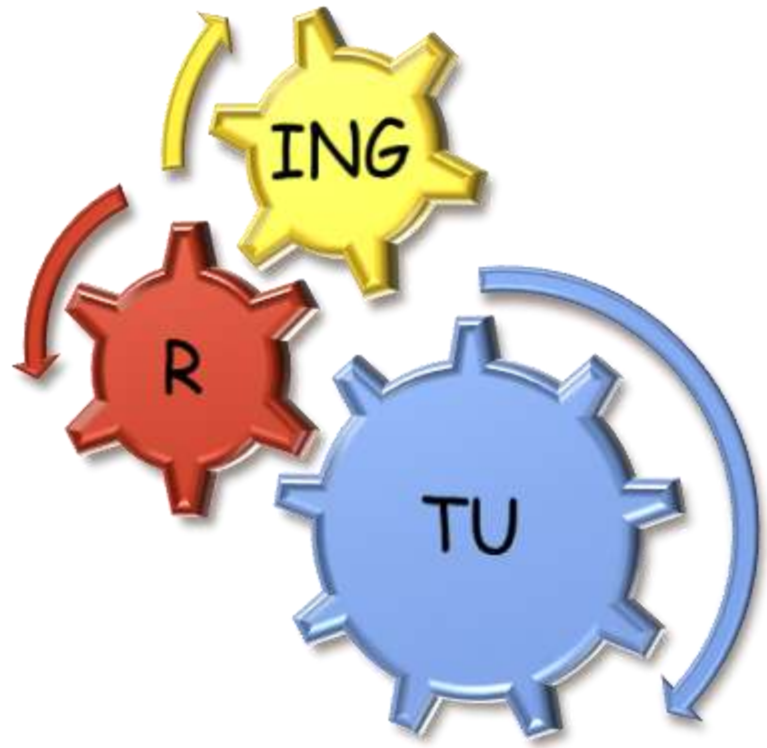


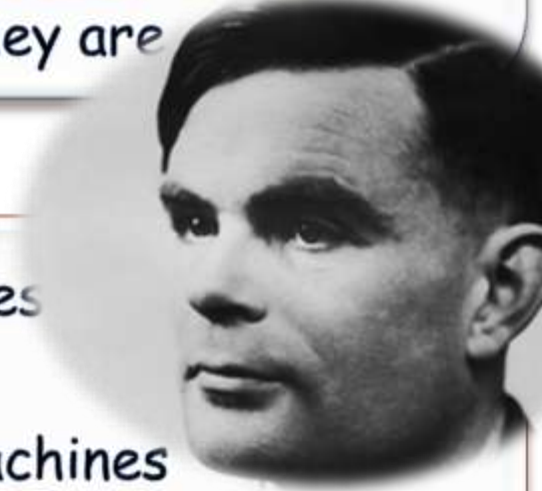
Machines

THE TIMES THEY ARE A - CHANGIN'

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

Goal:

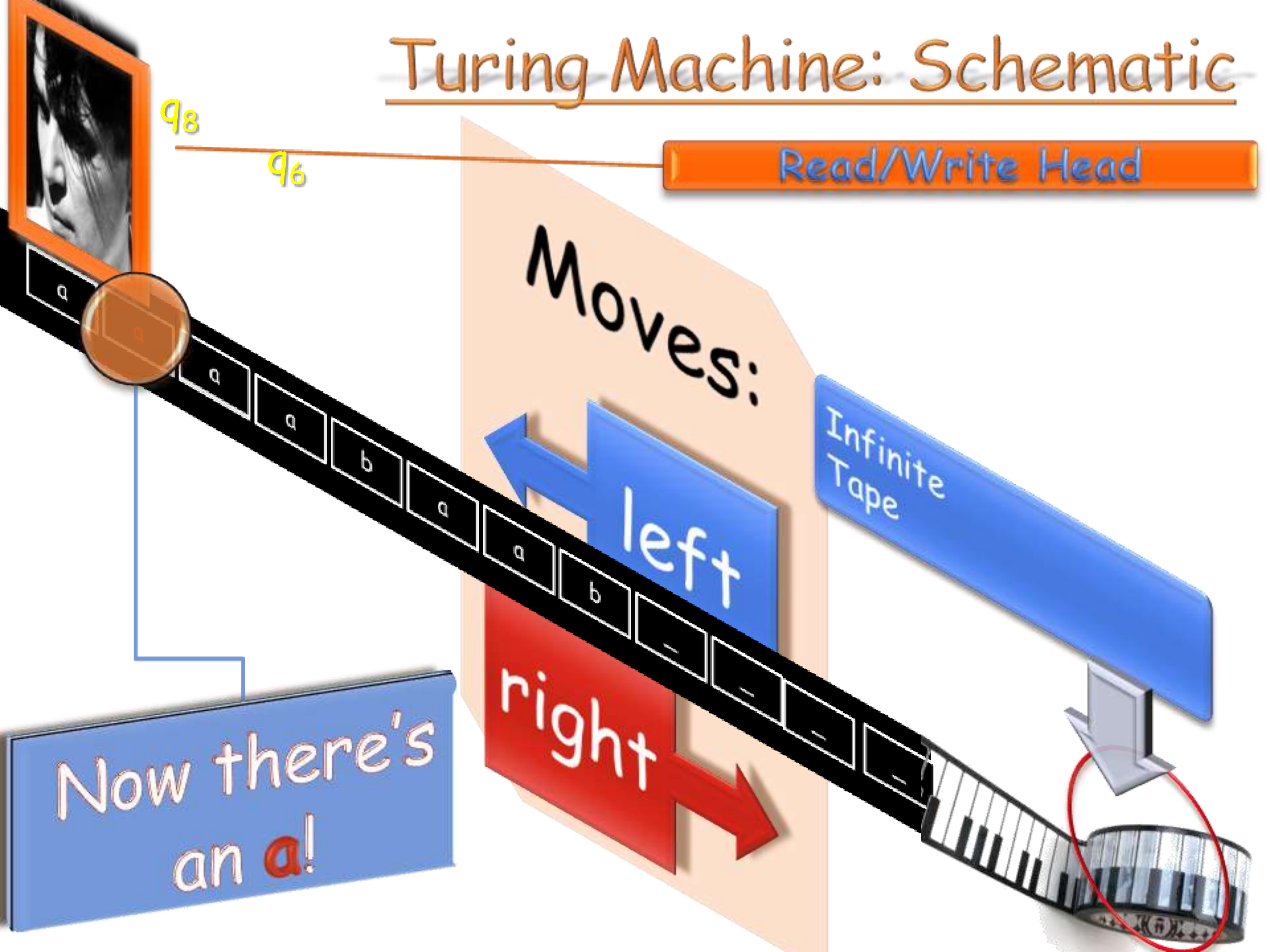
- Complexity Theory classifies computational problems according to the amount of resources (say time) required
- Revisit the computational model "Turing Machine", this time discuss bounds on its resources and how robust they are



Plan:

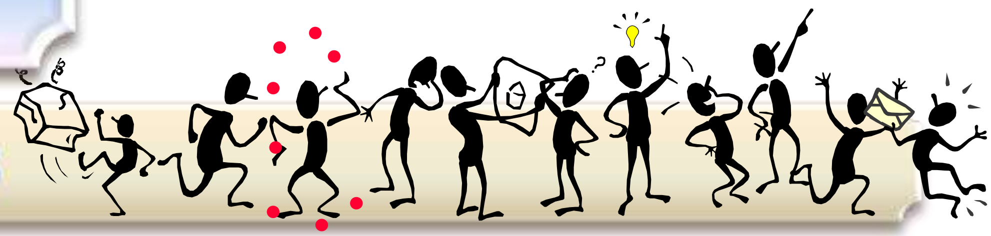
- Deterministic Turing machines
- Multi-tape Turing machines
- Non-deterministic Turing machines
- The Church-Turing hypothesis
- Complexity classes as bounded TMs.

Turing Machine: Schematic



TM: Formally

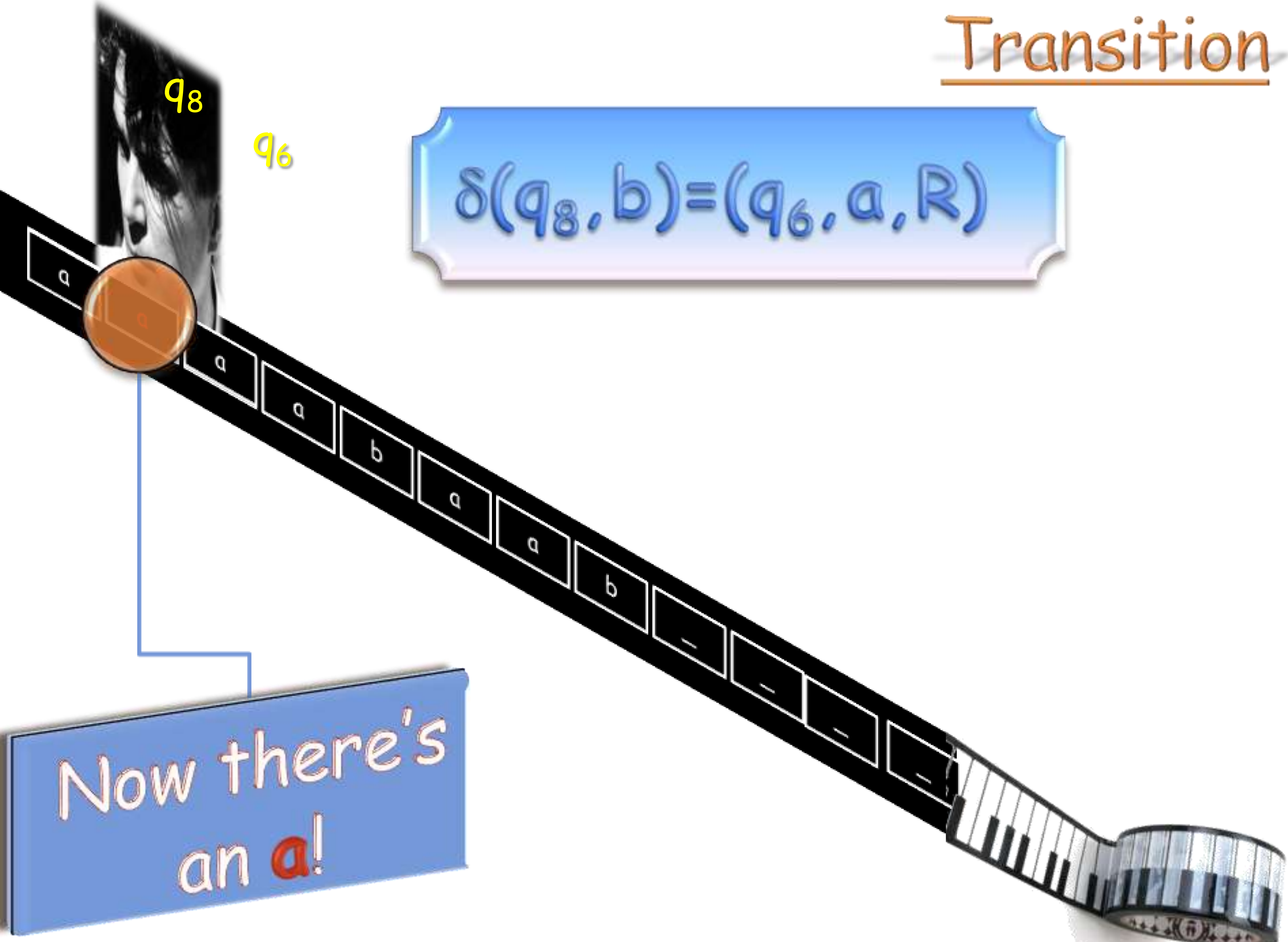
Syntactically, a TM consists of the following objects:



Q	finite set of states
Σ	input alphabet: a finite set — Excluding "_"
Γ	tape alphabet — $\Sigma\subseteq\Gamma$ and $\_\in\Gamma$
δ	$\delta:Q\times\Gamma\rightarrow Q\times\Gamma\times\{L,R\}$ - the transition function
q_0	start state
q_{acc}	$\in Q$ accept state
q_{rej}	$\in Q$ reject state — $q_{reject}\neq q_{accept}$

Transition

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



Computations

Initial Configuration

- For input "abaabaab"

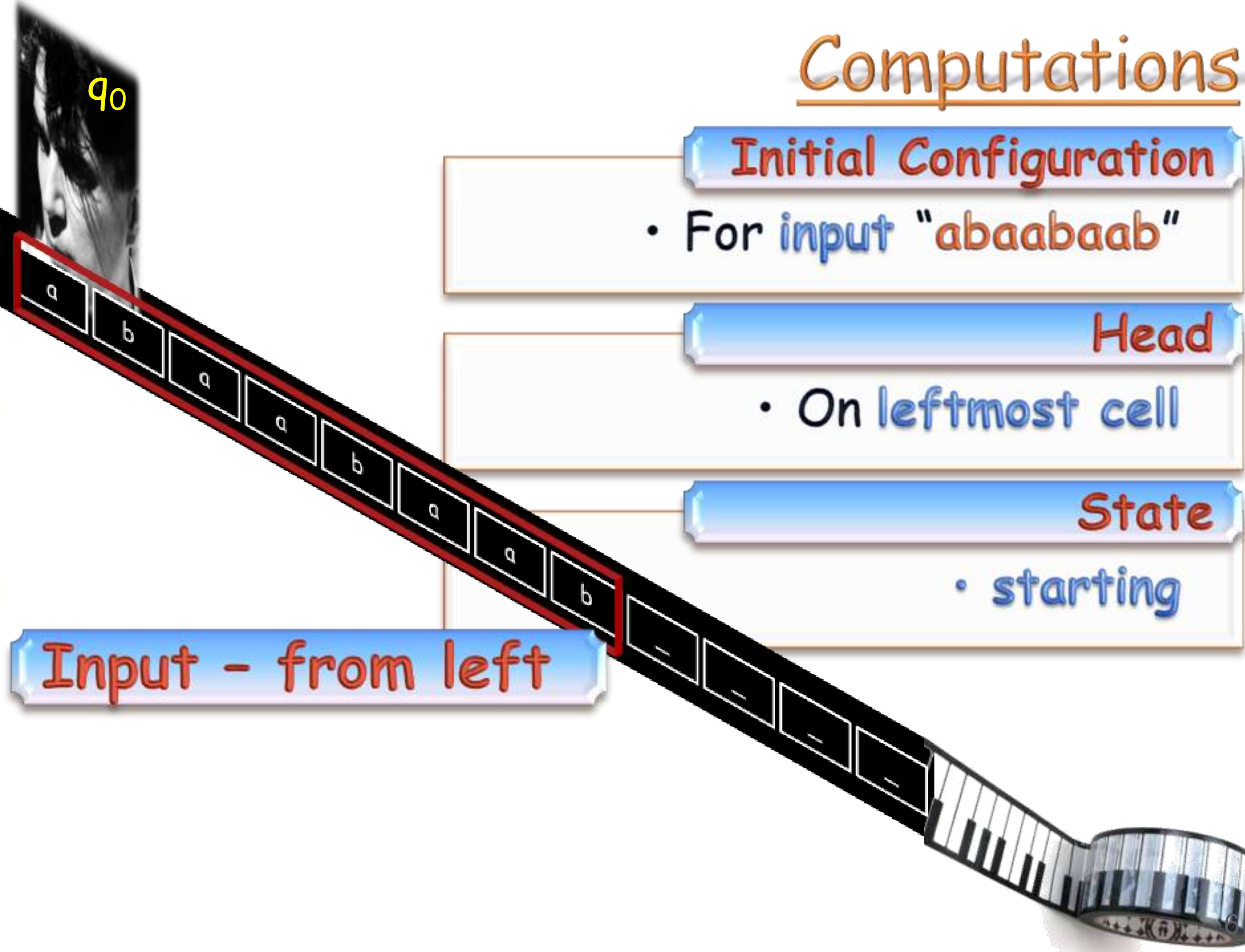
Head

- On leftmost cell

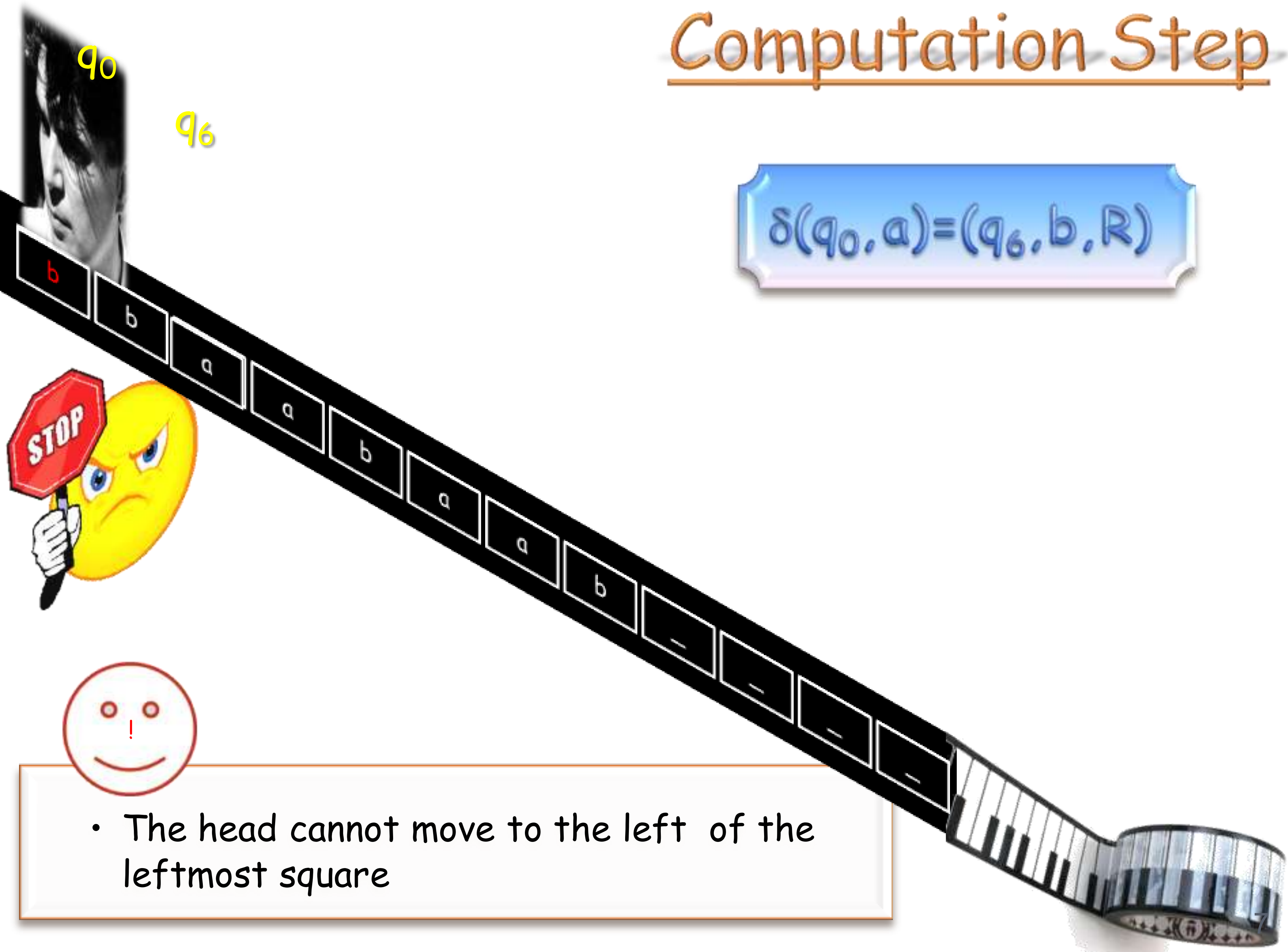
State

- starting

Input - from left



Computation Step



$$\delta(q_0, a) = (q_6, b, R)$$

- The head cannot move to the left of the leftmost square

Configurations

How many distinct configurations may a Turing machine that uses N cells have?

- the **content** of the **tape** → $|\Gamma|^N$
- the **position** of the **head** → N
- the **machine's state** → $|Q|$

$$|\Gamma|^N \times N \times |Q|$$

My first TM

$L=\{a^nb^nc^n \mid n\geq 0\}$

Examples:

Member of L: aaabbbccc

Non-Member of L: aaabbcccc

Q $= \{q_0, q_1, q_2, q_3, q_4, q_{accept}, q_{reject}\}$

Σ $= \{a,b,c\}$

Γ $= \{a,b,c,\_,X,Y,Z\}$

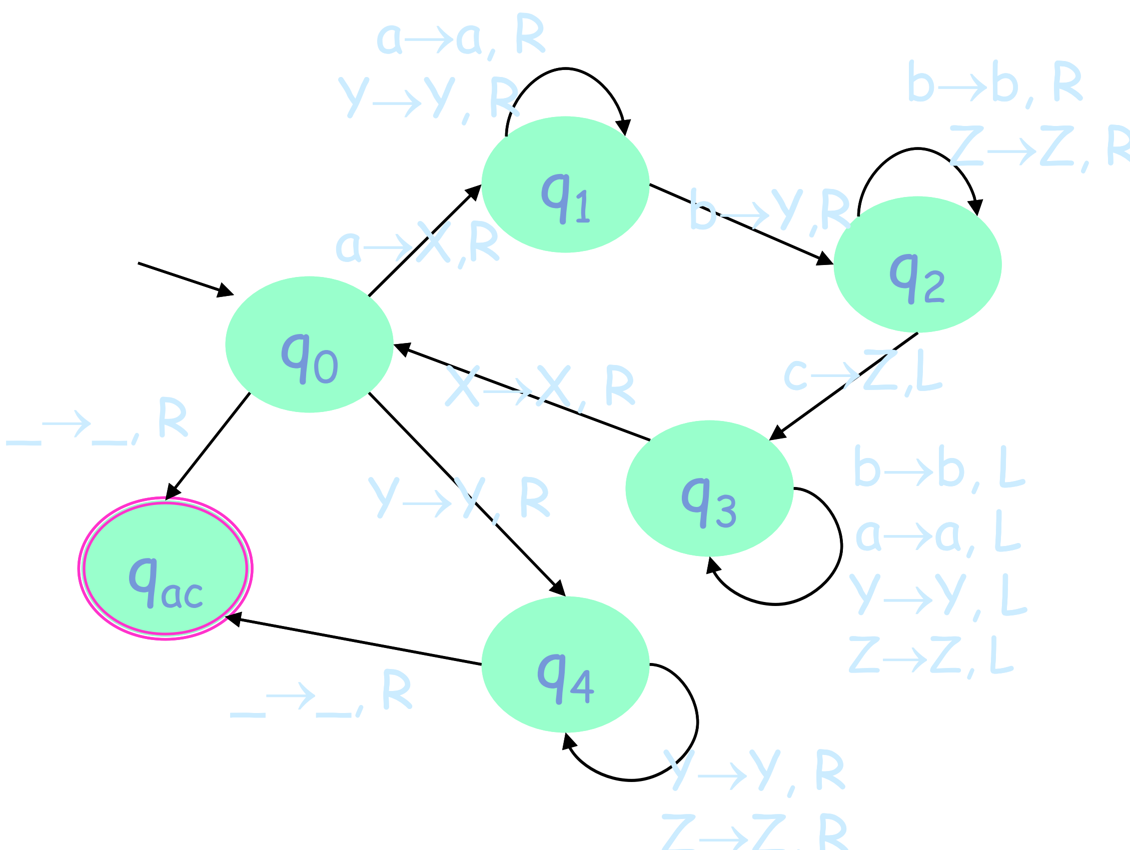
δ specified next...

q_0 - the start state.

q_{acc} $\in Q$ - the accept state.

q_{rej} $\in Q$ - the reject state.

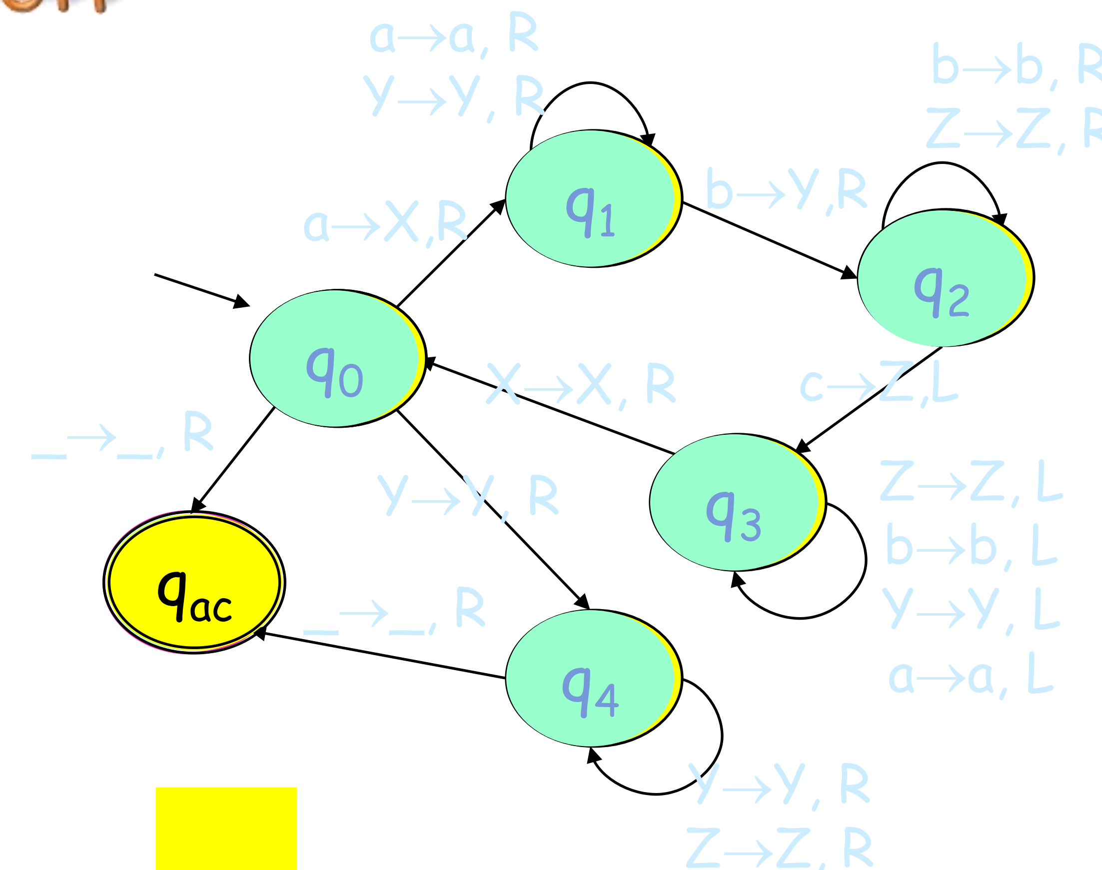
The Transitions Function

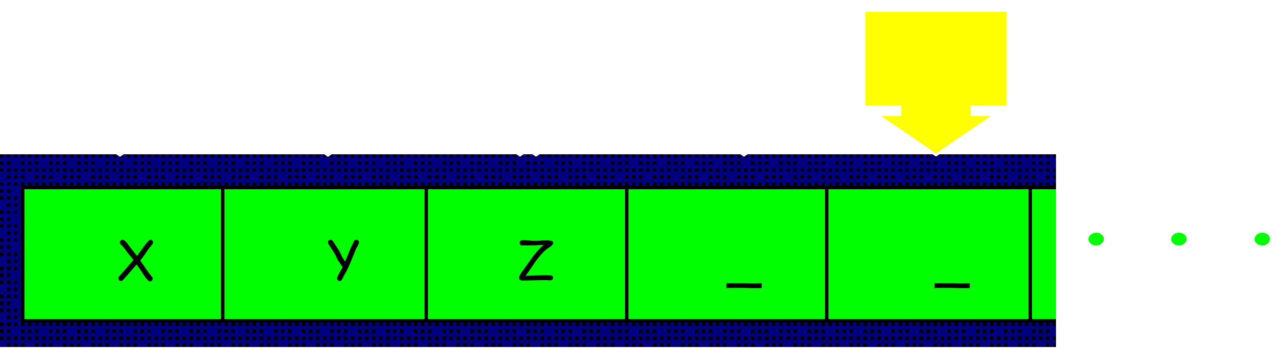


transitions not specified here yield q_{reject}

Complex

Demonstration





Equivalence between Types of TM

General:

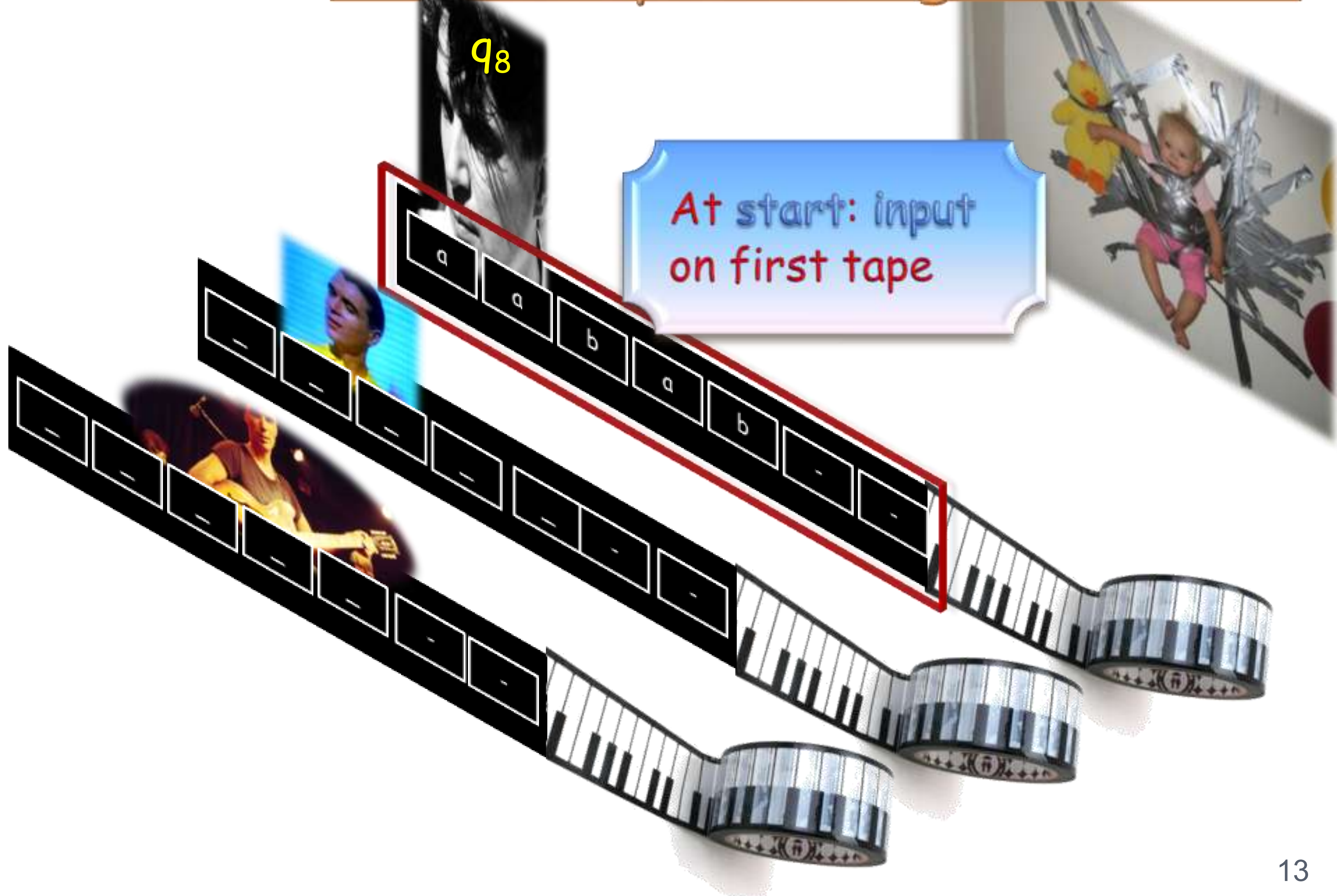
- Deterministic TMs are extremely powerful
- Ignoring polynomial blow-up in time/space, they are equivalent to many other models

Next

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

SIP 136-138

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	$\delta: Q\times\Gamma^k \to Q\times(\Gamma\times\{L,R\})^k$ – the transition …
q_0	start state
q_{acc}	$\in Q$ accept state
q_{rej}	$\in Q$ reject state

Excluding "_"

$\Sigma\subseteq\Gamma$ 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:

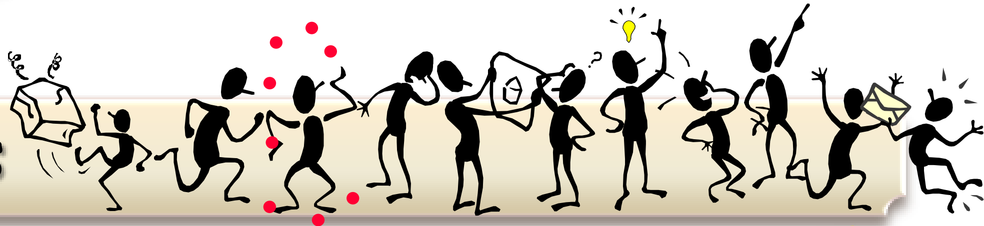
- Let us now consider a **non realistic** computational model: ***NONDETERMIONISTIC***

Which:

- can be simulated by **DTM**s
- However, with an **exponential blowup in time**.

Non-deterministic Turing Machines

Q	finite set of states
Σ	input alphabet: a finite set
Γ	tape alphabet
δ	$\delta: Q\times\Gamma \rightarrow P(Q\times\Gamma\times\{L,R\})$ - transition f…
q_0	start state
Q_{acc}	accept state
Q_{rej}	reject state



Excluding "_"

$\Sigma\subseteq\Gamma$ and $\_\in\Gamma$

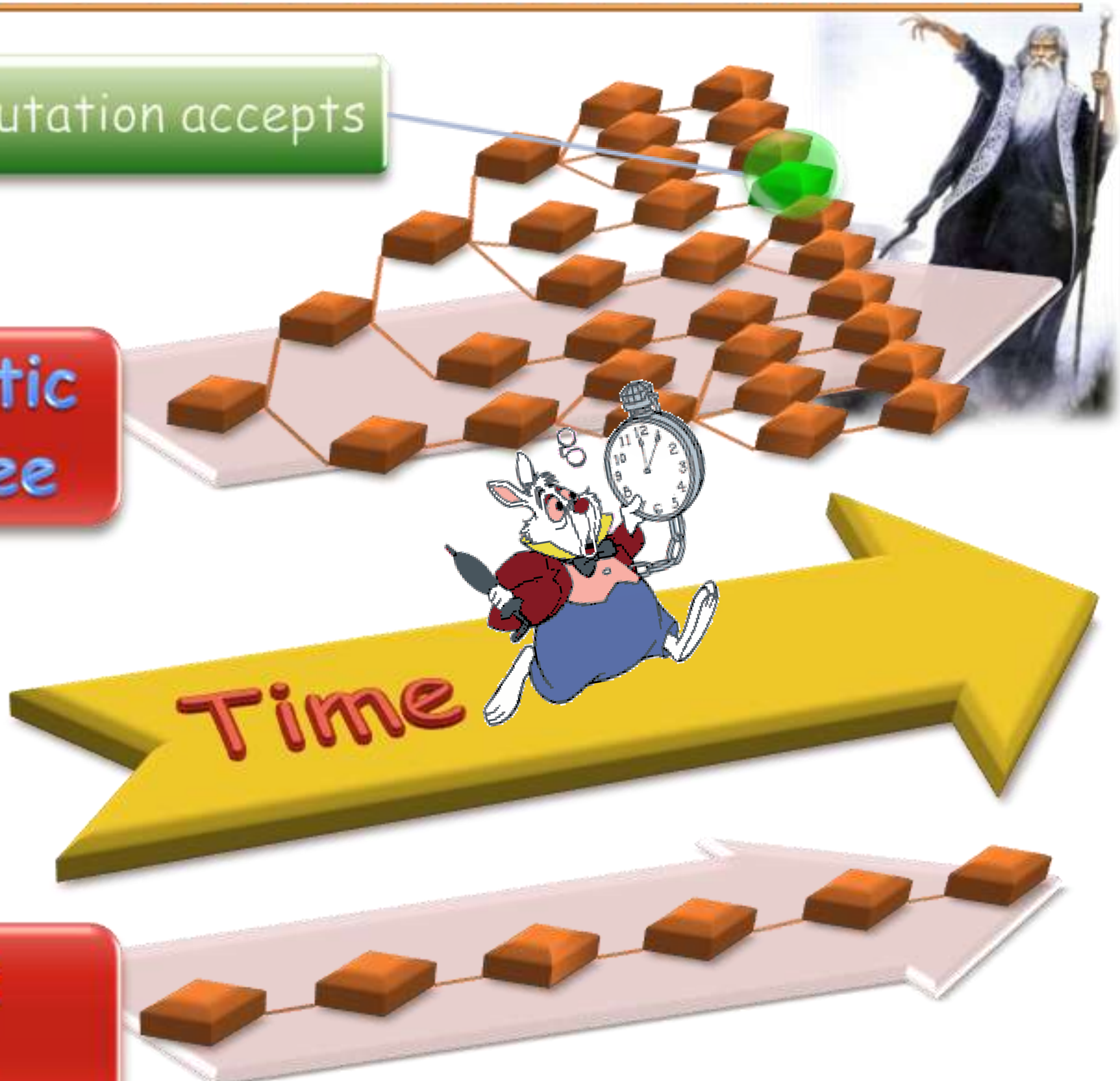
power set
$P(A)=\{B \mid B\subseteq A\}$

$q_{reject}\neq q_{accept}$

Deterministic vs. Nondeterministic

accepts if <u>any</u> 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

Nodeterministic

Guess

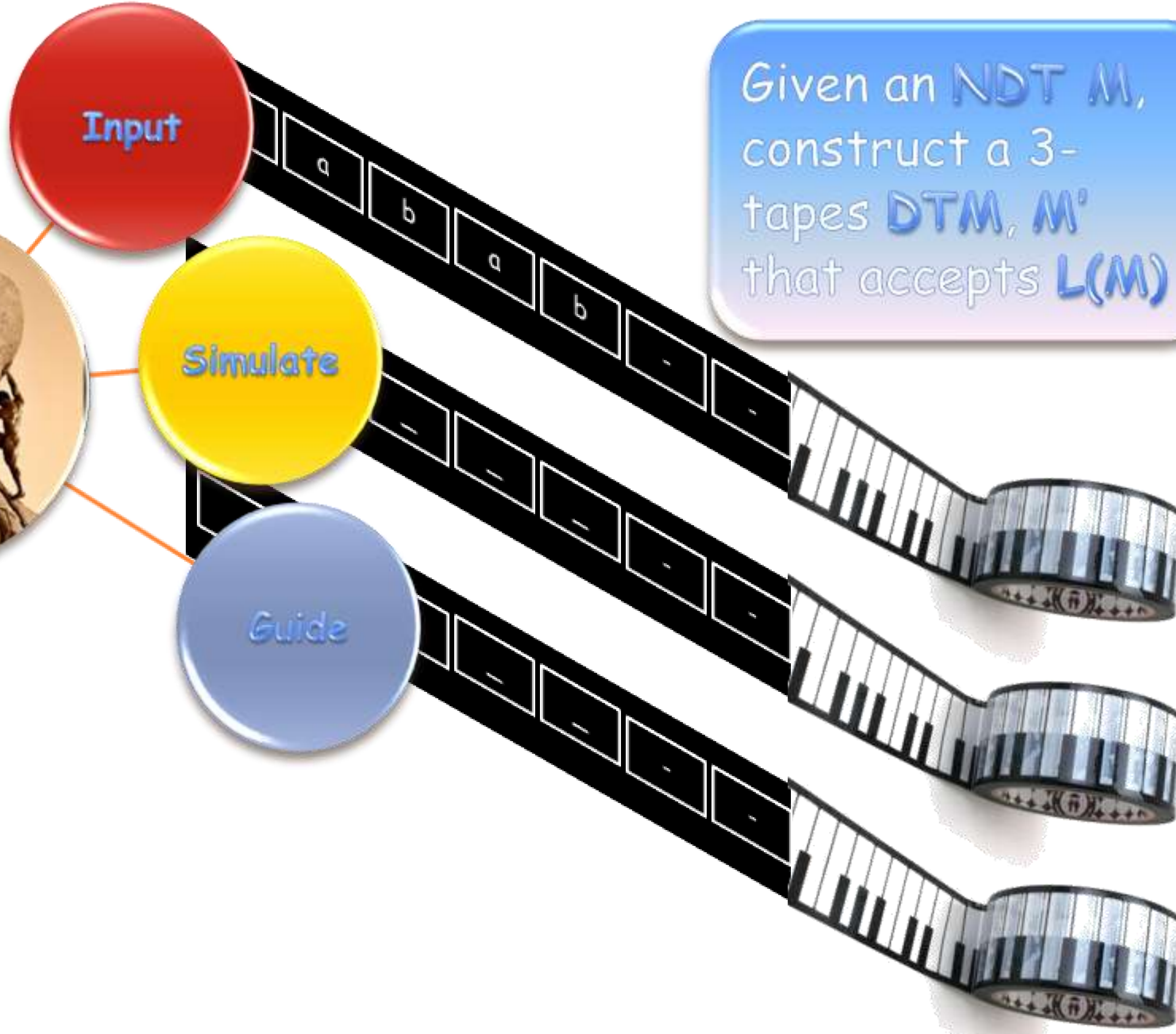
- Traverse from s to t
- A prime factorization
- Isomorphism

Verify

- Is it a path from s to t?
- Are primes whose product =N
- Does π transform G into G'?

SIP 138-140

Non-deterministic → Deterministic



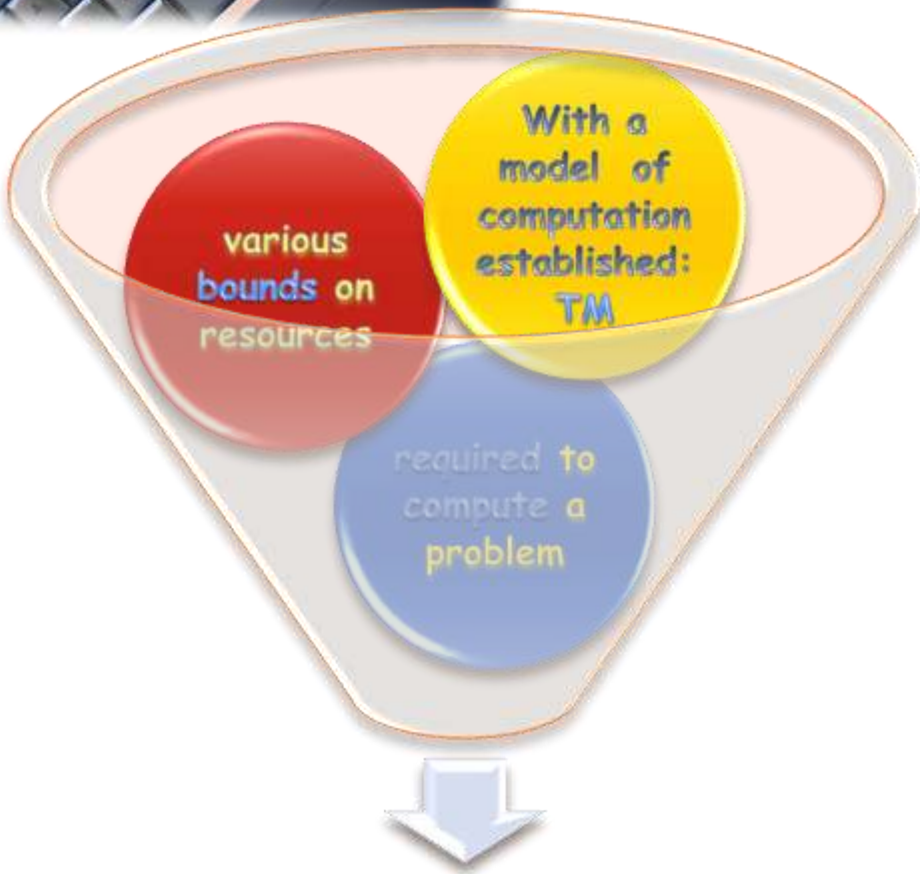
Given an NDT M, construct a 3-tapes DTM, M' that accepts L(M)

Let number of transition $\leq h$
Nondeterministic time $\leq t(n)$

Simulation

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



define **Complexity classes**

Time-Complexity

Definition:

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

Deterministic time:

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

Nondeterministic time:

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

Det. Polynomial time:

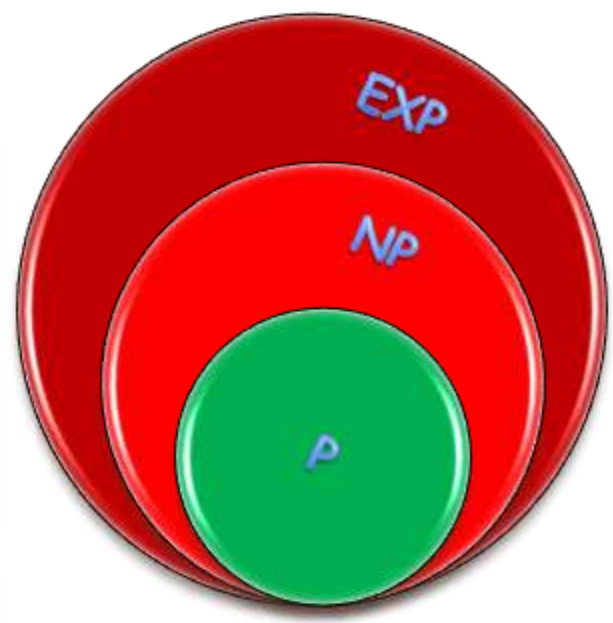
$$P \equiv \bigcup_k TIME\ [n^k]$$

Nondet. Polynomial time:

$$NP \equiv \bigcup_k NTIME\ [n^k]$$

Det Exponential time:

$$EXP \equiv \bigcup_k TIME\ [e^{n^k}]$$



Space-Complexity

Definition:

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

Deterministic space:

$$SPACE[t(n)] \cong \{L \mid L \text{ decided by } O(t(n))\text{-space } \textit{deterministic} \text{ TM}\}$$

Nondeterministic space:

$$NSPACE[t(n)] \cong \{L \mid L \text{ decided by } O(t(n))\text{-space non} \textit{deterministic} \text{ TM}\}$$

Det. Log space:

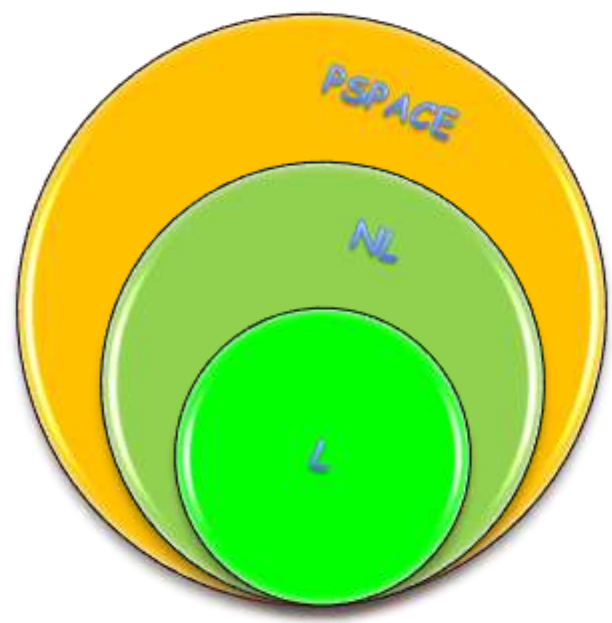
$$L \equiv SPACE[\log(n)]$$

Nondet. Log space:

$$NL \equiv NSPACE[\log(n)]$$

Det polynomial space:

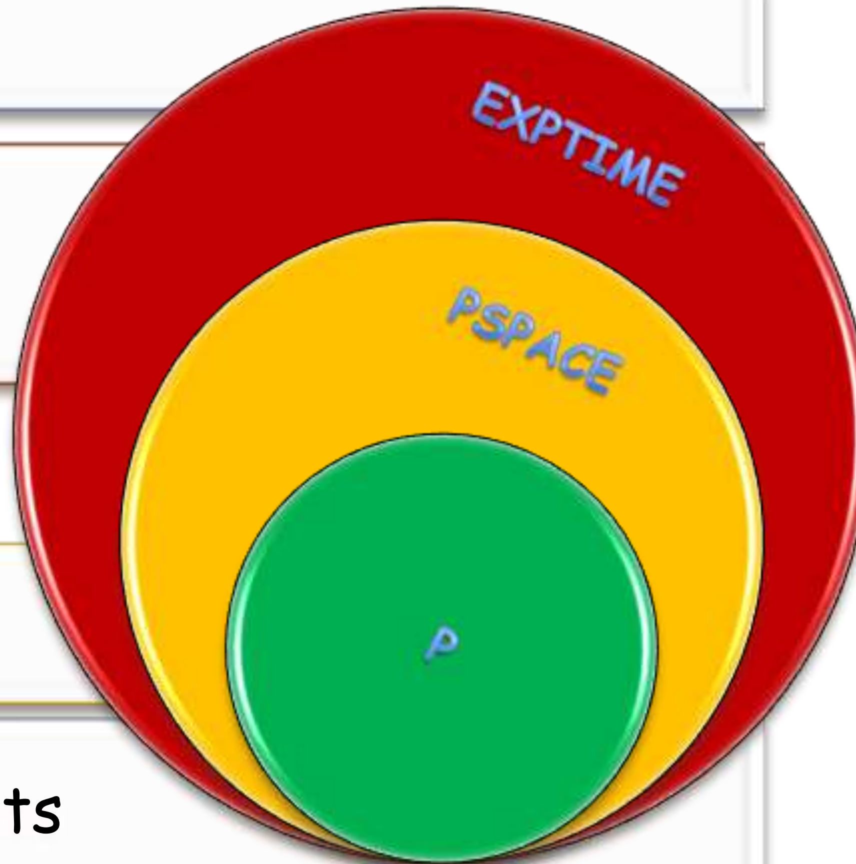
$$PSPACE \equiv \bigcup_k SPACE[n^k]$$



Space vs. Time

Claim:
- P⊆PSPACE

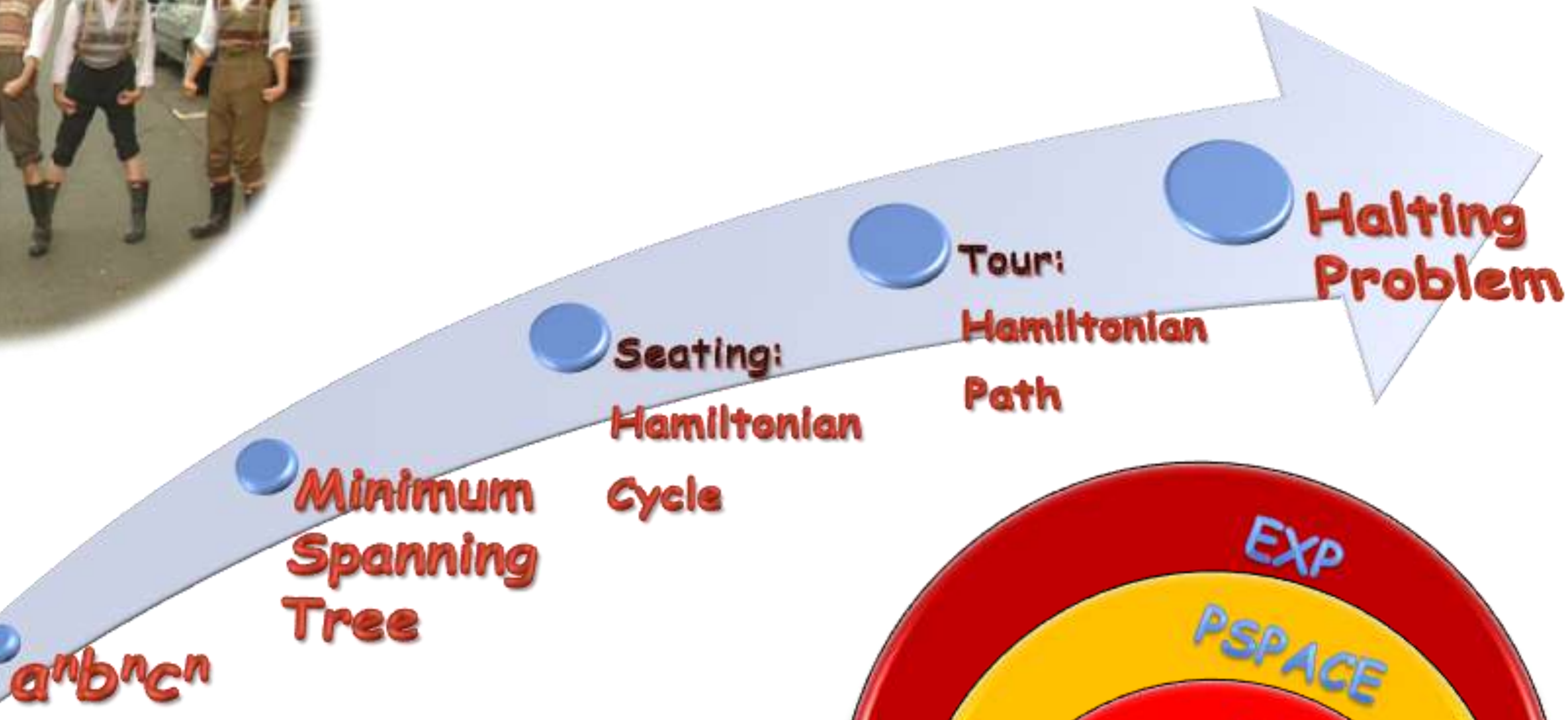
Proof:
- a TM that runs t(n) steps uses at most t(n) space■

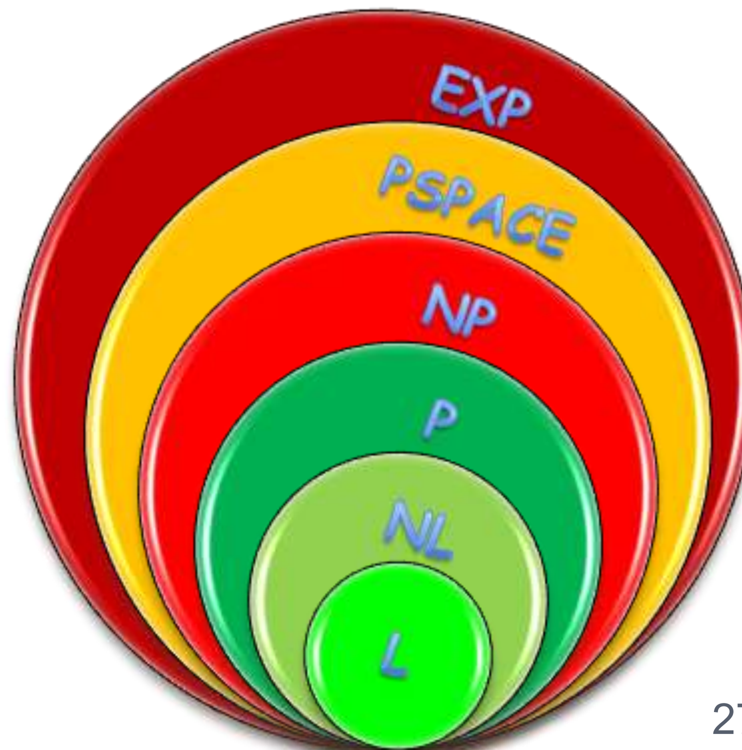


Claim:
- PSPACE⊆EXPTIME

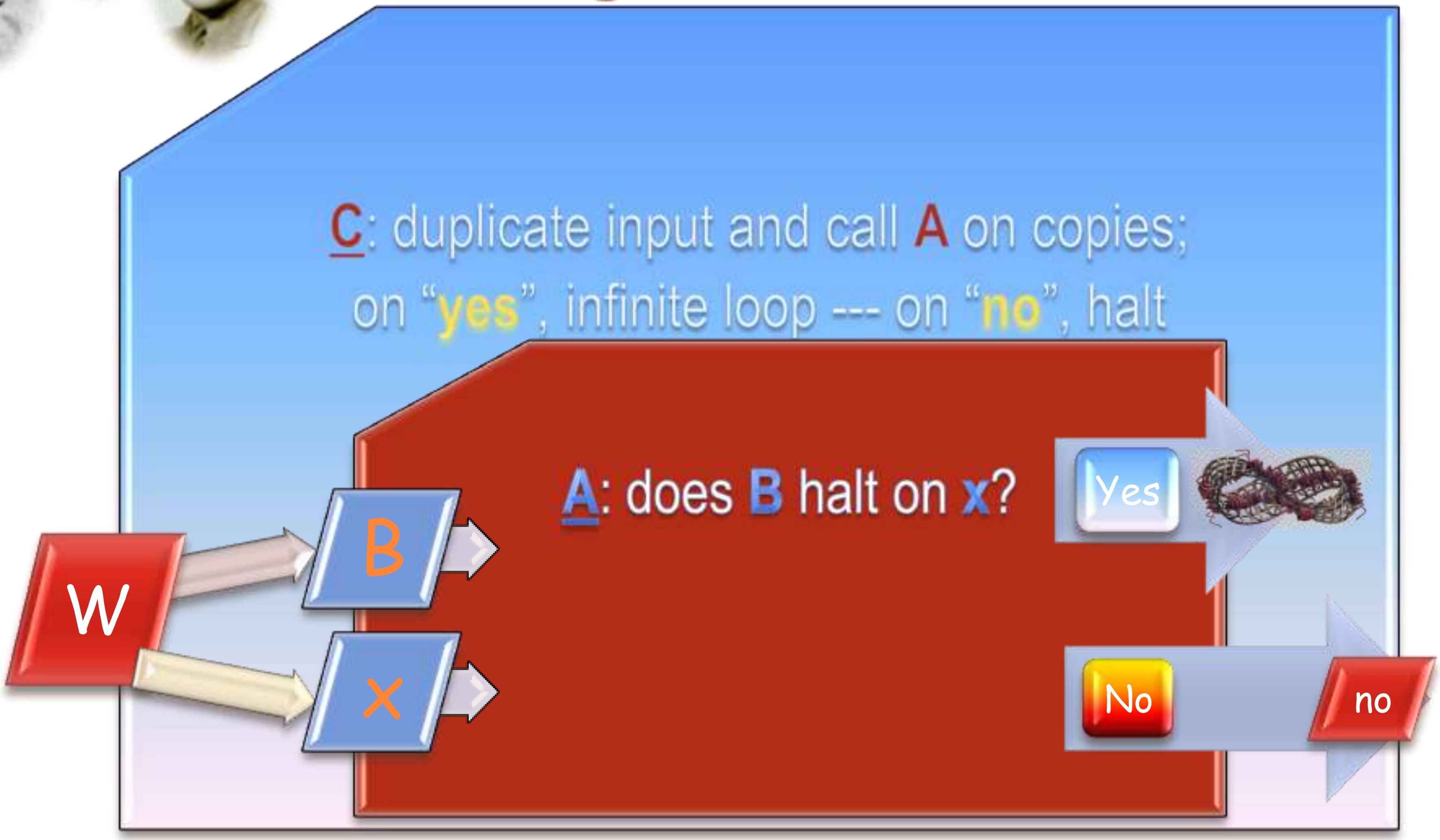
Proof:
- a deterministic run that halts must avoid repeating a configuration ⇒
- its running time is bounded from above by the number of configurations the machine has
- which, for a PSPACE machine, is exponential■

Name the Class



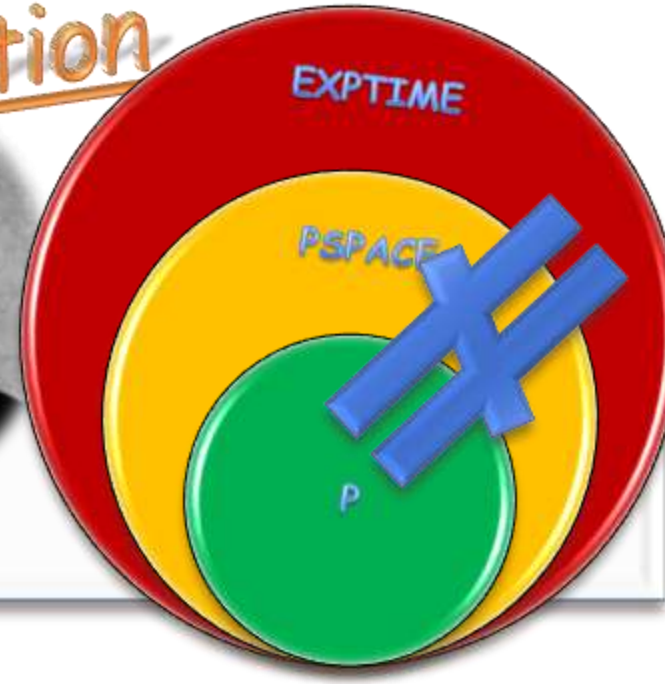


Halting Problem is Undecidable



Run C on (the representation of) C ⇒ contradiction

Diagonalization



Theorem:

- P ≠ EXPTIME

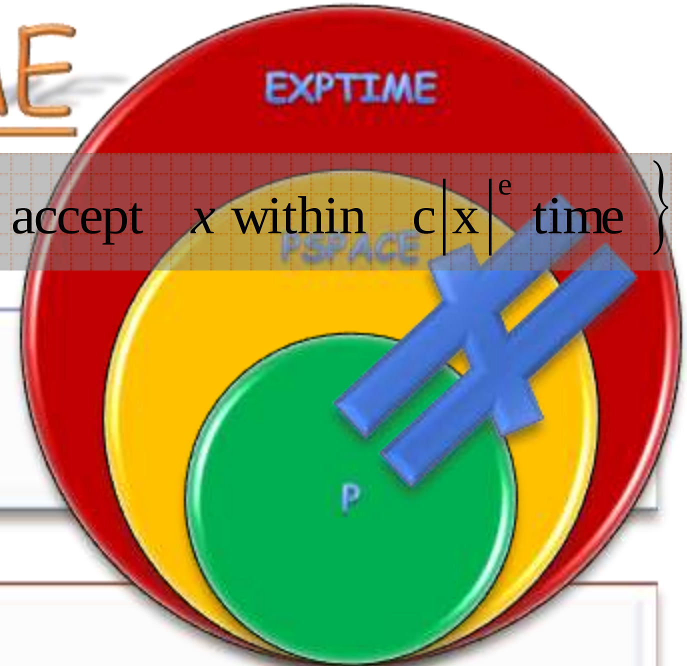
Proof:

- We construct a language L ∈ EXPTIME, which, however, is not accepted by any TM running in polynomial time:

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

P vs EXPTIME

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



Lemma:

- $L \in$ EXPTIME

Proof:

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

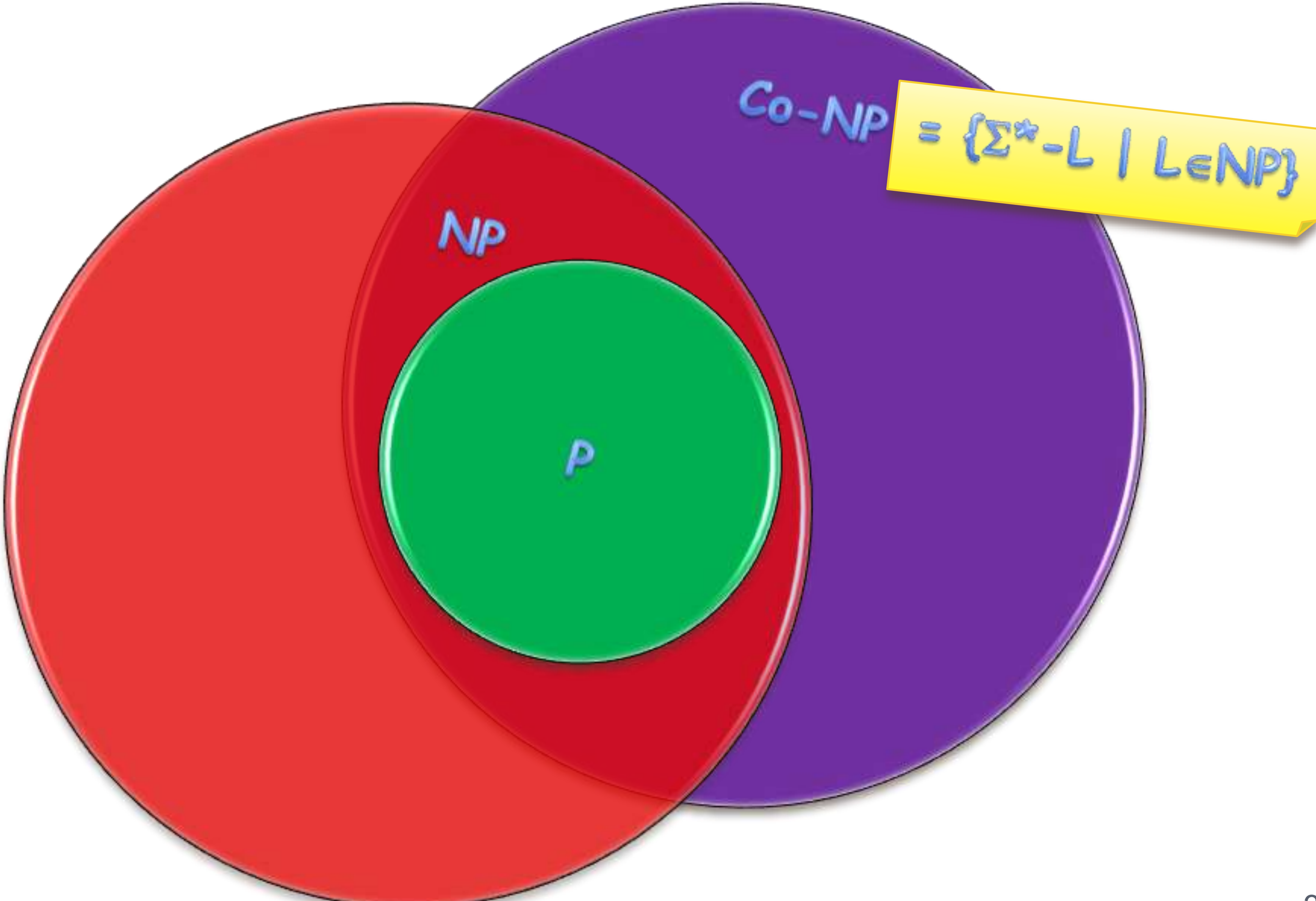
Lemma:

- $L \notin$ P

Proof:

- Assume a TM M that accepts $x \in L$ in time $c|x|^e$ ⇒ run it on the string "$\langle M \rangle \#1^c\#1^e\#$" ⇒ contradiction

P, NP and co-NP



Summary

presented two computational models:

1. deterministic Turing machines
2. non-deterministic Turing machines.

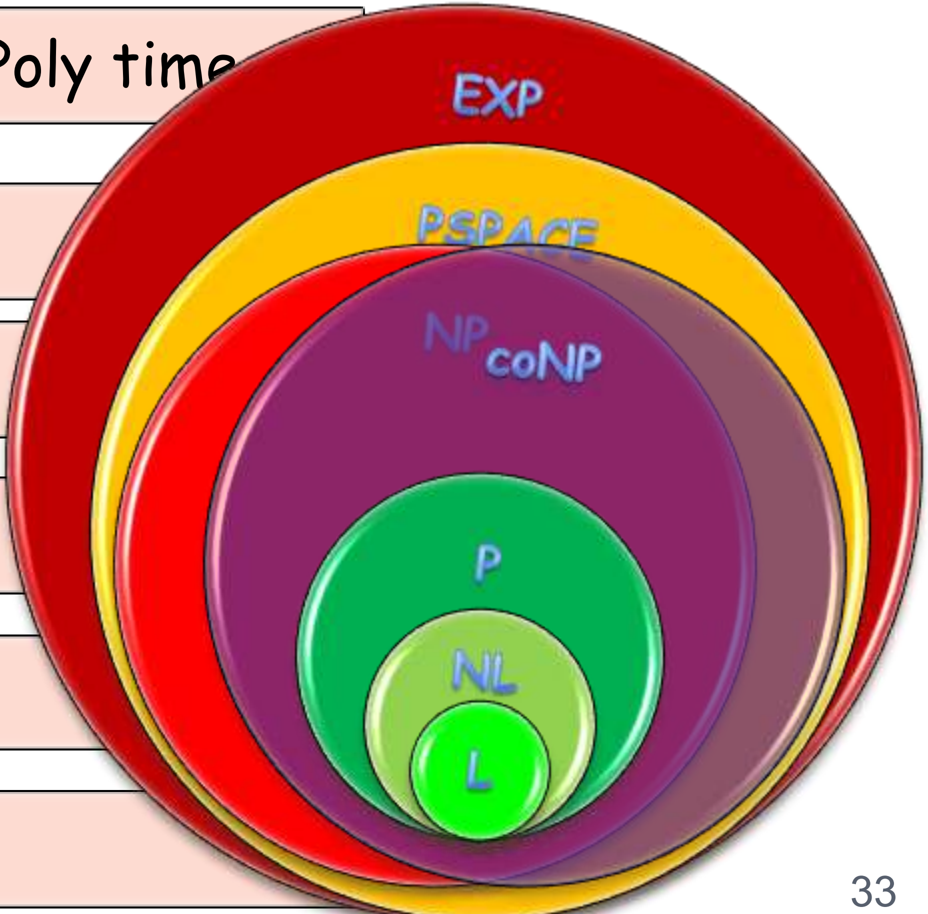
simulated NTM by DTM
with an exponential
blowup in time.

From now on: use pseudo-code instead of TMs

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

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



WWindex

Turing Machine

Church-Turing Hypothesis

Complexity Theory

Halting Problem

Non Deterministic TM

Complexity Classes

NP

co-NP

P

L

NL

EXPTIME

PSPACE


Cantor, Georg


Hilbert, David


Gödel, Kurt


Turing, Alan


Church, Alonzo