中国科学院大学
University of Chinese Academy of Sciences

CS101

Logic Thinking

Power and Limit of Computing

zxu@ict.ac.cn
zhangjialin@ict.ac.cn

Outline

- Foundation of logic
  - Propositional Logic
  - Predicative Logic
- Automata and Turing Machines
- Power and Limitation of Computing
  - Mechanical Theorem Proving
  - Church-Turing Hypothesis

1.4 POWER AND LIMIT OF COMPUTING

What does “computable” mean?

- The computable numbers problem
  - For any real number, is the number computable?
    - Here, “computable” means there exists a machine that can automatically generate the decimals of the number, up to any length (any desired number of decimal digits)
  - E.g., Alan Turing showed that the circular constant π is computable
    - Textbook contains a program to generate the first 800 decimal digits of π:
      - 31415926535897932384626433832795028841971693993751058209749445923052187816406286208998628034825342117067982148086513282306647093844609550582231725359408128481117450284102701938521105559644622948954930381964428810975665933446128475648233786783165271201909145648566923468758034861045432664821339360726024914127372458700660631558817488152092379709628292540917153643678925903600113305305488204665213841469519415170641609433057270365759591953092186117381932611793105118548074462379962422327495673518857527248912279381830119491298336733624406566430860213903734946395224737190702179860943702770539217176293176752384674818467660919405132000568127145263560827785771342757789609173637178721468440906122495343014654958537105079227968925892354201995611212902196086403441815981362977477130996051870721134999998372978049951059731732816096318 5

Turing machine is powerful

- A problem is Turing computable, if there is a Turing machine that correctly solves the problem

- Church-Turing Thesis: Assume a reasonable abstract computer X is given. Any problem computable in X is also Turing computable.

Computable = Turing computable

Turing machine is not omnipowerful

- The halting problem
  - Given the description of an arbitrary Turing machine M and an input string x, decide whether M will terminate or run forever.

- The halting problem is not Turing computable!

Halting problem

	1	2	3	4	5	6	7	8	9	…
1	0	0	0	0	0	0	0	0	0	…
2	1	1	1	1	1	1	1	1	1	…
3	0	1	0	0	0	0	0	0	0	…
4	0	0	1	1	1	1	1	1	0	…
5	1	0	0	0	0	0	0	0	0	…
6	1	1	1	1	1	1	1	1	1	…
7	1	1	0	0	1	1	1	1	1	…
…	.	.	.	.	.	.	.	.	.	.

Turing machine H can solve Halting problem:
H can compute this table

- i-th row, j-th column:
  - 1: Turing machine M_i will terminate on input j
  - 0: Turing machine M_i will not terminate on input j

Halting problem (2)

- Define Turing machine G
  - For any input i
  - if $H(i,i) = 0$, then halt

    else run forever

- Consider $G(G)$
  - If $G(G)$ terminates, it means $H(G,G) = 0$. Thus, G should halt on input G, contradiction!
  - If $G(G)$ does not terminate, similar argument

Three properties of an axiom system

- Completeness: every true statement can be proven
  - For every true statement, there exists a proof in this axiom system
- Consistency: there is no contradiction in the system
  - There is no statement that we can prove to be true and to be false in the same system
- Independence: no axiom can be derived from other axioms in the system
  - Not always needed

- Not all axiom systems have the three properties
  - Not all axiom systems have the first two properties!

Propositional logic

- "Be true" and "can be proved" are the same

- An axiom system for propositional logic

$$A \to (B \to A)$$

$$(A \to (B \to C)) \to ((A \to B) \to (A \to C))$$

$$((\neg A) \to (\neg B)) \to (B \to A)$$

Euclidean geometry

- Axiom system:
  - To draw a straight line from any point to any point.
  - To produce (extend) a finite straight line continuously in a straight line.
  - To describe a circle with any center and distance (radius).
  - That all right angles are equal to one another.
  - [The parallel postulate]: That, if a straight line falling on two straight lines make the interior angles on the same side less than two right angles, the two straight lines, if produced indefinitely, meet on that side on which the angles are less than two right angles.

- "Be true" and "can be proved" are the same

Gödel's Incompleteness Theorems

- Gödel's first incompleteness theorem:
  Any mathematical system that includes elementary number theory (natural numbers, addition, and multiplication) cannot have completeness and consistency at the same time.
- Gödel's second incompleteness theorem:
  For any mathematical system that includes elementary number theory, if it is consistent, then its consistency cannot be proved within itself.

"Be true" and "can be proved" are not the same!

Kurt Godel
1906-1978

Gödel's Incompleteness Theorems

- Consistency VS. Completeness

- "True" and "can be proved" are not the same!
  - "Can be proved" is "true"
  - "True" may not "can be proved"
- Given an axiom system, can we find an algorithm which can decide whether each statement is true or false?
  - NO
  - A mechanized proof system is impossible!
  - The power of computation is limited!

Incomputability is both negative and positive

- People have found ways to exploit such negative results for positive benefits
- Examples of ideas:
  - Incomputable problems provide opportunities for human intelligence
  - Computationally hard problems can be used to design computer and Internet games
  - If a privacy protection technique can be formulated as incomputable problems, one cannot use computers alone to break privacy protection

  - Incomputability results by Kurt Gödel and Alan Turing provide a foundational piece for future technology systems

Life after Google: The Fall of Big Data and the Rise of the Blockchain Economy

George Gilder