peano's axiomatic methods. Quine's decision to study his postgraduate education at Harvard University was linked to his interest in learning from Whitehead.

After Quine finished his studies in the USA, he traveled to Europe to learn the best mathematical logic methods from experts in the field during the first half of the twentieth century, such as Carnap and Tarski. The travel Quine did to Europe expanded his mind after socializing with the Vienna Circle and the Warsaw School of Logic, shaping his view of mathematical logic. However, his project was incomplete because he could not build a system of mathematical logic that could rival Russell's. Quine's luck changed when he got in contact with Zermelo's method for avoiding paradoxes. Influence of Continental and British logicians helped Quine develop his mathematical logic system. Unfortunately, Quine's system of mathematical logic had inconsistencies. His former doctoral student, Wang, helped him to eliminate those contradictions.

Professor Yuelin was one of the most important intellectual influences on Wang. Maybe Yuelin guided the decision of Wang to go to study in the USA because Yuelin studied at Columbia University during the 1920s. Furthermore, before the political tension between the USA and China in the 1950s, Chinese students were encouraged to study at U.S. universities.

Quine also was an important influence in Wang's work after he finished his doctoral studies. With Quine, Wang deepened his study in mathematical logic. Maybe the influence of Chinese universities and Harvard helped Wang correct the mathematical logic system proposed by Quine in 1940. Perhaps Quine's influence was stronger than Yuelin's because Wang continued a mathematical logic career the following years after finishing his doctoral studies at Harvard University.

Wang went to Zurich to work with Bernays in axiomatic set theory from 1950 to 1951. He also went to Oxford as a reader in philosophy from 1956 to 1961. Even though Wang was a brilliant philosopher with a bright future, his career was constrained to mathematical logic because the tension from 1950 to 1972 between the USA and China, limiting Wang's professional options.

The U.S. military financed AI during its first years because the field would serve U.S. national security and military interests. Wang could not enter the field of AI because the political relations between China and the USA were tense after the Korean War. The limits imposed on Chinese professionals in U.S. territory after the Korean War changed Wang's plan, focusing on studying the decision problem. Program P was the beginning of Turing's computational problem to be formalized in the tractable/intractable distinction. How did Wang's work influence the formalization of the P/NP distinction in computational complexity? Chapter 9 has the answer to that question.

Chapter 9
The birth of computational complexity

This chapter aims to explain the birth of computational complexity historically. Following chapters 1 and 8, this chapter will explain the influence philosopher Hao Wang exercised over his doctoral student Stephen Cook. Why is Wang's work vital for Cook? Because Wang discovered the impossibility of solving problems of predicate calculus in a few steps using Jacques Herbrand's pattern recognition, as explained in chapter 8.

Wang's research results were similar to Turing's computability, resulting in the formalization of computational intractability in the field of computational complexity. Cook used Wang's tractable/intractable distinction in his paper "The complexity of theorem proving procedures" (1971) to conceptualize computability theory. How Cook's work is linked to Turing's? Because Turing machines were the gold standard for studying the problem of computational complexity during the 1960s. While Cook was Wang's doctoral student, he focused his energy on showing how Turing machines helped to understand how humans solve addition and subtraction problems Has artificial intelligence (AI) any relationship with computational intractability? Yes, but it is not explicit at all. One has to study the relationships of both fields through games that shaped the study of human decision-making to make sense of the link between algorithmic problem solving with computational intractability. While chess was the basis for studying human problem solving, dominoes was for computational intractability. Why was the problem of computational intractability discovered outside ai if games such as chess and dominoes shaped the rationality of decision-making and problem-solving? The constraints imposed on research during the Cold War are the answer. When the AI community ignored Wang, he focused his energy on studying the decision problem, as chapter 8 explained.

The Wang's Program P influenced the discovery of computational intractability. Furthermore, Cook's P/NP distinction was formalized in computational complexity, part of the field of computer science. Cook developed his carrer doing computational complexity research. As was explained in chapter 2, computer science was institutionalized during the 1960s, while AI was in 1956. At least one decade separates both fields. Cook did not work within AI.

Historical studies related to computational complexity historical development are very scarce because the field has been studied mainly from technical and scientific standpoints, such as such as Stuart Russell and Peter Norvig's *Artificial intelligence: A modern approach* (2010).[203] Michael Mahoney's "Computers and mathematics: The search for a discipline of computer science" (1992) and "Computer science: The search for a mathematical theory" (1997) are pioneer works in that respect because they contextualized the development of Cook's P/NP distinction in intellectual history. They focus on Turing machines influence in measuring complexity functions. Mahoney's historiography main problem is the lack of external factors that influenced computational complexity development, such as international relations.

This chapter will explain the discovery of computational intractability in mathematical logic and its formalization in the field of computational complexity. The contribution of this chapter to the literature is the development of computational complexity concerning AI using games for extrapolating the laws of human decision-making.

Five parts make up this chapter. The first describes the use of games for understanding the laws of the human decision-making. The next explains the discovery of computational intractability using dominoes. The following part reviews the history of the development of the field of computational complexity before Cook's P/NP distinction. The fourth describes the formalization of the tractable/intractable distinction in computational complexity. Finally, the last part of the thesis analyses the possibility of merging AI with computational complexity to transcend the problem of computational intractability that exists in AI.

---

[203] Russell is a Smith-Zadeh professor in engineering at the University of California in Berkley, while Norvig is the research director at Google.
See http://people.eecs.berkeley.edu/~russell/ and https://norvig.com/ for more information

9.1 Games as the basis for modeling the human mind

Games were the first elements used by AI researchers because they considered games helpful in representing environmental complexity (McCorduck, 1979, pp. 146-147). Checkers and chess were some examples of games used by researchers interested in developing machine intelligence.[204]

Arthur Samuel worked until 1946 at Bell Laboratories after finding a position as a professor in electrical engineering at the University of Illinois at Urbana Champaign. During his time at Urbana, Samuel became interested in developing a computer checkers game simulation to raise money and to understand how computers learn.[205] In 1947, Samuel sought the help of some institutions like the Institute for Advanced Studies at Princeton to build a computer because private sector could not provide technology. Unfortunately, Samuel and his team had limited skills in building a computer (McCorduck, 1979, pp. 148-149; Nilsson, 2010, pp. 90-91; Samuel, 1963, pp. 71-72).

Samuel left his post at the university to join the International Business Machines Corporation (IBM) in 1949. In the corporation, he found the best place to use his skills to research how to build computers. Furthermore, Samuel took advantage of his employment at IBM to develop his checkers playing program (McCorduck, 1979, p. 150-151, Nilsson, 2010, p. 90).[206] For testing his checkers game, Samuel used several computers produced by IBM during the time he was working in the

---

[204] Chess was considered helpful for finding patterns that ruled the intellectual world. Researchers also considered chess a game that would help understand governing laws of human decision-making and problem-solving (Newell et al., 1963, p. 39).

[205] Samuel considered the limits of computer memory by the time he was programming his checkers playing machine. He considered storing the number of possible movements based on the players' recent moves, deleting those in the memory for a long time (Samuel, 1979, pp. 80-82). The best way Samuel found to reduce the storage problem was by saving the generalizations based on the previous experience during the game (Samuel, 1963, p. 83). The suggestion Samuel made at the end of his paper was to save the first moves of the play in memory, followed by the use of the learning algorithm (Samuel, 1963, pp. 94-95).

[206] Samuel programmed his game of checkers following the next logic. Before the machine decided which move to make, the program evaluated its possibilities for taking the next move (Samuel, 1963, p. 76).

company.[207] In the beginning, Samuel's work was unsuccessful because he did not know the rules of checkers. To solve his problem, he hired several checker masters after giving up reading checkers' literature. Samuel took that decision because checkers' rules were tough to program.[208] Unfortunately, players that worked with Samuel could barely help him. After all, communication between experts and Samuel was confusing (McCorduck, 1979, pp. 151-152). However, his effort was successful compared with chess game programming because rules of checkers were easier to program (McCorduck, 1979, pp. 152-153; Samuel, 1963, p. 72)[209] Samuel was excited. After all, his program defeated checkers' expert Robert Nealey (Nilsson, 2010, p. 93).[210]

Samuel's work inspired Alex Bernstein. Bernstein seemed to be the qualified person to program a chess-playing machine because he had previous experience with the game.[211] Bernstein was in touch with mathematical modeling while he worked in the U.S. Military. Bernstein was part of a team focused on developing USA's first missile air defense system. After his experience in the military, he went to Columbia University, but he left the institution after some time, feeling he was not part of the academic world.

Bernstein chose to work at IBM after dropping out from Columbia (McCorduck, 1979, pp. 154-155). By the time he started his duties at IBM, Bernstein became interested in programming a chess-playing machine.[212] Bernstein was not a good

---

[207] During the time Samuel worked at IBM, computer models 701 and 702 were in production (McCorduck, 1979, p. 150).

[208] The books and manuals written during the time Samuel worked at IBM were not very explicit about the principles of the game of checkers (McCorduck, 1979, p. 152).

[209] The rules are much more complex than checkers because the chess pieces have dissimilar moves from one another. Please check the report by Bernstein et al. (1958) related to the chess-playing machine built at IBM.

[210] Nealey was a checkers master from the U.S. State of Connecticut (Nilsson, 2010, p. 93).

[211] Allen Newell, John C. Shaw, and Herbert Simon also were interested in programming a chess-playing machine to simulate the process of problem-solving and decision-making. The game of chess inspired their Logic Theorist (LT) because they saw the process of logic problem solving quite similar to a game of chess. Bernstein's work inspired Newell, Shaw, and Simon (Newell et al., 1963, pp. 50-51).

[212] Bernstein's program was launched in an IBM model 704 (Newell et al., 1963, p. 48).

chess player, but he had the knowledge to start thinking about how to start programming the needed chess moves to simulate the game. Furthermore, he read Aron Nimzowitsch's (1886-1935) *My system* (1925). Nimzowitsch's oeuvre helped Bernstein to reduce the game strategies' complexity [213] Other than that, Turing's influence in Bernstein's work was present (McCorduck, 1979, pp. 155-156).[214] Unfortunately, chess is intractable because 10^{120} is the number of possible moves during a chess game.[215] (McCorduk, 1979, p. 157; Newell et al., 1963, p. 43). Berstein solved the problem by introducing rules of thumb.[216] Similarly, Newell, Shaw, and Simon's chess gaming machine could not guarantee a satisfactory result.[217] The works of chess and checkers inspired other researchers to program another type of games.

---

[213] In his book, the chess player Nimzowitsch wrote the vocabulary that explained different strategies grandmasters of the game used intelligibly (Bhutia, 2015).

[214] In 1953, Turing's "Digital computers applied to games" was published in the book *Faster than thought* (1953). Turing's work proposed a novel idea related to developing a computer-playing machine and the rules the program would run to simulate the chess-playing process (Turing, 1963).

[215] The time necessary to compute such several chess moves is greater than the life of the universe (McCorduck, 1979, p. 157).

[216] Samuel also introduced rules of thumb for programming his checkers playing machine because checkers is intractable (McCorduck, 1979, p. 157). Each game turn involves 10^{40} moves, producing a problem because computing the possible strategies could take centuries to calculate (Samuel, 1963, p. 72).

[217] Newell, Shaw, and Simon used a tree for representing each possible move the game of chess would have. They used a search algorithm to seek the best possible tactics. Unfortunately, the algorithm did not guarantee convergence in seeking data structures for finding the best possible moves (Newell et al., 1963, pp. 53-57). They used heuristics to mimic how humans play chess (Newell et al., 1963, pp. 62-63).

9.2 Computational intractability is born

In the 1960s, when Hao Wang worked at IBM, he developed the concept of tiling. Wang's idea emerged from a solitary game he created using dominoes. The result of his work was an exciting finding because Wang's game had an analogy with Turing machines (Wang, 1990, p. 204). Wang found the connection with the intractability problem using different types of tiles since rotating the domino tokens was not possible (Wang, 1990, pp. 204-205).[218] The tiling problem was very similar to the halting problem Alan Turing developed in the 1930s. Depending on the initial token configuration, a Turing machine could find a solution or not. Wang used the same principle in his dominoes game (Wang, 1990, pp. 205-206).[219] Wang's results were published in the paper "Dominoes and the AEA case of the decision problem" (1963).[220] Wang's tiling problem served as an inspiration to

---

[218] Before Wang introduced his tiling problem, he explained the infinity lemma using a game of chess. The lemma explained that chess has the problem of infinite moves. However, the possible moves from one node were limited. As a result of a particular node of the set of possible moves, the number of moves was countable. The concept of the infinity lemma also explained the infinity of possible options in the game developed by Wang (Wang, 1990, pp. 201-204).

[219] The analogy between the game of dominoes and the Turing machine was very significant. Wang showed that games could be analyzed to determine whether they were tractable. Wang (1990) explained how the tiling problem could be compared with a Turing machine

"The (unrestricted) domino problem. To find an algorithm to decide, for any given (finite) set of domino types, whether it is solvable. The origin-constrained domino problem. To decide, for any given set P of domino type and a member C thereof, whether P has a solution with the origin occupied by a domino of type C.

The diagonal- (row-, column-) constrained domino problem. To decide for any given set P of domino types and a subset Q thereof, whether P has a solution with the main diagonal (the first row, the first column) occupied by dominoes of types in Q" (Wang, 1990, p. 220). See (Wang, 1990, p. 221) for a more detailed explanation of the relationship between dominoes and the Turing machine.

[220] Wang said that the problem of dominoes could be reduced to an AEA formula. He wrote in his paper: "For all *x* there is a *y* such that for all *z*... followed by a logical combination of predicates without quantifiers such that the set has a solution if and only if the formula is or is not self contradictory. In other words, we can translate a domino question into a logical formula by specifying certain constraints and then determine if the domino set is solvable

researchers in computer science for understanding the theoretical foundations of computational intractability.

9.3 The creation of the field of computational complexity

One of the first persons that considered the limits of computation in terms of a mathematical function was mathematician Michael Oser Rabin (1931).[221] Rabin's "Degree of difficulty of computing a function and a partial ordering of recursive sets" (1960), financed by the Office of Naval Research Information Systems of the USA (ONR), compared the complexity of computing two functions based on proofs (Rabin, 1960, pp. 1-2). Rabin explained the process of measuring the complexity of a function in section one of his research, reaching two crucial conclusions. The first was linked to the concept of computing a function, while the other was related to the conditions that made a function computable (Rabin, 1960, pp. 4-8).[222] The second part of Rabin's thesis compared the degree of difficulty for computing two functions.[223] Finally, the last topic Rabin discussed in his work was related to the decision problem. He explained how solvable was a function compared to another in mathematical terms.

---

by seeing if the formula is or not self contradictory. Therefore, since the general domino problem is unsolvable then is no general method for deciding if an arbitrary AEA formula is self-contradictory" (Wang, 1990, p. 208).

[221] Mathematician Rabin graduated from the Hebrew University of Jerusalem. He is one of the main contributors to the development of the field of computer science, winning the Association for Computing Machinery (ACM) Turing Award for his work in the computational complexity area in 1976 (Hosch, 2020a).

[222] The first conclusion reached by Rabin was "A function *f* is computable (recursive) if and only if there exist a system L computing it". The second conclusion he wrote in his thesis was: "It is easy to verify that for a given system L which computes *f*, the function $F_L(n)$ F(n) is computable. If *f* is primitive recursive, then there exist a system (L,w) computing *f* such that $F_L(n)$ is primitive recursive" (Rabin, 1960, pp. 7-8).

[223] See (Rabin, 1960, pp. 8-14) for a more detailed mathematical explanation of the difficulty of computing two functions.

Rabin's work was an inspiration to other researchers interested in the problem of computational complexity like physicist Juris Hartmanis (1928) and mathematician Richard E. Stearns (1936).[224] Hartmanis and Stearns' "On the computational com plexity of algorithm" (1965).[225] provided a framework for classifying computational sequences based on their complexity according to Turing machine classification of computable and non-computable sequences.[226] Moreover, they used Turing machines for measuring the velocity of symbol printing in a tape (Hartmanis & Stearns, 1965, pp. 285-286). Hartmanis and Stearns also associated Turing machines and digital computers because the latter is an idealized form of the former (Hartmanis & Stearns, 1965, p. 285).[227]

The influence Hartmanis and Stearns had over Alan Cobham (1927) was of great importance to continue the development of the computational complexity field.[228] In 1964, Cobham gave a talk titled "The intrinsic computational difficulty of functions". His conference completely changed the perception of how computational complexity was understood until that moment because the

---

[224] Hartmanis was born in Riga, Latvia. He studied for a master's degree in mathematics from the University of Kansas City in 1951. Later, he obtained his Ph.D. in mathematics in 1955 from the California Institute of Technology (Caltech). Hartmanis taught at Cornell University and Ohio State University. During the time he worked at GE, he met Stearns. With him, Hartmanis won the ACM Turing Award in 1993 for establishing the foundations of the computational complexity field (Hosch, 2020b; Hosch, 2020c).

Stearns was born in Caldwell, New Jersey. He obtained his bachelor's degree in mathematics at Carleton College in 1958 and a Ph.D. in the same field at Princeton University in 1961. Stearns was a professor at State University of New York (SUNY) from 1978 to 2000. The last year he taught at SUNY, he became an emeritus professor at the university. Stearns worked at General Electric (GE) from 1961 to 1978 (Hosch, 2020c).

[225] Harmanis and Stearns published their paper while they worked at GE (National Academies of Sciences, Engineering, and Medicine, 2020, pp. 189-190).

[226] See (Hartmanis & Stearns, 1965, pp. 286-287) for a more detailed explanation of the limits of computability.

[227] See (Hartmanis & Stearns, 1965, p. 285) for a more detailed explanation.

[228] Mathematician Cobham graduated from Oberlin College. He developed the concept of measuring the complexity of a computational problem in polynomial time (also known as class P). Cobham worked at IBM. and as chair of computer science at Wesleyan University (Shallit, 2010).

discipline did not consider time and storage as variables for classifying computational functions. Cobham provided a metamathematical theory for making sense of the relation of complexity with storage and time (Cobham, 1965, pp. 24-27). After carefully studying the problem of Turing machines, Cobham concluded that computational intractability was part of the aritmetization of Turing machines. Moreover, Cobham proposed the classification of computational functions based on the use of storage and time (Cobham, 1965, pp. 26-29). Cobham influenced the development of theory of measurable computable and non-computable problems found in the work of the computer scientist Stephen Cook.

9.4 The P/NP problem

Stephen Arthur Cook is a mathematician and computer scientist who distinguished problems that could be solved in polynomial time or not. Cook was born in the city of Buffalo in the USA. His father was a chemist who worked as an adjunct professor at the University of Buffalo. At the same time, Cook's mother had occasional employment as English teacher at Erie Community College (Frana, 2002, p. 3). After high school, Cook decided to study at the University of Michigan, mainly because his parents were alumni of that institution. Cook did not have the intention to study mathematics at the university, but he changed his mind afterward. By the time Cook enrolled at Michigan, he chose the College of Engineering because Cook was interested in electronics. Cook's decision to change major happened later after talking with one of his mathematical mentors Nicholas Kazarinoff.[229] Kazarinoff suggested Cook switch majors because Cook would perform better if he studied mathematics (Frana, 2002, pp. 3-5). Cook heard Kazarinoff's advice, switching to the mathematics program and graduating in 1961 with a bachelor’s in computer science (Frana, 2002, pp. 5-6; Hosch, 2020d).[230] One year later, Cook earned his master’s in mathematics at Harvard University.

---

[229] To know more about Kazarinoff, please follow the next link: https://buffalonews.com/news/nicholas-d-kazarinoff-ub-math-professor/article_35403da5-3631-5455-8dad-7df1e461304d.html

During his master’s studies at Harvard, Cook met his future doctoral advisor Wang (Frana, 2002, p. 7). Cook and Wang were interested in studying the problem of computation. Cook read Rabin’s works while he was studying for his master’s at Harvard, while Wang did some research in theoretical computing during the time he worked at IBM. Both found common ground for collaborating. Cook began his Ph.D. work under the supervision of Wang.

´The result of Cook's research was a doctoral thesis inspired by Cobham's work related to the complexity of multiplication under the title *On the minimum computation of time functions* (1966) (Frana, 2002, p. 8).[231] During the time Cook was doing his thesis, he worked as a consultant for the Research and Development (RAND) Corporation in 1965. The report he wrote at the institution was part of the research linked to his Ph.D. thesis (Cook, 1966, p. i). Cook's memorandum showed how a theoretical machine could generate number theory proofs (Cook, 1956, p. iii). Cook explained in his work the limits of Herbrand-Skolem theorem at the moment of generating proofs of mathematical logic theorems. Even though the theorem was efficient in large domains, Herbrand-Skolem theorem could not obtain good results in a more restricted domain. For that reason, Cook proposed a solution to solving logical problems in the field of number theory through algebraization (Cook, 1965, p. v; Cook, 1965, pp. 1-3). One of the interesting aspects of Cook's proposal was introducing a new type of metamathematics that dealt with polynomial expressions such as “xyz+3+xz” (Cook, 1965, pp. 4-5).[232]

One year later, Cook presented his thesis, focusing on explaining why multiplication is a harder operation than addition. Cook wrote a common sense idea in the first pages of his doctoral dissertation. He explained that multiplication is far more complex than addition. Cook used some examples to sustain his hypothesis. He mentioned that circuitry for addition is far easier to build than

---

[230] Before the institutionalization of the field, computer science was part of the area of mathematics.

[231] Cook sent me his doctoral thesis through email (S. Cook, personal communication, June 22, 2018).

[232] Cook developed a metamathematical system that eliminated the axioms of associativity for addition and multiplication (Cook, 1965, pp. 4-5). The result of Cook's formulation was the transformation of one formula into another because excessive parenthesis was not necessary (Cook, 1965, pp. 11-12).

multiplication. Moreover, he said that since ancient antiquity, the number of pages needed to do the entire multiplication process required more time than those used for addition (Cook, 1966, pp. 1-2). Cook found in single tape Turing machines the theoretical framework for explaining the conditions algorithmic process a computer or a person does for solving problems of addition and subtraction (Cook, 1966, p. 3). The machine achieved its purpose by imposing a bounded time restriction on the number of possible elements to compute. The components Cook considered, in this case, were the storage like planar arrays and a set of modules that could read and modify the arrays (Cook, 1966, pp. 4-5).[233] Cook's thesis result was a success, showing that Turing machines could solve the problem of multiplying two numbers in measurable time (Cook, 1996, p. 51).

After graduating from Harvard in 1966, the University of California in Berkeley hired Cook as an assistant professor in mathematics (Frana, 2002, pp. 9-10). Cook worked at Berkeley from 1966 until 1970, deciding to move to the city of Toronto because he found a position as associate professor in computer science and mathematics at the institution (Frana, 2002, pp. 11-12).[234] His magnum opus was published while he was a professor at the University of Toronto. Cook's "The complexity of theorem-proving procedures" (1971) shows how a computer problem could be reduced to a polynomial degree of difficulty, allowing the complexity to be measured in polynomial time (Cook, 1971, p. 151). Cook starts his article by showing the relationship between recursivity and formulas written in propositional calculus. Cook continues by mentioning that a Turing machine would help explain whether a problem could be reduced to polynomial time. Cook

---

[233] Cook defended the idea of using Turing machines as models for computation. Jiri Becvar's "Real time and complexity problems in automata theory" (1965) said that real devices inspired by Turing machines could process a limited number of operations in a unit of time (Becvar, 1965, p. 480). Cook said that the problem of computation he wanted to address was a mathematical one, not related to physics or engineering. Moreover, Cook mentioned that Turing machine inspired one of the most important concepts of modern computing: stored program computer (Cook, 1966, pp. 13-14).

[234] Cook decided to resign his position at Berkeley because the university denied him a tenured position in the mathematics department (Frana, 2002, pp. 11-12). Since 1970, Cook has been working at the University of Toronto.

compares a Turing machine to an oracle in that respect.[235] The result of Cook's formulation is the distinction between problems that could be solved in polynomial time (P) or not (NP) (Cook 1971, p. 151).[236] Cook's P/NP distinction helps to separate tractable problems from intractable ones by means of time.[237] The concept was helpful, but it was not considered by those researchers that were developing AI.[238] The constraints imposed to research during the Cold War developed separate identities in AI and computational complexity. However, "Cold War rationality" is the link between them.

---

[235] Cook explained the concept of an oracle as: "A query machine is a multitape Turing machine with a distinguished tape called query tape, and three distinguished states called the query state, yes state and no state respectively. If M is a query machine and T is a set of strings, then a T-computation of M is a computation of M in which initially M is in the initial state and has an input string w on its input tape, and each time M assumes the query state there is a string u on the query tape, and the next step M assumes is the yes state if $u \in T$ and the no state if $u \notin T$. We think of an "oracle", which knows T, placing M in the yes state or the no state" (Cook, 1971, p. 151).

[236] Cook formalized the concept of polynomial time as follows "A set S of strings is Dominoes and the AEA case of the decision problems P-reducible (P for polynomial) to a set T of strings iff there is some query machine M and a polynomial Q(n) such that for each input string w, the T-computation of M with input w halts within Q(|w|) steps (|w| is the lenght of w), and ends in an accepting state iff $w \in S$ " (Cook, 1971, p. 151).

[237] Wang's "Dominoes and the AEA case of the decision problems" (1963) inspired Cook because Wang's work was the framework for identifying tractable and intractable problems (Cook, 1971, p. 156). Moreover, I exchanged emails with Cook, telling me that Wang was his primary source of inspiration while writing his paper (S. Cook, personal communication, June 22, 2018).

[238] By the time Cook wrote his paper, artificial intelligence and computer science were separate disciplines. The late institutionalization of the field and ignoring Wang's work by the artificial intelligence community could have influenced the strong disciplinary demarcation between artificial intelligence and computer science.

9.5 Is possible to solve the problem of computational intractability?

Games were used for modeling decision-making and problem-solving in AI. Moreover, games also influenced the works of Wang related to computational intractability, as described in this chapter. Even though AI and computational complexity share a common component, computational intractability, both fields are not in contact since they treat different types of problems.

AI and computational complexity could have benefited from one another if they established a dialog. AI researchers used games to simulate human problem-solving through interaction between specific types of environments with an intelligent algorithm. In contrast, computational complexity used games to demonstrate which problems are tractable and intractable.

Interactionist and mentalist approaches are schools that explain the separation between AI and computational complexity. Interactionism helps to understand how concepts and ideas are linked through interaction between an agent and the environment. Interactionism does not deny the separation between the individual and the environment. However, the action of interaction helps to understand the relationship between decisions a subject makes concerning the environment (Agre, 1997, p. 53). Games and "microworlds" are related because an agent require interaction to make sense why specific rules govern an environment.[239] For example, chess is considered the baseline for developing artificial intelligence programs because different strategies an agent has to think of for winning could serve to extrapolate rules of decision-making. Moreover, planning and strategy are necessary to play a successful game of chess (Ensmenger, 2011, p. 18).

The definition of interactionism is the opposite of mentalism because the latter focuses on explaining computing in a metamathematical form (Agre, 1997, pp. 51-52). Mentalism is associated with platonism because ideas of logic and language

---

[239] I disagree with categorizing "microworlds" as part of the mentalist approach Agre (1997) suggested. Even though "microworlds" require rules for manipulating objects of the environment, there is not an implicit separation between the rules and the actions taken by the agent. Chess requires knowing the rules before playing the game, as Ensmenger (2011) explains (Agre, 1997, p. 51; Ensmenger, 2011, p. 17). For that reason, I categorize games and "microworlds" as part of the interactionist approach.

are linked to the world of ideals (Agre, 1997, pp. 51-52). Turing machines are associated with mentalism because those machines explain computation without considering the limits of machine resources (Cogburn & Silcox, 2011, p. 71). Wang developed his tractable/intractable distinction using Turing machines. For that reason, computational complexity can be considered part of a metamathematical framework for explaining the intractability of problems computers must solve.[240] Even though mentalism and interactionism could benefit from each other, the separation between both theories happened because Cold War politics left Wang out of AI research.[241] It was not until the late 1980s that some planning algorithms implemented the P/NP distinction, giving some air to those AI researchers that wanted to make progress in the field (Russell & Norvig, 2010, p. 394).

AI would have benefited from computational complexity if those areas of knowledge walked together. To solve the contradiction between machine intelligence and computational intractability, I introduce an author not considered by mainstream history and philosophy of science, not to mention about the history of computing and AI. The person I am talking about is philosopher Theodor Adorno. His “negative dialectic” proposes the solution of a contradiction by considering the tension between a general and a particular.[242] Adorno’s negative dialectic has a strong relationship with “entangled history” because both focus on creating a new understanding of a problem based on mixing more than one element that shares something common.

For the case of this thesis, games used for developing a rationality during the Cold War are the common point between AI and computational complexity. Moreover,

---

[240] John McCarthy was the only researcher of those who attended the Dartmouth Research Project on Artificial Intelligence that defended the idea of using logic as the foundation of artificial intelligence. The critics of McCarthy said that his proposal was not possible to implement because McCarthy's work lacked a mathematical model (Mahoney, 2011, pp. 126-127). McCarthy believed that lambda calculus, similar to Turing machines, could be the foundation of AI (Mahoney, 2011, pp. 126-127).

[241] See section 8.2 for understanding the dispute between Herbert Simon and Wang in the context of the Cold War.

[242] See Adorno (1958) and Dussel (2015).

they share a common context: the Cold War. In that vein, “Entangled history” and Adorno’s “negative dialectic” can get along well. Constructivism does not fit in solving the contradiction because that historical standpoint only considers scientific practice.

My solution to the problem of computational intractability is internal, not external, focusing on the connection between data and metamathematics. Using the “negative dialectic”, a tension between the concept of machine intelligence developed in AI with the problem of computational intractability exists. We saw how the conception of machine intelligence found in ai was institutionalized at the Dartmouth Summer Research Project in chapter 5. Enthusiasm in the field lasted until the second half of the 1960s because AI entered a crisis, as seen in chapter 7. Using the words of Adorno, the concept entered into a contradiction. Some critics of the mainstream paradigm told the limits of heuristics, as Wang explained in his paper “Toward mechanical mathematics” (1960). Unfortunately, Wang’s proposal was ignored by the early artificial intelligence community because U.S-Chinese tense international relations from 1950 until 1972. Wang was perhaps conscious of the possible crisis AI was going to suffer, but context did not let Wang enter the field. In Adorno’s words, power did not let solve the contradiction, in this case, the fight for hegemony between the Soviet Union and the USA. Wang changed of plans, switching from the area of automatic theorem proving to study the decidability problem. Wang’s work resulted in the formalization of the tractable/intractable distinction after carefully studying the problem of dominoes. His work inspired his former doctoral student Cook to develop his P/NP distinction. In theory, the concept of machine intelligence found in AI could be merged with the conception of computational intractability found in computational complexity because both come from “Cold War rationality”. While the first is descriptive of a certain problem, the second explains whether the problem is tractable or not. The mixture of AI and computational complexity could help to transcend the intractability problem. As the chair of the artificial intelligence and machine learning institute of informatics at the Ludwig Maximilian University of Munich, Eyke Hüllermeier said in the conference *AI in flux* at the Deutsches Museum on December 1, 2021, the future for solving problems of soft artificial intelligence lies combining artificial intelligence data-driven methods with mathematical logic (Hüllermeier, 2021). I hope my proposal can serve that goal.

Conclusions

This chapter shows the connection between AI and computational intractability through games. Games were one of the most essential elements for developing “Cold War rationality” (Erickson, 2013). As was explained previously in chapters 4 and 6, chess was a crucial element in early artificial intelligence for understanding human problem-solving and decision-making because Simon and Newell saw a similarity between a game of chess with solving theorems in mathematical logic.

Private enterprise was also crucial to developing a rationality based on games. Researchers such as Samuel and Bernstein programmed games of checkers and chess at IBM. Turing’s influence was present in Bernstein developments. Maybe Wang benefited from IBM’s “Cold War rationality” developments for discovering the problem of computational complexity in mathematical logic.

Wang’s tiling problem is related to developing the tractable/intractable distinction. As was explained in chapter 8, Wang found some cues of his distinction from his Program P. His findings influenced the development of his tiling problem. Wang’s formulation served as an inspiration to his former doctoral student Cook to develop his P/NP distinction.

AI and computational complexity were developed using games. Those games served as a model for framing” Cold War rationality”. In that vein, merging both fields can help to transcend the problem of computational intractability. “Entangled history” has a relationship with Adorno’s “negative dialectic”. Using Adorno's “negative dialectic”, Cook’s P/NP distinction could overcome artificial intelligence's intractability by introducing Cook's concept in AI. In theory, my proposal would solve the problem of computational intractability in AI.

General conclusions

In the general assembly of the 26th International Congress of History of Science and Technology held virtually from July 25 to 31, 2021, the Hans Rausing professor of history and philosophy of science at Cambridge University, Hasok Chang, said to the participants that one of the saddest situations history and philosophy of science are suffering is the continuous separation between history and philosophy of science (Chang, 2021). This thesis has shown that saving the link between both fields is still possible. Using "entangled history", it has been possible to understand how the connection between science and politics shaped the development of the field of computational complexity and how this, in turn, was linked to discovering the problem of computational intractability. The formalization of the problem of computational complexity during the 1960s happened against the background of the tension between the USA and the People's Republic of China.

While AI was one of the sciences developed for military and national security purposes, the study of computational complexity was not. For that reason, both fields were born separately. AI was born in 1956 at the Dartmouth Summer Research Project on Artificial Intelligence held at Dartmouth College. Two of the attendants were political scientist Herbert Simon and physicist Allen Newell. Both presented at the event the first intelligent program that could prove theorems of mathematical logic. At that time, their "Logic theorist" program (LT) was the only functional AI program. For that reason, Simon and Newell became authorities in the field in the years to come.

Simon and Newell were the heirs of cybernetic research. In 1942, Simon had obtained his doctoral degree in political science. His thesis touched on the problem of human decision-making in organizations. After analyzing the limits of neoclassical economics, Simon concluded that human decision-making did and cannot use perfect rationality. Some years later, he tested his hypothesis using a servomechanism to develop a mathematical model for optimizing the use of resources. The result of his work was going to inspire Simon's new theory of human decision-making. With the help of Newell, both implemented Simon's theory of human decision making using heuristics in their LT. After presenting their program at the Dartmouth Summer Research Project, they focused on implementing human decision-making processes beyond proving theorems of mathematical logic. The result was their GPS. LT and GPS generated great enthusiasm in the early AI community, shaping the development of the field in the years to come.

AI was born in the context of the Cold War. The field was of great importance to the U.S. government to simulate human decision-making for supporting decisions to avoid nuclear conflict and guarantee the hegemony of the USA in the world. Therefore, several institutions received major funding after AI was born in 1956: MIT, Stanford University, the Stanford Research Institute, and the University of Carnegie Mellon. They shaped most of AI during the Cold War.

The results in computer vision and natural language processing were remarkable. Furthermore, AI helped to understand how intelligent agents could interact with objects in a controlled environment. "Microworlds" were a byproduct of the AI culture when Marvin Minsky and Seymour Papert worked together at the MIT Artificial Intelligence Group. Unfortunately, the confidence of those working in AI faded because algorithms developed during the second half of the 1960s could not solve real-world problems. Starting in the second half of the 1960s, several areas of AI such as, natural language processing, could not solve specific problems, such as translating Russian texts to English. Furthermore, some algorithms developed using "microworlds" could not solve real-world problems. AI algorithms failed mainly because of the reduced complexity of "microworlds". Before the second half of the 1960s, researchers were confident that "microworlds" could capture the real-world conditions of an environment. Unfortunately, "microworlds" failed to capture the essence of the environment the agent was interacting with. The U.S. government decided to cut the funding for those areas of AI that did not produce excellent results. Another reason AI failed during the 1970s was the neglect of points of view different from those close to the mainstream researchers in AI.

Some of those researchers ignored by the artificial intelligence community were considered even dangerous to U.S. national security. Several voices were critical of the LT developed by Simon and Newell, among them philosopher Hao Wang. Wang lived in the USA during the tension between the USA and the People's Republic of China. Therefore, Wang could not develop his career in AI.

Wang criticized LT's heuristics for not being a reliable method for problem-solving. Instead, he proposed using Herbrand's theorem to prove mathematical logic theorems included in Bertrand Russell and Alfred North Whitehead's *Principia mathematica* (1910-1913). Wang's Program P showed significant superiority over the LT because it could solve almost all of the theorems in Russell and Whitehead's book.

When the LT was conceived, Simon and Newell were linked to RAND. Both met each other at the *Systems Research Laboratory* for studying the process of decision-making. When they decided to program the LT, they used chess for understanding the process of human decision-making. Games, such as chess, were the basis for developing a rationality during the Cold War for several proposes, such as avoiding military conflict. Wang's work was not taken seriously since Simon and Newell's LT was better connected to the developments related to "Cold War rationality" (Erickson et al., 2013).

Furthermore, Wang was considered a menace to U.S. national security, given that China and the USA were in tension during the Cold War. The U.S. government viewed Chinese academics and professionals with distrust. AI research was part of the sciences funded by the U.S. Military for supporting its activities. For that reason, researchers working on AI had to be trusted, as historian of science Dereck De Solla (1922-1983) has explained in his book *Little science, big science* (1986).

After being ignored by the artificial intelligence community, Wang took another path to continue his career. He chose to study the decidability problem. Those interested in developing programs that could mimic human problem-solving at IBM maybe influenced the formalization of Wang's tractable/intractable distinction. Arthur Samuel and Alex Bernstein programmed games of checkers and chess, respectively. They wanted to simulate the process of decision-making and problem-solving. For that reason, Bernstein talked about his work at the

Dartmouth Summer Research Project on Artificial Intelligence. Unfortunately, IBM saw no profitability in the research related to games for modeling human decision-making, but the company kept the computing culture developed by Bernstein and Samuel. Wang worked at IBM for some years. The result of his work was the automatic theorem prover Program P. Wang's work at IBM inspired the formulation of his tiling problem. Wang ultimately also inspired Stephen Cook's "The complexity of theorem proving procedures" (1971).

Cook used Wang's concept to develop his P/NP distinction. Unfortunately, Cook developed his work outside the field of AI for two reasons. The first is the development of computational complexity in computer science. The institutionalization of computer science started later than that of AI research. While AI became a recognized discipline in 1956, computer science did so only in 1968. Moreover, AI developed separately from computational complexity. The second reason is the neglect of Wang's work by the early AI community. Therefore, Cook's P/NP distinction was not born in AI research. It took the artificial intelligence community some years to recognize the importance of Cook's work. Still, the separation of both fields continues to this day.

Even though AI and computational complexity were developed separately, in theory they could merge. The transcendence of the tension between AI and computational intractability could happen by merging data-driven methods with the field of computational complexity because the latter could explain whether a problem is tractable or not in a reasonable time. Computational complexity has the property of explainability, while AI provides methods for obtaining data. Science, history, and sociology often seem to be separated. However, this thesis has shown how the three ways of knowing could interact to produce a solution to a scientific and philosophical problem: the problem of computational intractability and its possible solution.

Bibliography

Adorno, T. (2013). *Introducción a la dialéctica.* Buenos Aires: Eterna Cadencia Editora.

Agar, J. (2001). *Turing and the universal machine.* Cambridge: Icon books.

Agar, J. (2003). *The government machine: A revolutionary history of the computer.* Cambridge and London: The MIT press.

Agar, J. (2012). *Science in the twentieth century and beyond.* Cambridge and Malden: Polity Press.

Agre, P. (1997). Computation and human experience. New York: Cambridge University Press.

Aiken, H., Oettinger, A. G., & Bartee, T. C. (1964). Proposed automatic calculating machine. *IEEE Spectrum*, 1(8) 62-69.

Aspray, W., & Williams, B. O. (1994). Arming American scientists: NSF and the provision of scientific computing facilities for universities, 1950-1973. *IEEE Annals of the History of Computing,* 16(4), 60-74.

Aspray, W. (1990). *An interview with Thomas A. Keenan OH 217.* Charles Babbage Institute. Available at: https://conservancy.umn.edu/handle/11299/107401

Augier, M., March, J. (2004). Herbert A. Simon, Scientist. In Mie Augier & James G. March, (Eds.), *Models of a Man: Essays in memory of Herbert A. Simon* (pp. 3-32). Cambridge: MIT Press.

Barnard, C. (1968). *The functions of the executive.* Cambridge and London: Harvard University Press.

Becvar, J. (1965). Real time and complexity problems in automata theory. *Kybernetica*, 1(6), 475-498.

Bernstein, A., Roberts, M., Arbuckle, T., & Belsky, M. (1958). A chess playing program for the IBM 704. *Procedings of the May 6-8, 1958, western joint computer conference: contrasts in* computers, 157-159.

Bell, J. (2015). The axiom of choice. *Stanford encyclopedia of philosophy.* Available at: https://plato.stanford.edu/entries/axiom-choice

Blanchette, P. (2007). The Frege-Hilbert controversy. *Stanford encyclopedia of Philosophy*. Retrieved from: https://plato.stanford.edu/entries/frege-hilbert/

Boden, M. (2006). *Mind as Machine: a history of cognitive science.* Oxford: Oxford University Press.

Boole, G. (1854). *An investigation of the laws of thought.* Cambridge: Macmillan and Co.

Buckley, P. (2010). *The Cambridge illustrated history of China.* New York: Cambridge University Press.

Burks, A., & Goldstine, H., & von Neumann, J. (1946). *Preliminary discussion of the logical design of an electronic computing instrument.* Princeton: Institute for Advanced Study.

Bhutia, T. (2015). Aron Nimzowitsch. *Encyclopaedia Britannica.* Retrieved from: https://www.britannica.com/biography/Aron-Nimzowitsch

National Academies of Sciences, Engineering, and Medicine. (2020). *Science the endless frontier. 75th anniversary edition.* Washington D.C: The National Academies Press.

Burris, S. (2018). George Boole. *Stanford encyclopedia of philosophy*. Retrieved from: https://plato.stanford.edu/entries/boole/

Caccetta, L., & Kulanoot, A. (2001). Computational aspects of hard Knapsack problems, *Non linear analysis*, 47(8), 5547-5558.

Campbell-Kelly, M. (2013). Remembering Michael S. Mahoney. *Perspectives on Science*, 21 (3), 379-383.

Carver, W. B. (1956). Obituary: William Deweese Cairns. *The American Mathematical Monthly,* 63(3), 204-205.

Chang, H. (2021). General assembly of the 26th International Congress of History of Science and Technology. *26th International Congress of History of Science and Technology*, Prague, Czech Republic.

Charniak, E. (1972). *Toward a model of children's story comprehension* (Doctoral dissertation). Retrieved from: https://dspace.mit.edu/handle/1721.1/6892

Christ, C. F. (1952). History of the Cowles Commission 1932-1952. *Economic Theory and Measurement,* 3-65.

Christ, C. (1994). The Cowles Commission's contributions to econometrics at Chicago, 1939-1955. *Journal of Economic Literature,* 32(1), 30-59.

Cohen, I. B. (2003). Mark I, Harvard. In *Encyclopedia of computer science* (pp. 1078-1080). John Wiley and Sons.

Cook, S. (1965). *Report on algebraic techniques and the mechanization of number theory (Memorandum RM-4319-PR)*. RAND Corporation.

Cook. S. (1966). *On the minimum computation of time functions* (Doctoral dissertation), Harvard University. Boston, United States of America.

Cook, S, (1971). The complexity of theorem proving procedures. In *Proceedings of the third annual ACM symposium on theory of computing*, 151-158.

Cook, S. (1983). An overview of computational complexity. *Communications of the ACM,* 26(6), 400-408.

Cogburn, J., & Silcox, M. (2011). Computability theory and ontological emergence. *American Philosophical Quarterly,* 48(1), 63-74.

Cooper, W. (2004). Memorial to Herbert Simon. In Mie Augier & James G. March (Eds.), *Models of a man: essays in memory of Herbert A. Simon* (pp. 67-74). Cambridge: MIT Press.

Copi, I. (1958). The Burali-Forti paradox. *Philosophy of Science*, 25(4), pp. 281-286.

Cortada, J. (2019). *IBM: The rise and fall and reinvention of a global icon.* Cambridge: MIT Press.

Church, A. (1935). Review of Willard van Orman Quine’s A system of logistic. *Bulletin of the American Mathematical Society*, 41, 598-603.

Cobham, A. (1965). The intrinsic computational difficulty of functions. In Yehoshua Bar-Hillel, editor, *Logic, methodology and philosophy of science: Proceedings of the 1964 International Congress (Studies in Logic and the Foundations of Mathematics)* (pp. 24-30). Amsterdam: North-Holland publishing.

Crevier, D. (1993). *The tumultuous history of the search for artificial intelligence*. New York: Basic Books.

Cumings, B. (2010). *The Korean war: a history.* New York: Modern library chronicles book.

Davis, M. (2011). Hao Wang's contributions to mechanized deduction and *Entscheidungsproblem*. In Charles Parsons & Montgomery Link (Eds.), *Hao Wang: logician and philosopher* (pp. 73-78). London: College Publications.

Dawson, P. L. (2005). *Luis de Flores and the Special Devices Division* (Doctoral dissertation). Available from ProQuest Dissertations & Theses Global. (304999162). Retrieved from https://www.proquest.com/dissertations-theses/luis-de-florez-special-devices-division/docview/304999162/se-2?accountid=15292

De León, M. (2016). El matemático que cuidaba las espaldas a Trotsky. *El País.* Available at:
https://elpais.com/elpais/2016/09/23/ciencia/1474627145_094617.html

DeMorgan, A. (1847). *Formal logic, or calculus of inference, necessary and probable.* London: Taylor and Walton.

De Mol, L. (2018). Turing machines. *Stanford encyclopedia of philosophy.* Retrieved from: https://plato.stanford.edu/entries/turing-machine/

Desmet, R. (2018). Alfred North Whitehead. *Stanford encyclopedia of philosophy.* Available at: https://plato.stanford.edu/entries/whitehead/

De Solla, D. (1986). *Little science, big science.* New York: Columbia University Press.

Dick, S. (2014). *After math: (Re)configuring minds, proof, and computing in the postwar United States* (Doctoral dissertation). Harvard University, Boston, United States of America.

Dick, S. (2015). Of models and machines: Implementig bounded rationality. *Isis,* 106(3), 623-634.

Dinneen, G. (1955). Programming pattern recognition. *Proceedings of the 1955 Western Joint Computer Conference*, 7, 94-100.

Dennis, M. (2020). Vannevar Bush: American engineer. *Encyclopaedia Britannica.* Available at: https://www.britannica.com/biography/Vannevar-Bush

Duignan, B. (2015). Occam's razor. *Encyclopaedia Britannica.* Available at: https://www.britannica.com/topic/Occams-razor

Dussel, E. (2015). *Filosofías del sur: Descolonización y transmodernidad.* Mexico D.F: Akal.

Dyson, F. (2013). Julian Bigelow: Bridging abstract logic and practical machines. *Institute for Advanced Study.* Available at: https://www.ias.edu/ideas/2013/george-dyson-julian-bigelow

Ebbinghaus, H. D. (2004). Zermelo in the mirror of the Baer correspondence, 1930-1931. *Historia Mathematica*, 31(1), 76-86.

Edwards, P. N. (1996). The closed world: Computers and the political discourse in Cold War America. Cambridge and London: MIT Press.

Edwards, L. D. (1998). Embodying mathematics and science: Microworlds as representations. *The Journal of Mathematical Behavior*, 17(1), 53-78.

Ensmenger, N. (2011). Is chess the drosophilia of artificial intelligence? A social history of an algorithm. *Social Studies of Science*, 42 (1), 5-30.

Erickson, P. (2015). *The world the game theorists made.* Chicago: University of Chicago Press.

Erickson, P., Klein J., Daston, L., Lemov, R., Sturm, T., & Gordin, M. (2013). *How reason almost lost its mind: the strange career of Cold War rationality.* Chicago and London: University of Chicago Press.

Feferman, A. (1993). *Politics, logic, and love: the life of Jean van Heijenoort.* Wellesley: A. K. Peters.

Feferman, S., Dawson Jr, J., Goldfarb, W., Parsons, C., & Sieg, W. (2013). *Kurt Gödel collected works: Volume IV Correspondence A-G.* Oxford: Claredon Press.

Forsythe, G. (1959). The role of numerical analysis in an undergraduate program. *The American Mathematical Monthly*, 66(8), 651-662.

Frana, P. (2002). *Oral history interview with Stephen Cook.* Charles Babbage Institute. Available at: https://hdl.handle.net/11299/107226

Frege, G. (2000). Begriffsschrift. A formula language, modeled upon that of arithmetic, for pure thought "(Original work published 1879)". Halle. Reprinted in Jean van Heijenoort (Ed.), *From Frege to Gödel. A source book in mathematical logic, 1879-1931* (pp. 1-82). Lincoln, NE: toExcel.

Gardner, H. (1987). *The mind's new science: a history of the cognitive revolution.* New York: Basic Books.

Garey, M., & Johnson, D. (1979). *Computers and intractability: a guide to the theory of NP-Completeness.* New York: W. H. Freeman and Company.

Gentzen, G. (1935). Untersuchungen über das logische Schließen I. *Mathematische Zeitschrift*, 39(1), 176-210.

Gibson, R. (2004). *The Cambridge companion to Quine.* New York: Cambridge University Press.

Glymour, C., & Eberhardt, F. (2016). Hans Reichenbach. *Stanford encyclopedia of philosophy.* Available at: https://plato.stanford.edu/entries/reichenbach/

Grant, H., Kleiner, I. (2015). Philosophy of mathematics: From Hilbert to Gödel. In Hardy Grant & Israel Kleiner (Eds.), *Turning points in the history of mathematics* (pp. 85-93). New York: Birkhäuser.

Grosch, H. (1977). *Computer oral history collection, 1969-1973, 1977.* Available at: https://amhistory.si.edu/archives/AC0196_abstracts_e-g.pdf

Gómez-Torrente, M. (2019). Alfred Tarski. *Stanford encyclopedia of philosophy.* Available at: https://plato.stanford.edu/entries/tarski/

Gödel, K. (2000a). The completeness of the axioms of the functional calculus of logic "(Original work published 1930)". Reprinted in Jean van Heijenoort (Ed.), *From Frege to Gödel. A source book in mathematical logic* (pp. 582-591)*, 1879-1931.* Lincoln, NE: toExcel.

Gödel, K. (2000b). Some metamathematical results on completeness and consistency, On formally undecidable propositions of Principia Mathematica and related systems I, and on completeness and consistency "(Original work published 1931)". Reprinted in Jean van Heijenoort (Ed.), *From Frege to Gödel. A source book in mathematical logic, 1879-1931* (pp. 592-617). Lincoln, NE: toExcel.

Golinski, J. (1998). *Making natural knowledge: constructivism and the history of science.* Cambridge: Cambridge University Press.

Goldstine, H., Pomere, J., & Smith, C. (1954). *Final progress report on the physical realization of an electronic computing instrument.* Princeton, NJ: The Institute for Advanced Study.

Grcar, J. F. (2011). John von Neumann's analysis of Gaussian elimination and the origins of modern numerical analysis. *SIAM review*, 53(4), 607-682.

Gregersen, E. (2020). *Lawrence Roberts: American computer scientist.* Encyclopaedia Britannica. Available at:
https://www.britannica.com/biography/Lawrence-Roberts.

Gruenberger, F. (1968). The history of the Johnniac. RAND memorandum RM-5654-PR.

Gigerenzer, G., & Sturm, T. (2007). Tools=theories=data? On some circular dynamics in cognitive science. In Mitchell Ash & Thomas Sturm (Eds.). *Psychology's territories: Historical and contemporary perspectives from different disciplines* (pp. 305-342). Mahwah: Lawrence Erlbaum Associates, Publishers.

Gupta, G. (2007). Computer science curriculum developments in the 1960s. *IEEE annals of the history of computing*, 29(2), 40-54.

Guzman Arenas, A. (2005). Adolfo Guzman Arenas. *Instituto Politécnico Nacional.* Available at: http://www.cic.ipn.mx/aguzman/.

Guzman Arenas, A. (1958). *Computer recognition of three-dimensional objects in a visual scene* (Doctoral dissertation). Retrieved from: https://dspace.mit.edu/handle/1721.1/6902

Haaparanta, L. (2009). *The development of modern logic.* Oxford: Oxford University Press.

Hafner, K. (2016, November 17). Jay W. Forrester dies at 98; a Pioneer in computer models. *The New York Times.* Available at: https://www.nytimes.com/2016/11/18/technology/jay-forrester-dead.html

Han, Y. (1993). An untold story: American policy toward Chinese students in the United States. *Journal of American-East Asian relations*, 2(1), 77-99.

Hartmanis, J., & Stearns, R. E. (1965). On the computational complexity of algorithms. *Transactions of the American Mathematical Society*, 117, 285-306.

Hebb, D. (1949). *The organization of behavior: A neurophysiological theory.* New York: John Willey and Sons.

Herbrand, J. (1930). *Recherches sur la théorie de la démonstration* (Doctoral dissertation). Retrieved from: http://www.numdam.org/article/THESE_1930__110__1_0.pdf

Herbrand, J. (1968). *Ecrits logiques.* Paris: Presses Universitaires de France.

Heinzmann, G. (2017). Henri Poincaré. *Stanford encyclopedia of philosophy.* Retrieved from: https://plato.stanford.edu/entries/poincare/

Hilbert, D. (1996). Axiomatic thought "(Original work published 1918)". Reprinted in William B. Ewald (Ed.), *From Kant to Hilbert, A source book in the foundations of mathematics* (pp. 1105-1114). Oxford: Claredon Press.

Hilbert, D., Ackermann, W. (1975). *Elementos de la lógica teórica.* Madrid: Editorial Tecnos.

Hilbert, D. (2000). On the foundations of logic and arithmetic "(Original work published 1904)". Reprinted in Jean van Heijenoort (Ed.), *From Frege to Gödel, A source book in mathematical logic, 1879-1931* (pp. 129-138). Lincoln, NE: toExcel.

Hilbert, D. (1968). *Grundlagen der Geometrie.* Stuttgart: B.G. Teubner.

Hilbert, David. (1902). Mathematical problems. *Bulletin of the American Mathematical Society.* 8(10). 437-479.

Hodges, A. (2008). Alan Turing and the Turing test. *Alan Turing: The enigma.* Available at: https://www.turing.org.uk/publications/testbook.html

Hosch, W. (2020a). Michael Oser Rabin. *Encyclopaedia Britannica*. Available at: https://www.britannica.com/biography/Michael-Oser-Rabin

Hosch, W. (2020b). Juris Hartmanis. *Encyclopaedia Britannica*. Available at: https://www.britannica.com/biography/Juris-Varlejs-Hartmanis

Hosch, W. (2020c). Richard E. Stearns. *Encyclopaedia Britannica*. Available at: https://www.britannica.com/biography/Richard-Edwin-Stearns

Hosch, W. (2020d). Stephen Arthur Cook. *Encyclopaedia Britannica*. Available at: https://www.britannica.com/biography/Stephen-Arthur-Cook

Hosch, W. (2020e). George Danzig: American mathematician. *Encyclopaedia Britannica*. Available at: https://www.britannica.com/biography/George-Dantzig

Hu, D. (2016). American influence on Chinese physics study in the early twentieth century. *Physics in Perspective*, 17(4), 268-297.

Hüllermeier,, E. (2021, December 1). *Algorithms, knowledge and data: on the evolution of AI systems design*. Paper presented at the AI in Flux, Munich.

Jevons, W. S. (1864). *Pure logic or the logic of quantity apart of quantity: with remarks on Boole's system and on the relation of logic and mathematics.* London: Edward Stanford.

Kennedy, H. (1980). *Peano: life and works of Giuseppe Peano.* Dordrecht: D. Reidel.

Kevles, D. (1977). The National Science Foundation and the debate over postwar research policy, 1942-1945: a political interpretation of Science—The endless frontier. *Isis,* 68(1), 4-26.

Kline, R. (2010). Cybernetics, automata studies, and the Datmouth Conference on Artificial Intelligence. *IEEE Annals of the History of Computing,* 33(4), 5-16.

Kline, R. (2015). *The cybernetics moment: or why we call our age the information age.* Baltimore: John Hopkins University Press.

Knuth, D. (1972). George Forsythe and the development of computer science. *Communications of the ACM,* 15(8), 721-726.

Köhler, E. (2011). Collaborating with Hao Wang on Gödel's philosophy. In Charles Parsons & Montgomery Link (Eds.), *Hao Wang: logician and philosopher* (pp. 55-72). London: College Publications.

Kruskal, J., & Newell, A. (1950). A model for organization theory. In *Tech. Rept. LOGS-103.* The RAND Corporation Santa Monica, Ca.

Lee, K., Wong, D., & Ng, K. (2013). Dr. Sun Yat-sen (1866-1925): medical doctor and China’s founding president. *Sigapore Medical Journal,* 54(6), 356-358.

Linsky, B. (2019). Principia mathematica. *Stanford encyclopedia of philosophy.* Available at: https://plato.stanford.edu/entries/principia-mathematica/

Lotha, G. (2019). Howard Aiken: American mathematician and inventor. *Encyclopaedia Britannica.* Retrieved from: https://www.britannica.com/biography/Howard-Aiken

Lotha, G. (2021a). John Mauchly. *Encyclopaedia Britannica.* Available at: https://www.britannica.com/biography/John-Mauchly

Lotha, G. (2021b). Sir Robert Alexander Watson-Watt: British physicist. *Encyclopaedia Britannica.* Retrieved from: https://www.britannica.com/biography/Robert-Alexander-Watson-Watt

Lotha, G. (2022a). Eugene Wigner: American physicist. *Encyclopaedia Britannica.* Retrieved from: https://www.britannica.com/biography/Eugene-Wigner

Lotha, G. (2022b). Bell Laboratories. *Encyclopaedia Britannica.* Available at: https://www.britannica.com/topic/Bell-Laboratories

MacBride, F., Marion, M., Frápolli, M., Edgington, D., Elliott, E., Lutz, S., & Paris, J. (2019). Frank Ramsey. *Stanford encyclopedia of philosophy.* Available at: https://plato.stanford.edu/entries/ramsey/#FounLogiMath

Mahoney, M. (2003). Computer science: The search for a mathematical theory. In John Kriege and Dominique Pestre (Eds.), *Companion encyclopedia of science in the twentieth century* (pp. 617-634). Oxford: Routledge.

Mahoney, M. (2011). *Histories of computing*. Cambridge: Harvard University Press.

Mar, G. (2015). Hao Wang's logical journey. *Journal of Chinese Philosophy*, 42, 540-561.

Markoff, J. (2012, March 23). David L. Waltz, Computer Science pioneer, dies at 68. *The New York Times.* Retrieved from: https://www.nytimes.com/2012/03/24/science/david-l-waltz-computer-science-pioneer-dies-at-68.html

Markowsky, G. (2021). Claude Shannon: American engineer. *Encyclopaedia Britannica.* Retrieved from: https://www.britannica.com/biography/Claude-Shannon

Massachussetts Institute of Technology (1999). Physicist George Valley Jr. is dead at 86. *MIT News.* Available at: https://news.mit.edu/1999/valley-1020

Merrill, D. (2006). The Truman doctrine: Containing communism and modernity. *Presidential Studies Quarterly,* 36(1), 27-37.

Masani, P. (1990). *Norbert Wiener 1894-1964.* Basel: Birkhäuser.

Merrill, D. (2012). *Augustus DeMorgan and the logic of relations.* Dordrecht and Boston and London: Kluwer.

Menger, K. (1930). Das Botenproblem. In Karl Menger (Ed.), *Ergebnisse eines Mathematischen Kolloquiums 2* (pp.11-12), Leipzig: Teubner, 11-12.

McCarthy, J., Minsky, M., Rochester, N., & Shannon, C. (2006). A proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955. *AI Magazine*, 27(4), 12-14.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4), 115-133.

McCorduck, P. (1979). *Machines who think.* San Francisco: W. H. Freeman and company.

Minsky, M. (1961). Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1), 8-30.

Minsky, M. (1972). *Computation: finite and infinite machines.* London: Prentice Hall.

Mosselmans, B. (2015). William Stanley Jevons. *Stanford encyclopedia of philosophy.* Retrieved from: https://plato.stanford.edu/entries/william-jevons/

National Research Concil. (1999). *Funding a revolution: Government support for computing research*. Washington, D.C: National Academy Press.

Neuber, M. (2018). Herbert Feigl. *Stanford encyclopedia of philosophy.* Available at: https://plato.stanford.edu/entries/feigl/

Newell, A., Shaw, J. C., & Simon, H. (1963). Chess-playing program and the problem of complexity. In Edward A. Feigenbaum & Julian Feldman (Eds.), *Computers and thought* (pp. 39-70). New York and San Francisco: McGraw-Hill.

Newell, A. (1955). The chess machine: an example of dealing with a complex task by adaptation. *Proceedings of the Western Joint Computer Conference,* 7, 101-108.

Newell, A., Shaw, J. C., & Simon, H. (1957). Programming the Logic Theory Machine. *Proceedings of the Western Joint Computer Conference*, 15, 218-239.

Newell, A., Shaw, J. C., & Simon, H. (1959a). Report on the General Problem Solving program (Report No. P-1584). RAND Corporation.

Newell, A., Shaw, J. C., & Simon, H. (1959b). *The process of creative thinking*. Santa Monica, CA: RAND Corporation.

Newell, A., Simon, H., & Perlis, A. (1967). Computer science. *Science.* 157 (3795), 1373-1374.

Nilsson, N. J. (2010). *The quest for artificial intelligence.* New York: Cambridge University Press.

Norberg, A. (1991). *Oral history interview with Terry Allen Winograd.* Stanford: The Charles Babbage Institute. Available at: https://conservancy.umn.edu/bitstream/handle/11299/107717/oh237taw.pdf

O'Connor, J., & Robertson, E. (1999). Ernst Friedrich Ferdinand Zermelo. *MacTutor history of mathematics archive.* Available at: https://mathshistory.st-andrews.ac.uk/Biographies/Zermelo/

O'Connor, J., & Robertson, E. (2000). Edward Vermilye Huntington. *MacTutor history of mathematics archive.* Available at: https://mathshistory.st-andrews.ac.uk/Biographies/Huntington/

O'Connor, J., & Robertson, E. (2001). Gerhard Gentzen. *MacTutor history of mathematics archive.* Available at: https://mathshistory.st-andrews.ac.uk/Biographies/Gentzen/

O'Connor, J., & Robertson, E. (2005). Thoraf Albert Skolem. *MacTutor history of mathematics archive.* Available at: https://mathshistory.st-andrews.ac.uk/Biographies/Skolem/

O'Connor, J., & Robertson, E. (2006). Jacques Herbrand. *MacTutor history of mathematics archive.* Available at: https://mathshistory.st-andrews.ac.uk/Biographies/Herbrand/

Parsons, C. (2011). Hao Wang. In Charles Parsons & Montgomery Link (Eds.), *Hao Wang: logician and philosopher* (pp. 7-14). London: College Publications.

Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. New York: Basic Books.

Peckhaus, V., & Kahle, R. (2002). Hilbert's paradox. *Historia Mathematica*, 29(2), 157-175.

Perry, T. S. (1995). Lotfi A. Zahde [fuzzy logic inventor biography], *IEEE spectrum*, 32(6), 32-35.

Platzman, G. W. (1979). The ENIAC computations of 1950: Gateway to numerical weather computations. *Bulletin of the American Meteorological Society*, 60(4), 302 312. https://doi.org/10.1175/1520-0477(1979)060<0302:TECOTN>2.0.CO;2.

Poincaré, H. (1906). Les mathématiques et la logique. *Revue de Métaphysique et de Morale*, 14(3), 294-317.

Poveda, J. (2013). Esquemas de votación Borda aplicados a la clasificación de imágenes utilizando los histogramas de color RGB, HSV, y el descriptor de distribución de color MPEG-7 (Bachelor's thesis). Universidad Politecnica Salesiana del Ecuador, Cuenca, Ecuador.

Poveda, J., & Robles, V. Combination of HSV and RGB and MPEG-7 descriptor: Preliminar results and future work. *Ingenius*, (8), 3-10.

Quinn, A. H. (1988). *Edgar Allan Poe: A critical biography.* Baltimore: The John Hopkins University press.

Quine, W. V. O. (1936a). Set-theoretic foundations of logic. *The Journal of Symbolic Logic*, 1(2), 45-57.

Quine, W. V. O. (1936b). Towards a calculus of concepts. *The Journal of Symbolic Logic*, 1(1), 2-25.

Quine, W. V. O. (1937). New foundations for mathematical logic. *The American Mathematical Monthly*, 44(2), 70-80.

Quine, W. V. O. (1972). *Lógica matemática.* Madrid: Revista de occidente.

Quine, W. V. O. (1974). *Methods of logic.* London: Routledge & Kegan Paul.

Quine, W. V. O. (1985). *The time of my life. An autobiography.* Cambridge: MIT Press.

Quintanilla, S. (2002). Arturo Rosenblueth y Norbert Wiener: dos científicos en la historiografía de la educación contemporanea. *Revista Mexicana de Investigación Educativa*, 7(15), 303-329.

Raatikainen, P. (2013). Gödel's Incompleteness Theorems. *Stanford encyclopedia of philosophy.* Retrieved from: https://plato.stanford.edu/entries/goedel-incompleteness/

Rabin, M. O. (1960). *Degree of difficulty of computing a function and a partial ordering of recursive sets* (Technical report No. 2). Hebrew University.

Ramsey, F. P. (1926). The foundations of mathematics. *Proceedings of the London Mathematical Society,* 2(25), 338-384.

Randeria, S. (2006). Entangled histories of uneven Modernities: civil society, caste solidarities and legal pluralism in post-colonial India. In John Keane (Ed.), *Civil Society- Berlin perspectives* (pp. 213-242). New York, Berghahn.

Roberts, L. G. (1963). *Machine perception of three dimensional solids.* (Doctoral dissertation). Retrieved from: https://dspace.mit.edu/handle/1721.1/11589

Rodriguez, E. (2017a). Augustus DeMorgan: English mathematician and logician. *Encyclopaedia Britannica*. Retrieved from:
https://www.britannica.com/biography/Augustus-De-Morgan

Rodriguez, E. (2017b). Paul Isaak Bernays: Swiss logician and mathematician. *Encyclopaedia Britannica.* Available at: https://www.britannica.com/biography/Paul-Isaak-Bernays

Rodriguez, E. (2017c). Oswald Veblen: American mathematician. *Encyclopaedia Britannica.* Available at: https://www.britannica.com/biography/Oswald-Veblen

Rodriguez, E. (2022a). Manhattan project. *Encyclopaedia Britannica.* Available at: https://www.britannica.com/event/Manhattan-Project

Rodriguez, E. (2022b). J. Presper Eckert Jr. *Encyclopaedia Britannica.* Available at: https://www.britannica.com/biography/J-Presper-Eckert-Jr

Rodriguez, E. (2022c). Jean Piaget: Swiss psychologist. *Encyclopaedia Britannica.* Retrieved from: https://www.britannica.com/biography/Jean-Piaget

Rogers, B. (2020). Sir A. J. Ayer: British philosopher. *Encyclopaedia Britannica*. Available at: https://www.britannica.com/biography/A-J-Ayer

Russell, B. (2000). Letter to Frege "(Original work published 1902)". Halle. Reprinted in Jean van Heijenoort (Ed.), *From Frege to Gödel, A source book in mathematical logic, 1879-1931* (pp.124-125). Lincoln, NE: toExcel.

Russell, B. (1906). Les paradoxes de la logique. *Revue de Métaphysique et de Morale,* 14(5), 627-650.

Russell, B. (1907). On some difficulties in the theory of transfinite numbers and order types. *Proceedings of the London Mathematical Society,* 2(1), 29-53.

Russell, B. (1908). Mathematical logic as based on the theory of types. *American journal of mathematics,* 30(3), 222-262.

Russell, B., & Whitehead, A. N. (1927). *Principia mathematica.* Cambridge: Cambridge University Press.

Russell, B. (1996). *The principles of mathematics.* New York: W. W. Norton & company.

Russell, B. (2010). *The basic writings of Bertrand Russell.* London and New York: Routledge.

Russell, S., & Norvig, P. (2010). *Artificial intelligence: a modern approach.* New York: Prentice Hall.

Samuel, A. (1963). Some studies in machine learning using checkers. In Edward A. Feigenbaum & Julian Feldman (Eds.), *Computers and thought* (pp. 71-105). New York and San Francisco: McGraw-Hill.

Scanlan, M. (2000). "The known and unknown H. M. Sheffer". *Transactions of the Charles S. Pierce Society,* 36(2), 193-224.

Schaffer, S., & Shapin, S. (1985). *Leviathan and the air pump: Hobbes, Boyle, and the experimental life.* Oxford: Princeton University Press.

Selfridge, O. (1955). Pattern recognition and modern computers. *Procedings of the 1955 Joint Computer Conference,* 7, 91-93.

Shannon, C. E. (2001). A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review,* 5(1), 3-55.

Shannon, C. E. (1950). A chess-playing machine. *Scientific American*, 182(2), 48-51.

Shannon, C.E. (1953). Computers and Automata. *Proceedings of the IRE*, 41(10), 1234-1241.

Shallit, J. (2010). Alan Cobham. *Recursivity.* Available at: http://recursed.blogspot.com/2010/04/alan-cobham.html

Simon, H., Divine, W., Cooper, E., & Chernin, M. (1941) *Determining work loads for professional staff in a public welfare agency.* Berkeley: Bureau of Public Administration, University of California.

Simon, H., Shepard, R., & Sharp, F. (1943). *Fire losses and fire risks.* Berkeley: Bureau of Public Administration, University of California.

Simon, H. (1943). *Fiscal aspects of metropolitan consolidation.* Berkeley: Bureau of Public Administration, University of California.

Simon, H. (1947). Effects of increased productivity upon the ratio of urban to rural population. *Econometrica. Journal of Econometric Society,* 15(1)*,* 31-42.

Simon, H. (1952). On the definition of the causal relation. *The Journal of philosophy*, 49(16), 517-528.

Simon, H. (1996). *Models of my life.* London: MIT Press.

Simon, H. (1997a). *Administrative behavior.* New York: The Free Press.

Simon, H. (1997b). Allen Newell (1927-1992): A bibliographical memoir by Herbert A. Simon. *Bibliographical memoirs of members of the National Academy of Sciences*, 71, 141-173.

Simon, H. (2006). *Las ciencias de lo artificial.* Granada: Editorial Comares.

Newell, A., & Simon H. (1956). The logic theory machine: a complex information processing system. *IRE Transactions on Information Theory*, 2(3), 61-79.

Simons, P. (2014). Jan Lukasiewicz. *Stanford encyclopedia of philosophy.* Available at: https://plato.stanford.edu/entries/lukasiewicz/

Simons, P. (2015). Stanislaw Leśniewski. *Stanford encyclopedia of philosophy.* Available at: https://plato.stanford.edu/entries/lesniewski/

Skolem, T. (1930). Einige Bemerkungen zu der Abhandlung von E. Zermelo: "Über die Definitheit in der Axiomatik. *Fundamenta Mathematicae*, 15, 337-341.

Sobel, I. (1970). *Camera models and machine perception.* PhD thesis. Stanford University.

Spade, P. V. (2019). William of Ockham. *Stanford encyclopedia of philosophy.* Available at: https://plato.stanford.edu/entries/ockham/

Stefferud, E. (1963). *The logic theory machine: a model heuristic program* (No. RM 3731 CC). Santa Monica: RAND Corporation.

Strong, J. V. (1976). The infinite ballot box of nature: DeMorgan, Boole, and Jevons on probability and the logic of induction. In *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association* (Vol. 1976, No. 1, pp. 197-211). Philosophy of Science Association.

Stueck, W. (1995). *The Korean war: an international conflict.* New Jersey: Princeton University Press.

Tarski, A. (1983). Some observations on the concepts of w-consistency and w-completeness "(Original work published 1933)". Reprinted in John Corcoran (Ed.), *Logic, semantics, and metamathematics. Papers from 1923 to 1938* (pp. 279-295). Indiana, Hackett Publishing Company.

Thompson, K. O. (2013). Jin Yuelin: Formidable, formative philosopher of twentieth-century China. *China Review International,* 20(1/2), 40-46.

Turing, A. (1936). On computable numbers with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(1), 230-265.

Turing, A. (1950). Computing machinery and intelligence. *Mind*, 59 (236), 433-460.

Turing, A. (1963). Digital computers applied to games "(Original work published 1953)". In Bertram Vivian Bowden (Ed.), *Faster than thought: a symposium on digital computing machines* (pp. 286-310). London: Sir Isaac Pitman & Sons.

Van Heijenoort, J. (2000). *From Frege to Gödel: A source book in mathematical logic, 1879-1931.* Cambridge and London: Harvard University Press.

Von Neumann, J., & Goldstine, H. (1947). Numerical inverting of matrices of high order. *Bulletin of the American Mathematical Society*, 53(11), 1021-1099.

Waltz, D. (1972). *Generating semantic descriptions from drawing of scenes with shadows.* PhD thesis. Massachussetts Institute of Technology.

Wang, H. (1960a). Toward mechanical mathematics. *IBM Journal of Research and Development*, 4(1), 2-22.

Wang, H. (1960b). Proving theorems by pattern recognition. *Communications of the ACM* , 3(4), 220-234.

Wang, H. (1961). Proving theorems by pattern recognition. *The Bell System Technical Journal*, 40(1), 1-41.

Wang, H. (1990). *Computation, logic, philosophy: A collection of essays.* Dordrecht: Kluwer.

Wang, H. (1991). Reflexiones sobre Kurt Gödel. Madrid: Alianza Editorial.

Wang, H. (2011a). Memories related to professor Jin Yuelin. In Charles Parsons & Montgomery Link (Eds.), *Hao Wang: logician and philosopher* (pp. 27-38). London: College Publications.

Wang, Z. (2010). Transnational science during the Cold War: The case of Chinese/American scientists. *Isis,* 101(2), 367-377.

Watson, I. (2012). How Alan Turing invented the computer age. *Scientific American.* Available at: https://blogs.scientificamerican.com/guest-blog/how-alan-turing-invented-the-computer-age/

Weiner, J. (1999). *Frege.* Oxford: Oxford University Press.

Wiener, N. (1970). *Extrapolation, interpolation, sand smoothing of stationary series.* Cambridge: The MIT Press.

Wiener, N. (1948). *Cybernetics or control and communication in the animal and the machine.* Paris: Herman & Cie Editeurs.

Winograd, T. (1971). *Procedures as a representation for data in a computer program for understanding natural language.* PhD thesis. Available at: https://dspace.mit.edu/handle/1721.1/7095

Young, G. (2013). Servomechanism. *Encyclopaedia Britannica.* Retrieved from: https://www.britannica.com/technology/servomechanism

Yost, J.R. (2013). Appropriation and independence: BTM, Burroughs, and IBM at the advent of the computer industry. *IEEE annals of history of computing*, 35(4), 5-17.

Yufan, H., & Zhihai, Z. (1990) China's decision to enter the Korean War: History revisited. *The China quarterly,* 121, 94-115.

Zach, R. (2003). Hilbert's Program. *Stanford Encyclopedia of Philosophy*. Retrieved from: https://plato.stanford.edu/entries/hilbert-program/#Hilbert:05/

Zermelo, E. (2000). Investigations in the foundations of set theory "(Original work published 1908)". Reprinted in Jean van Heijenoort (Ed.), *From Frege to Gödel, A source book in mathematical logic, 1879-1931* (pp. 199-215). Lincoln, NE: toExcel.

Zermelo, E. (1908). Untersuchungen über die Grundlagen der Mengenlehre I. *Mathematische annalen,* 65(2), 261-281.

Zhaowu, H. (2011). Remembering Wang Hao. In Charles Parsons & Montgomery Link (Eds.), *Hao Wang: logician and philosopher* (pp. 49-53). London: College publications.

Zhang, M., & Qiang, P. (2012). Jialin Xie wins China's top science award. *International linear collider.* Available at: http://newsline.linearcollider.org/2012/02/23/jialin-xie-wins-chinas-top-science-award