Today's lectures: Turing Machines

Reading:

October 31: Turing Machines and Effective Computability
Read: *Kozen* lecture 28 (or *Sipser* 3.1).

November 2: More Turing Machines
Read: *Kozen* lecture 29 (or *Sipser* 3.1 again).

November 4: Review and Examples.

November 7: Modified Turing Machines
Read: *Kozen* lecture 30 (or *Sipser* 3.2).

November 9: Second Midterm: in class.

November 14: Diagonalization and the Halting Problem
Read: *Kozen* lecture 31 (or *Sipser* 4.2).

November 16: Decidability
Read: *Kozen* lecture 32 (or *Sipser* 4.1).

November 18: Review and Examples.

November 21: Reductions
Read: *Kozen* lecture 33 (or *Sipser* Chapter 5).

November 23: Gödel's Theorem
Read: *Kozen* lecture 38 (or *Sipser* 6.2).

November 25: Review and Examples.

November 28: Everything Else About Turing Machines.

November 30: Theorem Proving.

December 2: Something Fun.

I. The Formalist Programme

A. Formalizing Mathematics

1. During the second half of the 19^{th} century, mathematicians discovered that much of mathematics could be formalized using set theory. This removed ambiguity, provided a rigorous basis for intuitive ideas, revealed connections between various branches of mathematics, and led to new results. Overall it was quite successful.

2. Once everything is viewed in terms of sets, one ends up needing sets that contain sets as members – just as we needed the power set of the states of an NFA to construct the equivalent DFA.

a. Let $\mathbb{S}$ be the set of all sets. Obviously, this is what we need as our domain of discourse if we want to talk about sets in general.

b. Note that $\mathbb{S} \in \mathbb{S}$. In other words, $\mathbb{S}$ contains itself. There are other situations in which a set ends up containing itself.

c. Now, let $T = \{A \in \mathbb{S} |\ A \notin A\}$.
T is the set of all sets that don't contain themselves. Is $T \in T$?

- If $T \in T$, then T contains itself. Therefore, $T \notin T$ by the definition of T.
- On the other hand, if $T \notin T$, then T doesn't contain itself. Therefore, $T \in T$ by the definition of T.

Either choice leads to a contradiction. This is known as "Russell's Paradox" (after the mathematician Bertrand Russell).

3. Mathematicians generally blamed Russell's Paradox on the vaguaries of natural language. We defined $\mathbb{S}$ with an English sentence, and the definition of T uses $\mathbb{S}$.

a. Perhaps, we could find a way to define sets that didn't require resorting to natural language.

b. If we're lucky, we might find a definition that avoids creating problems like Russell's paradox.

c. Many mathematicians attempted to do this. The most valiant effort was probably that by Russell and Whitehead when they wrote *Principia Mathematica*. However, even *Principia* is unsatisfying. It makes a start at putting all of mathematics into a formal framework, but it shows that it takes lots of work to cover very simple ideas. It left the question open, could the formalization of all of mathematics be completed if enough effort were invested?

B. The Hilbert Questions

1. The setting: International Conference of Mathematicians, Paris, 1900

2. Hilbert described a vision for mathematics

a. Develop a formal approach to mathematics that is

i. Sound: it is not possible to prove a contradiction or falsehood.

ii. Complete: all true statements can be proven.

iii. Decidable: given a true statement, its proof can be derived in a finite number of steps.

b. He then mentioned 23 open problems in mathematics that should be solved by this approach. He started with the continuum hypothesis, and included Fermat's Last theorem and many other famous problems.

Here's an excerpt from the speech. Note his confidence that "we have, nevertheless, the firm conviction that their solution must follow by a finite number of purely logical processes."

> If we do not succeed in solving a mathematical problem, the reason frequently consists in our failure to recognize the more general standpoint from which the problem before us appears only as a single link in a chain of related problems. ...Occasionally it happens that we seek the solution under insufficient hypotheses or in an incorrect sense, and for this reason do not succeed. The problem then arises: to show the impossibility of the solution under the given hypotheses, or in the sense contemplated. Such proofs of impossibility were effected by the ancients, for instance when they showed that the ratio of the hypotenuse to the side of an isosceles right triangle is irrational. In later mathematics, the question as to the impossibility of certain solutions plays a preeminent part, and we perceive in this way that old and difficult problems, such as the proof of the axiom of parallels, the squaring of the circle, or the solution of equations of the fifth degree by radicals have finally found fully satisfactory and rigorous solutions, although in another sense than that originally intended. It is probably this important fact along with other philosophical reasons that gives rise to the conviction (which every mathematician shares, but which no one has as yet supported by a proof) that every definite mathematical problem must necessarily be susceptible of an exact settlement, either in the form of an actual answer to the question asked, or by the proof of the impossibility of its

solution and therewith the necessary failure of all attempts. Take any definite unsolved problem, such as the question as to the irrationality of the Euler-Mascheroni constant C, or the existence of an infinite number of prime numbers of the form 2^{n+1}. However unapproachable these problems may seem to us and however helpless we stand before them, we have, nevertheless, the firm conviction that their solution must follow by a finite number of purely logical processes.

From: http://aleph0.clarku.edu/ djoyce/hilbert/problems.html.

II. Turing Machines

A. The Machinery

1. A tape

a. The tape holds the initial input.

b. The Turing machine can overwrite symbols on the tape.

2. A finite state controller – at each step the controller:

a. reads a symbol from the tape

b. based on the symbol and current state: the machine

i. writes a symbol at the position from which it just read,

ii. moves the read/write head one position to the left or to the right, and

iii. updates its state

B. Formal Descriptions of Turing Machines

1. The ingredients

a. A tape alphabet, Γ.

b. An input alphabet, $\Sigma \subset \Gamma$.

We need a subset so that the machine can tell where the input ends.

c. A set of states for the controller, Q.

d. A transition function, $\delta : (Q \times \Gamma) \to (Q \times \Gamma \times \{L, R\})$

e. Some special states and symbols:

s: the start state.

t: the accepting state.

r: the rejecting state.

$\sqcup$: a blank.

$\vdash$: the left tape end-marker.

2. The tuple: $(Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$

Nine pieces, but we've defined them all above. Note that Q, Γ, and δ really capture the operation of the machine. The other pieces are there to say how to start the machine, how to know where the input ends, and how to know when the machine is done.

3. Configurations

The state of the machine is determined by the state of the finite-state controller, $q \in Q$, the contents of the tape, $x \in \Gamma^*$, and the position of the read/write head, $n \in \mathbb{N}$.

a. We write a configuration as a tuple, (q, x, n).

b. Noting that the tape extends infinitely to the right with blanks, we typically write $x = z\sqcup^\omega$.

C. An Example

1. Figure 1 shows a Turing machine that recognizes the language $a^n b^n c^n$.

2. Note that this language is neither regular nor context-free.

3. The machine is a slight simplification of the one presented by Kozen. It operates as follows:

a. state s: Make a left-to-right pass over the input string, and append a $\dashv$.

b. state 1: Return to the left end marker.

c. states $2 - 5$: Make a left-to-right pass replacing the first a with a $\sqcup$, and likewise for the first b and the first c. If the tape is of the form $\sqcup^* \dashv$, then accept. Otherwise, if the tape isn't of the form $(a + \sqcup)^*(b + \sqcup)^*(c + \sqcup)^* \dashv$, then reject. Otherwise, go back to state 1, return to the left end-marker, and make another pass.

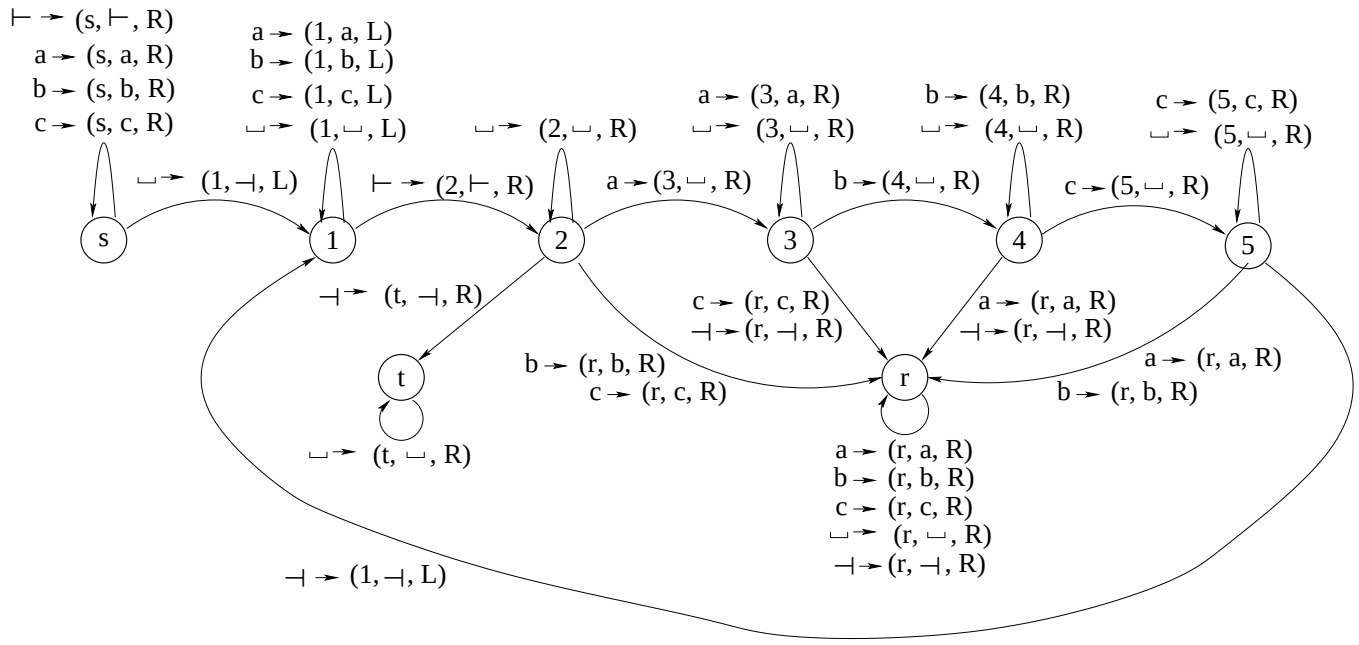

Figure 1: A Turing Machine That Recognizes $a^n b^n c^n$

4. Figure 2 shows the sequence of configurations when processing aaabbbccc. The position of the read/write head is indicated by underlining the corresponding tape symbol.

III. Where We Go From Here

A. What Turing Machines Can Do

B. What Turing Machines Can't Do

C. Universality

Step	State	Tape
0	s	$\vdash$aaabbbcccc$\sqcup^\omega$
1	s	$\vdash$a$\underline{\text{a}}$abbbcccc$\sqcup^\omega$
2	s	$\vdash$aa$\underline{\text{a}}$bbbcccc$\sqcup^\omega$
3	s	$\vdash$aaa$\underline{\text{a}}$bbbcccc$\sqcup^\omega$
4	s	$\vdash$aaab$\underline{\text{b}}$bbcccc$\sqcup^\omega$
5	s	$\vdash$aaabb$\underline{\text{b}}$bcccc$\sqcup^\omega$
6	s	$\vdash$aaabb$\underline{\text{b}}$bcccc$\sqcup^\omega$
7	s	$\vdash$aaabbb$\underline{\text{c}}$bcccc$\sqcup^\omega$
8	s	$\vdash$aaabbbc$\underline{\text{c}}$bcccc$\sqcup^\omega$
9	s	$\vdash$aaabbbcc$\underline{\text{c}}$cccc$\sqcup^\omega$
10	s	$\vdash$aaabbbcccc$\underline{\sqcup}\sqcup^\omega$
11	1	$\vdash$aaabbbcccc$\underline{\sqcup}\sqcup^\omega$
12	1	$\vdash$aaabbbcc$\underline{\text{c}}$cc$\sqcup\sqcup^\omega$
13	1	$\vdash$aaabbbc$\underline{\text{c}}$ccc$\sqcup\sqcup^\omega$
14	1	$\vdash$aaabbb$\underline{\text{b}}$cccc$\sqcup\sqcup^\omega$
15	1	$\vdash$aaabb$\underline{\text{b}}$bcccc$\sqcup\sqcup^\omega$
16	1	$\vdash$aaa$\underline{\text{b}}$bbcccc$\sqcup\sqcup^\omega$
17	1	$\vdash$aa$\underline{\text{a}}$bbbcccc$\sqcup\sqcup^\omega$
18	1	$\vdash$a$\underline{\text{a}}$abbbcccc$\sqcup\sqcup^\omega$
19	1	$\vdash$$\underline{\text{a}}$aabbbcccc$\sqcup\sqcup^\omega$
20	1	$\vdash$$\underline{\vdash}$aaabbbcccc$\sqcup\sqcup^\omega$
21	2	$\vdash$$\underline{\text{a}}$aabbbcccc$\sqcup\sqcup^\omega$
22	3	$\vdash\sqcup\underline{\text{a}}$abbbcccc$\sqcup\sqcup^\omega$
23	3	$\vdash\sqcup$a$\underline{\text{a}}$bbbcccc$\sqcup\sqcup^\omega$
24	3	$\vdash\sqcup$aa$\underline{\text{b}}$bbcccc$\sqcup\sqcup^\omega$
25	4	$\vdash\sqcup$aa$\sqcup\underline{\text{b}}$bcccc$\sqcup\sqcup^\omega$
26	4	$\vdash\sqcup$aa$\sqcup$b$\underline{\text{b}}$cccc$\sqcup\sqcup^\omega$
27	4	$\vdash\sqcup$aa$\sqcup$bb$\underline{\text{c}}$ccc$\sqcup\sqcup^\omega$
28	5	$\vdash\sqcup$aa$\sqcup$bb$\sqcup\underline{\text{c}}cc\sqcup\sqcup^\omega$
29	5	$\vdash\sqcup$aa$\sqcup$bb$\sqcup$c$\underline{\text{c}}$cc$\sqcup\sqcup^\omega$
30	5	$\vdash\sqcup$aa$\sqcup$bb$\sqcup$cc$\underline{\sqcup}\sqcup^\omega$
31	1	$\vdash\sqcup$aa$\sqcup$bb$\sqcup$c$\underline{\text{c}}$cc$\sqcup\sqcup^\omega$
32	1	$\vdash\sqcup$aa$\sqcup$bb$\underline{\sqcup}$cc$\sqcup\sqcup^\omega$
33	1	$\vdash\sqcup$aa$\sqcup$b$\underline{\text{b}}\sqcup$cc$\sqcup\sqcup^\omega$
34	1	$\vdash\sqcup$aa$\sqcup\underline{\text{b}}b\sqcup$cc$\sqcup\sqcup^\omega$
35	1	$\vdash\sqcup$aa$\underline{\sqcup}$bb$\sqcup$cc$\sqcup\sqcup^\omega$
36	1	$\vdash\sqcup$a$\underline{\text{a}}\sqcup$bb$\sqcup$cc$\sqcup\sqcup^\omega$
37	1	$\vdash\sqcup\underline{\text{a}}\sqcup$bb$\sqcup$cc$\sqcup\sqcup^\omega$
38	1	$\vdash\underline{\sqcup}$aa$\sqcup$bb$\sqcup$cc$\sqcup\sqcup^\omega$
39	1	$\vdash\underline{\vdash}\sqcup$aa$\sqcup$bb$\sqcup$cc$\sqcup\sqcup^\omega$
40	2	$\vdash\underline{\sqcup}$aa$\sqcup$bb$\sqcup$cc$\sqcup\sqcup^\omega$
41	2	$\vdash\sqcup\underline{\text{a}}a\sqcup$bb$\sqcup$cc$\sqcup\sqcup^\omega$
42	3	$\vdash\sqcup\sqcup\underline{\text{a}}\sqcup$bb$\sqcup$cc$\sqcup\sqcup^\omega$
43	3	$\vdash\sqcup\sqcup$a$\underline{\sqcup}$bb$\sqcup$cc$\sqcup\sqcup^\omega$
44	3	$\vdash\sqcup\sqcup$a$\sqcup\underline{\text{b}}b\sqcup$cc$\sqcup\sqcup^\omega$
45	4	$\vdash\sqcup\sqcup$a$\sqcup\sqcup\underline{\text{b}}\sqcup$cc$\sqcup\sqcup^\omega$
46	4	$\vdash\sqcup\sqcup$a$\sqcup\sqcup$b$\underline{\sqcup}$cc$\sqcup\sqcup^\omega$
47	4	$\vdash\sqcup\sqcup$a$\sqcup\sqcup$b$\sqcup\underline{\text{c}}c\sqcup\sqcup^\omega$
48	5	$\vdash\sqcup\sqcup$a$\sqcup\sqcup$b$\sqcup\sqcup\underline{\text{c}}\sqcup\sqcup^\omega$
49	5	$\vdash\sqcup\sqcup$a$\sqcup\sqcup$b$\sqcup\sqcup$c$\underline{\sqcup}\sqcup^\omega$
50	1	$\vdash\sqcup\sqcup$a$\sqcup\sqcup$b$\sqcup\sqcup\underline{\text{c}}\sqcup\sqcup^\omega$

Step	State	Tape
51	1	$\vdash\sqcup\sqcup$a$\sqcup\sqcup$b$\sqcup\sqcup$c$\sqcup\sqcup^\omega$
52	1	$\vdash\sqcup\sqcup$a$\sqcup\sqcup$b$\underline{\sqcup}\sqcup$c$\sqcup\sqcup^\omega$
53	1	$\vdash\sqcup\sqcup$a$\sqcup\sqcup\underline{\text{b}}\sqcup\sqcup$c$\sqcup\sqcup^\omega$
54	1	$\vdash\sqcup\sqcup$a$\sqcup\underline{\sqcup}$b$\sqcup\sqcup$c$\sqcup\sqcup^\omega$
55	1	$\vdash\sqcup\sqcup$a$\underline{\sqcup}\sqcup$b$\sqcup\sqcup$c$\sqcup\sqcup^\omega$
56	1	$\vdash\sqcup\sqcup\underline{\text{a}}\sqcup\sqcup$b$\sqcup\sqcup$c$\sqcup\sqcup^\omega$
57	1	$\vdash\sqcup\underline{\sqcup}a\sqcup\sqcup$b$\sqcup\sqcup$c$\sqcup\sqcup^\omega$
58	1	$\vdash\underline{\sqcup}\sqcup$a$\sqcup\sqcup$b$\sqcup\sqcup$c$\sqcup\sqcup^\omega$
59	1	$\vdash\underline{\vdash}\sqcup\sqcup$a$\sqcup\sqcup$b$\sqcup\sqcup$c$\sqcup\sqcup^\omega$
60	2	$\vdash\underline{\sqcup}\sqcup$a$\sqcup\sqcup$b$\sqcup\sqcup$c$\sqcup\sqcup^\omega$
61	2	$\vdash\sqcup\sqcup$a$\sqcup\sqcup$b$\sqcup\sqcup$c$\sqcup\sqcup^\omega$
62	2	$\vdash\sqcup\underline{\sqcup}a\sqcup\sqcup$b$\sqcup\sqcup$c$\sqcup\sqcup^\omega$
63	2	$\vdash\sqcup\sqcup\underline{\text{a}}\sqcup\sqcup$b$\sqcup\sqcup$c$\sqcup\sqcup^\omega$
64	3	$\vdash\sqcup\sqcup\sqcup\underline{\sqcup}\sqcup$b$\sqcup\sqcup$c$\sqcup\sqcup^\omega$
65	3	$\vdash\sqcup\sqcup\sqcup\sqcup\underline{\sqcup}b\sqcup\sqcup$c$\sqcup\sqcup^\omega$
66	3	$\vdash\sqcup\sqcup\sqcup\sqcup\underline{\text{b}}\sqcup\sqcup$c$\sqcup\sqcup^\omega$
67	4	$\vdash\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\underline{\sqcup}\sqcup$c$\sqcup\sqcup^\omega$
68	4	$\vdash\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\underline{\sqcup}c\sqcup\sqcup^\omega$
69	4	$\vdash\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\underline{\text{c}}\sqcup\sqcup^\omega$
70	5	$\vdash\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\underline{\sqcup}\sqcup\sqcup^\omega$
71	1	$\vdash\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\underline{\sqcup}\sqcup\sqcup^\omega$
72	1	$\vdash\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\underline{\sqcup}\sqcup\sqcup\sqcup^\omega$
73	1	$\vdash\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\underline{\sqcup}\sqcup\sqcup\sqcup\sqcup^\omega$
74	1	$\vdash\sqcup\sqcup\sqcup\sqcup\sqcup\underline{\sqcup}\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup^\omega$
75	1	$\vdash\sqcup\sqcup\sqcup\sqcup\underline{\sqcup}\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup^\omega$
76	1	$\vdash\sqcup\sqcup\sqcup\underline{\sqcup}\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup^\omega$
77	1	$\vdash\sqcup\sqcup\underline{\sqcup}\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup^\omega$
78	1	$\vdash\sqcup\underline{\sqcup}\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup^\omega$
79	1	$\vdash\underline{\sqcup}\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup^\omega$
80	1	$\vdash\underline{\vdash}\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup^\omega$
81	2	$\vdash\underline{\sqcup}\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup^\omega$
82	2	$\vdash\sqcup\underline{\sqcup}\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup^\omega$
83	2	$\vdash\sqcup\sqcup\underline{\sqcup}\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup^\omega$
84	2	$\vdash\sqcup\sqcup\sqcup\underline{\sqcup}\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup^\omega$
85	2	$\vdash\sqcup\sqcup\sqcup\sqcup\underline{\sqcup}\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup^\omega$
86	2	$\vdash\sqcup\sqcup\sqcup\sqcup\sqcup\underline{\sqcup}\sqcup\sqcup\sqcup\sqcup\sqcup^\omega$
87	2	$\vdash\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\underline{\sqcup}\sqcup\sqcup\sqcup^\omega$
88	2	$\vdash\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\underline{\sqcup}\sqcup\sqcup^\omega$
89	2	$\vdash\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\underline{\sqcup}\sqcup^\omega$
90	2	$\vdash\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\underline{\sqcup}\sqcup^\omega$
91	t	$\vdash\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\underline{\sqcup}\sqcup^\omega$

Figure 2: Configurations for the TM from figure 1 accepting aaabbbccc $\in a^n b^n c^n$

The Greatest Math Mistake of the Century

The following is a myth based on a true story. It contains inaccuracies that exist only because the story would be long, complicated and incomprehensible to most if I told it accurately. I have tried to correct some of the inaccuracies in the footnotes for you purists out there. Links are to the History of Mathematics page.

I think it was Benjamin Franklin, who once said, "The only bad mistakes are the ones you don't learn from." A lesson for us all.

That being the case, I shall tell you the story of a good math mistake, because we learned from this one. In fact, we learned so much that this math mistake is not only good, it is great, perhaps the greatest math mistake of the century. It was made by David Hilbert early in the century.

Hilbert was a great mathematician, who mastered all the fields of math there were, because you could still do that back then. So it is surprising that such a genius could make such a blunder.

Hilbert wanted to solve every arithmetic problem there is[1]. Or, rather, find a method that could be used to solve every one. It seemed obvious that they were all solvable! That was Hilbert's big mistake.

"Yeah, how hard can it be?", said Bertrand Russell[2]. "$1 + 1 = 2$, everything follows from that", said Alfred North Whitehead[3]. Many years passed before *Principia Mathematica* came about[4] (it was harder than they thought), "but Hilbert is right, and here's the proof", said they.

"Not so fast", said a tall gangly Kurt Gödel[5] in the back row with an Austrian accent. "The problem of solving every arithmetic problem is itself an arithmetic problem, and proving that all arithmetic is solvable is also an arithmetic problem; Hence, **proving all arithmetic is *not* solvable is also an arithmetic problem**. And, if this last problem is solved then we have proven arithmetic false, but if we cannot solve this last problem, then arithmetic is, by counter example, incomplete." Kurt Gödel's incompleteness theory proved Hilbert wrong[6].

Russell and Whitehead took up Philosophy, which made them much more popular, since no one understood *Principia* anyway.

A few years later a British track star with the unlikely name of Al Turing[7] asked an unlikely question[8] "What if we limited math to what could be done by a computer, would it be subjected to Gödel's restrictions?"
"What's a computer?", they responded.
"Well, it is a machine that can do math for you.", said Al.
"Interesting!", they responded.
"You see if you limit the eigenstates to what is mechanical you can use a similar trick Gödel used...."
"What kind of a machine?"
"Just any computing machine, now as I was saying..."
"Wait a second, are you saying it is possible to build a machine to do math for us?"
"Yes, and with it I can prove..."

---

[1] Hilbert's challenge can be found in his famous 1901 lecture on 23 unsolved problems in Mathematics. Hilbert was a founder of the Formalist school of mathematics which believed it was possible to solve every math problem. I use the term "arithmetic" where in reality it was a general class of logic structure called formal systems, what we call "arithmetic" being an example of a formal system, others include set theory, linear algebra, polynomial algebra, symbolic logic, and all computer languages. Noam Chomsky of MIT believes that human languages may also be formal in structure.

[2] See http://www-groups.dcs.st-and.ac.uk/∼history/Mathematicians/Russell.html.

[3] See http://www-groups.dcs.st-and.ac.uk/∼history/Mathematicians/Whitehead.html.

[4] *Principia Mathematica* was started in 1910 and finished in 1913.

[5] See http://www-groups.dcs.st-and.ac.uk/∼history/Mathematicians/Godel.html.

[6] Kurt Gödel gave his lecture in 1931.

[7] See http://www-groups.dcs.st-and.ac.uk/∼history/Mathematicians/Turing.html.

[8] Turing's work *On Computable Numbers* was published in 1936.

"Hold on a minute," they responded, "Are you telling us that we have rooms filled with accountants using slide rules and abacuses figuring out all our companies finances and all this time we could have just used some bloody machine do it for us?"
"Well they can't do everything, as I was about to show you. . . "
"Where can we get one of these computers?"
"Oh forget it!", responded Al dejectedly.

What Turing was trying to say was this: In Kurt Gödel's theory, he used a technique of enumerating every arithmetic problem there is. $1 + 1 = 2$ is enumerated to 45236, etc. Using the same technique, we can create a theoretical operating system, for a theoretical device that can do math automatically. The "Turing Machine" as described was completely impractical and could never be built. But, it did not rule out the possibility of a practical design. A problem for someone else.

The someone else was Turing's professor, John Von Neumann[9]. He actually attended Gdel's original lecture. And, may be the only man, besides Hilbert, to understand it at the time. Since they all spoke German.

"A computer you say. . . Hmm, very interesting.", said Von Neumann, "I have not seen one, but if I were to build one, I would need a input device of some sort." "Oooh", they said. "And some kind of control mechanism, electrical if possible." "Ahhh", they said. "It would be better if it were binary, that way memory could be created easily." "Wow, memory!" "And of course you would need a means of receiving output, like a television." "What's a television?"

The (*ahem!*) "Von Neumann Machine" was born. John may be famous for many things, Humility was not one of them. Luckily "computer" had a better ring to it.

Later, Von Neumann did actually help design the first computer, with a couple of other "John"s namedEckert and Mauchly[10]. But, just in case the computer started taking control of the world like in those sci-fi stories, Von Neumann helped design the Atom Bomb the previous year. He then invented Game Theory, which proved that the bomb should never be used, unless you are really really mad[11]. Same goes for computers for that matter.

Eckert and Mauchly went on to actually build the first programmable computer in Philadelphia. Just across the bridge from New Jersey where Kurt Gödel lived[12]. Small world.

Later we learned Alan Turing built a programmable electronic computer predating the one in Philly *Turing's digital electronic computer "Colossus" was completed in Britain in 1943, followed by Colossus II in 1944. Other computers predating ENIAC were built by Charles Babbage and Ada Lovelace (Britain,1842, mechanical, never finished), Herman Hollerith (America,1890, Mechanical), Vannevar Bush (America,1930 analog electro-mechanical), Konrad Zuse (Germany,1939, digital electro-mechanical), Helmut Hoelzer (Germany,1941, analog electronic), Howard Aiken and Grace Hopper (America, 1942, electro-mechanical), and John Atanasoff and Clifford Berry (America, 1942, digital electronic, never finished).* . It helped win the war, but Turing could not take credit or make any money from his invention.¡br¿ It was Top Secret.

Alan Turing committed suicide[13] after a sex scandal. (He was gay – long before it became trendy.) John Von Neumann died of radiation[14] after standing too close to the Atom Bomb. Kurt Gödel, after mathematically proving food is bad for you, died of starvation[15]. They were all completely crazy when they left this mortal realm. . . Computers can do that.

David Hilbert lived a long and happy life[16], having never lived to see a computer. Never knowing how big the consequences of his great little mistake were. We unfortunately were not so lucky.

The rest, as they say, is history. Later Bill Gates came along and took over the world, Where is Von Neumann's bomb when you really need it?

---

[9] See http://www-groups.dcs.st-and.ac.uk/∼history/Mathematicians/Von_Neumann.html. Von Neumann taught at Princeton University in the late 30's while Turing was a grad student. He worked on designs of "Von Neumann Machines" through the early 1940's. Other Pre-realization computer pioneers included Norbert Weiner, Howard Aiken, Alonzo Church, and Alwin Walther of Germany. The latter working with computer pioneer Konrad Zuse. The two worked for Germany during W.W.II completely unaware of the work of Turing or Von Neumann.

[10] In truth, Von Neumann merely played an advisory role on ENIAC. His role on UNIVAC was more significant. See http://www-groups.dcs.st-and.ac.uk/∼history/Mathematicians/Eckert_John.html.

[11] Von Neumann worked for the Manhattan Project at the end of W.W.II, his game theory argued strategies for the Cold War that inspired the movies *Dr. Strangelove* and *War Games*.

[12] During the 40's and early 50's, Gödel worked with Albert Einstein at Princeton University. He was portrayed in the 1996 movie *IQ* by an actor who looked nothing like him.

[13] Died 1954 in England.

[14] Died 1957 of Brain Cancer in Washington D.C.

[15] Died 1978 in Princeton, New Jersey.

[16] Hilbert retired in 1930 and despite the war, died peacefully in Germany in 1943.