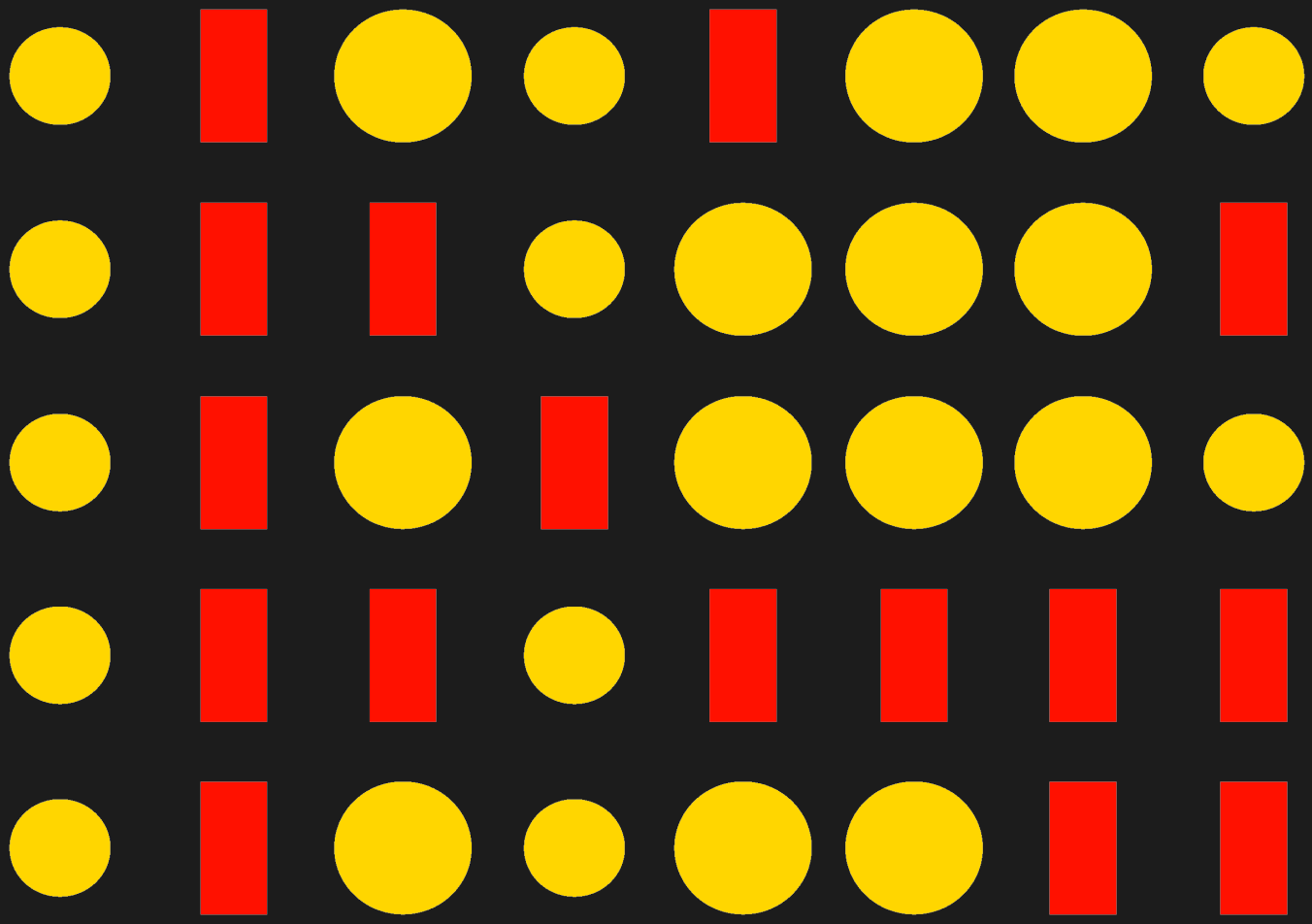

HaPoC

Warsaw University
of Technology
October 18–20

2023

PW

7th International Conference on the
History and Philosophy of Computing

Contents

About HaPoC-7

As a technological and scientific field, computing is constantly expanding in scope and depth, affecting all aspects of contemporary life. Our perceptions and conceptions of the field are also constantly evolving, though not always in lockstep with technological and scientific developments. HaPoC conferences are a place to develop ways of both reflecting on and thinking ahead of these developments.

Achieving this goal requires a broad perspective on computing, in which social and cultural aspects play a central role. For these reasons, thinking about computing technologies and practices requires an interdisciplinary approach in which art is as necessary as engineering, anthropological insights as important as mathematical models, and the critical perspectives of history, philosophy, ethics and law as crucial as the formal models of theoretical computer science.

Since 2011, the biennial History and Philosophy of Computing (HaPoC) conference series has contributed to building such an interdisciplinary community and environment. We aim to bring together historians, philosophers, computer scientists, social scientists, legal scholars, designers, engineers, practitioners, artists, logicians, mathematicians, each with their own experience and expertise, to participate in the collective construction of a comprehensive and forward-looking image of computing.

Warsaw University of Technology is honoured to host the 7th edition of HaPoC in October 2023 at the Faculty of Administration and Social Sciences.

Timetable

Wednesday, 18 October

Opening	Mała Aula
08:00–08:45	Registration, morning coffee
08:45–09:00	Welcome by Anna Zalcewicz, Dean of the faculty, and the organisers
Keynote 1	**Mała Aula**
09:00–10:00	Corey J. Maley: Computation, computability, and representation
Session 1: Physical Computation	**Mała Aula**
10:00–10:30	Nick Wiggershaus: Physical programmability
10:30–11:00	Matej Kohar: Physical computation from an engineering standpoint
11:00–11:30	Coffee Break
Session 2: History and Philosophy of Computing	**Mała Aula**
11:30–12:00	Oliver Marshall: Alonzo Church's early formalism
12:00–12:30	Bernardo Gonçalves: Lady Lovelace's Objection: The Turing-Hartree disputes over the meaning of digital computers, 1946-1951
12:30–13:00	Melissa Antonelli and Paolo Pistone: On probabilistic and counting computation: A historical study
13:00–14:30	Lunch Break
Session 3: History of Computing	**Room 213**
14:30–15:00	Helena Durnova: Czechoslovak origins of Sofsem
15:00–15:30	Zbigniew Stachniak: Personal computing paradigm – 50 years after
15:30–16:00	Paweł Grabarczyk: Finding a new home. The history of the domestication of Atari computers in Poland
16:00–16:30	Coffee Break
Session 4: Computing and Physics	**Room 213**
16:30–17:00	Jean-Philippe Martinez and Robert Harlander: The emergence of algebraic programs for the calculation of Feynman diagrams in high energy physics
17:00–17:30	Marien Raat: Computability in nested closed timelike curves
Session 5: Ethics and Society	**Room 213**
17:30–18:00	Shannon Brick: Is individual data sovereignty exploitative?
18:00–18:30	Martin Baesler: The importance of judgment – Hannah Arendt and Joseph Weizenbaum on the use of computer power
20:00–open	Artistic Event "Homerun" at Academy of Fine Arts

Thursday, 19 October

08:45–09:00	Morning coffee
Keynote 2	**Room 213**
09:00–10:00	Giuseppe Primiero: On the past and the future of (HaPo)Computing
Session 6: History of Programming	**Room 213**
~~10:00–10:30~~	Amelie Mittlmeier: *talk cancelled*
10:30–11:00	Tomas Petricek: The rise and fall of extensible programming languages
11:00–11:30	Coffee Break
Session 7: Aesthetics	**Room 213**
11:30–12:00	Pierre Depaz: Metaphors and mental spaces in the cognitive aesthetics of source code
12:00–12:30	Caterina Moruzzi and Oreste Campagner: AI and art: Non-human, still human, or all too human?
12:30–13:00	Edgar Lejeune: Computing without computers in the 1970s? Jacques Bertin's 'graphics' as a counterpart to 'traditional' humanities computing in France
13:00–14:30	Lunch Break
Session 8: Algorithms	**Room 213**
14:30–15:00	Alberto Naibo and Thomas Seiller: Formalising the notion of algorithm
15:00–15:30	A.G. Holdier: Agentic and algorithmic context collapse
15:30–16:00	Philippos Papayannopoulos: Different conceptions of algorithms and the challenges of developing a definition
16:00–16:30	Coffee Break
Session 9: Machine Learning	**Room 213**
16:30–17:00	Koray Karaca: Model-choice accountability and deep machine learning models
17:00–17:30	Chiara Manganini and Giuseppe Primiero: The ontology of Machine Learning systems
17:30–19:00	HaPoC General Assembly – Chair: Juan Luis Gastaldi Room 213
20:00–open	Conference Dinner

Friday, 20 October

08:45–09:00	Morning coffee
Session 10: Philosophy of Artificial Intelligence	Room 213
09:00–09:30	Darrell Rowbottom, Andre Curtis-Trudel and William Peden: Does the No Miracles Argument apply to AI?
09:30–10:00	Amira Moeding: Empiricism and rationalism in language processing and AI research: On the nature of the machine
10:00–10:30	Jo Ann Oravec: Cheating (and cheating detection) in the advent of ChatGPT: How Artificial Intelligence is reframing academic production
10:30–11:00	Caroline Lawitschka and Philip König: This abstract was written by ChatGPT: Empowering intellectual workers through AI-generated content
11:00–11:30	Coffee Break / Brunch
Session 11: Cognitive Science	Room 213
11:30–12:00	Phillip Kieval: Getting real about neural data science: Neural manifolds and exploratory concept formation
12:00–12:30	Michał Piekarski: Information as constraint for neurocognitive mechanisms. Argument from the predictive processing framework
12:30–12:45	Coffee Break / Brunch
Keynote 3	Room 213
12:45–13:45	Marcin Miłkowski: Peer review of code
13:45–14:00	Closing of the conference

Abstracts

On probabilistic and counting computation: A historical study

Melissa Antonelli (HIIT, University of Helsinki) and Paolo Pistone (University of Bologna)

From the 1950s on, studies on randomized algorithms started spreading and, nowadays, probabilistic computation is pervasive in almost every areas of CS. In particular, in the 1970s, “quantitative” models were formally defined, for example probabilistic Turing machines (PTMs) by Santos and Gill, or counting and threshold machines by Valiant and Simon. Since the beginning, the development of such machine models has been related with inquiries about computational power and resource issues. Indeed, new probabilistic (e.g. BPP and PP) and counting (e.g. #P) classes were introduced. Clearly, enumerating and probabilistic computations are strongly linked; in 1975, Simon proved that, within certain conditions, PTMs and threshold machines yield the same complexity class. Although these results are well-known, the corresponding terminology is sometimes imprecise or misleading. For instance, the terms “counting” and “threshold machines” are usually used as interchangeable, and PTMs are defined in different ways - either as deterministic TMs accessing a random-bit source in the form of an oracle or as NTMs with transition functions to be chosen with (possibly equal) probability. Being this the root of potential misunderstanding, we aim to clarify a few notions (and the differences between them) crucial in this field, by introducing their original formulations in the context where they first appeared.

Keywords: Randomized Computation, History of Computing, Probabilistic and Counting Machines

The importance of judgment – Hannah Arendt and Joseph Weizenbaum on the use of computer power

Martin Baesler (University of Freiburg)

Joseph Weizenbaum warned against using computers to replicate human thinking, in spite of and in fact because this would lead to more efficient problem-solving methods. What he put forward as the ‘computer metaphor’ resembles Hannah Arendt’s critique of the calculation processes that replace political judgment. How can the concept of political judgment help unmask the computer metaphor, and in which way can it clarify what the responsible use of computers would be? In this paper, I will create a dialogue between Arendt’s reflections on the impact of computers as tools being used to enhance scientific viewpoints on the human world and Joseph Weizenbaum’s critique of the computer metaphor. Arendt’s concept of political judgment helps to clarify what the responsible use of highly improved intelligent computer systems in the political sphere would mean. The responsible use of computers is usually believed to require an understanding or at least

insight into algorithmic decision-making. This, however, becomes more and more difficult when faced with 'self-learning' machines. In my reading of Arendt's political judgment, it is crucial that we become a distanced spectator of the use of algorithms and create representations of points of view that reveal the controversiality behind standards of algorithmic problem-solving. By taking the responsibility and importance of judgment seriously we can avoid the fallacy of the computer metaphor.

Keywords: Algorithmic Decision-Making, Political Judgment, Computer Metaphor

Is individual data sovereignty exploitative?

Shannon Brick (Georgetown University)

This paper is about models of data sovereignty which empower data subjects to claim payment, from online platforms, for their personal data. It argues that these arrangements are exploitative. However, the claim is not that online platforms are exploiting users. Rather, the claim is that when users of an online platform receive payment in exchange for their personal data, those users are exploiting non-users of the platform. The paper starts by explaining the movement towards what I call 'individual data sovereignty.' The origins of this movement are located in the perception that online platforms like Google exploit users when they use our personal data to generate powerful, and therefore profitable, predictions about human behavior. The second part of the paper explains why individual sovereignty realizes a new form of horizontal exploitation. Key to mounting this argument is recognizing the relational character of all personal data. In the third and final part of the paper, the moral failure of individual data sovereignty is traced to an inaccurate understanding of the reasons for which current models of data capitalism are problematic. It is claimed that current models do not exploit individual users. The problem is that data capitalists take advantage of a collective action problem to expropriate for themselves data that is best understood as a collective epistemic resource.

Keywords: Data, Sovereignty, Capitalism, Exploitation

Metaphors and mental spaces in the cognitive aesthetics of source code

Pierre Depaz (Paris-3 Sorbonne Nouvelle)

While programmers often refer to the beauty of code, definitions of such instances primarily take place through empirical examples , broad heuristics or as negative definitions. Indeed, the aesthetic aspects of source code—its positively valued formal representation—have been present since at least Donald Knuth's magnum opus, and always related to the functionality of the executed code. This contribution, building on findings from our doctoral research, draws on literature studies and cognitive psychology to highlight how the formal presentation of source code—i.e. its surface structure—enables the understanding of the computational processes described—its deep structure. As such, since source code is written for both humans and machine, we argue that

the role of aesthetics in source code is to facilitate the comprehension of what the program does. Building on Nelson Goodman's hypothesis that aesthetic presentation enables a specific form of intellectual communication , we will present how metaphors and linguistic structure contribute to the creation of mental spaces. Thus, we present a framework to consider how one reads and writes source code, and to complement a more empirical approach on expliciting the role of aesthetic thinking in the understanding of computational actions.

Keywords: source code, aesthetics, cognition, programming

Czechoslovak origins of Sofsem

Helena Durnova (Masaryk University)

Now an internationally recognised conference Sofsem (SOFtware SEMinar) was first held in Czechoslovakia in November 1974. It was organised by the UN Research Computing Center in Bratislava, conceived in 1967 and it was not the first gathering providing space for theoretical discussions of computer science in the geographical area: the MSFCS conferences (Mathematical Foundations of Computer Science) were held alternately in Poland and in Czechoslovakia since 1972. While MFCS were scientific conferences (with proceedings published in English), Sofsem was rather a training space for prospective top-level Czechoslovak programmers, with course material in Czech and Slovak. For the first decade, the chair of the programme committee was Jozef Gruska (from the UN Research Computing Center) and the whole series was opened with an extensive talk on structured programming delivered by Jiří Hořejš. Already that talk was accompanied by sketches mocking the programmers' work. Further insider jokes were a part and parcel of the so-called Antisofsem, a social evening which took place on the last night of the event. In my talk, I will explore the prehistory and early history of this enterprise as well as its connections to the national and international players.

Keywords: Sofsem (SOFtware Seminar), MFCS (Mathematical Foundations of Computer Science), UN Research Computing Centre in Bratislava, Jozef Gruska, Jiří Hořejš, cultural history of programming

Lady Lovelace's Objection: The Turing-Hartree disputes over the meaning of digital Computers, 1946-1951

Bernardo Gonçalves (University of São Paulo)

Should machines be emancipated from slavery and given 'fair play' to 'compete with men in all purely intellectual fields'? Or should this be considered part of the fashion to 'decry human reason' and 'a path which leads straight to Nazism'? Can machines think? Or can they only 'do whatever we know how to order it to perform'? In the postwar years, 1946-1951, these ethical and ontological questions were debated by the British computer pioneers Alan Turing and Douglas Hartree, who differed in their interpretations of the meaning and significance of digital computers as a new piece of science and technology. Hartree emphasized the unprecedented calculation speeds of digital

computers, and envisioned applications in physics, logistics, and warfare. Turing, who envisioned applications in biology and cognition, emphasized the potential for computers to outperform humans intellectually. This included capabilities considered distinctly human, which Hartree denied by mobilizing an earlier dictum of Ada Lovelace. I will examine the Turing-Hartree disputes and draw a parallel between their positions and their perspectives on postwar Britain. I will suggest that Hartree saw digital computers as a means of increasing order in society, while Turing saw them as a means of creating diversity and democratizing the distribution of power.

Keywords: Alan Turing, Douglas Hartree, Early modern computing, Postwar Britain, Computers and society, History and future of computing

Finding a new home. The history of the domestication of Atari computers in Poland

Paweł Grabarczyk (IT University of Copenhagen)

This paper presents a historical case study of the domestication of computing education and computer technology in a non-Western country. The aim of the paper is to show that the process of implanting a Western-born technology outside of its original economic, cultural, and societal context may result in the creation of a new, grassroots computing culture that differs substantially from its original (Western) form. Specifically, I analyze the case of the adoption and adaptation of the Atari 8-bit platform in communist Poland “behind the iron curtain” (mid 1980s). The official computing education in Polish schools rarely went beyond LOGO and BASIC languages, using computers more as electrified enhancements of lessons of mathematics than their own stand-alone subject. In stark contrast, enthusiast magazines published in the country taught people how to crack programs, write their own software in assembler, hack hardware, and build their own DIY solutions. Even though the case study is historic in nature, I suggest that it can be used to illustrate a general problem: the need to acknowledge and include computing education grassroots movements that are typically neglected in developing countries, but end up being formative for generations of programmers. I suggest that this type of clash of computing educational practices is a likely consequence of the adoption of Western technology in a non-Western context.

Keywords: Computing history, Atari 8-bit computers, Polish history, Computing education, Grass-roots movements

Agentic and algorithmic context collapse

A.G. Holdier (University of Arkansas)

”Context collapse” occurs when discursive spaces are crowded such that speech acts performed in one leak into another, paradigmatically facilitated by the hyperconnectivity of contemporary communication technology. Previous theorists have distinguished two kinds of collapse in terms of a speaker’s preferences: what I call authorial collapse is intended by the speaker (for a variety of reasons, each revolving around the encouragement of interpersonal connections), while adversarial

collapse is not (and manifests primarily as a privacy violation). In this paper, I articulate a second, heretofore unexplored dimension of context collapse grounded in the phenomenon's contextual kinematics: agentic collapse is caused by a social agent, whereas algorithmic collapse occurs beyond the control of any individual. After analyzing the resultant fourfold taxonomy and its implications for both intimacy and social cohesion in the age of artificial intelligence, I develop the notion of metacontexual (or ecological) context collapse and explain its role in perspectival conflicts.

Keywords: Context Collapse, Privacy, Online Speech, Ethics of Computing, Intimacy

Model-choice accountability and deep machine learning models

Koray Karaca (University of Twente)

In recent years, the need for algorithmic accountability (AA) has become a pressing issue in contexts where automated decision (AD) systems are used to make high-stake decisions about human subjects. In this talk, I will address the epistemological dimension of AA that concerns the choice of DML models (to be) used in the design of AD systems. I shall refer to the kind of AA that applies to the choice of DML models as model-choice accountability (MCA). I shall characterize MCA as the obligation to provide necessary adequacy-for-purpose (AfP) justifications as to why the chosen model is adequate for the design of the relevant AD system. As I shall suggest, demonstrating MCA boils down to finding relevant AfP justifications for the choice of a model that is essential to the design of an AD system. This is basically a manifestation of the problem of justifying model choice, for the solution of which an appropriate account of model evaluation is needed. To this end, I will draw on Wendy Parker's recent account (2020) of model evaluation, as it considers AfP as the sole criterion of model choice.

Keywords: deep machine learning models, algorithmic accountability, adequacy for purpose

Getting real about neural data science: Neural manifolds and exploratory concept formation

Phillip Kieval (University of Cambridge)

Recent research suggests that manifolds play an important role in neural computations. Manifolds are continuous, low-dimensional structures embedded in high-dimensional neural activity. Investigators purport to uncover these structures by using data-analytic techniques to reduce the dimensionality of patterns of neural activity and subsequently reveal the underlying dynamics that are functionally relevant to a specific task. These data-analytic practices, however, introduce a range of implicit assumptions which threaten to cloud our analyses with seemingly arbitrary modelling choices. In this paper, I argue that these techniques produce data models which aim to represent a set of phenomena in an ongoing process of stabilization. Neuroscientists use extent experimental and data-analytic methods of measuring and visualizing neural activity to operationally define the concept of a neural manifold. Hence, neuroscientists must make data modelling choices to tentatively fix the referent of the concept. Attributing mechanistic significance to neural manifolds,

however, requires novel methods of intervening on the manifolds themselves to establish their difference-making role in neural computations.

Keywords: neural computation, neural manifolds, exploratory experiments, operational definition, concept formation

Physical computation from an engineering standpoint

Matej Kohar (Technische Universität Berlin)

Usually, the problem of physical computation is formulated as: "Under what conditions does a given physical system S perform/implement (a given) computation C?" In this paper, I present an alternative tripartite formulation of the problem of physical computation reflecting computer engineering/design concerns: "Given an abstract computation C, what conditions must a physical system S satisfy so that a given user U can use S to perform C?" I argue that on this formulation, the relative importance of corollary issues such as pancomputationalism, simultaneous implementation, and the problem of computational explanation decreases. Conversely, further considerations surface regarding the relation between the computing system and its intended user. I introduce six criteria which specify the relevant notion of use: the input criterion, the state transition criterion, the result criterion, resetability, reliability and efficiency. I show that the current accounts of physical computation formulated as answers to the original formulation of the problem do not guarantee the fulfilment of these criteria. This shows that the alternative formulation cannot be reduced to the original one, and constitutes a separate problem faced by computer engineers. I conclude that different questions of physical computation arise across different scientific disciplines like computer engineering or cognitive science.

Keywords: computational implementation, computer engineering, computing systems

This abstract was written by ChatGPT: Empowering intellectual workers through AI-generated content

Caroline Lawitschka (University of Vienna) and Philip König (SBA Research)

With this contribution, written by ChatGPT, we want to investigate the potential of AI-driven tools to facilitate interdisciplinary collaboration, knowledge sharing, and learning in an ever-evolving academic landscape. By reducing the burden of time-consuming tasks, AI has the potential to enable researchers to engage in cross-disciplinary work, bridging gaps between seemingly unrelated fields, and fostering communication between experts from different disciplines. We discuss the implications of AI for generalists and specialists and the role of AI-generated content in promoting interdisciplinary understanding. Moreover, we examine the transformative potential of AI-driven tools in democratizing knowledge and enhancing interdisciplinary endeavours, further strengthening the connections between disciplines. By exploring the advantages of AI tools in providing flexible, responsive, and efficient learning experiences customized to individual needs, we emphasize the significance of AI in an increasingly interconnected academic environment. By examining

these issues, we aim to stimulate interdisciplinary discussions on the transformative potential of generative AI and encourage conversations on responsible harnessing of this technology for the advancement of intellectual work and learning in the age of AI. With this we want to set the stage for further exploration of AI's role in shaping the future of research and education, fostering a more interconnected, informed, and innovative academic community.

Keywords: ChatGPT, Interdisciplinary Collaboration, AI in Research and Education

Computing without computers in the 1970s? Jacques Bertin's 'graphics' as a counterpart to 'traditional' humanities computing in France

Edgar Lejeune (Centre Alexandre Koyré (EHESS & CNRS))

In the 1960s, the French cartographer Jacques Bertin (1918-2010) developed what he referred to as 'graphics', a set of graphical methods designed to help scholars dealing with huge data tables. Bertin's idea was that anyone could process scientific information - and thus analyze it - without mathematical knowledge nor without computers, using just the principles of visual perception. In order to help scholars doing so, he formalized in his classical 1967 book, *La Sémiologie graphique*, clear graphical rules and imagined a set of physical devices, such as his famous reorderable matrices. In the following years, Bertin and its collaborators extensively considered the differences between 'graphics' and comparable computer science-related procedures, such as factorial analysis. They also imagined ways for their methods to compete with the development of computer science and thus tried to digitize 'graphics' using cathodic screens, keyboards and customized typewriters. This communication will rely on a set of published articles and personal archives in order to discuss to what extent Jacques Bertin's 'graphics' could be considered a counterpart to more 'traditional' forms of computing in the late 1960s and 1970s. By doing so, I will also address by its margin the more general question of the definition of computing.

Keywords: History of computing, Digital Humanities, Information Vizualisation

The ontology of machine learning systems

Chiara Manganini and Giuseppe Primiero (both Università degli Studi di Milano)

Along with the debate on the nature of computational artefacts at large, the philosophy of computer science has recently addressed the ontological question of what constitutes a physical computational artefact. Despite their nuances, all the analyses proposed so far largely rest on the central notions of specification, implementation, and correctness. In this paper, we extend this debate to machine learning (ML) systems, showing that all the three concepts need to undergo a substantial revision when it comes to predictive computational artefacts. The fundamental shift that a ML ontology must accommodate is the new epistemological role played by specification, defined as the set of the artefact functional requirement(s). In predictive contexts, in fact, specifications are discovered through the actual training process rather than fixed from the beginning, this having

deep consequences on the relevant notions of correctness and implementation involved. With the help of a real-world example, we hence propose a new account of the ontology of ML systems, showing that this revised framework allows to formulate new and systematic insights on the notions of correctness and miscomputation. Finally, the relationship between miscomputation and bias is explored, opening the way to a new understanding of ML fairness, particularly relevant in decision-making contexts.

Keywords: Ontology of predictive computational artefacts, Philosophy of computing, Machine learning systems, Miscomputation, Bias

Alonzo Church's early formalism

Oliver Marshall (UNAM)

Throughout his career, Church was concerned with providing a foundation for mathematics that meets two basic conditions: it is adequate for extant mathematics and avoids paradoxes. Church's attempts to achieve this are especially illuminating because he discusses, at various points, the philosophical outlook that he weaves into his presentation of formal logic and formalized mathematics. They also had surprising implications for the theory of computation. In this talk, we describe Church's A Set of Postulates for the Foundations of Logic (1932-33) and his subsequent paper "Richard's Paradox" (1934), in which he formulates his early philosophy of mathematics and logic. This shows the influence of Russell and Hilbert and contains many ideas that would preoccupy Church throughout his career – such as a concern with intensional contexts and the demand that a formal object language suffices for mathematical proof and logical analysis to be public and communicable without the need for further clarification in the meta-language. We argue that this philosophy explains his reaction to Richard's paradox and Gödel's first theorem. These results are "deplorable," "nihilistic," and imply "that the whole program of the mathematical logician is futile" (1934: 97).We show that rather than deriving Richard's paradox as a version of the Liar, Church himself poses it in terms of the difference between enumerable and non-enumerable sets. We compare this with his subsequent work on the Entscheidungsproblem.

Keywords: Church, Lambda calculus, function in intension, proposition, formalism, paradox, incompleteness, undecidability

The emergence of algebraic programs for the calculation of Feynman diagrams in high energy physics

Jean-Philippe Martinez (Technische Universität Berlin) and Robert Harlander (RWTH Aachen)

Today, our best theory to describe the behavior and interaction of subatomic particles is quantum field theory. Practical calculations of scattering or decay processes of elementary particles within this theory are mostly based on perturbative methods, for which a powerful computational tool was introduced in the late 1940s, the Feynman diagrams. They provide a pictorial representation of the

mathematical expressions of the perturbative expansion of quantum field theory. The test of the adequacy between theory and experiment led, in the 1960s, to the computation of higher orders in the perturbative expansion. This greatly increased the number of diagrams to be considered, as well as their internal complexity. The resulting calculations proved "hopelessly cumbersome" and computers became necessary. For this purpose, algebraic programs were specifically developed, such as SCHOONSCHIP, REDUCE and ASHMEDAI. This paper examines the origins of these programs and their early success. It also discusses the perspectives they raised in the 1970s, at a time when the birth of computational physics as an independent discipline led the community to produce a first collective assessment of these new methods. Finally, we will briefly review the current situation in this area to assess the extent to which expectations have been met.

Keywords: algebraic program, SCHOONSCHIP, REDUCE, ASHMEDAI, Feynman diagrams, Quantum field theory, High energy physics

"History is a touchy thing." The Conference on the History of Programming Languages, 1978

Amelie Mittlmeier (Ludwig-Maximilians-Universität Munich)

In 1978, a group of computer scientists organized a conference on the history of programming languages. The documents on the preparation and execution of the conference have survived comprehensively and provide insight into the conflicts that arose among the participants during the preparation of the event. For the computer professionals answered the question of how and by whom their own history could (or should) be researched and presented in rather different ways. The organizers' claim to present an accurate and comprehensive picture of the past was met with skepticism and sometimes harsh criticism. Fundamental questions about how to approach and deal with historical events were vigorously debated. The exchange between the organizers of the conference and 'professional' historians shows how even among the latter there was no unanimity in this respect - that instead the field of the history of science and technology was also in a phase of transformation. In this respect, the conference not only illuminates the early self-historicization of the still young discipline of computer science, but also provides insights into the incipient establishment of research on the History of Computing within the History of Science and Technology.

Keywords: History of Computer Science, self-historicization, History of Science and Technology

Empiricism and rationalism in language processing and AI research: On the nature of the machine

Amira Moeding (University of Cambridge)

I ask how epistemological virtues in Computational Linguistics (CL), artificial intelligence research (ai) and computer science could turn from explanation and transparency to quantifiable standards due to the rise of what actors in CL have called the 'empiricist paradigm' and how the resulting

changes influenced conceptions of ‘data,’ ‘intelligence,’ and ‘automation’ in ai research. The paper, thus, recovers the history of this ‘paradigm shift’ which occurred between 1987 and 1998 in computational linguistics, mainly in continuous speech recognition (csr) and machine translation (mt) and traces the consequences of this change for evaluative standards in ai research and, more broadly, computer science (Wilks 2009, 7- 14; Liberman 2009, 595). The intellectual shift introduced by the Continuous Speech Recognition Group at IBM has been examined before (Li 2017). Yet, the approach has not been contextualized within the broader debate about rationalist methods, i.e., encoding rules derived from theories about human language understanding, versus empiricist methods, i.e., automating stochastic prediction from language data. The paper illuminates how Noam Chomsky ended statistical or empiricist research in language processing during the late 1950s and how different actors at AT&T and IBM in the late 1980s were able to (re-)introduce a focus on data in ai research.

Keywords: Language Processing, artificial intelligence, empiricism, rationalism, Noam Chomsky, IBM, epistemic virtues, scientific representation, Big Data, DARPA

AI and art: Non-human, still human, or all too human?

Caterina Moruzzi (University of Konstanz) and Oreste Campagner (Università di Bologna)

AI models for the generation of content are disrupting traditional approaches to creative production, sparking debates about their impact on the role of the artist and on the nature and function of art. In this paper, we provide a systematisation of theoretical considerations that can be derived from the rapid developments of the field of AI-generated content. By situating these reflections within the context of existing debates on creativity, aesthetics, and the philosophy of art, we offer a systematic approach to thinking about the impact of AI on the nature and function of art and on the dynamics of interaction between artists, users, and machines. Our account indicates that anthropocentrism in art is challenged: in the creation of content with AI, the agency of the creative process is shared between humans and machines at an unprecedented level. But what is the role of the human component in the co-agency relationship? Are they artists or technologists? We argue that the introduction of AI in art challenges the modern distinction between ars and techné, and we suggest an expansion of the classical definition of the creator from an individual to a collective dimension that can simultaneously involve prompters, machines and programmers.

Keywords: Creativity, Art, Generative Artificial Intelligence, Artist

Formalising the notion of algorithm

Alberto Naibo (Université Paris 1 Panthéon-Sorbonne) and Thomas Seiller (CNRS)

It is well-know that the Church-Turing plays a crucial role in founding what is usually called a theory of computation, i.e. a theory dealing with effective computable functions. However, the Church-Turing thesis is not enough in order to ground a theory of algorithms, that is a theory where the main objects of studies are algorithms. The reason is that, despite some attempts, there

does not exist a satisfying and generally accepted formal definition of what an algorithm is. In this presentation we will introduce a new framework for thinking of a notion of algorithm and open the way of a formal characterisation of it. The proposed approach is based on a conceptual analysis allowing us to trace a precise mathematical distinction between three notions: computation, program, and algorithms. The mathematical tools we use, graphings, allow us to account both for infinite aspects and finite description of computation and programs – in between dynamical systems and labelled graphs. Using this formalism, we will give an example of formalisation of programs as well as the possible algorithms they implement.

Keywords: algorithms, philosophy of computer science, computability

Cheating (and cheating detection) in the advent of ChatGPT: How Artificial Intelligence is reframing academic production

Jo Ann Oravec (University of Wisconsin-Whitewater)

Cheating is a growing academic and ethical concern in higher education. The technological "arms race" that involves cheating-detection system developers versus technology-savvy students is attracting increased attention to cheating issues; it is also generating iterations of technological innovations as corporations, higher educational institutions, and legal experts attempt to control the situation. Artificial intelligence (AI) has added new dimensions to academic cheating challenges as students (as well as faculty and staff) can easily access powerful systems for generating content that can be presented in assignments, exams, or published papers as their own. AI methodology is also providing some emerging anti-cheating approaches, including facial recognition and watermarking. From an ethical and dramaturgical perspective, this paper provides an overview of human/AI collaboration approaches and frames some educational misuses of ChatGPT as forms of "misattributed co-authorship." As with other kinds of collaborations, the work that students produce with AI assistance can be presented in transparent and straightforward modes or (unfortunately) in opaquer and ethically-problematic ways. However, rather than just for catching or entrapping students, the emerging varieties of technological cheating-detection strategies can be used to assist students in learning how to document and attribute properly their AI-empowered as well as human-human collaborations.

Keywords: artificial intelligence, ChatGPT, ethical analysis, dramaturgical perspective, academic cheating, cheating detection, collaboration, co-authorship

Different conceptions of algorithms and the challenges of developing a definition

Philippos Papayannopoulos (IHPST (Université Paris 1 Panthéon-Sorbonne & CNRS))

I examine the classical idea of "algorithm" as a sequential, step-by-step, deterministic procedure (i.e., the idea of "algorithm" that was already in use by the 1930s), with respect to the following themes: (a) its relation to the notion of an "effective procedure", (b) its different roles and uses in

logic, computer science, and mathematics (focused on numerical computation), and (c) its different formal definitions that have been proposed by practitioners in these areas. I argue that the term "algorithm" is actually conceptualized and used in contrasting ways between the above areas. More specifically, I argue that inherent in the practices of logic, computer science, and mathematics is a tension between (a) a symbolic, representation-sensitive, and finitary conception of an "algorithm" and (b) an abstract, representation-independent, and model-relative conception. While the former conception identifies algorithms with effective procedures (in the sense of "effective" captured by Church's thesis), the latter conception (which goes far back in mathematical practice) accepts also algorithms that go beyond the Turing limit (e.g., they admit uncomputable primitive operations). After elaborating on this tension, I discuss the challenges and prospects for the community toward adopting a final foundational theory of (classical) "algorithms".

Keywords: Definitions of algorithms, Formalization of mathematical concepts, Effective procedures, Numerical algorithms, Algorithms vs computations

The rise and fall of extensible programming languages

Tomas Petricek (Charles University)

The term extensible programming language is not something that would be familiar to programmers and computer scientists today. Yet, in the1960s and 1970s, extensible programming languages were a household name. Two symposia on extensible languages were held in 1969 and 1971; accounts published in those report that there are over 50 extensible languages; Harvard caried out an Extensible Languages Research Program and "A Survey of Extensible Programming Languages" reviewing the work in the area was published in 1974. What were extensible programming languages, where did the term go and why? In this paper, I give a brief summary of the answer. Extensible programming languages was an umbrella term for languages with a wide range of extensibility mechanisms. All those mechanisms shared the same motivation – to let the language user adapt it to their particular needs – but they differed significantly in their technical realization, ranging from (what we would today call) syntactic macro mechanisms to new procedure and new datatype definitions. Work on all the techniques that were, in the late 1960s and early 1970s, falling under the term extensible programming languages continues to this day, but it is rarely, if ever, treated as belonging to the same research programme. Extensible programming languages provide an intriguing case of a short-lived research programme that brought distinct ideas together, but never managed to integrate them in a way that would ensure its longevity.

Keywords: History of Programming, Extensible Programming Languages, Macros

Information as constraint for neurocognitive mechanisms. Argument from the predictive processing framework

Michał Piekarski (Cardinal Stefan Wyszynski University in Warsaw)

Recently, there has been a discussion among theorists of mechanistic explanation about the necessity to include constraints and free energy flows into the explanations (Bechtel, 2018; Bechtel

& Bollhagen, 2021; Winning & Bechtel, 2018). This observation is directly related to the existence of control mechanisms that are non-autonomous and entail the existence of heterarchical networks. According to this approach a full explanation of how the brain works, i.e. what makes neural mechanisms active and able to perform their functions, requires taking into account information constraints which can be described in the terms of variational free energy (Winn & Bishop, 2005; Smith, Friston & White, 2022). In my presentation, I will consider how this idea can be applied into the framework of predictive processing (Clark, 2016; Hohwy, 2020). This framework provides a computational model of cognitive mechanisms in terms of hierarchical generative models which minimize prediction errors. I will argue that without reference to informational variational free energy, we would not be able to explain not only the homeostatic nature of neural computational mechanisms, but also their energy consumption, which is related to their ability to transmit information (cf. Laughlin, 2001).

Keywords: information, predictive processing, mechanisms, explanation, constraints, variational free energy, generative model, Bayesian inference

Computability in nested closed timelike curves

Marien Raat (Utrecht University)

Some solutions of general relativity include closed timelike curves (CTC's), which could be used for backwards time travel. Building on earlier research by Aaronson, Bavarian and Gueltrini, I consider how computability theory would change in an universe with arbitrarily curved spacetimes, where CTC's can branch off from other CTC's. I assume that the CTC's that show up in such a spacetime follow the consistency condition introduced by Deutsch. I introduce a computational model based on this spacetime structure. Then I show that the computational power of such a spacetime depends on how many levels deep the CTC's are nested within each other. For every CTC nested within each other, the Turing degree that is computable in the spacetime is increased by one Turing jump. Showing what is uncomputable in such spacetimes turns out to be harder. I show a flaw in the proof of uncomputability in a spacetime with a single CTC by Aaronson, Bavarian and Gueltrini. I then discuss ways in which the proof might be corrected. Lastly I will show how this proof would extend to the more general case of arbitrary curved spacetimes in which any amount of CTC's might be present.

Keywords: hypercomputation, closed timelike curves, time loops, ctc, computability, models of computation, computability theory

Does the No Miracles Argument apply to AI?

Darrell Rowbottom, Andre Curtis-Trudel and William Peden (all Lingnan University)

The no miracles argument for scientific realism rests on the notion that the approximate truth of science's theories best explains its persistent predictive success. In contemporary science, however, machine learning systems, such as AlphaFold2, have also been remarkably predictively

successful. We might therefore ask what best explains such successes. Might such AIs accurately represent critical aspects of their targets in the world? And if so, does a variant of the no miracles argument apply to these AIs? We argue for an affirmative answer to these questions. Our argument proceeds in two main steps. First, we reconstruct a version of a typical frequency-based no miracles argument (NMA). In the second step, we argue that if we relax certain assumptions made in traditional versions of the NMA, a parallel argument can be constructed for contemporary AI systems.The existence of an AI-specific NMA has certain important consequences for projects in explainable AI and attempts to use AI systems to arrive at scientific understanding. However, we do not argue that the standard NMA is sound. And so our result might not be a boon for scientific realism, because the AI-based NMA could be a reductio ad absurdum of the standard NMA.

Keywords: Artificial Intelligence, Scientific Realism, No miracles, deep learning

Personal computing paradigm – 50 years after

Zbigniew Stachniak (York University)

On September 25, 1973, a Toronto-based company Micro Computer Machines (or MCM) unveiled its small portable microprocessor-powered computer designed specifically for personal use – the MCM/70. With the launch of the MCM/70, the company was offering a new personal computing paradigm – these were inexpensive, easy-to-operate computers, such as the MCM/70, that were to make widespread, personal information processing a reality. Initially, MCM maintained that a PC operated in a powerful and user-friendly operating environment would bridge the gap between the roles of the computer owner, end-user, operator, and applications software developer – it would fuse all these roles into a new owner-as-programmer paradigm. This standpoint shifted in time toward a personal computer owner as a software consumer operating a PC in a user-friendly and networked environment – a prelude to our current information- and Internet-centric culture. In this talk I will present the MCM's evolving and shifting point of view on personal computing and examine the paradigm's universal as well as period-dependent features.

Keywords: PC, personal computing, software

Physical programmability

Nick Wiggershaus (Lille University)

The philosophical discourse regarding programmability is largely underdeveloped. Reviewing the literature uncovers that only a limited amount of scholarship has examined the physical properties that enable a system to be programmed. This is a sorry condition, for we seem to be unable to fully answer such questions as: How are programs integrated into the causal nexus? What does it mean for a physical system to be programmable? In the interest of answering these questions, I develop the here newly introduced notion of physical programmability, viz. 'The degree to which the selected operations of an automaton can be reconfigured in a controlled way.' Subsequently, I explain the significance of the variables in above's characterization. Accordingly, the function of (i) automaton; (ii) selected operation; (iii) reconfigured in a controlled way (iv) the degree to which, are

discussed in detail. The result is fourfold: One, the domain of systems that can be bestowed with the property of being physically programmable is limited to 'material automata'. Two, the selected operation of these material automata is explained best through the neo-mechanistic framework. While my original focus was physical computation, one may also 'plug in' different operations. This way, one may apply the notion of physical programmability to music boxes without having to subscribe to the idea that they compute. Three, I expanded the understanding of 'reconfiguration in a controlled way' by establishing a connection between mechanisms and manipulability theory. Fourth, by discussing various examples of automata, I show that physical programmability is a gradual notion and comes in different degrees.

Keywords: Programmability, Interventionism, Automata, Physical Computation