CS420

Introduction to the Theory of Computation

Lecture 12: Turing Machines

Tiago Cogumbreiro

Today we will learn...

- Introduce Turing Machines
- Design Turing Machines
- Define Turing machines
- Configuration
- Configuration history

> Section 3.1

You might enjoy this...

- A Mind for Numbers, Barbara Oakley. (audio book is free @ UMB Library)

1. Recap

- Deterministic Finite Automaton that recognize Regular Languages
- Pushdown Automaton that recognize Context-Free Languages

2. Turing Machines

- Introduced to research into the foundations of mathematics
- characterizes **computation**
- can represent any computable machine unbounded by time and space

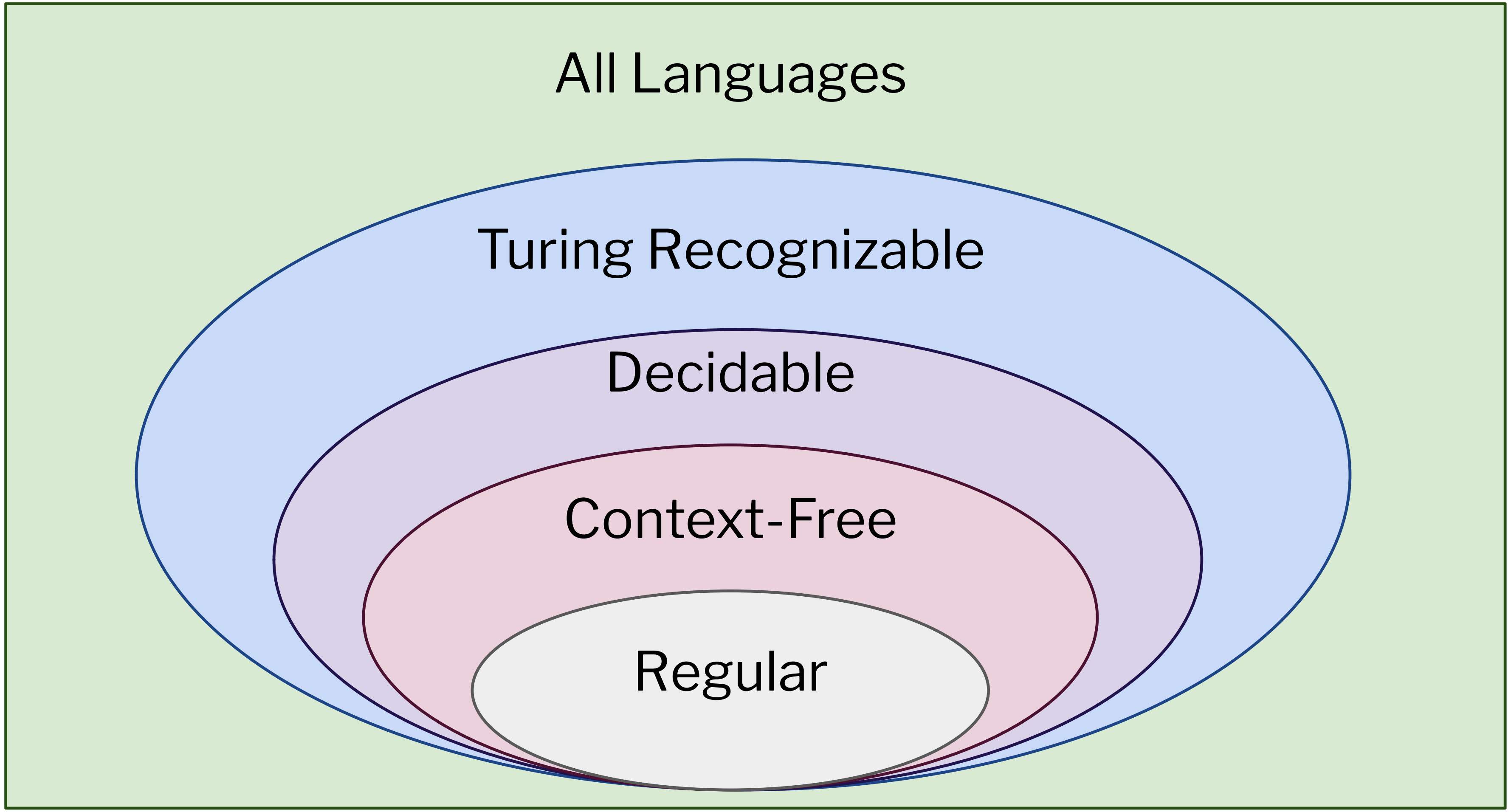
In general, describes problems of the form:

> Decide for any given x whether or not x has property P

Next lecture

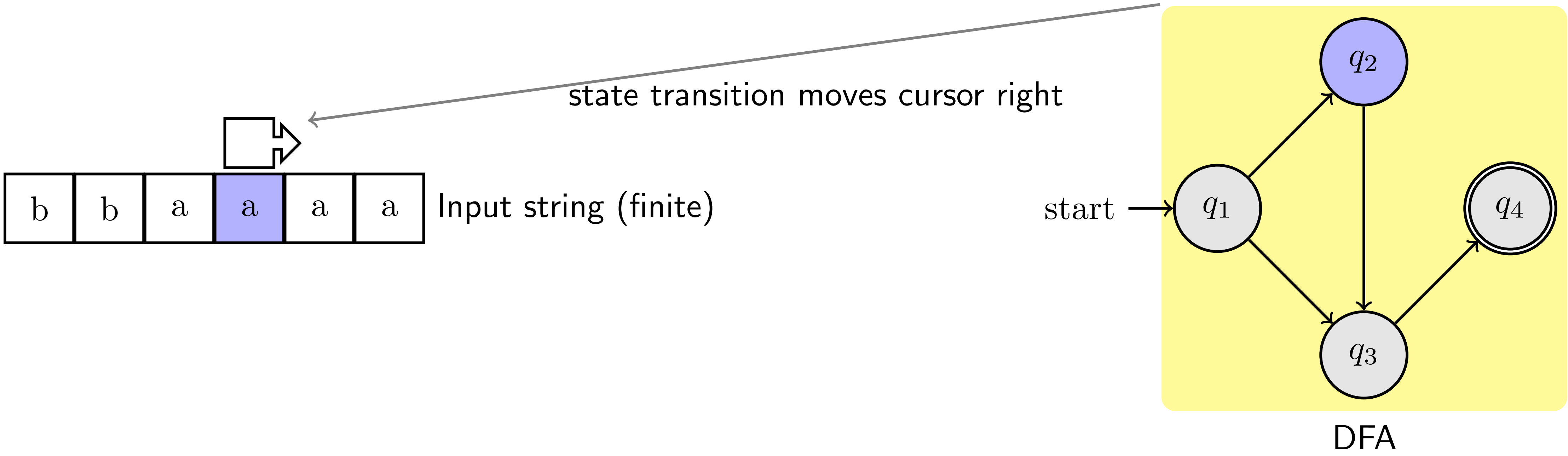
- Historical background on Turing machines

The big picture



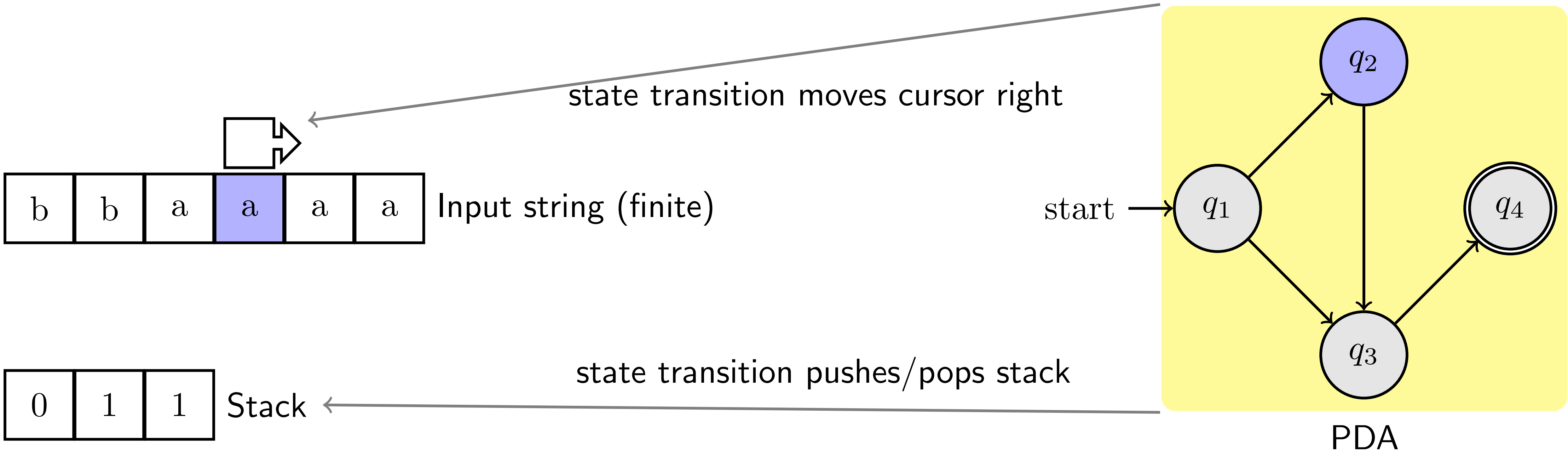
Recall DFA operation

- Automaton processes a finite input string (acceptance)
- Transition moves the cursor forward
- Final state accepts the string if the cursor is at the end



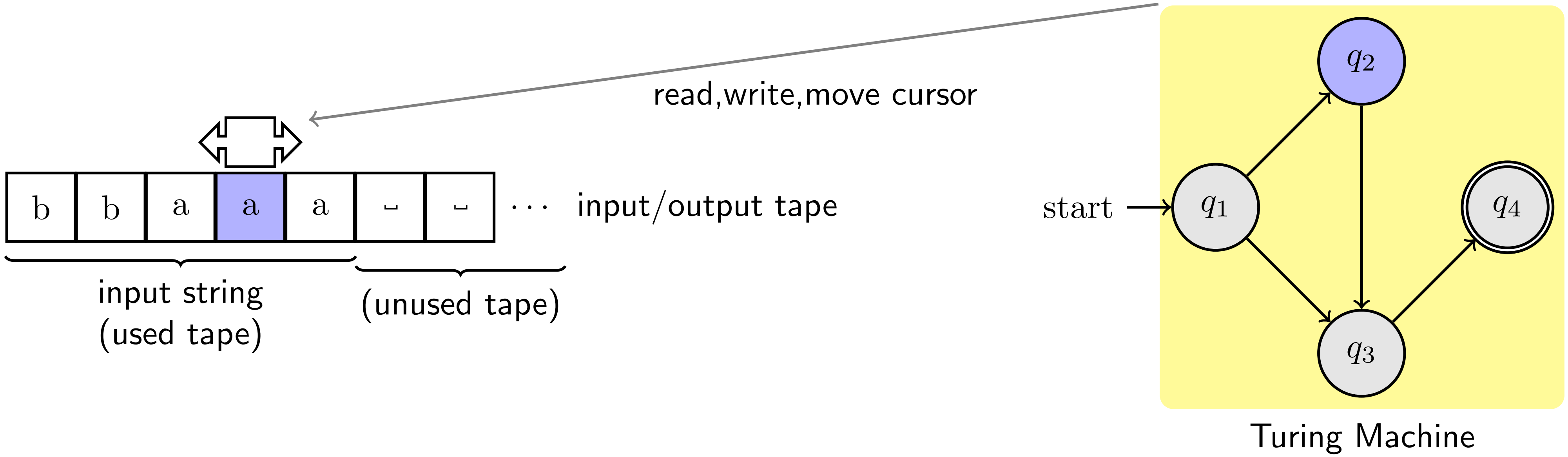
Recall PDA operation

- Automaton processes a finite input string (acceptance) and a stack
- Transition may move the cursor forward and may push/pop the stack
- Final state accepts the string if the cursor is at the end



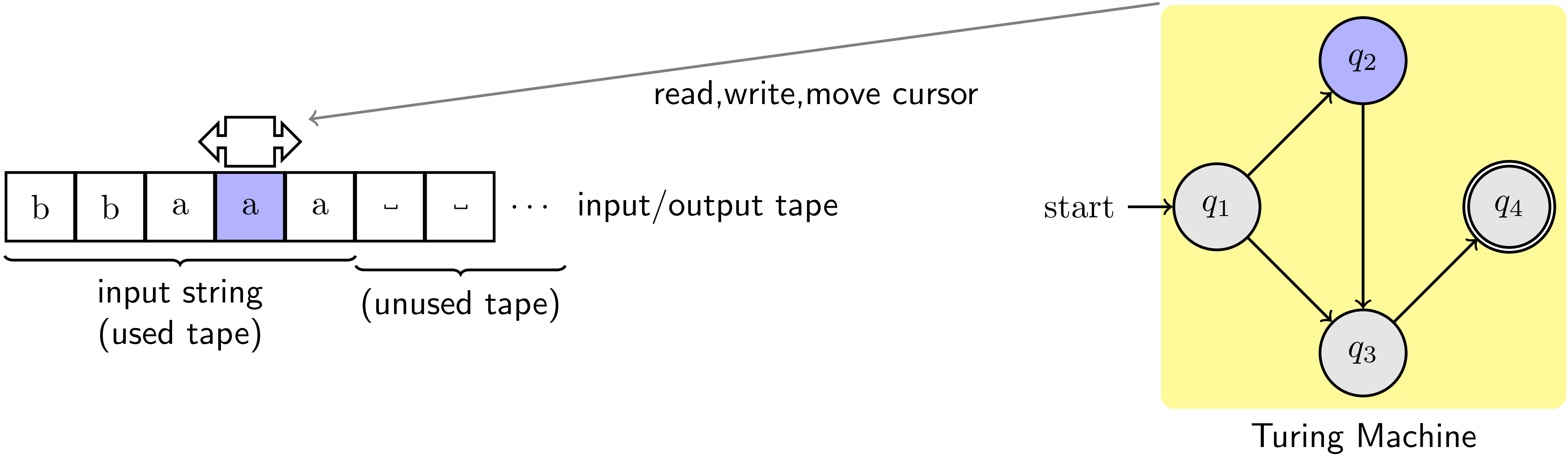
Turing Machine operation

- Automaton processes an **infinite tape**
- Transition may move the cursor forward **or backward**
- Elements of the tape may be written or read
  (tape combines the input string and the stack)
- Tapes may contain a special character called black, notation ␣ (akin to `NULL`)



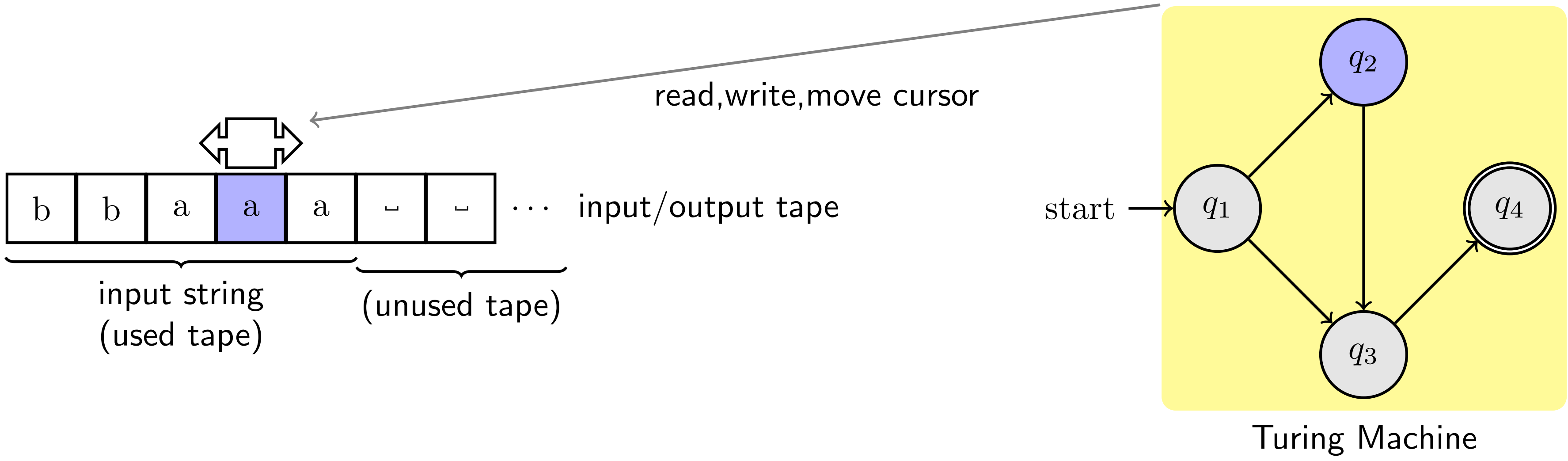
Turing Machine operation

- The **tape head** (or cursor) points to a position in the tape (akin the instruction pointer in a processor)
- Transition: read $\to$ write, move direction
  $q \xrightarrow{a \to b,\text{R}} q'$



Turing Machine control

- The automaton (the turing machine) is known as the **control** or the **program**
- The automaton is deterministic (nondeterminism has same expressiveness!)
- A single initial state
- A single accept state
- A reject state



Turing Machines acceptance

Given a tape (with an) and a Turing machine, there are three kinds of answers:

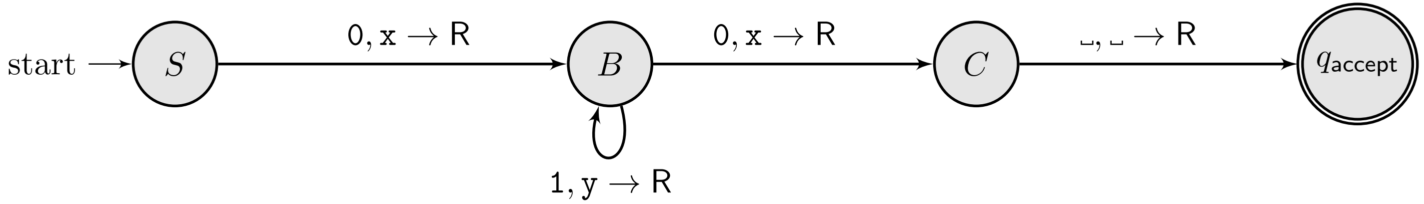
- **Accept**
  Whenever the machine reaches the accept state, the automaton halts and the input string is accepted.
- **Reject**
  Whenever the machine reaches the reject state, the automaton halts and the input string is rejected.
- **Loop forever**
  The machine keeps doing transitions in a loop, never accepting nor rejecting the input string.

> While a PDA and a DFA can either accept or reject a string, a Turing machines can also loop forever!

Examples

Example 1

$\boxed{L = 01^{\star}0}$



- Deterministic (only one outgoing edge **per input**)
- Convention: missing transitions go to reject state (hidden).

Example

State	Tape
S	01110
B	x1110
B	xy110
B	xyy10
B	xyyy0
C	xyyyx_
q_{accept}	xyyyx _

Simulate

Example 2

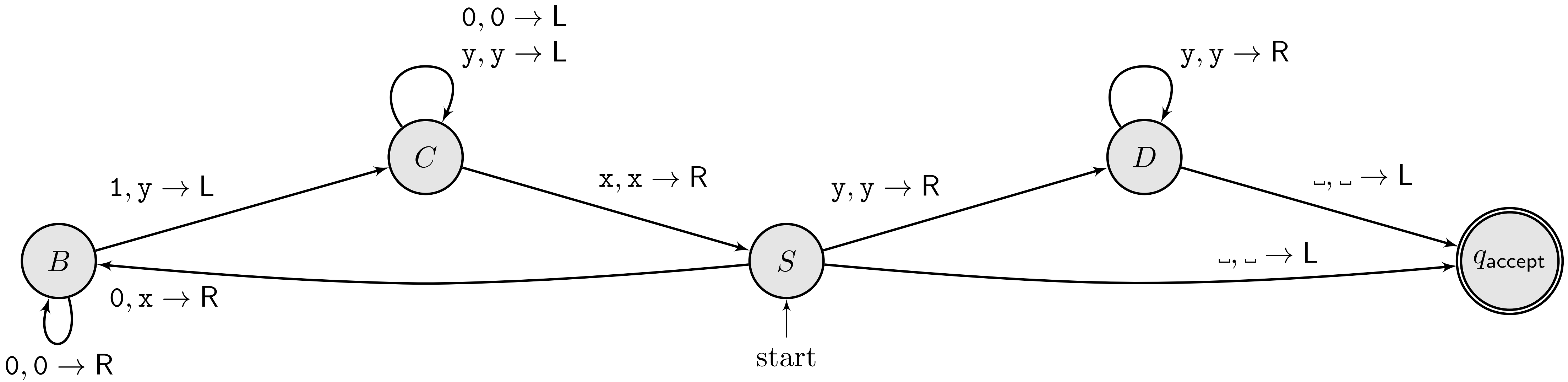
$$L_1 = \{0^n 1^n \mid n \geq 0\}$$

Mark 0 seek and mark 1 and cycle back.

- Start (S): `if 0 {write X; move right; goto B}; if Y {skip right; goto D}`
- Seek 0 (B): `while 1 or X {skip right};if 1 {write Y; move right; goto C}`
- Seek 1 (C): `while 0 or Y {skip left};if X {skip; move right; goto S}`
- Check valid (D): `while Y {skip right};if ␣ {skip;move right; goto accept}`

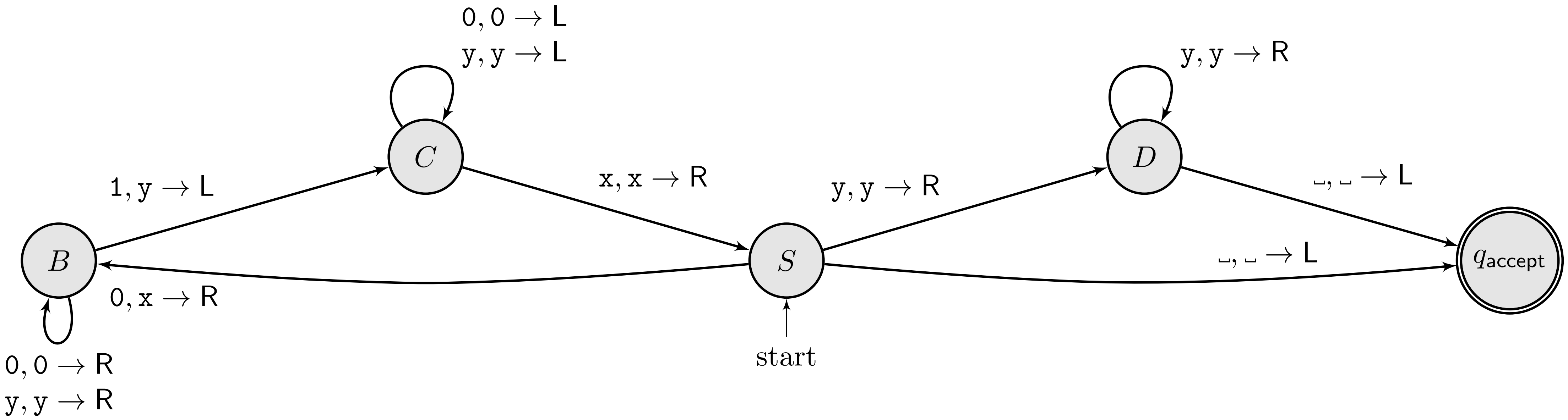
Tape	*State*	*Rule*
0011	S	read 0; write X; move right; goto B
X011	B	skip right while 1 or x; if 1 {write Y; move right; goto C}
X0Y1	C	skip left while 0 or y; if x {skip; move right; goto S}
X0Y1	S	read 0; write x; move right; goto B
XXY1	B	skip right while 1 or x; if 1 {write Y; move right; goto C}
XXYY	C	skip left while 0 or y; if x {skip; move right; goto S}
XXYY	S	read y; skip right; goto D
XXYY␣	D	read ␣, goto accept

Example 2



State	Tape
S	0011
B	X011
B	X011
C	X0Y1

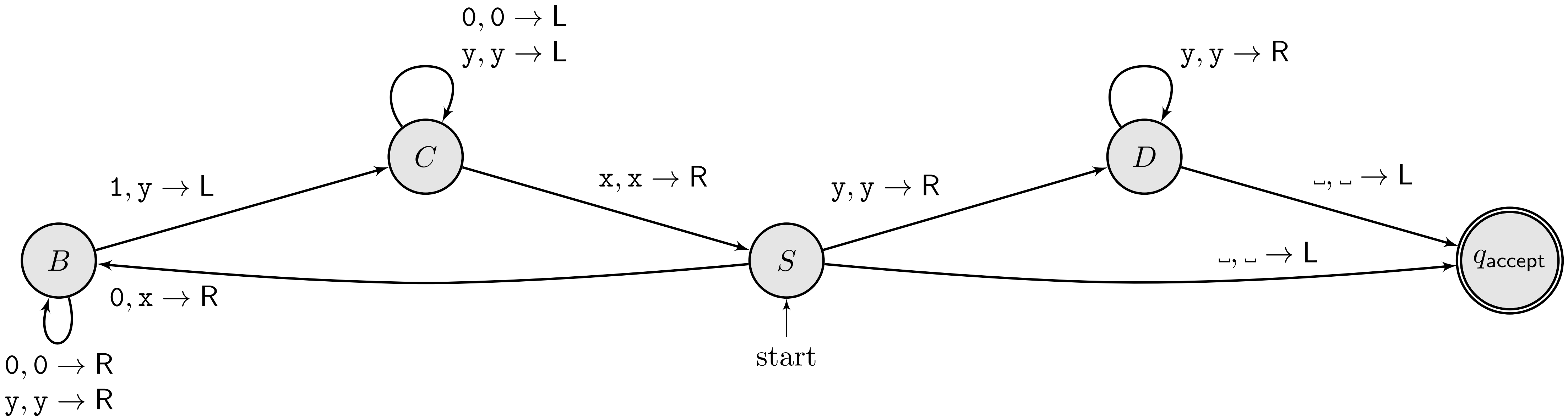
Example 2



State	Tape
S	0011
B	X011
B	X011
C	X0Y1

State	Tape
C	X0Y1
S	X0Y1
B	XXY1
B	XXY1

Example 2

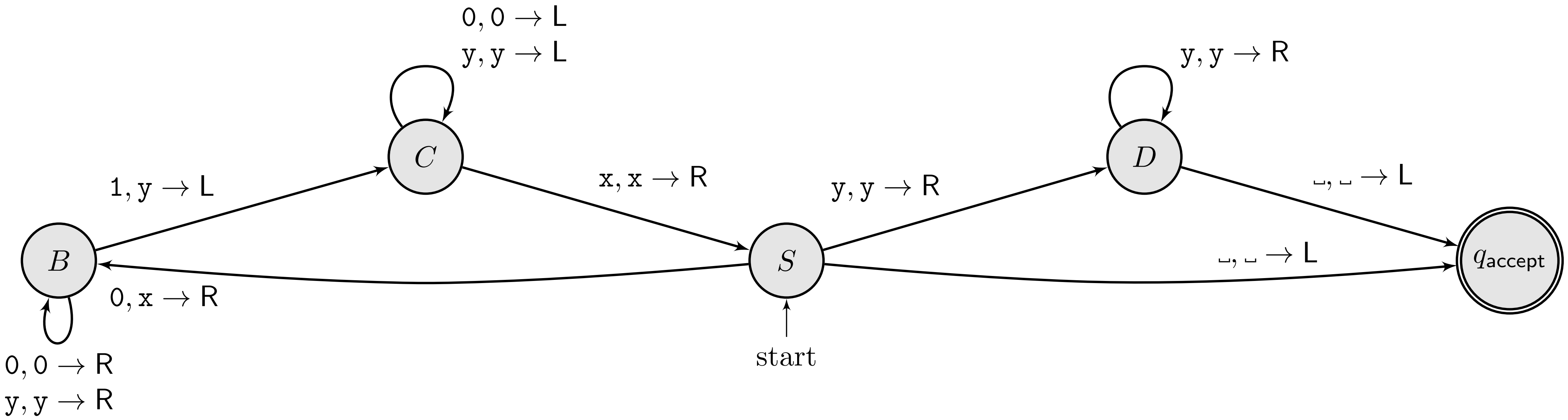


State	Tape
S	0011
B	X011
B	X011
C	X0Y1

State	Tape
C	X0Y1
S	X0Y1
B	XXY1
B	XXY1

State	Tape
C	XXYY
C	XXYY
S	XXYY
D	XXYY

Example 2



State	Tape
S	0011
B	X011
B	X011
C	X0Y1

State	Tape
C	X0Y1
S	X0Y1
B	XXY1
B	XXY1

State	Tape
C	XXYY
C	XXYY
S	XXYY
D	XXYY

State	Tape
D	XXYY␣

Accept!

Simulate

Example 3

$L_3 = \{a^n b^n c^n \mid n \geq 0\}$

Example 3

$L_3 = \{a^n b^n c^n \mid n \geq 0\}$

- START: Skip marks **right** until we: i) read a; mark it; go to `A`; ii) read blank, accept.
- A: Skip **right** until read b; mark it; go to `Bs`
- B: Skip **right** until read c; mark it; go to `Cs`
- C: Skip **right** until read blank; move left; go to `REWIND`
- REWIND: Skip **left** until we reach blank, go to `START`

Simulate

Turing Machines

Definition 3.3

A Turing machine is a 7-tuple $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$

1. Q set of states
2. Σ input alphabet not containing the blank symbol $\sqcup$
3. Γ the tape alphabet, where $\sqcup \in \Gamma$ and $\Sigma \subseteq \Gamma$
4. $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{\mathsf{L}, \mathsf{R}\}$ transition function
5. $q_0 \in Q$ is the start state
6. q_{accept} is the accept state
7. q_{reject} is the reject state ($q_{reject} \neq q_{accept}$)

Configuration

> A configuration is a snapshot of a computation. That is, it contains all information necessary to resume (or replay) a computation from any point in time.

A configuration consists of

- the tape
- the head of the tape
- the current state

Configuration

Textual notation

> We write the table and place the current state **before** (left of) where the head of the tape points to:

In the following example, the head points to position no.5, the tape is `0130045`, and the current state is q_3:

Recall example 1

State	***Tape***	***Configuration***
S	`01110`	`S 01110`
B	`x1110`	
B	`xy110`	
B	`xyy10`	
B	`xyyy0`	
B	`xyyyx_`	

Fill in the configuration...

Example 1 configuration

State	Tape	Configuration
S	`01110`	`S 01110`
B	`x1110`	`x B 1110`
B	`xy110`	`xy B 110`
B	`xyy10`	`xyy B 10`
B	`xyyy0`	`xyyy B 0`
B	`xyyyx_`	`xyyyx B`

Configuration history

> The configuration history (sequence of configurations), describes all configurations from the initial state until a current state.

Definition

We say that C_1 **yields** C_2

Example

Configuration history
`S 01110`
`x B 1110`
`xy B 110`
`xyy B 10`
`xyyy B 0`
`xyyyx B`

More examples

- $L_5 = \{w\#w \mid w \in \{a,b\}^\star\}$
- $L_6 = \{w \mid w \text{ is a palindrome}\}$
- $L_7 = \{a^n b^{2n} \mid n \geq 0\}$