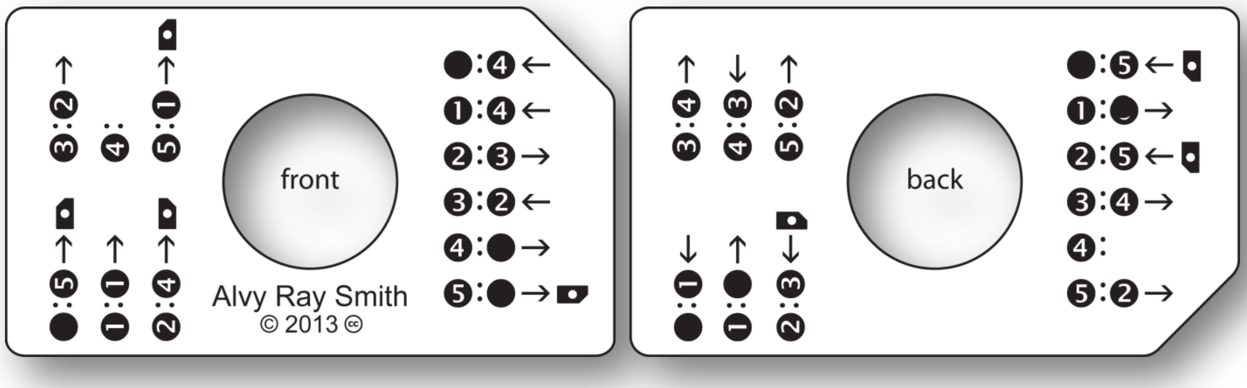
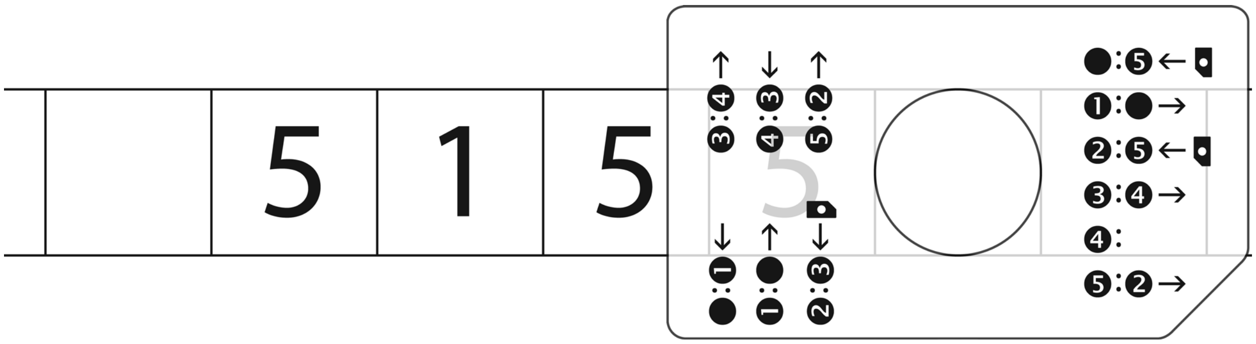
# A Business Card Universal Turing Machine[1]

*by Alvy Ray Smith*

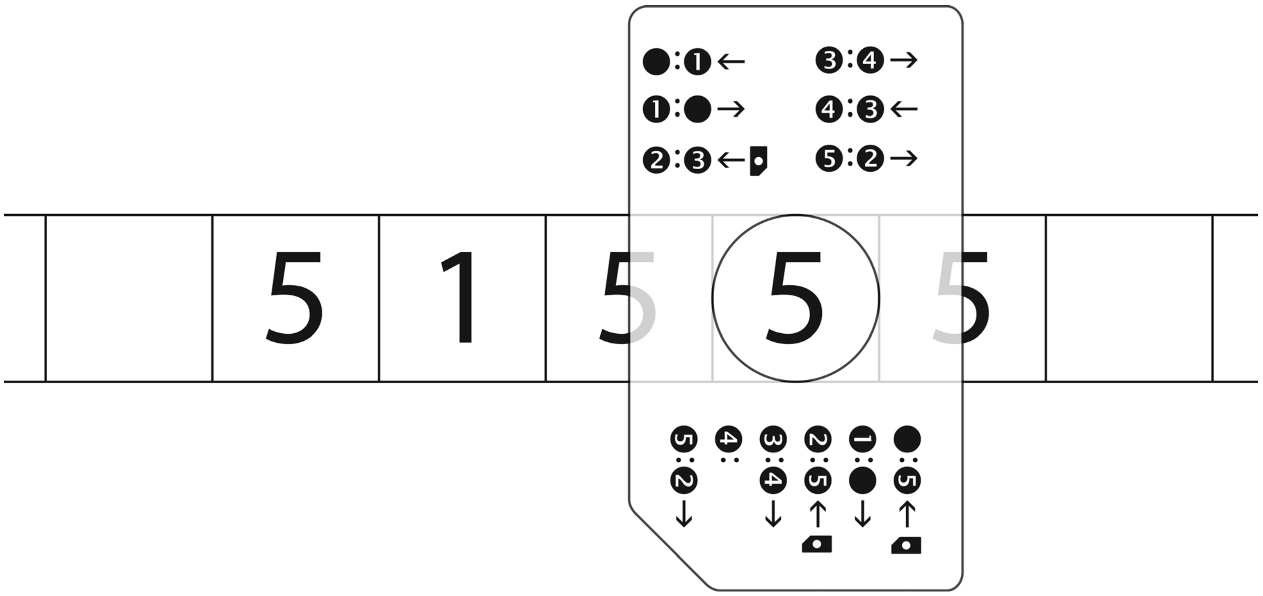
You have in your hands a simple device that can compute anything whatsoever that's computable. It's a realization of one of the world's great ideas—Alan Turing's universal computer. This piece of computer hardware is a business card with a hole in the center and one corner cut off, as depicted here.



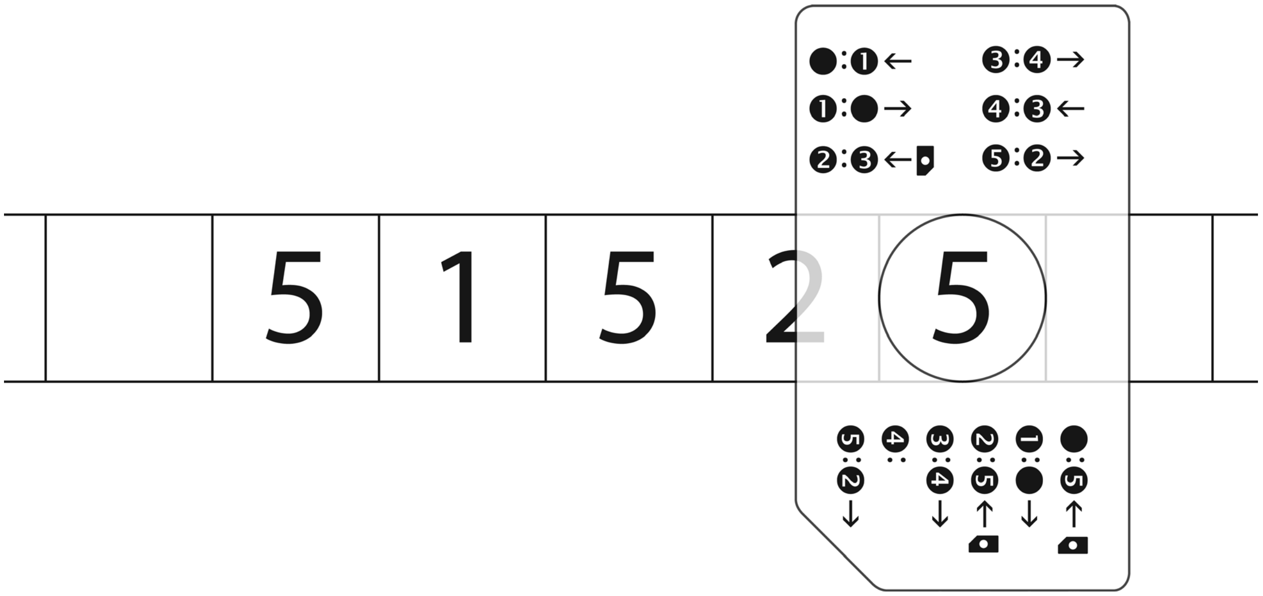
Imagine that there's a paper tape running from left to right. It's divided into squares, and you can see one square through the hole in the card. The tape is mostly blank, but there are typically one or more squares with symbols on them. In this implementation the symbols are the blank and the numerals 1, 2, 3, 4, and 5. Here's a picture of a tape with just four marks on it: 5155. The rest of the tape is blank. The card is shown slightly transparent so that you can see the tape through it.



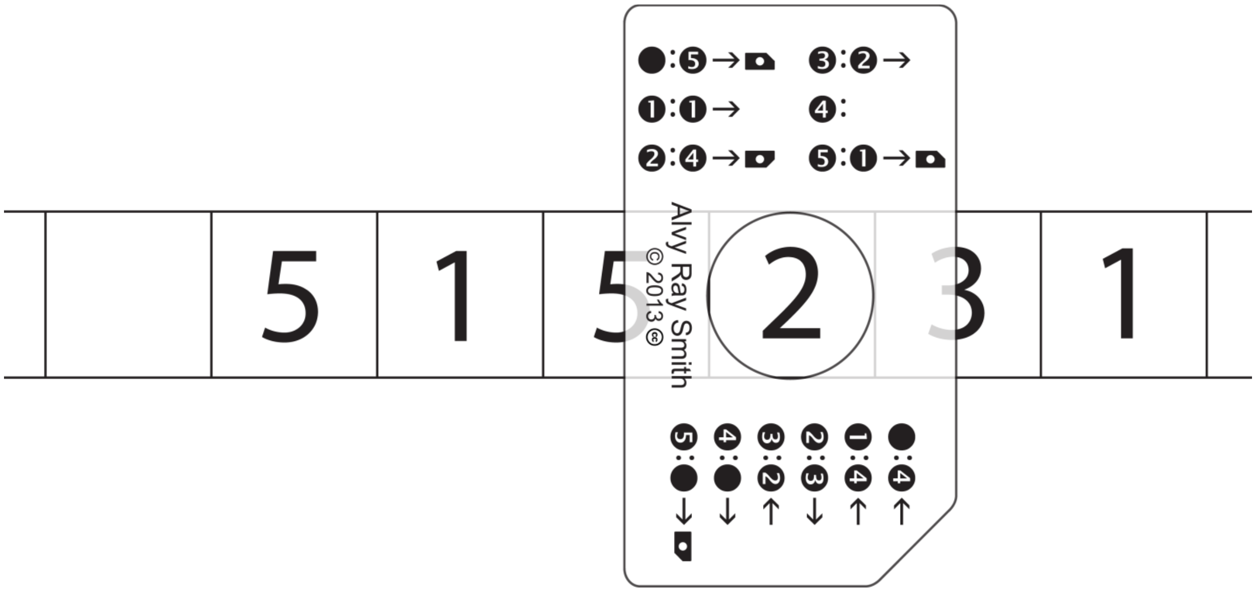
This business card computer works like this. Find the rule for the symbol in the hole. It's the one at the upper right in this case. (Pay no attention to rules written sideways.) It says to write a 5 in the blank space, then move the card left by one square, then rotate the card to match the little glyph which represents the card itself. Here's the result.



The rule that now applies is the one at the lower right. (Recall that you are to ignore the rules written sideways.) It says to replace the 5 with a 2, then move right one square. There's no little card glyph this time, so the card remains in the same orientation. Here's the result.



Skipping ahead three steps gives this configuration.



And so on and on. If you pursue this further, you'll find that the card will eventually end up in its original orientation (the "back" position), with a 4 in the hole. The rule that applies in this case is empty right of the colon, which means nothing else happens. The computarion halts.

In the 1930s, a "computer" was a human—usually a woman. In 1936 Alan Turing captured what a computer did with pencil and paper when she carefully executed a "systematic process"—such as adding a list of several dozen numbers—perhaps interrupting the effort with a tea break. A Turing machine was the model he devised that captured the essential simplicity of what she did. He ignored her tedium as irrelevant.

Turing's model of a "computer" was a very simple device—a nonphysical, mathematical idea—which we now call a *Turing machine*. He defined each of his machines to have only four things: a one-dimensional tape divided into squares, a finite alphabet of symbols for it, a tape scanner with a finite number of states, and an "instruction table" that determines what to do with each combination of scanner state and tape symbol. And one other thing. The tape is as long as necessary in either direction. There's always another square if you need it. From this simple machine concept came all of computing. It surprised Turing's professor at Cambridge University and might surprise you—because it truly is a very simple idea, with profound consequences.

To see that the business card device is a (realization of a) Turing machine, notice that the tape scanner is the business card itself with a hole in it. The alphabet is the six symbols (counting the blank). The four sets of rules form the instruction table.

The only part of Turing's definition that you might find nonintuitive is "state." That's a mathematical notion that captures, for example, the two stable positions of a light switch. In the case at hand, the four possible vertical or horizontal orientations of the business card are its four states.

But the business card device isn't just any old ordinary Turing machine. It's "universal." What does that mean?

Turing's great idea wasn't only that a Turing machine could execute a systematic process—that what we mean by a "systematic process" is embodied exactly by a Turing machine. Turing's master stroke was to show that there was a *single* Turing machine that could do what *any* other Turing machine could do. It could perform *all* systematic processes in other words. It's one machine that can compute anything whatsoever that's computable. That's what *universal* means in computation. The modern computer is a descendant of this particular kind of Turing machine, the universal Turing machine.

Turing's trick for making one of his machines universal was clever and simple. He wanted his universal machine to compute what any other Turing machine would compute. Let's call this arbitrary other machine T. So he placed a description of T's instruction table—its set of rules—on the tape of the universal machine. Then he designed the scanner of the universal machine—*its* instruction table—to read the description of T off the tape and do what T would have done to the tape. That's a systematic process so has to be possible—exactly Turing's key idea. The part of the tape of the universal machine to the right of the description of T serves as T's tape. In modern terminology, Turing stored the program of T in the memory of the universal machine, and stored the data there too. To change which machine the universal machine simulated—that is, which computation it performed—he simply changed the program, the description part of the tape. His universal Turing machine was what we now call a *stored-program* computer, since it stored the program and the data in the same way, and both in the memory of the machine. This is what is meant by a computer today.

The business card machine is a universal Turing machine, with four states and six symbols. The point is that a computer is a simple idea. As you might suppose, it's programming the computer that's the hard part. And although speed is not part of the idea of computation, it's the vast increase in speed—and concomitant decrease in size—that has made computers the useful and ubiquitous companions they now are.

**Acknowledgement:** I based the design of this device on one of the simplest universal Turing machines ever discovered, by Yurii Rogozhin in 1996.[2] I thank my colleague Tom Griffiths of UC Berkeley for inspiring me to realize Rogozhin's UTM(4,6)—as it's formally called—as a business card, with states represented by orientation. I thank Dan Garcia of UC Berkeley for finding the example computation used here. His complete simulation follows. The second column of steps follows the first column, and then the third column follows them. The first item in each step is the current orientation of the card (LU = front, or landscape mode with notch up; LD = back, or landscape mode with notch down; PR = front rotated, or portrait mode with notch right; PL = back rotated, or portrait mode with notch left), and the outlined square shows the hole location at each step:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| LD | 5 | 1 | 5 | 5 | $\boxed{\phantom{0}}$ | | | | |
| PL | 5 | 1 | 5 | $\boxed{5}$ | 5 | | | | |
| PL | 5 | 1 | 5 | 2 | $\boxed{5}$ | | | | |
| PL | 5 | 1 | 5 | 2 | 2 | $\boxed{\phantom{0}}$ | | | |
| PL | 5 | 1 | 5 | 2 | $\boxed{2}$ | 1 | | | |
| PR | 5 | 1 | 5 | $\boxed{2}$ | 3 | 1 | | | |
| LD | 5 | 1 | 5 | 4 | $\boxed{3}$ | 1 | | | |
| LD | 5 | 1 | 5 | 4 | 4 | $\boxed{1}$ | | | |
| LD | 5 | 1 | 5 | 4 | 4 | | $\boxed{\phantom{0}}$ | | |
| PL | 5 | 1 | 5 | 4 | 4 | $\boxed{\phantom{0}}$ | 5 | | |
| PL | 5 | 1 | 5 | 4 | $\boxed{4}$ | 1 | 5 | | |
| PL | 5 | 1 | 5 | $\boxed{4}$ | 3 | 1 | 5 | | |
| PL | 5 | 1 | $\boxed{5}$ | 3 | 3 | 1 | 5 | | |
| PL | 5 | 1 | 2 | $\boxed{3}$ | 3 | 1 | 5 | | |
| PL | 5 | 1 | 2 | 4 | $\boxed{3}$ | 1 | 5 | | |
| PL | 5 | 1 | 2 | 4 | 4 | $\boxed{1}$ | 5 | | |
| PL | 5 | 1 | 2 | 4 | 4 | | $\boxed{5}$ | | |
| PL | 5 | 1 | 2 | 4 | 4 | | 2 | $\boxed{\phantom{0}}$ | |
| PL | 5 | 1 | 2 | 4 | 4 | | $\boxed{2}$ | 1 | |
| PR | 5 | 1 | 2 | 4 | 4 | $\boxed{\phantom{0}}$ | 3 | 1 | |
| LU | 5 | 1 | 2 | 4 | 4 | 5 | $\boxed{3}$ | 1 | |
| LU | 5 | 1 | 2 | 4 | 4 | $\boxed{5}$ | 2 | 1 | |
| LD | 5 | 1 | 2 | 4 | 4 | | $\boxed{2}$ | 1 | |
| PL | 5 | 1 | 2 | 4 | 4 | $\boxed{\phantom{0}}$ | 5 | 1 | |
| PL | 5 | 1 | 2 | 4 | $\boxed{4}$ | 1 | 5 | 1 | |
| PL | 5 | 1 | 2 | $\boxed{4}$ | 3 | 1 | 5 | 1 | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| PL | 5 | 1 | $\boxed{2}$ | 3 | 3 | 1 | 5 | 1 | | |
| PR | 5 | $\boxed{1}$ | 3 | 3 | 3 | 1 | 5 | 1 | | |
| PR | 5 | 1 | $\boxed{3}$ | 3 | 3 | 1 | 5 | 1 | | |
| PR | 5 | 1 | 2 | $\boxed{3}$ | 3 | 1 | 5 | 1 | | |
| PR | 5 | 1 | 2 | 2 | $\boxed{3}$ | 1 | 5 | 1 | | |
| PR | 5 | 1 | 2 | 2 | 2 | $\boxed{1}$ | 5 | 1 | | |
| PR | 5 | 1 | 2 | 2 | 2 | 1 | $\boxed{5}$ | 1 | | |
| LU | 5 | 1 | 2 | 2 | 2 | 1 | 1 | $\boxed{1}$ | | |
| LU | 5 | 1 | 2 | 2 | 2 | 1 | $\boxed{1}$ | 4 | | |
| LU | 5 | 1 | 2 | 2 | 2 | $\boxed{1}$ | 4 | 4 | | |
| LU | 5 | 1 | 2 | 2 | $\boxed{2}$ | 4 | 4 | 4 | | |
| LU | 5 | 1 | 2 | 2 | 3 | $\boxed{4}$ | 4 | 4 | | |
| LU | 5 | 1 | 2 | 2 | 3 | | $\boxed{4}$ | 4 | | |
| LU | 5 | 1 | 2 | 2 | 3 | | | $\boxed{4}$ | | |
| LU | 5 | 1 | 2 | 2 | 3 | | | | $\boxed{\phantom{0}}$ | |
| LU | 5 | 1 | 2 | 2 | 3 | | | $\boxed{\phantom{0}}$ | 4 | |
| LU | 5 | 1 | 2 | 2 | 3 | | $\boxed{\phantom{0}}$ | 4 | 4 | |
| LU | 5 | 1 | 2 | 2 | 3 | $\boxed{\phantom{0}}$ | 4 | 4 | 4 | |
| LU | 5 | 1 | 2 | 2 | $\boxed{3}$ | 4 | 4 | 4 | 4 | |
| LU | 5 | 1 | 2 | $\boxed{2}$ | 2 | 4 | 4 | 4 | 4 | |
| LU | 5 | 1 | 2 | 3 | $\boxed{2}$ | 4 | 4 | 4 | 4 | |
| LU | 5 | 1 | 2 | 3 | 3 | $\boxed{4}$ | 4 | 4 | 4 | |
| LU | 5 | 1 | 2 | 3 | 3 | | $\boxed{4}$ | 4 | 4 | |
| LU | 5 | 1 | 2 | 3 | 3 | | | $\boxed{4}$ | 4 | |
| LU | 5 | 1 | 2 | 3 | 3 | | | | $\boxed{4}$ | |
| LU | 5 | 1 | 2 | 3 | 3 | | | | | $\boxed{\phantom{0}}$ |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LU | 5 | 1 | 2 | 3 | 3 | | | | $\boxed{\phantom{0}}$ | 4 | |
| LU | 5 | 1 | 2 | 3 | 3 | | | $\boxed{\phantom{0}}$ | 4 | 4 | |
| LU | 5 | 1 | 2 | 3 | 3 | | $\boxed{\phantom{0}}$ | 4 | 4 | 4 | |
| LU | 5 | 1 | 2 | 3 | 3 | $\boxed{\phantom{0}}$ | 4 | 4 | 4 | 4 | |
| LU | 5 | 1 | 2 | 3 | $\boxed{3}$ | 4 | 4 | 4 | 4 | 4 | |
| LU | 5 | 1 | 2 | $\boxed{3}$ | 2 | 4 | 4 | 4 | 4 | 4 | |
| LU | 5 | 1 | $\boxed{2}$ | 2 | 2 | 4 | 4 | 4 | 4 | 4 | |
| LU | 5 | 1 | 3 | $\boxed{2}$ | 2 | 4 | 4 | 4 | 4 | 4 | |
| LU | 5 | 1 | 3 | 3 | $\boxed{2}$ | 4 | 4 | 4 | 4 | 4 | |
| LU | 5 | 1 | 3 | 3 | 3 | $\boxed{4}$ | 4 | 4 | 4 | 4 | |
| LU | 5 | 1 | 3 | 3 | 3 | | $\boxed{4}$ | 4 | 4 | 4 | |
| LU | 5 | 1 | 3 | 3 | 3 | | | $\boxed{4}$ | 4 | 4 | |
| LU | 5 | 1 | 3 | 3 | 3 | | | | $\boxed{4}$ | 4 | |
| LU | 5 | 1 | 3 | 3 | 3 | | | | | $\boxed{4}$ | |
| LU | 5 | 1 | 3 | 3 | 3 | | | | | | $\boxed{\phantom{0}}$ |
| LU | 5 | 1 | 3 | 3 | 3 | | | | | $\boxed{\phantom{0}}$ | 4 |
| LU | 5 | 1 | 3 | 3 | 3 | | | | $\boxed{\phantom{0}}$ | 4 | 4 |
| LU | 5 | 1 | 3 | 3 | 3 | | | $\boxed{\phantom{0}}$ | 4 | 4 | 4 |
| LU | 5 | 1 | 3 | 3 | 3 | | $\boxed{\phantom{0}}$ | 4 | 4 | 4 | 4 |
| LU | 5 | 1 | 3 | 3 | 3 | $\boxed{\phantom{0}}$ | 4 | 4 | 4 | 4 | 4 |
| LU | 5 | 1 | 3 | 3 | $\boxed{3}$ | 4 | 4 | 4 | 4 | 4 | 4 |
| LU | 5 | 1 | 3 | $\boxed{3}$ | 2 | 4 | 4 | 4 | 4 | 4 | 4 |
| LU | 5 | 1 | $\boxed{3}$ | 2 | 2 | 4 | 4 | 4 | 4 | 4 | 4 |
| LU | 5 | $\boxed{1}$ | 2 | 2 | 2 | 4 | 4 | 4 | 4 | 4 | 4 |
| LU | $\boxed{5}$ | 4 | 2 | 2 | 2 | 4 | 4 | 4 | 4 | 4 | 4 |
| LD | | $\boxed{4}$ | 2 | 2 | 2 | 4 | 4 | 4 | 4 | 4 | 4 |

[1] Design 2013 by Alvy Ray Smith, protected under a Creative Commons copyright.

[2] Rogozhin, Yurii. Small universal Turing machines. *Theoretical Computer Science* 168(1996):215–240.