The Search for the Perfect Language

• I'll tell you how the search for certainty led to incompleteness, uncomputability & randomness,
• and the unexpected result of the search for the perfect language.
Bibliography

• Umberto Eco, *The Search for the Perfect Language*, in Italian, French, English...


• David Malone, *Dangerous Knowledge*, BBC TV, 90 minutes, [Google video](#)
Umberto Eco, The Search for the Perfect Language

- Language of creation (Hebrew?)
- How God summoned things into being by naming them
- Expresses inner structure of world
- Kabbalah
- Ramon Llull ~ 1200
  - Combine concepts in all possible ways mechanically
  - Wheels
- Source of all (universal) knowledge!
Leibniz

- Mechanical reasoning
- Characteristica universalis
- Calculus ratiocinatar
- Settle disputes: "Gentlemen, let us compute!"
- Computer to multiply
- Binary arithmetic, base-two
- Christian Huygens: "The calculus is too mechanical!"
Georg Cantor

• Mathematical theology!
• Infinite sets, infinite numbers
• Transfinite ordinals: 0, 1, 2, 3 ... $\omega$, $\omega+1$ ... $2\omega$
  ... $\omega^2$ ... $\omega^\omega$ ... $\omega^{\omega\omega\omega\omega}$...
• Transfinite cardinals: Integers = $\aleph_0$, reals = $\aleph_1$, $\aleph_2$, $\aleph_3$ ... $\aleph_\omega$ ... $\aleph_\omega^\omega$ ... $\aleph_\omega^{\omega\omega\omega}$...
• Hilbert/Poincaré: Set theory = heaven/disease!
Bertrand Russell

• Cantor's diagonal argument: Set of all subsets gives next cardinal.
• Russell's paradox: Is the set of all subsets of the universal set bigger than the universal set?!
• Problem: Set of all sets that are not members of (in) themselves.
• Member of itself or not?!
David Hilbert

• Formalize **all** math = TOE = Theory of Everything
• Objective not subjective **truth**
• Absolute truth — **black** or **white**, not **gray**
• Formal axiomatic theory for all mathematics:
  – Has proof-checking algorithm.
  – Has algorithm to generate all the theorems in the theory.
• Meta-mathematics: Study from outside.
Positive Work on Hilbert's Program

- John von Neumann
- Zermelo-Fraenkel
John von Neumann

- Monist ontology (Spinoza!)
- $0 = \{\}$ empty set
- $1 = \{0\} = \{\{\}\}$
- $2 = \{0, 1\} = \{\}, \{\{\}\}\}$
- $\omega = \{0, 1, 2, 3, ...\}$
- Everything is a set!
- Including ordered pairs, sequences & functions.
- All math built from empty set.
Abraham Fraenkel
(Hebrew University)

• Zermelo-Fraenkel set theory = Abstract, formal set theory
• First-order logic
• Axiom of choice
• Avoids sets that are too big.
Negative Work on Hilbert's Program

- Gödel, 1931
- Turing, 1936
- Algorithmic Information Theory (AIT), 1960s → present
Kurt Gödel, 1931

- "This statement is false!"
- True or false?
- "This statement is unprovable!"
- Provable or unprovable?
- No TOE for mathematics!
- Incompleteness!
Alan Turing, 1936

• Universal Turing machine = Universal programming language
• The Halting Problem!
• Uncomputability ⇒ Incompleteness
Algorithmic Information Theory

• Program-Size Complexity =
  – minimum size program,
  – minimum number of yes/no decisions God has to make to create something.

• \( U(\pi_C p) = C(p) \)  **Simulation!**

• Given program \( p \) for computer \( C \), concatenate prefix \( \pi_C \) to \( p \).

• Universal machine \( U \) simulates the computation \( C \) performs given \( p \).

• Most concise, expressive
  – universal Turing machine,
  – universal programming language.

• **The Halting Probability** \( \Omega \).

• Randomness = Irreducible complexity \( \Rightarrow \) Incompleteness!
Negative Conclusions

• Hilbert's search for the perfect language giving all math knowledge
• (= formal axiomatic theory for all of mathematics)
• cannot succeed!
• Every formal axiomatic theory is incomplete, as shown by Gödel, Turing, Ω.
• Search for certainty ⇒ Incompleteness, uncomputability & randomness.
• Nevertheless, mathematicians remain enamored with formal proof:
• See *AMS Notices* special issue on formal proof, December 2008.
Positive Conclusions

• However, perfect languages have emerged for computing, not for reasoning.
• Universal Turing machines and universal programming languages can all simulate each other.
• Most expressive, concise languages: $U(\pi_C p) = C(p)$.
• AIT's program-size complexity is a measure of conceptual complexity, of complexity of ideas.
• Furthermore, viewed from the perspective of the Middle Ages, programming languages give us the God-like power to breathe life into (some) inanimate matter:
  • Hardware $\approx$ **Body**, Software $\approx$ **Soul**!