Machines

THE TIMES THEY ARE A-CHANGIN'

COME GATHER 'ROUND PEOPLE
WHEREVER YOU ROAM
AND ADMIT THAT THE WATERS
Goal:

- Complexity Theory classifies computational problems according to the amount of resources (say time) required.
- Revisit the computational model “Turing Machine”, this time discuss bounds on its resources and how robust they are.

Plan:

- Deterministic Turing machines
- Multi-tape Turing machines
- Non-deterministic Turing machines
- The Church-Turing hypothesis
- Complexity classes as bounded TMs.
Now there's an a!
Syntactically, a **TM** consists of the following objects:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q$</td>
<td>finite set of states</td>
</tr>
<tr>
<td>$\Sigma$</td>
<td>input alphabet: a finite set</td>
</tr>
<tr>
<td>$\Gamma$</td>
<td>tape alphabet</td>
</tr>
<tr>
<td>$\delta$</td>
<td>$\delta:Q \times \Gamma \rightarrow Q \times \Gamma \times {L,R}$ - the transition function</td>
</tr>
<tr>
<td>$q_0$</td>
<td>start state</td>
</tr>
<tr>
<td>$q_{acc}$</td>
<td>$\varepsilon Q$ accept state</td>
</tr>
<tr>
<td>$q_{rej}$</td>
<td>$\varepsilon Q$ reject state</td>
</tr>
</tbody>
</table>

- Excluding "_"
- $\Sigma \subseteq \Gamma$ and 
- $\_ \in \Gamma$
- $q_{reject} \neq q_{accept}$
Transition

$\delta(q_8, b) = (q_6, a, R)$

Now there's an $a$!
Initial Configuration
- For input “abaabaab”

Head
- On leftmost cell

State
- starting

Input - from left
• The head cannot move to the left of the leftmost square
How many distinct configurations may a Turing machine that uses $N$ cells have?

- The content of the tape
- The position of the head
- The machine's state

$|\Gamma|^N \times N \times |Q|$
\[ L = \{a^nb^nc^n \mid n \geq 0\} \]

**Examples:**
- Member of \( L \): \( aaabbbccc \)
- Non-Member of \( L \): \( aaabbeccc \)

\[ Q = \{q_0, q_1, q_2, q_3, q_4, q_{\text{accept}}, q_{\text{reject}}\} \]

\[ \Sigma = \{a, b, c\} \]

\[ \Gamma = \{a, b, c, \_, X, Y, Z\} \]

specified next...

- the start state.

\( q_0 \in Q \) - the accept state.

\( q_{\text{acc}} \in Q \) - the reject state.
The Transitions Function

transitions not specified here yield

q_{reject}

Complex
Equivalence between Types of TM

General:

- Deterministic TMs are extremely powerful
- Ignoring polynomial blow-up in time/space, they are equivalent to many other models

Next:

- Let us consider one such model in particular: Multi-Tape TM.
Multi-Tape Turing Machines

At start: input on first tape
Multi-Tape TMs

- **Q**: finite set of states
- **Σ**: input alphabet: a finite set
- **Γ**: tape alphabet
- **δ**: $\delta: Q \times \Gamma^k \rightarrow Q \times (\Gamma \times \{L, R\})^k$ - the transition function
- **q₀**: start state
- **q_{acc}**: $\varepsilon Q$ accept state
- **q_{rej}**: $\varepsilon Q$ reject state

- $\Sigma \subseteq \Gamma$ and $\_ \in \Gamma$
- $k$ - the number of tapes - is some constant
- $q_{reject} \neq q_{accept}$
The Church-Turing Hypothesis

**Theorem:**

- Multi-tape machines are polynomially equivalent to single-tape machines.

**Hypothesis:**

- We can state a much stronger claim concerning the robustness of the Turing machine model:

Intuitive notion of algorithm

=

Turing machine
Next:

• Let us now consider a non-realistic computational model: NONDETERMIONISTIC

Which:

• can be simulated by DTMs
• However, with an exponential blowup in time.
Non-deterministic Turing Machines

- Finite set of states
- Input alphabet: a finite set
- Tape alphabet
- Start state
- Accept state
- Reject state

\[ Q \times \Sigma \rightarrow P(Q \times \Gamma \times \{L,R\}) \text{ - transition function} \]

- \[ \Sigma \subseteq \Gamma \text{ and } _\_ \in \Gamma \]

- Start state
- Accept state
- Reject state

\[ q_{\text{reject}} \neq q_{\text{accept}} \]
Deterministic vs. Nondeterministic

accepts if any computation accepts

Non-deterministic computation tree

Deterministic computation
Witness Verification Program

Nondet. TM

magically guess which transitions to take to eventually accept if possible

A verifier

Verifies a witness to the fact that $x$ is in $L$
Traverse from s to t

A prime factorization

Isomorphism

Is it a path from s to t?

Are primes whose product = N

Does \( \pi \) transform \( G \) into \( G' \)?
Non-deterministic $\rightarrow$ Deterministic

Given an NDT $M$, construct a 3-tapes DTM, $M'$ that accepts $L(M)$.
Let number of transition $\leq h$
Nondeterministic time $\leq t(n)$

1. Write 0 on the guide tape

2. Copy the input to the simulation tape

3. Simulate $M$: choose each transition by the corresponding digit on the guide tape (if valid)

4. Accept if $M$ accepts

5. Add 1 to the number on the guide tape (in base $h$)
   If reached $h^{t(n)} + 1$ – reject

6. Go to step 2
Complexity Classes

With a model of computation established: TM

various bounds on resources

required to compute a problem

define Complexity classes
Definition:

- Let $t : \mathbb{N} \rightarrow \mathbb{N}$ be a complexity function

Deterministic time:

$\text{TIME}\ [t(n)] \equiv \{ L \mid L \text{ decided by } O(t(n))\text{-time deterministic TM} \}$

Nondeterministic time:

$\text{NTIME}\ [t(n)] \equiv \{ L \mid L \text{ decided by } O(t(n))\text{-time non deterministic TM} \}$

Det. Polynomial time:

$P \equiv \bigcup_k \text{TIME} [n^k]$

Nondet. Polynomial time:

$NP \equiv \bigcup_k \text{NTIME} [n^k]$

Det Exponential time:

$EXP \equiv \bigcup_k \text{TIME} \left[ e^{n^k} \right]$
Definition:

- Let $t: \mathbb{N} \rightarrow \mathbb{N}$ be a complexity function

**Deterministic space:**

$SPACE \quad [t(n)] \equiv \{ L \mid L \text{ decided by } O(t(n)) \text{- space deterministic TMs} \}$

**Nondeterministic space:**

$NSPACE \quad [t(n)] \equiv \{ L \mid L \text{ decided by } O(t(n)) \text{- space nondeterministic TMs} \}$

**Det. Log space:**

$L = SPACE \quad [\log(n)]$

**Nondet. Log space:**

$NL = NSPACE \quad [\log(n)]$

**Det polynomial space:**

$PSPACE = \bigcup_k SPACE \quad [n^k]$
• a deterministic run that halts must avoid repeating a configuration
• its running time is bounded from above by the number of configurations the machine has
• which, for a PSPACE machine, is exponential
Name the Class

Halting Problem

Minimum Spanning Tree

Seating: Hamiltonian Path

Tour: Hamiltonian Cycle

anbncn

EXP

PSPACE

NP

P

NL

L
Halting Problem is Undecidable

C: duplicate input and call A on copies; on “yes”, infinite loop --- on “no”, halt

A: does B halt on x?

Run C on (the representation of) C ⇒ contradiction
Theorem:

• $P \neq \text{EXPTIME}$

Proof:

• We construct a language $L \in \text{EXPTIME}$, which, however, is not accepted by any $\text{TM}$ running in polynomial time:

$$L \cong \{ x \mid x = \langle M \rangle \#1^c \#1^c \# , M \text{ doesn't accept } x \text{ within } c|x|^c \text{ time} \}$$
**Lemma:**

- $L \subseteq \text{EXPTIME}$

**Proof:**

- In particular, $L$ can be decided in time $|x| \cdot |x|^{|x|}$

---

**Lemma:**

- $L \not\subseteq P$

**Proof:**

- Assume a TM $M$ that accepts $x \in L$ in time $c|x|^e \Rightarrow$ run it on the string $\langle M \rangle #1^c #1^e#$ $\Rightarrow$ contradiction
P, NP and co-NP

\[ \text{Co-NP} = \{ \Sigma^* - L \mid L \in \text{NP} \} \]
presented two computational models:
1. deterministic Turing machines
2. non-deterministic Turing machines.

simulated NTM by DTM
with an exponential blowup in time.

The Church-Turing hypothesis:
Deterministic TMs equivalent to our intuitive notion of algorithms.

From now on: use pseudo-code instead of TMs
Defined complexity classes via bounds on TMs:

- **P**
  - Polynomial time

- **NP**
  - Nondeterministic Poly time

- **coNP**
  - Complement of NP

- **EXPTIME**
  - Exponential time

- **L**
  - Logarithmic space

- **NL**
  - Nondet. Log space

- **PSPACE**
  - Polynomial Space
<table>
<thead>
<tr>
<th>Turing Machine</th>
<th>Church-Turing Hypothesis</th>
<th>Complexity Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Halting Problem</td>
<td>Non Deterministic TM</td>
<td></td>
</tr>
<tr>
<td>Complexity Classes</td>
<td>NP</td>
<td>co-NP</td>
</tr>
<tr>
<td></td>
<td>P</td>
<td>L</td>
</tr>
<tr>
<td></td>
<td>EXPTIME</td>
<td>PSPACE</td>
</tr>
</tbody>
</table>

- Cantor, Georg
- Hilbert, David
- Gödel, Kurt
- Turing, Alan
- Church, Alonzo