Theory of Computation

(Turing Machines)

Rezaul Chowdhury
(Slides by Pramod Ganapathi)
Department of Computer Science
State University of New York at Stony Brook
Fall 2021

Contents

Contents

The Accidental Birth of Computer Science! (Video)

Turing Machines

What was the aim of Alan Turing?

Turing's aim

Design a model that is:

- Simple,
- Intuitive,
- Generic, and
- Formalizes computation performed by a human mind

How does a human compute?

- Write input on a paper
- Do the computation (think and write the intermediate results on the paper)
- Write output on the paper

How did Turing formalize human computation?

- Turing = Named after Alan Mathison Turing
- Machine = Computing machine

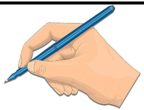
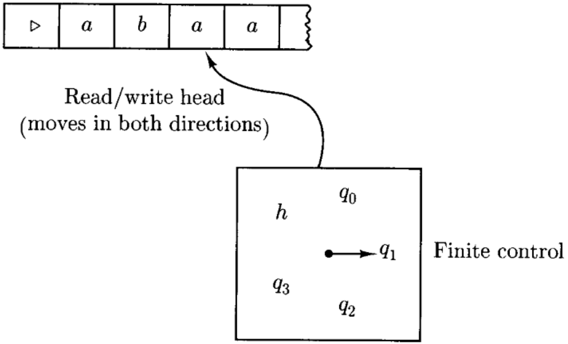
Human computation	Machine computation
	Tape
	Tape head
	Transition function

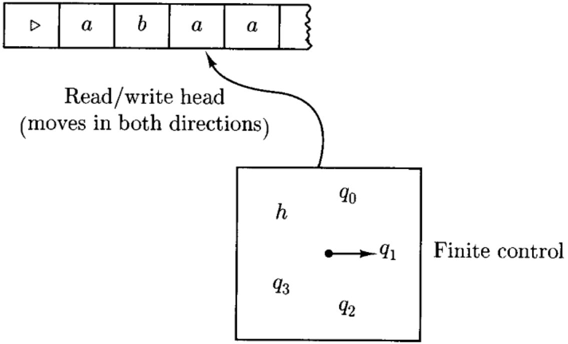
Diagram of a Turing machine (TM)



Source: Lewis and Papadimitriou. Elements of the Theory of Computation.

Operation	Explanation
Write	(Optionally) writes a new symbol at the current tape position.
Move	(Optionally) moves either left or right.
Think	(Optionally) changes to a new state.

Diagram of a Turing machine (TM)



Source: Lewis and Papadimitriou. Elements of the Theory of Computation.

Concept	Meaning
Tape	Simulates unlimited sheets of paper for computation.
Tape head	Read/write onto a tape cell. Moves left/right.
States	Simulates states of a human mind.
Input	Finite number of symbols initially on the tape.
Output	Finite number of symbols finally on the tape.
Computation	State transitions based on rules and input symbols.

What is a Turing machine (TM)?

Definition

A Turing machine (TM) M is a 6-tuple
$M = (Q, \Sigma, \Gamma, \delta, q_0, H)$, where,

1. Q: A finite set (set of states).
2. Σ: A finite set (input alphabet). Σ excludes $\triangleright, \Box, \leftarrow, \rightarrow$.
3. Γ: A finite set (tape alphabet).
   $\Sigma \cup \{\triangleright, \Box\} \subseteq \Gamma$. Γ excludes $\leftarrow, \rightarrow$.
4. $\delta : (Q - H) \times \Gamma$ to $Q \times (\Gamma \cup \{\leftarrow, \rightarrow\})$ is the transition function such that the tape head never falls off or erases $\triangleright$ symbol
   $\triangleright$ Time (computation)
5. q_0: The start state (belongs to Q).
6. $H = \{q_{\text{acc}}, q_{\text{rej}}\}$: The set of halting states (subset of Q).

Some notes on the Turing machine

Symbols

- $\triangleright$: Left end symbol
- $\square$: Blank symbol
- $\leftarrow, \rightarrow$: Left and right movement symbols
- Σ : Represents input/output/special symbols
- Γ : Represents symbols that can be present on the tape

Transition

- M never falls off the left end of the tape i.e.,
  when the current symbol is $\triangleright$, the tape head has to move right
- M stops when it reaches an accept or a reject state i.e.,
  δ is not defined for states in H

Construct TM to erase the input string

Problem

- Construct a TM to erase the input string

Construct TM to erase the input string

Problem

- Construct a TM to erase the input string

Solution

- Language recognizers such as DFA's cannot perform computational tasks such as erasing strings.
  So, no DFA can be used for erasing strings.
- Language generators such as CFG's cannot perform computational tasks such as erasing strings.
  So, no CFG can be used for erasing strings.
- TM's are more powerful than language recognizers and language generators.
  A TM can be used for erasing strings.

Construct TM to erase the input string

Problem

- Construct a TM to erase the input string

Solution (continued)

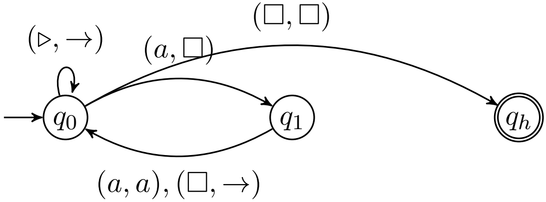
Time	State	Tape					
0	q_0	$\triangleright$	a	a	a	$\square$	$\cdots$
1	q_0	$\triangleright$	a	a	a	$\square$	$\cdots$
2	q_1	$\triangleright$	$\square$	a	a	$\square$	$\cdots$
3	q_0	$\triangleright$	$\square$	a	a	$\square$	$\cdots$
4	q_1	$\triangleright$	$\square$	$\square$	a	$\square$	$\cdots$
5	q_0	$\triangleright$	$\square$	$\square$	a	$\square$	$\cdots$
6	q_1	$\triangleright$	$\square$	$\square$	$\square$	$\square$	$\cdots$
7	q_0	$\triangleright$	$\square$	$\square$	$\square$	$\square$	$\cdots$
8	q_h	$\triangleright$	$\square$	$\square$	$\square$	$\square$	$\cdots$

Construct TM to erase the input string

Problem

- Construct a TM to erase the input string

Solution (continued)



	Current symbol (Γ)		
Current state $(Q - H)$	$\triangleright$	a	$\square$
q_0	$(q_0, \rightarrow)$	$(q_1, \square)$	$(q_h, \square)$
q_1	$-$	(q_0, a)	$(q_0, \rightarrow)$

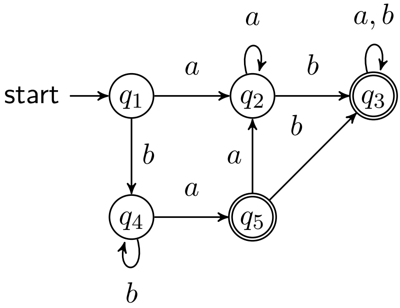
Construct TM for a regular language

Problem

- Construct a DFA that accepts all strings from the language $L = \{\text{strings containing } ab \text{ or end with } ba\}$

Solution

- Expression: $((a|b)^*ab(a|b)^*) \mid ((a|b)^*ba)$
- DFA:

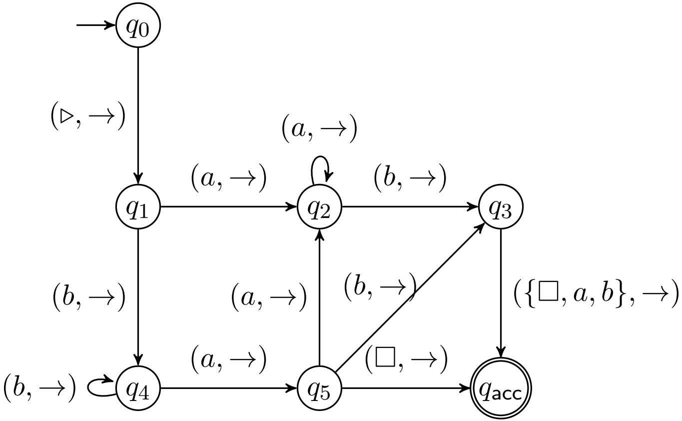


Construct TM for a regular language

Problem

- Construct a Turing machine that accepts all strings from the language $L = \{$strings containing ab or end with $ba\}$

Solution



Construct TM for a regular language

Problem

- Construct a Turing machine that accepts all strings from the language $L = \{\text{strings containing } ab \text{ or end with } ba\}$

Solution (continued)

	Current symbol (Γ)			
Current state ($Q - H$)	$\triangleright$	a	b	$\square$
q_0	$(q_1, \rightarrow)$	$-$	$-$	$-$
q_1	$-$	$(q_2, \rightarrow)$	$(q_4, \rightarrow)$	$-$
q_2	$-$	$(q_2, \rightarrow)$	$(q_3, \rightarrow)$	$-$
q_3	$-$	$(q_{\text{acc}}, \rightarrow)$	$(q_{\text{acc}}, \rightarrow)$	$(q_{\text{acc}}, \rightarrow)$
q_4	$-$	$(q_5, \rightarrow)$	$(q_4, \rightarrow)$	$-$
q_5	$-$	$(q_2, \rightarrow)$	$(q_3, \rightarrow)$	$(q_{\text{acc}}, \rightarrow)$

Construct TM for a regular language

Problem

- Construct a Turing machine that accepts all strings from the language $L = \{$strings containing ab or end with $ba\}$

Solution (continued)

- TM accepts the string bba because it enters the q_{acc} state

Time	State	Tape						
0	q_0	$\triangleright$	b	b	a	$\square$	$\square$	$\cdots$
1	q_1	$\triangleright$	b	b	a	$\square$	$\square$	$\cdots$
2	q_4	$\triangleright$	b	b	a	$\square$	$\square$	$\cdots$
3	q_4	$\triangleright$	b	b	a	$\square$	$\square$	$\cdots$
4	q_5	$\triangleright$	b	b	a	$\square$	$\square$	$\cdots$
5	q_{acc}	$\triangleright$	b	b	a	$\square$	$\square$	$\cdots$

Construct TM for a regular language

Problem

- Construct a Turing machine that accepts all strings from the language $L = \{$strings containing ab or end with $ba\}$

Solution (continued)

- TM rejects the string bbb because it enters the q_{rej} state

Time	State	Tape						
0	q_0	$\triangleright$	b	b	b	$\square$	$\square$	$\cdots$
1	q_1	$\triangleright$	b	b	b	$\square$	$\square$	$\cdots$
2	q_4	$\triangleright$	b	b	b	$\square$	$\square$	$\cdots$
3	q_4	$\triangleright$	b	b	b	$\square$	$\square$	$\cdots$
4	q_4	$\triangleright$	b	b	b	$\square$	$\square$	$\cdots$
5	q_{rej}	$\triangleright$	b	b	b	$\square$	$\square$	$\cdots$

Construct TM for a regular language

Problem

- Construct a Turing machine that accepts all strings from the language $L = \{$strings containing ab or end with $ba\}$

Solution (continued)

- TM accepts the string $aabbbbb$ because it enters the q_{acc} state
- Unlike DFA and CFG, a TM can accept a string without scanning the string completely

Time	State	Tape								
0	q_0	$\triangleright$	a	a	b	b	b	b	b	$\square$	$\cdots$
1	q_1	$\triangleright$	a	a	b	b	b	b	b	$\square$	$\cdots$
2	q_2	$\triangleright$	a	a	b	b	b	b	b	$\square$	$\cdots$
3	q_2	$\triangleright$	a	a	b	b	b	b	b	$\square$	$\cdots$
4	q_3	$\triangleright$	a	a	b	b	b	b	b	$\square$	$\cdots$
5	q_{acc}	$\triangleright$	a	a	b	b	b	b	b	$\square$	$\cdots$

Construct TM for a regular language

More problems

Use the TM to check acceptance of the following strings:

- ϵ
- aba $\triangleright$ contains ab and ends with ba
- aaa
- aab
- baa

The LEGO Turing Machine!
(Video)

TM's with different features

Problem

- Suppose you have TM's with the following characteristics.
  What are the computational powers of the TM's?

$\leftarrow$ movement	$\rightarrow$ movement	Write	Computational power
✗	✓	✗	**?**
✓	✓	✗	**?**
✗	✗	✓	**?**
✗	✓	✓	**?**
✓	✓	✓	TM

Construct TM for copying strings

Problem

- Construct a Turing machine that copies a string from the language $L = \Sigma^*$ where $\Sigma = \{a, b\}$.

Construct TM for copying strings

Problem

- Construct a Turing machine that copies a string from the language $L = \Sigma^*$ where $\Sigma = \{a, b\}$.

Solution

- Language recognizers such as DFA's cannot perform computational tasks such as copying strings.
  So, no DFA can be used for copying strings.
- Language generators such as CFG's cannot perform computational tasks such as copying strings.
  So, no CFG can be used for copying strings.
- TM's are more powerful than language recognizers and language generators.
  A TM can be used for copying strings.

Construct TM for copying strings

Solution (continued)

State	Tape									
q_0	$\triangleright$	b	b	a	$\square$	$\square$	$\square$	$\square$	$\square$	$\cdots$
q_2	$\triangleright$	b	b	a	$\#$	$\square$	$\square$	$\square$	$\square$	$\cdots$
q_2	$\triangleright$	b	b	a	$\#$	$\square$	$\square$	$\square$	$\square$	$\cdots$
q_5	$\triangleright$	2	b	a	$\#$	$\square$	$\square$	$\square$	$\square$	$\cdots$
q_6	$\triangleright$	2	b	a	$\#$	b	$\square$	$\square$	$\square$	$\cdots$
q_7	$\triangleright$	b	b	a	$\#$	b	$\square$	$\square$	$\square$	$\cdots$
q_5	$\triangleright$	b	2	a	$\#$	b	$\square$	$\square$	$\square$	$\cdots$
q_6	$\triangleright$	b	2	a	$\#$	b	b	$\square$	$\square$	$\cdots$
q_7	$\triangleright$	b	b	a	$\#$	b	b	$\square$	$\square$	$\cdots$
q_4	$\triangleright$	b	b	1	$\#$	b	b	$\square$	$\square$	$\cdots$
q_6	$\triangleright$	b	b	1	$\#$	b	b	a	$\square$	$\cdots$
q_7	$\triangleright$	b	b	a	$\#$	b	b	a	$\square$	$\cdots$
q_3	$\triangleright$	b	b	a	$\#$	b	b	a	$\square$	$\cdots$
q_{acc}	$\triangleright$	b	b	a	$\#$	b	b	a	$\square$	$\cdots$

Construct TM for copying strings

Problem

- Construct a Turing machine that copies a string from the language $L = \Sigma^*$ where $\Sigma = \{a, b\}$.

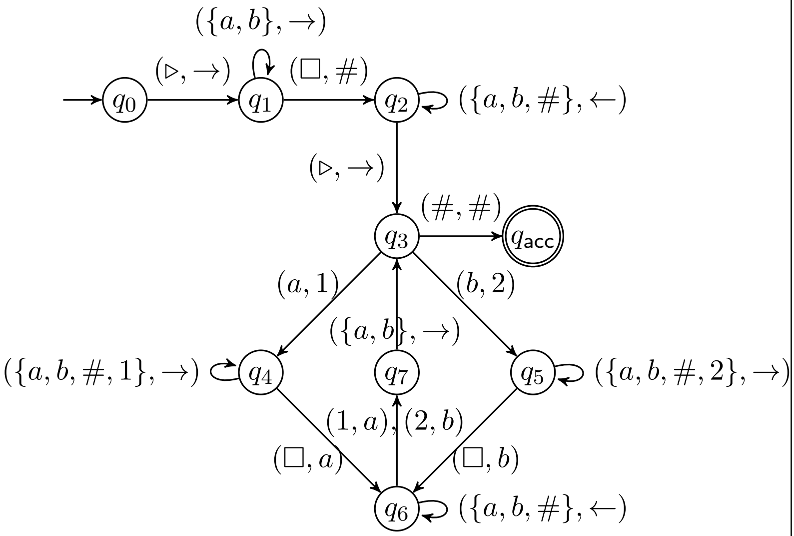
Solution (continued)

- $\Gamma = \Sigma \cup \{\triangleright, \square, \#, 1, 2\}$
- Cells with "$-$" means that the TM terminates in q_{rej} state

	Current symbol (Γ)						
State	$\triangleright$	a	b	$\#$	1	2	$\square$
q_0	$(q_1, \rightarrow)$	$-$	$-$	$-$	$-$	$-$	$-$
q_1	$-$	$(q_1, \rightarrow)$	$(q_1, \rightarrow)$	$-$	$-$	$-$	$(q_2, \#)$
q_2	$(q_3, \rightarrow)$	$(q_2, \leftarrow)$	$(q_2, \leftarrow)$	$(q_2, \leftarrow)$	$-$	$-$	$-$
q_3	$-$	$(q_4, 1)$	$(q_5, 2)$	$(q_{\text{acc}}, \#)$	$-$	$-$	$-$
q_4	$-$	$(q_4, \rightarrow)$	$(q_4, \rightarrow)$	$(q_4, \rightarrow)$	$(q_4, \rightarrow)$	$-$	(q_6, a)
q_5	$-$	$(q_5, \rightarrow)$	$(q_5, \rightarrow)$	$(q_5, \rightarrow)$	$-$	$(q_5, \rightarrow)$	(q_6, b)
q_6	$-$	$(q_6, \leftarrow)$	$(q_6, \leftarrow)$	$(q_6, \leftarrow)$	(q_7, a)	(q_7, b)	$-$
q_7	$-$	$(q_3, \rightarrow)$	$(q_3, \rightarrow)$	$-$	$-$	$-$	$-$

Construct TM for copying strings

Solution (continued)



Construct TM for copying strings

More problems

Use the TM to copy the following strings:

- ϵ
- a
- b
- aab

Construct TM for accepting $L = \{a^n b^n c^n \mid n \geq 1\}$

Problem

- Construct a Turing machine to accept all strings from the language $L = \{a^n b^n c^n \mid n \geq 1\}$

Construct TM for accepting $L = \{a^n b^n c^n \mid n \geq 1\}$

Problem

- Construct a Turing machine to accept all strings from the language $L = \{a^n b^n c^n \mid n \geq 1\}$

Solution

Language $A = \{abc, aabbcc, aaabbbccc, \ldots\}$

- No DFA can accept this language. $\triangleright$ Use pumping lemma
- No CFG can accept this language. $\triangleright$ Use pumping lemma
- A TM accepts this language.

Construct TM for accepting $L = \{a^n b^n c^n \mid n \geq 1\}$

Problem

- Construct a Turing machine to accept all strings from the language $L = \{a^n b^n c^n \mid n \geq 1\}$

Solution (continued)

State	Tape								
q_0	$\triangleright$	a	a	b	b	c	c	$\square$	$\cdots$
q_2	$\triangleright$	x	a	b	b	c	c	$\square$	$\cdots$
q_3	$\triangleright$	x	a	y	b	c	c	$\square$	$\cdots$
q_4	$\triangleright$	x	a	y	b	z	c	$\square$	$\cdots$
q_2	$\triangleright$	x	x	y	b	z	c	$\square$	$\cdots$
q_3	$\triangleright$	x	x	y	y	z	c	$\square$	$\cdots$
q_4	$\triangleright$	x	x	y	y	z	z	$\square$	$\cdots$
q_5	$\triangleright$	x	x	y	y	z	z	$\square$	$\cdots$
q_{acc}	$\triangleright$	x	x	y	y	z	z	$\square$	$\cdots$

Problem

- Construct a Turing machine to accept all strings from the language $L = \{a^n b^n c^n \mid n \geq 1\}$

Solution (continued)

- $\Gamma = \Sigma \cup \{\triangleright, \square, x, y, z\}$
- Cells with "$-$" means that the TM terminates in q_{rej} state

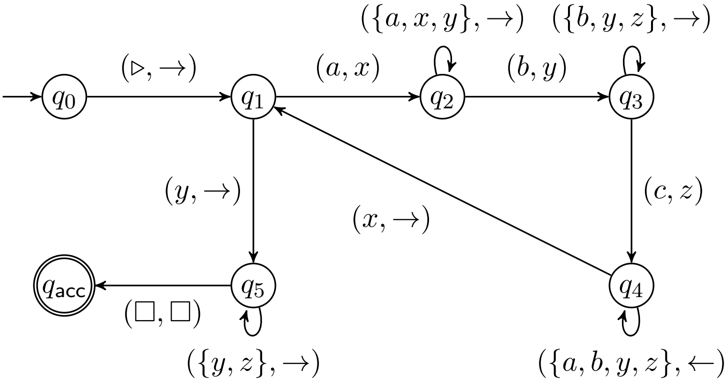
	Current symbol (Γ)							
St.	$\triangleright$	a	b	c	x	y	z	$\square$
q_0	$(q_1, \rightarrow)$	$-$	$-$	$-$	$-$	$-$	$-$	$-$
q_1	$-$	(q_2, x)	$-$	$-$	$-$	$(q_5, \rightarrow)$	$-$	$-$
q_2	$-$	$(q_2, \rightarrow)$	(q_3, y)	$-$	$(q_2, \rightarrow)$	$(q_2, \rightarrow)$	$-$	$-$
q_3	$-$	$-$	$(q_3, \rightarrow)$	(q_4, z)	$-$	$(q_3, \rightarrow)$	$(q_3, \rightarrow)$	$-$
q_4	$-$	$(q_4, \leftarrow)$	$(q_4, \leftarrow)$	$-$	$(q_1, \rightarrow)$	$(q_4, \leftarrow)$	$(q_4, \leftarrow)$	$-$
q_5	$-$	$-$	$-$	$-$	$-$	$(q_5, \rightarrow)$	$(q_5, \rightarrow)$	$(q_{\text{acc}}, \square)$

Construct TM for accepting $L = \{a^n b^n c^n \mid n \geq 1\}$

Problem

- Construct a Turing machine to accept all strings from the language $L = \{a^n b^n c^n \mid n \geq 1\}$

Solution (continued)



Construct TM for accepting $L = \{a^n b^n c^n \mid n \geq 1\}$

More problems

- Use the TM to check acceptance of the following strings: abc, aa, c, $abbc$, $aabc$, $abcc$, ϵ.
- Construct a TM that accepts language $L = \{a^n b^n c^n \mid n \geq 0\}$.

How are TM's different from DFA's and PDA's?

Feature	DFA	PDA	TM
Memory size	Finite	Infinite	Infinite
Halts?	✓	✓	✓, ✗
Input scanning	Left-to-right	Left-to-right	Arbitrary
#Passes	1 pass	1 pass	Any
Halting	End of input	End of input	Accept state
Computing power	Least	Medium	Highest
Language recognizer?	✓	✓	✓
Function calculator?	✗	✗	✓
Decide RL's?	✓	✓	✓
Decide CFL's?	✗	✓	✓
Decide REL's?	✗	✗	✓

What is a computation?

Example computation of a TM M

Time	Configuration	State	Tape						
0	C_0	q_0	$\triangleright$	b	b	b	$\square$	$\square$	$\cdots$
1	C_1	q_1	$\triangleright$	b	b	b	$\square$	$\square$	$\cdots$
2	C_2	q_4	$\triangleright$	b	b	b	$\square$	$\square$	$\cdots$
3	C_3	q_4	$\triangleright$	b	b	b	$\square$	$\square$	$\cdots$
4	C_4	q_4	$\triangleright$	b	b	b	$\square$	$\square$	$\cdots$
5	C_5	q_{rej}	$\triangleright$	b	b	b	$\square$	$\square$	$\cdots$

- Configurations: Information about the current state, tape head position, and the tape content. e.g.: $(q_4, \triangleright bbb\underline{\square})$
  $C_0, C_1, C_2, C_3, C_4, C_5$
- Starting, accepting, rejecting, halting configurations
- Computation: Sequence of successive configurations
  $C_0 \vdash_M C_1 \vdash_M C_2 \vdash_M C_3 \vdash_M C_4 \vdash_M C_5$
- Computation time/length: 5, written as $C_0 \vdash_M^5 C_5$
- Yields: e.g.: $C_1 \vdash_M^* C_4$

Let's think!

Deep problems

- How does a computer really work?
  Sequence of computer states, i.e., a computation.
- How does a human brain really work?
  The brain consists of, say, 86 billion neurons. Each neuron might temporarily store some information. Each neuron is connected to, say, 10,000 neurons. So, there might be 860 trillion neural connections. Suppose we represent the brain using a dynamic graph. Then, the human thinking and the human experience might just be sequences of configurations of neurons, i.e., a giant computation.
- How does the universe really work?
  The number of atoms in the observable universe is, say, 10^{80} in a higher dimensional space. The number of types of atoms might be finite. An atom moves from one point in space to another. Then, the happenings in the observe universe might just be a sequence of configurations, i.e., a gigantic computation.

Let's think!

Deep problems

- Suppose the workings of computers, human brains, and the observable universe is really a computation. What does that mean?

  We can simulate them on a real computer, if we have enough resources such as memory and processing power.
- What is the biggest assumption when we defined a Turing machine?

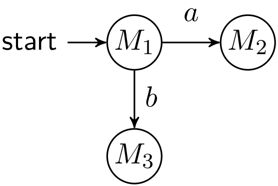
  Finiteness
- What happens when we move out of this assumption?

  ?

How to construct complicated TM's?

Constructing complicated TM's

- We can make complicated TM's from simpler TM's using the structure of a finite automaton
- Nodes of the automaton are the simpler TM's
- A connection $M_i \xrightarrow{k} M_j$ means when TM M_i halts and the current tape symbol is k, then TM M_j can start.



- Can you think of some examples of complicated TM's built from simpler TM's?

The Story of Information
(Video)
(try to watch the entire video from the start)

Universal Turing Machines

Universal Turing machine (UTM)

Definition

- A Universal Turing machine (UTM) MU can simulate the execution of any Turing machine M on any input w.

Working of UTM $U(\langle M, w \rangle)$

- Halt iff M halts on input w.
- If M is a deciding/semideciding machine, then
  - If M accepts, accept.
  - If M rejects, reject.
- If M computes a function, then $U(\langle M, w \rangle)$ must equal $M(w)$.

Simulating a TM

Theorem

- Given any Turing machine M and an input string w, there exists a Turing machine M' that simulates the execution of M on w and
  - Halts iff M halts on w, and
  - If it halts, returns whatever result M returns.

Universal machines

Problem

- We can construct a universal TM that accepts the language $L = \{\langle M, w\rangle \mid M \text{ is a TM and } w \in L(M)\}$
- Can we construct a universal DFA that accepts the language $L = \{\langle M, w\rangle \mid M \text{ is a DFA and } w \in L(M)\}$?
- Can we construct a universal CFG that accepts the language $L = \{\langle M, w\rangle \mid M \text{ is a CFG and } w \in L(M)\}$?

Universal machines

Problem

- We can construct a universal TM that accepts the language $L = \{\langle M, w\rangle \mid M \text{ is a TM and } w \in L(M)\}$
- Can we construct a universal DFA that accepts the language $L = \{\langle M, w\rangle \mid M \text{ is a DFA and } w \in L(M)\}$?
- Can we construct a universal CFG that accepts the language $L = \{\langle M, w\rangle \mid M \text{ is a CFG and } w \in L(M)\}$?

Solution

- No! $\triangleright$ Why?
- No! $\triangleright$ Why?

Compilers

Problem

- Suppose you discover a new programming language X. You want to write and compile your first program in X. Of course, a compiler for X is not available as it is a new language discovered by you. You know that you can use C++ or Java to write your compiler. But, is it possible to write your compiler in X that can be used to compile your program written in X?

Compilers

Problem

- Suppose you discover a new programming language X. You want to write and compile your first program in X. Of course, a compiler for X is not available as it is a new language discovered by you. You know that you can use C$++$ or Java to write your compiler. But, is it possible to write your compiler in X that can be used to compile your program written in X?

Solution

- Yes! $\triangleright$ How?

Universal Turing Machine in Minecraft!
(Video)

A Computer that Runs on Marbles!
(Video)