Tr

Haym Salomon @SalomonCrypto

Jul 3 · 22 tweets · SalomonCrypto/status/1543571618298880002

(1/21) Computer Science Fundamentals: Turing Machines and Turing Completeness.

Want to learn just how powerful @ethereum really is? You need to understand this core concept!

(2/21) It's July 2022, if you're still here reading this you already know how much potential @ethereum has; De-Fi is the first manifestation of @VitalikButerin's 2014 vision.

Programmable money is possible because Ethereum is the world computer.

Haym Salomon
@SalomonCrypto · **Follow**

(1/22) Let's talk about #blockchain. Having trouble telling grandma why you're selling her house for @VitalikButerin internet money? This is how I explain @ethereum to my friends IRL.

2:44 PM · May 7, 2022

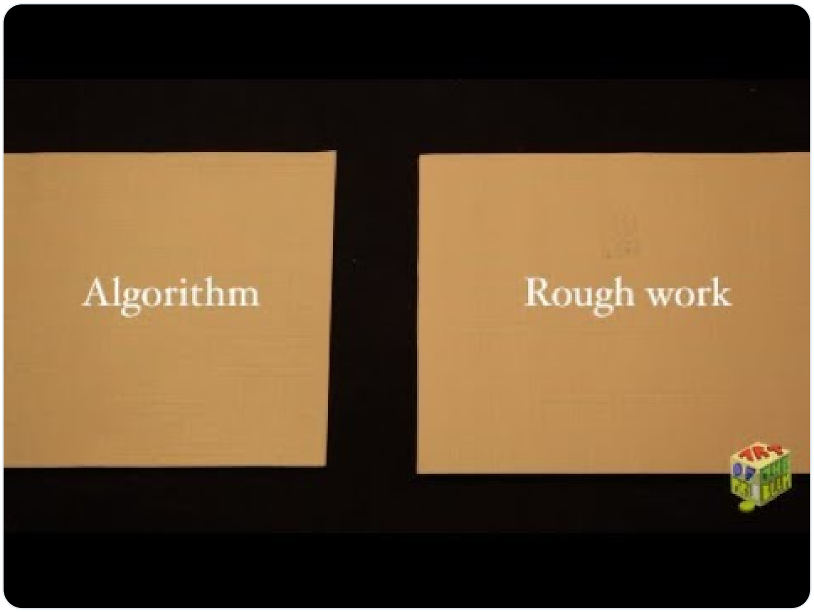
Read the full conversation on Twitter

23 Reply Copy link

Read 2 replies

(3/21) But to understand The World Computer you have to understand The Proto-Computer: The Turing Machine.

Invented by Alan Turing in the 1930s who named it an "a-machine," a Turing machine is an (abstract) model of computation.



https://www.youtube.com/embed/-ZS_zFg4w5k

(4/21) The concept is built around a simple hypothetical machine. It contains 3 key components:

- The tape, representing memory
- The head, keeping track of the machine position
- The instructions, telling the machine how to process the data

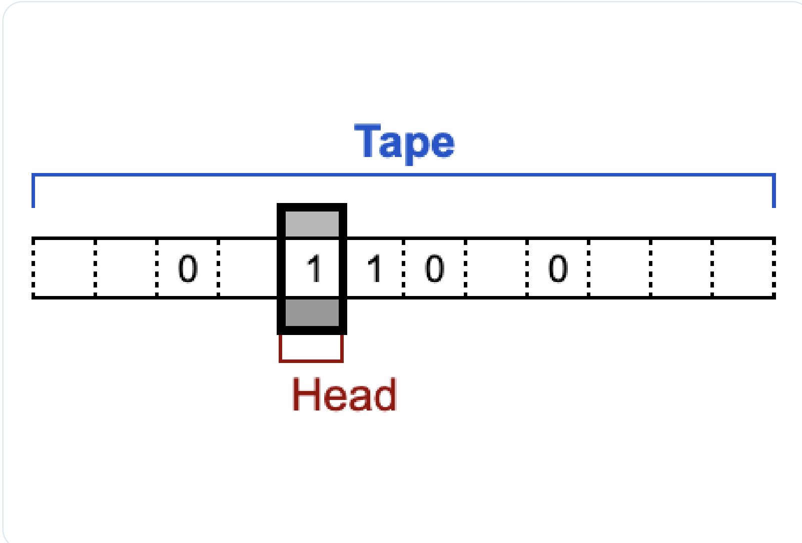
(5/21) The instructions are a set of conditions and associated operations.

There are 3 possible operations:
- Read the symbol under the head
- Edit (write and/or erase) the symbol under the head
- Move left or right one square

Let's look at an example.

(6/21) A Turing machine might look like this. While we are going to stick to 0s and 1s, a Turing machine can process any letters, numbers or symbols.

Next, let's create an instruction.

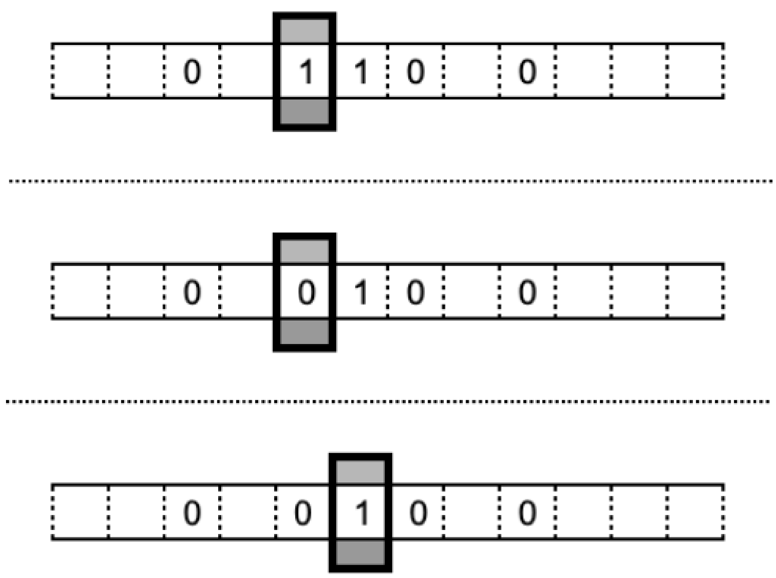


(7/21) Our goal is to switch the 1s to 0s and vice versa.

Step 1 - Read the tape and see if it is 1 or a 0
Step 2 - Write the new symbol
Step 3 - Move to the next number

Congratulations! We've created our first instruction.

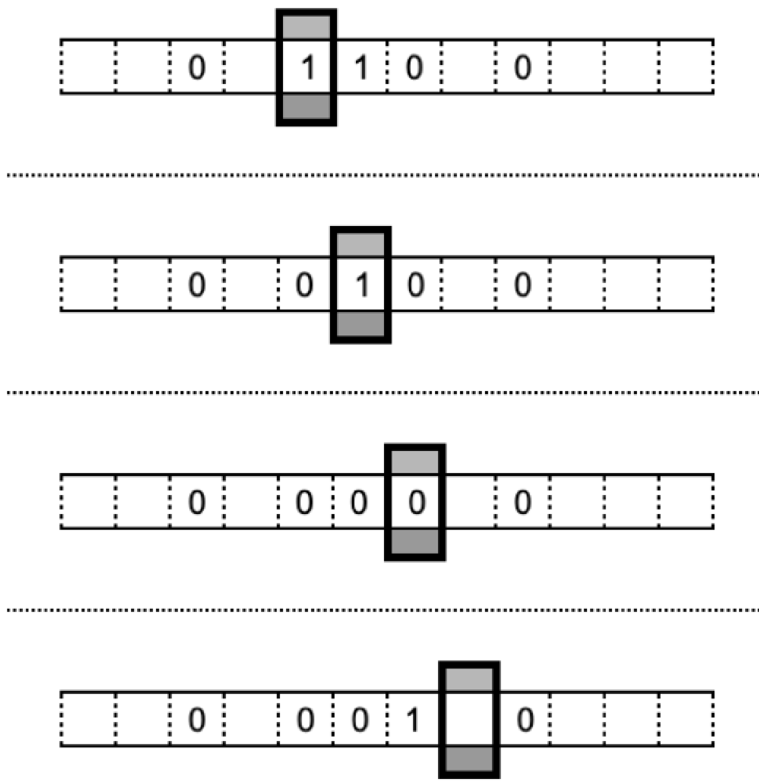
Read	Write	Move
(blank)	(blank)	no movement
1	0	head to the right
0	1	head to the right



(8/21) If we run our instruction two more times, we'll find ourselves at a blank space. Our Turing machine will follow our instruction and wait endlessly.

At this point we have to introduce the final concept required for a Turning machine: the machine state.

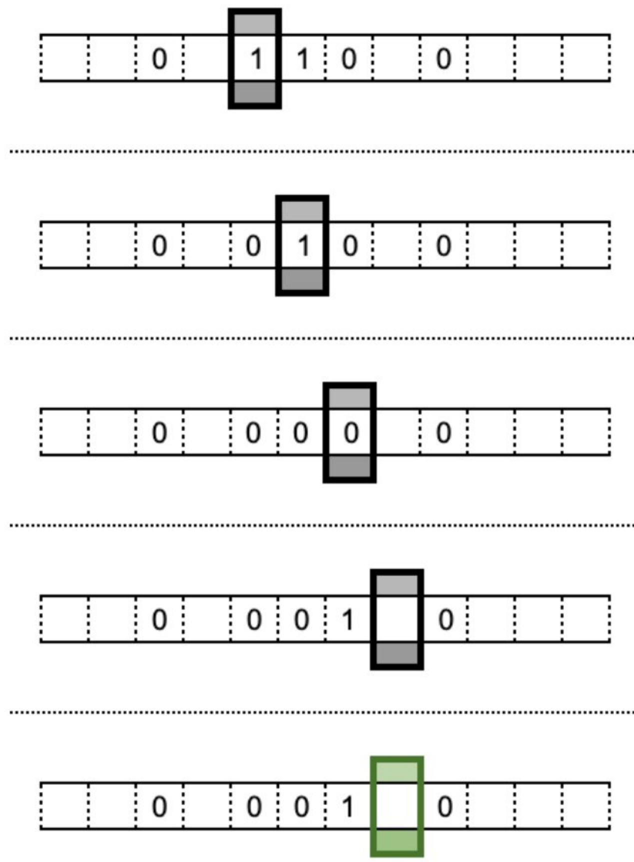
Read	Write	Move
(blank)	(blank)	no movement
1	0	head to the right
0	1	head to the right



(9/21) Now that the Turing machine can understand state, it can make decisions based on the context of the data on the tape.

In this example we introduced a state that instructs the machine to halt, but we can use the concept to begin to implement more complex functionality.

State	Read	Write	Move	Next State
0	(blank)	(blank)	no movement	END
0	1	0	head to the right	State 0
0	0	1	head to the right	State 0



(10/21) Consider this Turing Machine instruction set.

In state 0, the head moves right inverting each symbol. When it reaches an empty space, it will move to state 1.

In state 1, the head moves left and once again inverts the symbols. At an empty space the program ends.

State	Read	Write	Move	Next State
0	(blank)	(blank)	no movement	State 1
0	1	0	head to the right	State 0
0	0	1	head to the right	State 0
1	(blank)	(blank)	no movement	**END**
1	1	0	head to the left	State 1
1	0	1	head to the left	State 1

(11/21) This the end of our example, which is a little silly. All it does it invert the number back and forth, ending in the original state.

With the introduction of more states to our program, we can instruct the Turing machine to perform more complex functions.

(12/21) In fact, with enough states and a detailed enough instruction set, a Turing machine is capable of ANY computation a modern computer can do.

Take a look at your Macbook. If you can do something on that computer, you can (theoretically) do it on a Turing machine.

(13/21) In computability theory, Turing-completeness is a designation that describes a system with formal and specific requirements. For now we are going to keep things high level.

Let's use this definition: a system is Turing-Complete if it can simulate a Turing machine.

(14/21) Over the last century, Turing, Church and the giants who've built on their work have mathematically proven that if you can simulate a Turing machine on a computational system, you can also simulate that system on a Turing machine.

(15/21) Put simply, if a system is Turing-complete it is capable of anything that any other Turing-complete system can do.

Here are two pieces of information:
- most modern programming languages are Turing-complete
- Solidity is Turing-complete

Are things starting to click yet?

(16/21) Turing-completeness allows us to say with mathematical certainty that @ethereum is capable of anything that your computer can do.

Ethereum is the World Computer, crypto is programmable money, De-Fi is possible...

...because solidity is Turing-complete!

(17/21) I can already hear the peanut gallery:

"30 txns per second???"
"Have you seen gas fees???"
"Have you tried [other blockchain]???"

Go back and ask Dr. Turing what folks back then said about computing. I bet he'd have the same answer:

"We are still building."

(18/21) By their very nature, blockchains force developers to choose between decentralization, security and scalability.

But @ethereum has an answer; it will specialize in decentralization and security and offload execution to rollups.

Haym Salomon
@SalomonCrypto · **Follow**

(1/19) I believe that @Ethereum will be the only (sizable) settlement layer of the internet and of global finance.

I also believe in a multi-chain world.

Let's talk through Crypto-Economic Security, one of the reasons I think alt L1s will eventually become Ethereum L2s.

> **Haym Salomon** @SalomonCrypto
> Replying to @SalomonCrypto
> (20/22) However, I am willing to bet that, long term, Solana, Terra and Fantom will ultimate settle value on Ethereum. Just like @arbitrum, @loopringorg and @StarkWareLtd are doing from day one.

4:31 PM · May 25, 2022

Read the full conversation on Twitter

6 Reply Copy link

Read 1 reply

(19/21) So look, there's no way around it: we are still early. So early that it is easy to miss what is going on.

But missing what's going on would be a tragedy because something very special is happening in crypto.

A new type of computer is being born.

(20/21) As we march deeper into the 2020s, I'm sorry to say that the world is getting bleaker and bleaker.

Macro economics are breaking down, the old ways of working are failing, resources are not available where they are needed.

We need to find a new to coordinate.

(21/21) As I look to the future, I see a lot of reasons to be afraid. But no matter how dark things get, I still wake up every day, excited to log in.

Excited to build.

I believe in @ethereum.

Haym Salomon
@SalomonCrypto · **Follow**

I believe in @ethereum. I believe in DeFi. I believe in builders.

Interested in learning more? So will our grandkids. Follow along as I try to take notes for future historians:

bit.ly/SalomonTweets

5:26 PM · Jun 20, 2022

38 Reply Copy link

Read 2 replies

Like what you read? Help me spread the word by retweeting the thread (linked below).

Follow me for more explainers and as much alpha as I can possibly serve.

Haym Salomon
@SalomonCrypto · **Follow**

(1/21) Computer Science Fundamentals: Turing Machines and Turing Completeness.

Want to learn just how powerful @ethereum really is? You need to understand this core concept!

12:25 PM · Jul 3, 2022

Read the full conversation on Twitter

58 Reply Copy link

Read 1 reply

• • •