Sep 27, 2021

Computation & Complexity

Decidability

Q: What is decidability? And how can it be explained in Laypersons terms? Can it be illustrated with a diagram? What type of problems are undecidable? How is the Game of chess undecidable

Group #2

DOYLE, SAOIRSE ANTHONY

CHENG, CHEN

LIU, QING

Structure

A. Introduction – DOYLE, SAOIRSE ANTHONY
B. Historical Background – CHENG, CHEN
C. Basic Theory – CHENG, CHEN
D. Lay Persons Account – DOYLE, SAOIRSE ANTHONY
E. Practical Implication – LIU, QING
F. Recent Development – LIU, QING
G. Conclusion – All
H. Next Topics – LIU, QING
I. References – All
J. Questions – All
K. PowerPoint – CHENG, CHEN
L. Speech – DOYLE, SAOIRSE ANTHONY

Introduction

Good Afternoon, In this presentation we will be discussing the topic of decidability in Computer Science. We will explore what decidability means for Computer Science, what kind of problems are decidable or undecidable, and the wider implications for logic. We will use the game of chess as an example of an undecidable problem.

Historical Background

In term of the origin of the decidability problem, It was David Hilbert, a German mathematician and one of the most influential mathematicians of the 19th and early 20th centuries, with who one should associate the beginning of research on the decidability – He drew attention of mathematicians to this problem and made it into a central problem of mathematical logic. He called it was Entscheidungsproblem (what literally means: “the decision problem”).

It appeared in a sense already in his famous lecture at the Congress of Mathematicians in Paris in August 1900.The origin of the Entscheidungsproblem goes back to Gottfried Leibniz, who was a German polymath active as a mathematician, philosopher, scientist, and diplomat in the seventeenth century.

In 1928, David Hilbert and Wilhelm Ackermann posed the Entscheidungsproblem as a challenge. Remarkably, David Hilbert was the Doctoral advisor of Wilhelm Ackermann who was born in Germany and was awarded a Ph.D. by the University of Göttingen in 1925. Although David Hilbert cooperated with his student for a long time, Hilbert's support vanished when Ackermann got married.

Basic Theory

Decision theory is the study of an agent's choices. Turing Machine that was invented by Alan Turing in 1936 and it is used to accept Recursive Enumerable Languages (generated by Type-0 Grammar) is the prerequisite of Decision theory.

And a decision problem is a yes-or-no question on an infinite set of inputs. In logic, a true/false decision problem is decidable if there exists an effective method for deriving the correct answer. In short, a problem is said to be Decidable if we can always construct a corresponding algorithm that can answer the problem correctly.

We can intuitively understand Decidable problems by considering a simple example. Suppose we are asked to compute all the prime numbers in the range of 1000 to 2000. To find the solution of this problem, we can easily devise an algorithm that can enumerate all the prime numbers in this range.

Regarding Decidability in terms of a Turing machine, a problem is said to be a Decidable problem if there exists a corresponding Turing machine which halts on every input with an answer- yes or no. It is also important to know that these problems are termed as Turing Decidable since a Turing machine always halts on every input, accepting or rejecting it.

Lay Persons Account

In simple terms, Decidability is the question of whether an algorithm can be derived, such that the correct answer is always produced, even for different inputs. (cs.dartmouth.edu ,2006)

For example, in a simple game of Tic Tac Toe, there are only 9 places on the board, that either player can choose. Based off where the previous player chooses to move, there will always exist an optimal choice of move for an AI to make which will, every single time, lead to a win for the AI or at the very least, a draw. (Zero Sum Games, Stanford.edu)

However, in a game such as Scrabble, no such perfect algorithm exists, as there is an element of luck/randomness involved in the game. Even a very well made AI, or an extremely intelligent player, is not guaranteed to win each time a game is played, as the maximum score each player is capable of receiving, is directly limited by which letters are chosen at random which makes it impossible to guarantee a specific player will win or even draw the game each time. (Rigney, 2012)

Chess, while decidable in theory is for practical purposes, an undecidable game. Much like tic tac toe, each player starts off with the same number of pieces, and the same range of moves possible, so in theory, there is a perfect way to play chess which is guaranteed to win each time.

However, in practice this is impossible, as the branching factor of chess is much too large. In tic tac toe, the number of possible moves, is small enough that all possible permutations can be calculated with ease. In the case of chess, each possible move, leads to on average 35 other possible moves. So in the first turn each player has 35 possible moves, but each of those 35 moves lead to on average 35 more moves meaning for the second turn there are 35x35 =1225 possible moves and 42875 possible positions in the third move. The permutations quickly become much too large for even modern computers to calculate in reasonable time.

This difference comes about from how much information each player has about the other. In tic tac toe or chess, each player starts off in exactly the same position, and has the same set of possible moves they can make. However, in Scrabble for example, there is a lack of information caused by the randomness. Player 1 has no knowledge of which letters Player 2 has and vice versa.

This topic has much wider ranging applications than games. The existence of a set of problems for which no perfect algorithm exists has huge implications. This means that even with infinite computing power, and infinite run time, there are some problems we will never be able to obtain a perfect solution for.

Practical Implications

According to the paper found, the decidability theory is theoretically applied to identify/decide if two different computer networks are similar (in terms of efficiency, high performance computing and scalability) or not. A finite automata is designed for each network and two different scenarios are considered for demonstration.

The results show that the theory can be effectively used to make such comparisons between different computer networks. In simple terms, if there is a shopping bag, decidability is that one is able to check whether or not there is some salad in the bag.

Recent Developments

Recent decades have seen a remarkable development in the use of computer systems and computer networks. With their advances our reliance on hardware and software has amplified, and so has our susceptibility to their malfunction.

For fear of any major network failure, it is at all times good to recognize similar networks so that switching can be done to reduce damages. Theoretical computer science aims to model and understand the intricacy of computer systems, and thereby creates the basis for their formal verification: to mathematically prove that a system satisfies its requirement. So this is where Decidability can make a difference.

With the help of Decidability, authors can focus on the decision whether the given two different computer networks are similar or not, that is, authors check if the two networks are equivalent (equal) in terms of speed of communication, efficiency, high performance computing, and scalability, or not.

Conclusion

As we have shown, the problem of decidability has wide ranging consequences for computer science, maths and logic.

For a long time it was assumed that any problem could be solved given enough time and the right algorithm, but the question of decidability raises the possibility that there will be some problems we may never have a solution for. Is there a limit to the technological advances we can make? Or the scientific discoveries we can find? We will have to wait and see.

Next Topics

What is a recognisable languages and how is it connected to Turning Machines? How do we show that a language is recognisable? Is there a Layperson's explanation or example to illustrate this property?

References

1. "David Hilbert - Wikipedia." Accessed September 25, 2021. https://en.wikipedia.org/wiki/David_Hilbert.
2. "Entscheidungsproblem - Wikipedia." Accessed September 25, 2021. https://en.wikipedia.org/wiki/Entscheidungsproblem.
3. "Decidability vs. undecidability. Logico-philosophico-historical remarks." Accessed September 25, 2021. https://core.ac.uk/download/pdf/235272271.pdf.
4. "Gottfried Wilhelm Leibniz - Wikipedia." Accessed September 25, 2021. https://en.wikipedia.org/wiki/Gottfried_Wilhelm_Leibniz.
5. "Wilhelm Ackermann - Wikipedia." Accessed September 25, 2021. https://en.wikipedia.org/wiki/Wilhelm_Ackermann.
6. "Decision theory - Wikipedia." Accessed September 25, 2021. https://en.wikipedia.org/wiki/Decision_theory.
7. Dartmouth College “CS 4, Summer 2006: Lecture 22: Decidability” [online] Available at https://www.cs.dartmouth.edu/~cbk/classes/4/06x/notes/22-decidability.html [Accessed 25 September 2021]

8. Stanford University “Zero-Sum Games” [online] Available at: https://cs.stanford.edu/people/eroberts/courses/soco/projects/1998-99/game-theory/zero.html [Accessed 25 September 2021]
9. Rigney R. “You May Win Every Time, But You Haven't Solved This Game Yet” [online] Available at https://www.wired.com/2012/11/letterpress-solved-games/ [Accessed 25 September 2021]
10. "The Application of Decidability Theory to Identify Similar Computer Networks" https://www.researchgate.net/publication/256017856. [Accessed 25 September 2021]
11. "Decidable and Undecidable problems in Theory of Computation." 20 Nov. 2019, https://www.geeksforgeeks.org/decidable-and-undecidable-problems-in-theory-of-computation/. [Accessed 25 September 2021]
12. "Turing Machine in TOC - GeeksforGeeks." 5 Jul. 2021, https://www.geeksforgeeks.org/turing-machine-in-toc/. [Accessed 25 September 2021]