Gödel and Computability

Luís Moniz Pereira

Centro de Inteligência Artificial – CENTRIA
Universidade Nova de Lisboa, Portugal

Abstract

- We discuss the influence of Gödel’s results on the surfacing of the rigorous notion of computability set forth by Turing.

- We debate the limits of Artificial Intelligence spurned by Roger Penrose on the basis of Gödel's theorems, and the views of Gödel himself.

- We touch upon the use of logic as a tool with which to approach the description of mind.

Hilbert's Questions - 1

- The research programme Hilbert presented in 1928 to the international congress of mathematics in Bologna was essentially an extension of the work he had initiated in the 1890s.
- He raised profound and hard questions *about* systems of the kind Russell had produced. Was there a way to find out what could and could not be demonstrated from within such a system?
- Hilbert's approach was *formalist*, for he treated mathematics like a game, a matter of form. The allowed steps in a demonstration are seen the same way as the possible moves in a game, with the axioms being the initial state of the game.

Hilbert's Questions - 2

He made his questions very precise:

- **First, was mathematics *complete* ?**

  I.e. whether all mathematical statements (cf. “all integers are the sum of four square numbers”) could be either demonstrated or falsified.

- **Second, was mathematics *consistent* ?**

  I.e. whether any contradictory statements (cf. ‘2+2=5’) can be demonstrated by correct application of valid steps of derivation rules.

- **Third, was mathematics *decidable* ?**

  Can one ascertain if, in principle, there is a well-defined method which, when applied to any assertion, can be guaranteed to produce a correct decision about its truth.

Gödel’s Answers - 1

- Kurt Gödel was able to show that every formalization of arithmetic must be *incomplete*: that there are assertions that cannot be demonstrated nor rebutted.
- Starting from Peano’s axioms for the integer numbers, he extended them to a simple type theory, in such a way that the obtained system could represent sets of integers, sets of sets of integers, and so forth.
- However, his argument could be applied to any formal system sufficiently rich to include number theory, and the details of the axioms were not crucial.

Gödel’s Answers - 2

- Then he showed that all operations in a ‘proof’, the game-like deduction rules, are representable in arithmetic. This means such rules are constituted only by operations like counting and comparing, in order to verify if an expression (itself represented arithmetically) had been correctly replaced by another.
- Gödel showed the formulas of his system could be codified as integers, to represent assertions ***about*** integers.
- Next Gödel showed how to codify demonstrations as integers, so that he had a whole theory of arithmetic, codified ***from within*** arithmetic. He showed the property of ‘being a demonstration’ or being ‘provable’ was as much arithmetical as that of ‘being square’ or ‘being prime’.

Gödel’s Answers - 3

- The result of this codification process was that it became possible to write arithmetical assertions that referred to *themselves*, just like when a person says “I’m lying”.
- In fact, Gödel constructed a specific assertion that had exactly that property, as it effectively said “This assertion is not provable.”
- It followed that it could not be shown *true*, as that would lead to a contradiction. Neither could it be shown *false*, by the same reason.
- It was thus an assertion that could neither be proved nor refuted by logical deduction from the axioms. Gödel had just showed that arithmetic was *incomplete*.

Gödel's Answers - 4

- There is something special about that particular assertion of Gödel, more precisely that, considering it is not provable, it is in some sense *true*. But to assert that it is *true* requires an observer that can look at the system from the outside. It cannot be shown *from within* the axiomatic system.
- An important issue is that this argument assumes arithmetic is *consistent*. In fact, if arithmetic were inconsistent then *any* assertion would be provable, since in first order logic everything follows from contradiction.
- Gödel showed that arithmetic, once formalized, had to be either *inconsistent* or *incomplete*.

Gödel, computability, and Turing - 1

- Hilbert's third question, that of *decidability*, still remained open, although now it had to be formulated in terms of *provability*, instead of *truth*.
- Gödel's results did not eliminate the possibility that there could be some way to distinguish the provable from the non-provable assertions. That the peculiar self-referencing assertions of Gödel might in some way be separated from the rest.
- Could there be a well-defined *method* (i.e. a *mechanizable* procedure), applicable to any mathematical statement, that could decide if that statement was, or was not, derivable in a given formal system?

Gödel, computability, and Turing - 2

- It was not clear this problem would admit a definitive answer. Gödel was probably surprised by Turing’s solution, more elegant and conclusive than he had expected.
- Gödel fully understands, at the beginning of the ‘30s, that the concept of formal system is intimately tied up with that of mechanizable procedure. He considers Turing’s 1936 work on computable numbers as an important complement of his own work on the limits of formalization.
- Over the years, Gödel regularly credited Turing’s article as the definitive work that captures the intuitive concept of computability, the only author to present persuasive arguments about the adequacy of the precise concept he himself defined.

Gödel, computability, and Turing - 3

- Regarding the concept of mechanizable procedure, Gödel's incompleteness theorems naturally begged for an exact definition (as Turing would come to produce) by which one could say that they applied to every formal system, i.e. every system on which proofs could be verified by means of an automatic procedure (not just Russell's).
- In reality, Hilbert's programme included the *Entscheidungsproblem* (literally, 'decision problem'), which aimed to determine if there was a procedure to decide if, in elementary logic, any proposition was derivable or not by Frege's rules for first-order logic.
- This requires a precise concept of automatic procedure, in case the answer is negative (as is the case).

Gödel, computability, and Turing - 4

- To this end, in 1934 Gödel introduces the concept of general recursive functions, which was later shown to capture the intuitive concept of mechanizable computability. Gödel suggested the concept and Kleene worked on it.
- This concept, once formalized and somewhat extended, gave rise to the definition of ‘recursive function’. It was later verified that these were exactly equivalent to the computable functions of Turing.
- Once the concept, as defined by Turing, is accepted as the correct one, a simple step suffices to see that, not only Gödel’s incompleteness theorems apply to formal systems in general, but also to show that the *decision problem* is insoluble, as proved by Turing himself.

Gödel, computability, and Turing - 5

- In his Ph.D. thesis, Turing tries finding a way out from the power of Gödel's incompleteness theorem. The fundamental idea was that of adding to the initial system successive axioms. Each 'true but not demonstrable' assertion is added as a new axiom.
- However, in this way, arithmetic acquires the nature of a Hydra, because, once the new axiom is added, a new assertion of that type will be produced that takes it now into consideration. It is then not enough to add a <u>*finite*</u> number of axioms, but it is necessary to add an infinite number, which was clearly against Hilbert's finitary dream.
- If it were possible to produce a finite generator of such axioms, then the initial theory would also be finite and, as such, subject to Gödel's theorem.

Gödel, computability, and Turing - 6

- This work, nevertheless, had a pleasantly persistent side effect, namely the introduction of the concept of '<u>oracle</u> Turing machine', precisely so it could be allowed to ask and obtain from the exterior the answer to an insoluble problem from within it (e.g. identifying a Gödel assertion). It introduced the notion of relative computability, or relative insolvability, opening a new domain in mathematical logic and in computer science.
- The connection, made by S.A. Cook, in 1971, between Turing machines and the propositional calculus would give rise to the study of central questions about computational complexity.

Mens ex-machina - 1

- Gödel's incompleteness theorem demonstrates the rigidness of mathematics and the limitations of formal systems and, according to some, of computer programs.
- It relates to the issue of whether mind surpasses machine. Thus, the growing interest given to computers and Artificial Intelligence (AI) has lead to a general increase in interest about Gödel's work.
- But, so Gödel himself recognizes, his theorem does not settle the issue of knowing if mind surpasses machine. Actually, Gödel's work seems to favour (instead of countering) the <u>*mechanist*</u> position (and even <u>*finitism*</u>) as an approach to the automation of formal systems.

Mens ex-machina - 2

- Gödel contrasts *insight* with *proof*. A proof can be explicit and conclusive for it has the support of axioms and of rules of inference. In contrast, insights can be communicated only via "pointing" at things.
- Any philosophy expressed by an exact theory can be seen a special case of the application of Gödel's conceptual realism. Its objective should be to give a clear perspective of all the basic metaphysical concepts.
- Gödel claims this task consists in determining, *through intuition*, the primitive metaphysical concepts ***C*** and in making correspond to them a set of axioms ***A*** (so that only ***C*** satisfies ***A***, and the elements in ***A*** are implied by the original intuition of ***C***). He further admits that, from time to time, it would be possible to add new axioms.

Mens ex-machina - 3

- Gödel also advocates an ‘optimistic rationalism’. He appeals to (1): “The fact that those parts of mathematics which have been systematically and completely developed … show an astonishing degree of beauty and perfection.”
- So (2): It is <u>*not*</u> the case “that human reason is irredeemably irrational by asking itself questions to which it cannot answer, and emphatically asserting that only reason can answer them.” (3) follows: There are no “undecidable questions of number theory for the human mind.” So (4): “The human mind surpasses all machines.”
- However, the inference from (1) to (2) seems to be obtained from accidental successes in very limited fields to justify an anticipation of universal success. Besides, both (2) and (3) concern only a specific and delimited part of mind and reason which refer just to mathematical issues.

Mens ex-machina - 4

Gödel understands that his incompleteness theorem by itself does not imply the human mind surpasses all machines. An additional premise is necessary. He gives 3 suggestions:

a) It is sufficient to accept his ‘optimistic realism’.

b) Appealing “to the fact that *the mind, and the way it’s used, is not static, but finds itself in constant development,*” “there is no reason to claim that” the number of mental states “cannot converge to infinity in the course of its development.”

c) Believing there is a mind separate from matter, and that such will be demonstrated “scientifically (maybe by the fact that there are insufficient nerve cells to account for all the observable mental operations).”

Mens ex-machina - 5

- There is a known ambiguity between the notion of *mechanism* confined to the mechanizable (in the sense of computable or recursive) and the notion of *materialist mechanism*. Gödel enounces two propositions:

  (i) The brain operates basically like a digital computer.

  (ii) The laws of physics, in their observable consequences, have a finite limit of precision.

- He is holds that (i) is very likely, and that (ii) is practically certain. Perhaps the interpretation Gödel assigns to (ii) is what makes it compatible with the existence of non-mechanical physical laws, and in the same breath he links it to (i) in the sense that, as much as we can observe of the brain's behaviour, it functions like a digital computer.

Is mathematical insight algorithmic?

- R. Penrose (1994) claims that it is *not*, and supports much of his argument, as J. R. Lucas before him, on Gödel’s incompleteness theorem: It is *insight* that allows us to *see* that a Gödel assertion, undecidable in a given formal system, is accordingly true. How could this intuition be the result of an algorithm?

- Penrose insists that his argument would have been “certainly considered by Gödel himself in the 1930s and was never properly refuted since then …”

Is mathematical insight algorithmic?

In his Gibbs lecture in 1951, delivered to the *American Mathematical Society,* Gödel openly contradicts Penrose:

“On the other hand, on the basis of what has been proven so far, it remains possible that a theorem proving machine, indeed equivalent to mathematical insight, can exist (and even be empirically discovered), although that cannot be *proven*, nor even proven that it only obtains correct theorems of the finitary number theory.”

Is mathematical insight algorithmic?

- In the 1930's Gödel was especially careful in avoiding controversial statements, limiting himself to what could be proven. His Gibbs lecture was a veritable surprise. Gödel insistently argued that his theorem had important philosophical implications. But, as the above citation makes clear, he never stated that mathematical insight could be shown to be non-algorithmic.
- It is likely Gödel would agree with Penrose that mathematical insight could not be the product of an algorithm. In fact, Gödel apparently believed that the human mind could not even be the product of natural evolution. But Gödel never claimed such conclusions were a consequence of his theorem.

Is mathematical insight algorithmic?

- In this stance, AI would be primarily interested in what is feasible from the viewpoint of computability, whose formal concern involves only a very limited part of mathematics and logic.
- However, the study of the limitations of AI cannot be reduced to this restriction in its scope.
- In this regard, it is essential to distinguish between *algorithms for problem-solving*, and *algorithms simpliciter*, as sets of rules to follow in a systematic and automatic way, eventually self-modifiable, and *without* necessarily having a specific and well-defined problem to solve.

Logic consciousness - 1

- If we ask “Can we introduce the unconscious in computers?” some will answer computers are totally unconscious. In truth, we don’t know how to introduce consciousness in algorithms, because we use computers as unconscious appendices to our consciousness.
- The question is premature, since we can only refer to the human conception of unconscious after introducing consciousness into the machine. We understand much better the computational unconscious than our own.
- These questions point to the complexity of thought processes, including those of creativity and intuition (which in great measure we don’t understand), and pose a much richer challenge to AI (that helps by providing an indispensable symbiotic mirror).

Logic consciousness - 2

- Translation into a computational construction of some functional model, of an introspective and thus self-referent consciousness, would be permitted using whatever methodologies and programming paradigms currently at our disposal.
- Given this realization, one might be inclined to ask why the use a logic paradigm, via logic programming, which is precisely the one we prefer (e.g. Lopes and Pereira (2006) + ongoing work).
- There are several arguments which can be raised against its use. Hence we try to reproduce here the most relevant, rebut them, and present our own in its favour.

Logic consciousness - 3

- The first argument to be raised in these discussions is that regular human reasoning does not use logic, there existing complex, non-symbolic processes in the brain that supposedly cannot be emulated by symbolic processing.
- On this line of thought, many models have been produced based on artificial neural networks, on emergent properties of purely reactive systems, and many others, in an attempt to escape the tyranny of GOFAI ('Good Old Fashioned AI').
- There is a catch, however, to these models: Their implementation by its proponents ends up, with no particular qualms, being made on a computer, thus using symbolic processing to simulate these other paradigms.

Logic consciousness - 4

- The relationship of this argument to logic is ensured by the philosophical stance of functionalism: logic itself can be implemented on top of a symbol processing system, independently of the particular physical substrate supporting it.
- Once a process is described in logic, we can use its description to synthesize an artefact with those properties. So long it is a computational model, any attempt to escape logic will not prove itself to be inherently more powerful.
- Moreover, there is an obvious human ability to understand logical reasoning, one developed during the course of brain evolution. Its most powerful expression is science itself, and the knowledge amassed from numerous disciplines, each of which with their own logic nuances dedicated to reasoning in their domain.

Logic consciousness - 5

- Humans can use language without learning grammar. But if we are to understand linguistics, knowing the logic of grammar, syntax and semantics is vital. Humans do use grammar without explicit knowledge of it, but that does not mean it cannot be described.
- Talking about the movement of electrons, we do not mean a particular electron knows the laws it follows, but we are certainly using symbolic language to describe the process. It is even possible to use the description to implement a model and a simulation which exhibits the same behaviour.
- Likewise, even if human consciousness does not operate directly on logic, it does not mean we won't be forced to use logic to provide a rigorous *description* of that process.

Logic consciousness - 6

- Once obtained a sufficiently rigorous description of the system of consciousness, we are supposedly in possession of all *our* current (temporary) knowledge of that system, reduced to connections between minimal black boxes, inside which we know not yet how to find other essential mechanisms.
- No one has managed to adequately divide the black box of our consciousness about consciousness in the brain, but maybe we can provide a sufficiently rigorous description for it, that models a functional system its equivalent.
- When a division of that epistemic cerebral black box into a diversity of others is achieved later on, we are sure to be able, to describe new computational models equivalent to the inherent functional model.

Logic programming - 1

- In struggle for rigorous description, AI has made viable the proposition of turning logic into a programming language.
- Logic can presently be used as a specification language which is not only executable, but on top of which we can demonstrate properties and proofs of correction that validate the descriptions produced.
- Facing the challenge, AI developed logic beyond the confines of monotonic cumulativity, far removed from the artificial paradises of the well-defined, and well into the real world purview of incomplete, contradictory, arguable, reviseable, distributed and updatable, demanding, among other, the study and development of non-monotonic logics, and their computer implementation.

Logic programming - 2

- Over the years, enormous amount of work has been carried out on individual topics, such as logic programming language semantics, belief revision, preferences, evolving programs with updates, and many other issues that are crucial for a computational architecture of the mind.
- We have a state-of-the-art from whence we can begin addressing the more general issues with the tools already at hand, unifying such efforts into powerful implementations exhibiting promising new computational properties.
- Computational logic has shown itself capable to evolve to meet the demands of the difficult descriptions it is trying to address.

Logic and Cognition

- The use of the logic paradigm also allows us to present the discussion of our system at a sufficiently high level of abstraction and generality for productive interdisciplinary discussions, both on its specification and derived properties.
- The language of logic is universally used both by the natural sciences and the humanities, and more generally is at the core of any source of human derived common knowledge, so that it provides us with a common ground on which to reason about our theories.
- As the field of cognitive science is essentially a joint effort on the part of several distinct knowledge fields, we believe such language and vocabulary unification efforts are not just useful, but mandatory.

Reference (cf. home page):

L. M. Pereira, **Gödel e a Computabilidade**
Invited paper, special issue commemorative of Kurt Gödel's birth centenary, ***Boletim da Sociedade Portuguesa de Matemática***, nr.55:77-90, October 2006. **English version here**

The End

[thank you for your attention]