Turing Machines

Lecture 8
Thursday, September 17, 2020

LaTeXed: July 30, 2020 14:58

8.1
In the search for thinking machines

"Most General" computer?

1. DFAs are simple model of computation.
2. Accept only the regular languages.
3. Is there a kind of computer that can accept any language, or compute any function?
4. Recall counting argument. Set of all languages:
   $\{L \mid L \subseteq \{0,1\}^*\}$ is ~~countably infinite~~ / uncountably infinite
5. Set of all programs:
   $\{P \mid P$ is a finite length computer program$\}$:
   is countably infinite / ~~uncountably infinite~~.
6. **Conclusion:** There are languages for which there are no programs.

"Most General" computer?

1. DFAs are simple model of computation.
2. Accept only the regular languages.
3. Is there a kind of computer that can accept any language, or compute any function?
4. Recall counting argument. Set of all languages:
   $\{L \mid L \subseteq \{0,1\}^*\}$ is ~~countably infinite~~ / uncountably infinite
5. Set of all programs:
   $\{P \mid P$ is a finite length computer program$\}$:
   is countably infinite / ~~uncountably infinite~~.
6. **Conclusion:** There are languages for which there are no programs.

"Most General" computer?

1. DFAs are simple model of computation.
2. Accept only the regular languages.
3. Is there a kind of computer that can accept any language, or compute any function?
4. Recall counting argument. Set of all languages:
   $\{L \mid L \subseteq \{0,1\}^*\}$ is ~~countably infinite~~ / uncountably infinite
5. Set of all programs:
   $\{P \mid P$ is a finite length computer program$\}$:
   is countably infinite / ~~uncountably infinite~~.
6. **Conclusion:** There are languages for which there are no programs.

What can be computed?

Most General Computer:

1. If not all functions are computable, which are?
2. Is there a "most general" model of computer?
3. What languages can they recognize?

History: Formalizing mathematics

1. 19th century: Ooops. Math is a mess. Oy.
   Fix calculus, invented set theory (Cantor), etc.
2. David Hilbert (1862–1943)
   1. 1900: The list of 23 problems.
   2. Early 1900s – crisis in math foundations
      attempts to formalize resulted in paradoxes, etc.
   3. 1920: Hilbert's Program: "mechanize" mathematics.
   4. Finite axioms, inference rules turn crank, determine truth needed: axioms consistent & complete
   5. Hilbert: "No one shall expel us from the paradise that Cantor has created.".
3. Kurt Gödel (1906–1978)
   German logician, at age 25 (1931) proved: "There are true statements that can't be proved or disproved". (i.e., "no" to Hilbert)
   Shook the foundations of mathematics/philosophy/science/everything.

History: Formalizing mathematics

1. 19th century: Ooops. Math is a mess. Oy.
   Fix calculus, invented set theory (Cantor), etc.
2. David Hilbert (1862–1943)
   1. 1900: The list of 23 problems.
   2. Early 1900s – crisis in math foundations
      attempts to formalize resulted in paradoxes, etc.
   3. 1920: Hilbert's Program: "mechanize" mathematics.
   4. Finite axioms, inference rules turn crank, determine truth needed: axioms consistent & complete
   5. Hilbert: "No one shall expel us from the paradise that Cantor has created.".
3. Kurt Gödel (1906–1978)
   German logician, at age 25 (1931) proved: "There are true statements that can't be proved or disproved". (i.e., "no" to Hilbert)
   Shook the foundations of mathematics/philosophy/science/everything.

History: Formalizing mathematics

1. 19th century: Ooops. Math is a mess. Oy.
   Fix calculus, invented set theory (Cantor), etc.
2. David Hilbert (1862–1943)
   1. 1900: The list of 23 problems.
   2. Early 1900s – crisis in math foundations
      attempts to formalize resulted in paradoxes, etc.
   3. 1920: Hilbert's Program: "mechanize" mathematics.
   4. Finite axioms, inference rules turn crank, determine truth needed: axioms consistent & complete
   5. Hilbert: "No one shall expel us from the paradise that Cantor has created.".
3. Kurt Gödel (1906–1978)
   German logician, at age 25 (1931) proved: "There are true statements that can't be proved or disproved". (i.e., "no" to Hilbert)
   Shook the foundations of mathematics/philosophy/science/everything.

More history: Turing...

Alan Turing (1912–1954):

1. British mathematician
2. cryptoanalysis during WW II (enigma project)
3. Defined a computing model/program. In 1936 (age 23) provided foundations for investigating fundamental question of what is computable, what is not computable.
4. Gay, suicide.
5. Movies, UK apology.
6. Proved the halting theorem: Deciding if a computer program stops on a given input can not be decided by a program.

Turing original paper...

Is quite readable. Available here:
https://www.cs.virginia.edu/~robins/Turing_Paper_1936.pdf

THE END

...

(for now)