Computability Theory

Turing Machines

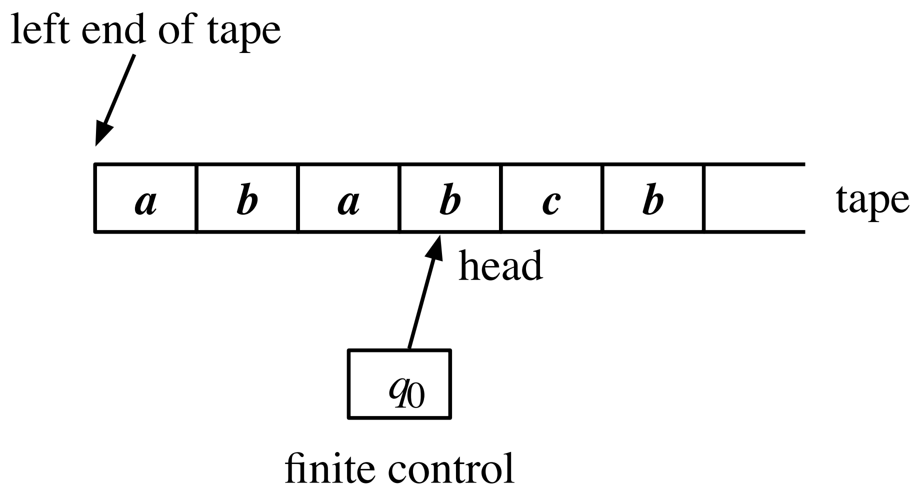
A **Turing machine** is a step-wise computing device consisting of:

1. an infinitely long **tape** divided into **tape squares** (or **tape cells**),
2. a **head** for scanning the tape, and
3. a **finite control** that maintains the current state.

Turing Machines

A **Turing machine** is a step-wise computing device consisting of:

1. an infinitely long **tape** divided into **tape squares** (or **tape cells**),
2. a **head** for scanning the tape, and
3. a **finite control** that maintains the current state.



Computation by a Turing Machine

In each computational step, a Turing machine:

- reads the symbol written at the current head location, and then,
- depending on the symbol and the current state
    - determines its next state,
    - write a symbol at the current head location, and
    - moves the head to either the next or the prior tape square.

Turing Machines (cont'd)

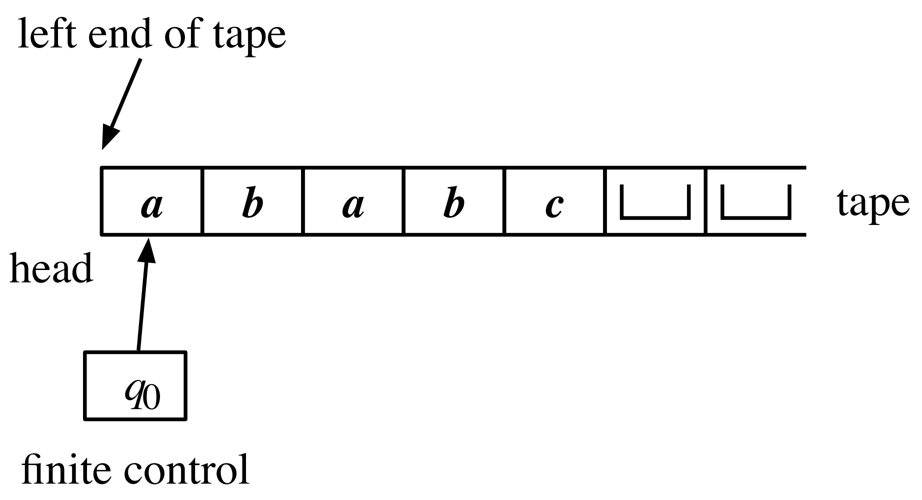
A **Turing machine** is a 7-tuple $(Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$, where Q, Σ, Γ are finite sets,

1. Q is a set of states,
2. Σ is the input alphabet,
3. Γ is the tape alphabet such that $\Sigma \subset \Gamma$. There is a special symbol $\sqcup$ in $\Gamma - \Sigma$.
4. $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ is the transition function,
5. $q_0 \in Q$ is called the **initial state**,
6. $q_{\text{accept}} \in Q$ is called the **accept state**, and
7. $q_{\text{reject}} \in Q$ is called the **reject state**

Initial Condition of a Turing Machine

At the beginning

1. the tape contains its input in the leftmost squares,
2. the rest of the tape is filled with $\sqcup$,
3. the head is at the leftmost cell, and
4. the state is q_0.



Termination of a Turing Machine

The Turing machine halts when it enters either q_{accept} or q_{reject}. We say that the machine **accepts** when the former occurs and **rejects** when the latter occurs.

Computation of a Turing Machine in One Step (Again)

In each computational step, a Turing machine:

- reads the symbol written at the current head location, and then,
- depending on the symbol and the current state
    - determines its next state,
    - write a symbol at the current head location, and
    - moves the head to either the next or the prior tape square.

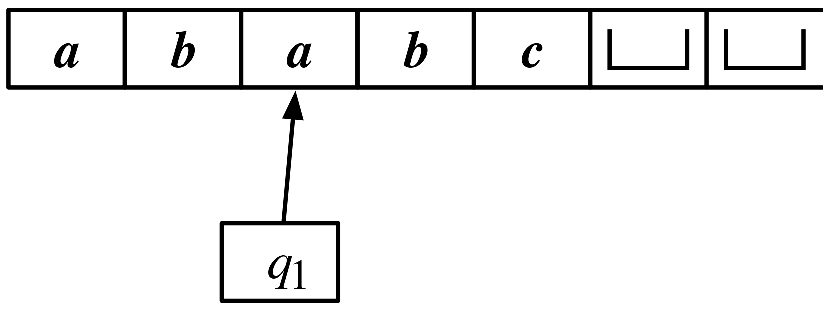
Computation of a Turing Machine in One Step (Again)

In each computational step, a Turing machine:

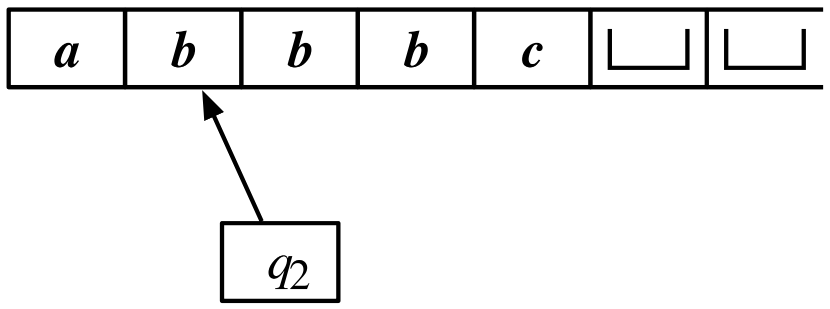
- reads the symbol written at the current head location, and then,
- depending on the symbol and the current state
    - determines its next state,
    - write a symbol at the current head location, and
    - moves the head to either the next or the prior tape square.

However, if the instruction is to move the head to the left and the current head position is the leftmost square, then the head stays at the leftmost square.

Example



becomes



The Languages of Turing Machines

A Turing machine M **accepts** (**rejects**) a string x if it enters the accepting (rejecting) state on input x.

A Turing machine M **recognizes** a language L, if for every input x, M **on** x **accepts if** $x \in L$ **and does not accept if** $x \notin L$.

A language is **Turing-recognizable** (or **recursively enumerable**) if there is a Turing machine that recognizes it.

The Languages of Turing Machines (cont'd)

A Turing machine M **decides** a language L, if M recognizes L and halts on all inputs.

In other words, a Turing machine M **decides** a language L if, for all x, M on x accetps if $x \in L$ and rejects if $x \notin L$.

A language is **Turing-decidable** (or simply **decidable**) if there is a Turing machine that decides it.

Example 1: $L = \{w\#w \mid w \in \{0,1\}^*\}$

L is not context free.

To decide whether an input $z \in \{0,1,\#\}^*$ is in L our Turing machine M operates as follows:

- M attempts to match letter by letter the word to the left of $\#$ (call the word u) and the word to the right of $\#$ (call the word v).
- When a letter in u matches a letter in v at the corresponding position, both letters are erased using a special letter $\mathtt{x}$.
- When all the letters of u and v have been matched, M accepts.

Components of M

M in itself has the following components:

- an automaton for $\mathtt{x}^*\#\mathtt{x}^*$;
- an automaton for $\mathtt{x}^*0\{0,1\}^*\#\mathtt{x}^*0\{0,1\}^*\#$;
- an automaton for $\mathtt{x}^*1\{0,1\}^*\#\mathtt{x}^*1\{0,1\}^*\#$;

Intuitively, M runs these automata concurrently.

The last two automata replace the letter immediately after each run of $\mathtt{x}$ with an $\mathtt{x}$ if the two letters match.

Method for Matching

The tape contents are always of the form $\mathtt{x}^*\{0,1\}^*\#\mathtt{x}^*\{0,1\}^*$, where the number of $\mathtt{x}$'s on the left is equal to the number of $\mathtt{x}$'s on the right.

Action in the Very First Step

- If the symbol being scanned is $\sqcup$, then immediately reject z because $z = \epsilon$.
- Otherwise, enter the loop.

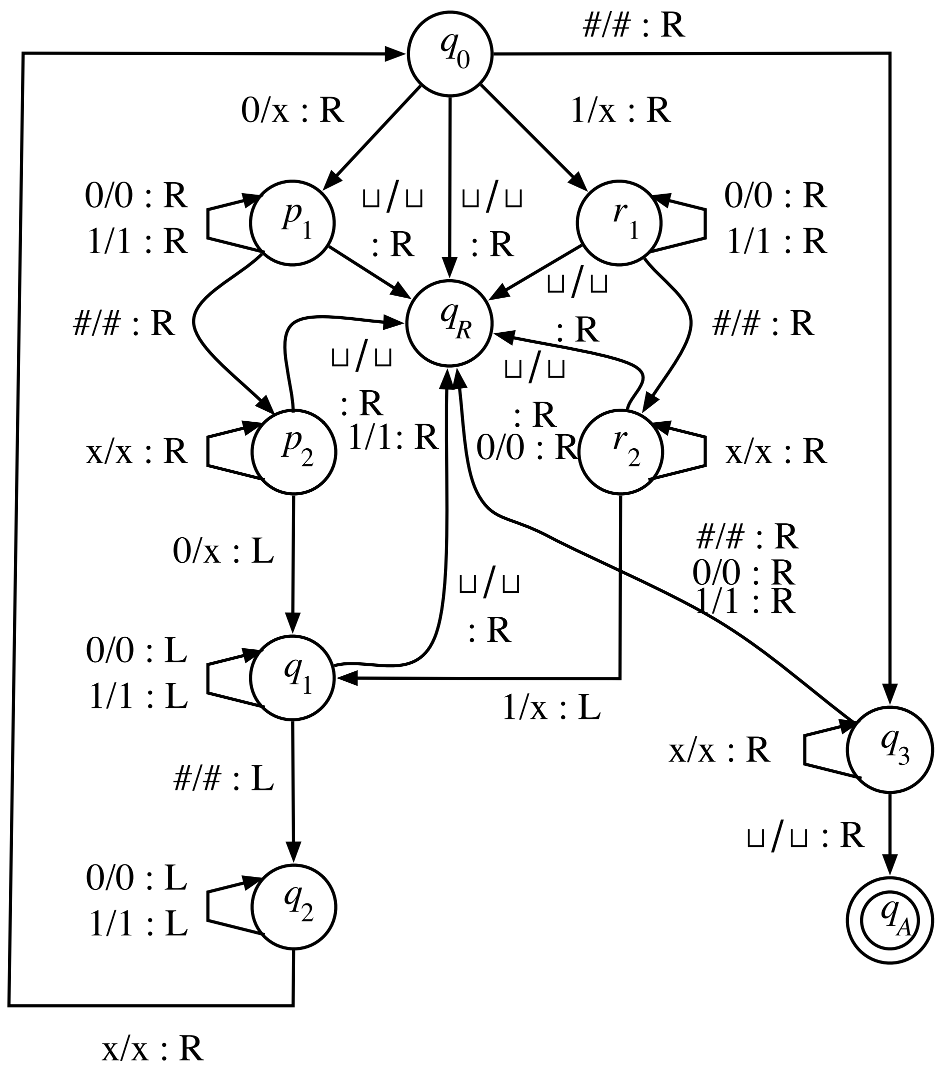
Main Loop

- (Case 1) If the letter at the current head position is a $\#$, then:
    - scan to the right until a letter other than $\mathtt{x}$ is found, and then
    - accept if that non-$\mathtt{x}$ is $\sqcup$; reject otherwise.

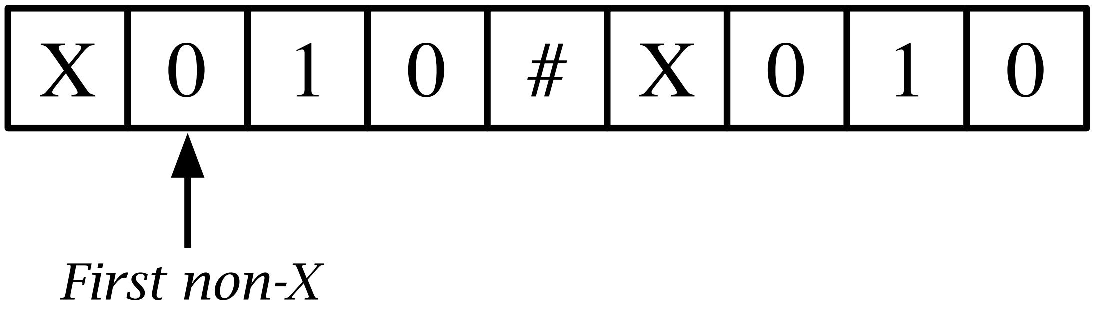
Method for Matching

- (Case 2) If the letter at the current head location is a 0 or a 1, then:
  - memorize the letter, say c, and write an $\mathtt{x}$,
  - scan to the right until a $\#$ is found,
  - scan to the right until a non-$\mathtt{x}$ is found, and then
  - if that non-$\mathtt{x}$ is not a c, then reject,
  - replace the non-$\mathtt{x}$ with an $\mathtt{x}$,
  - scan to the left until a $\#$ is found,
  - scan to the left until an $\mathtt{x}$ is found,
  - move to the right thereby locating the head to the very first non-$\mathtt{x}$.

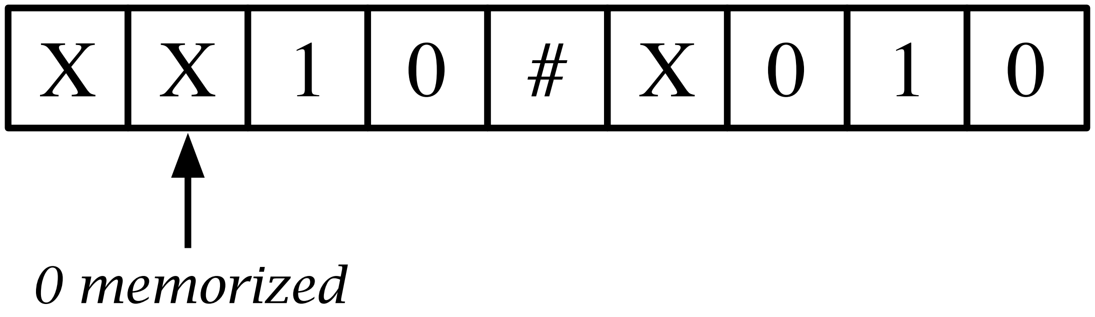
Example $\{w\#w \mid w \in \{0,1\}^*\}$



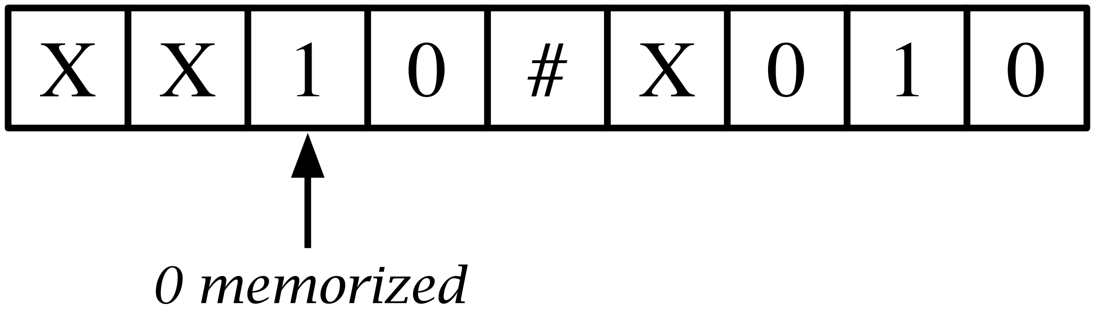
The Main Loop



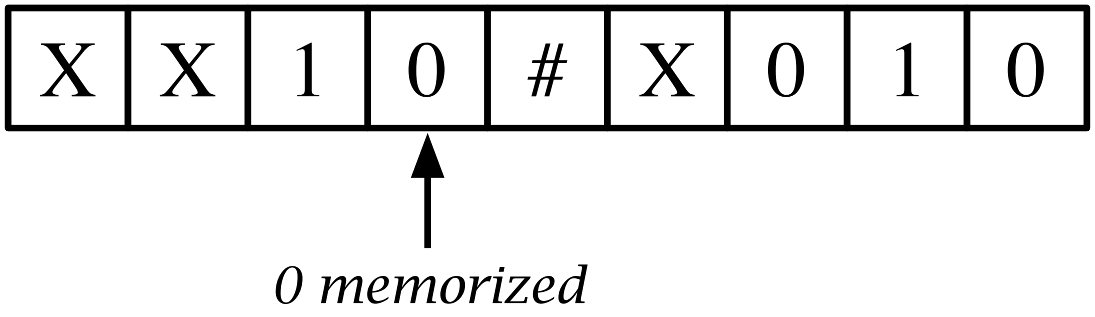
The Main Loop



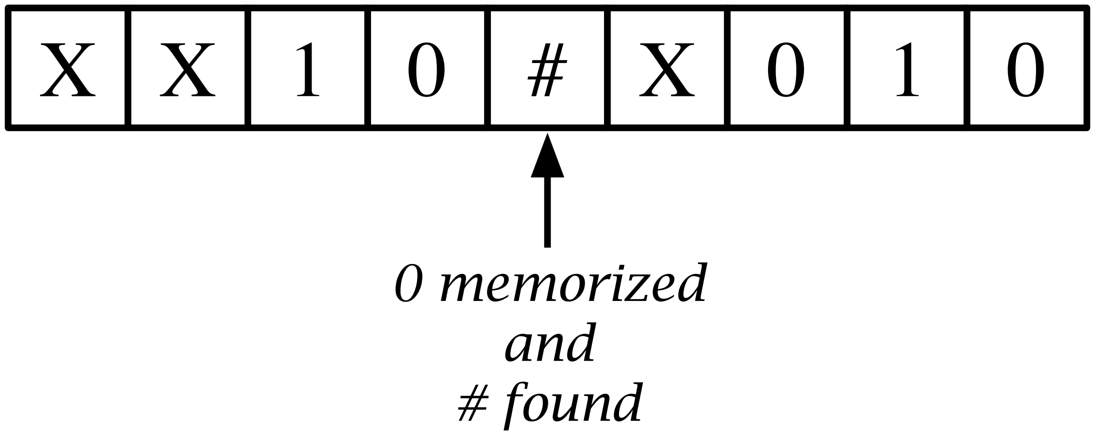
The Main Loop



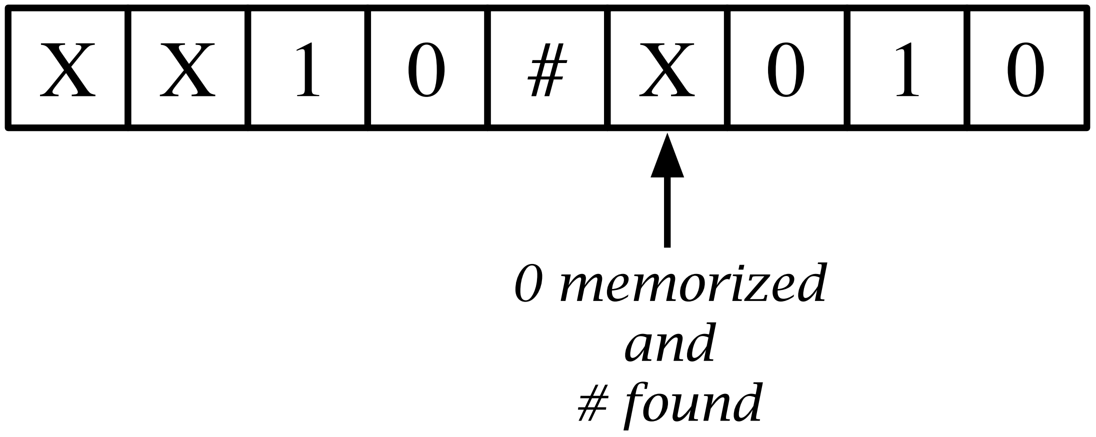
The Main Loop



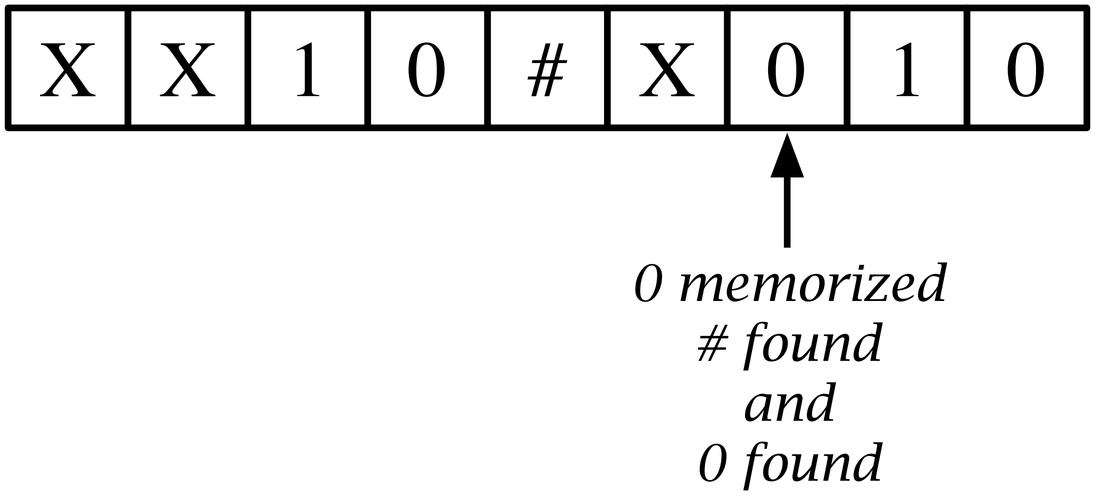
The Main Loop



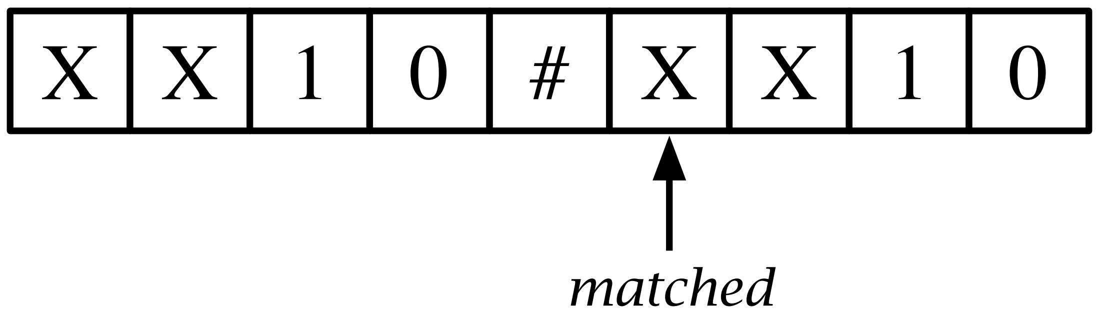
The Main Loop



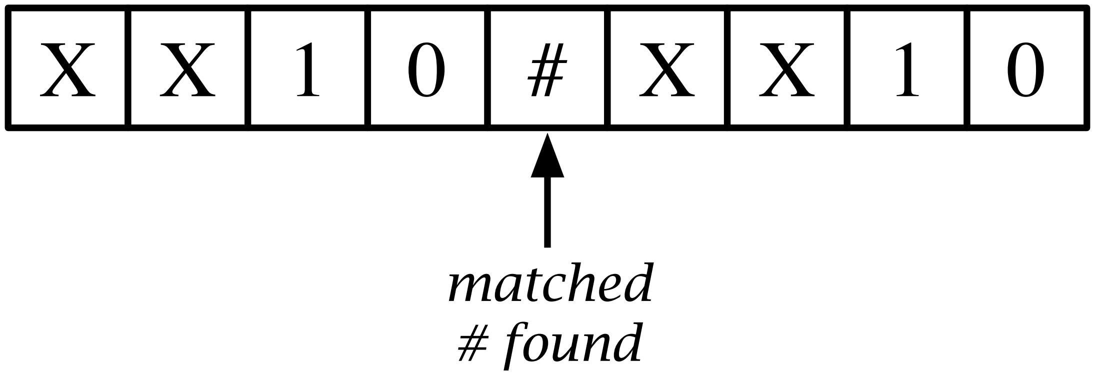
The Main Loop



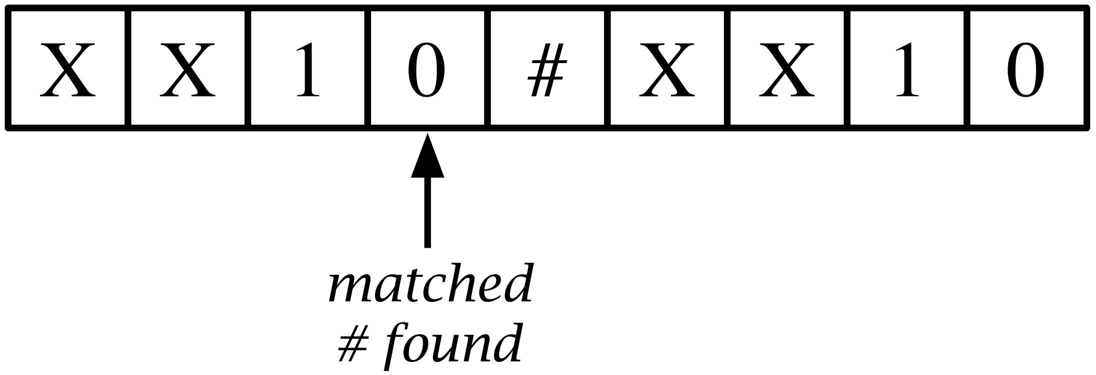
The Main Loop



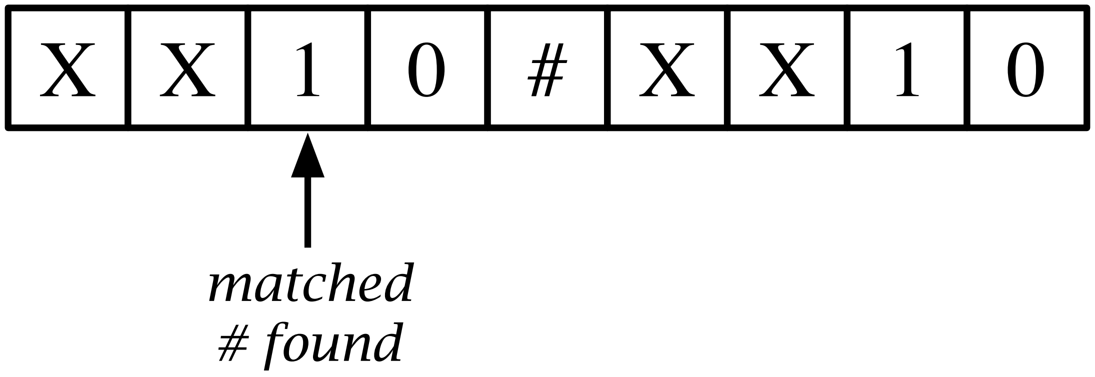
The Main Loop



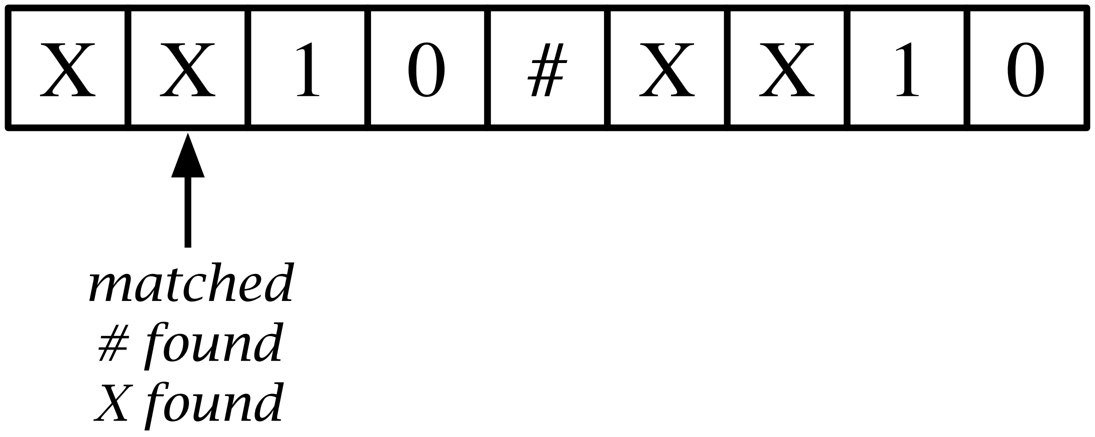
The Main Loop



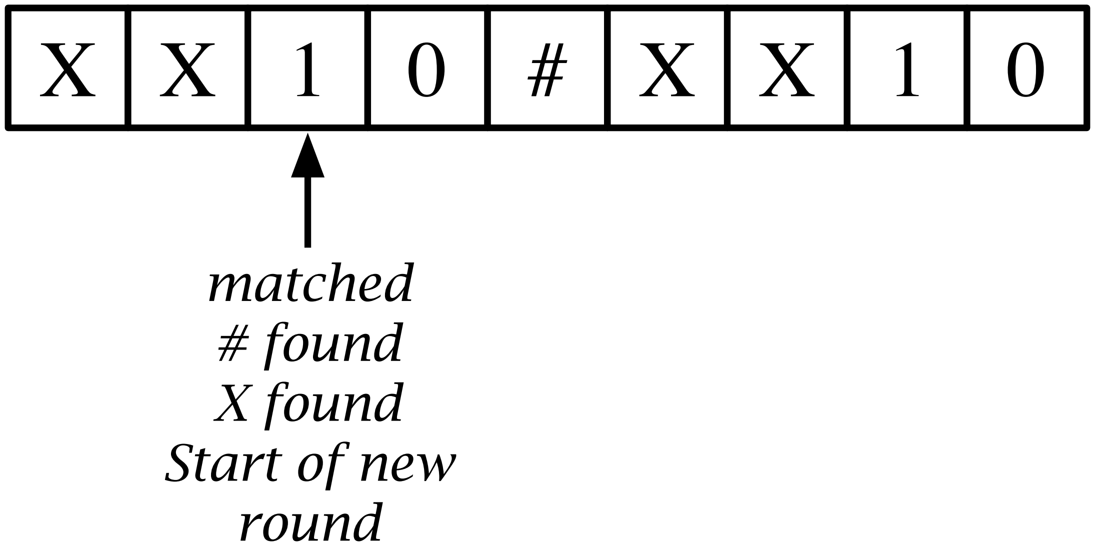
The Main Loop



The Main Loop



The Main Loop



Example 2: $\{w\#w\#w \mid w \in \{0,1\}^*\}$

- Instead of $\mathtt{x}$, use special symbols $\mathtt{x}_0$ and $\mathtt{x}_1$ used for crossing out a 0 and a 1, respectively.
- Execute the algorithm for the previous example with these substitutions and treating the second $\#$ as $\sqcup$.
- The algorithm halts in the accept state if and only if the input is in the form $w\#w\#y$ for some $w, y \in \{0,1\}^*$.
- If accept, move on to the next stage.

Example 2: $\{w\#w\#w \mid w \in \{0,1\}^*\}$

- Move the head to the symbol immediately to the right of the first $\#$.
- Execute the algorithm for the previous example by treating $\mathtt{x}_0$ as 0 and $\mathtt{x}_1$ as 1.
- Accept if and only if this simulation accepts.

Configurations

A **configuration** of a Turing machine is the setting of its tape, head, and state.

If the tape contents are $a_1, \ldots, a_m, \sqcup, \ldots,$ the head is located on the kth square, and the state is q, then write

$$a_1 \ \ldots \ a_{k-1} q a_k \ \ldots \ a_m$$

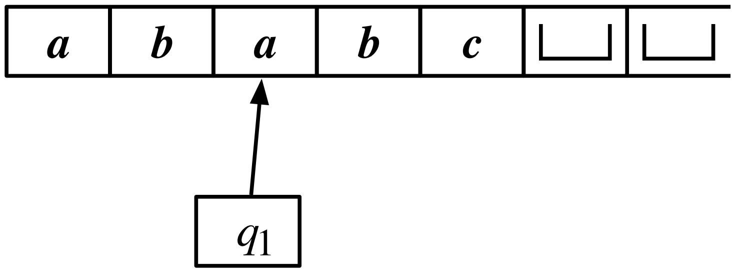
to represent the configuration as a word.

Note that Q is treated as an alphabet.

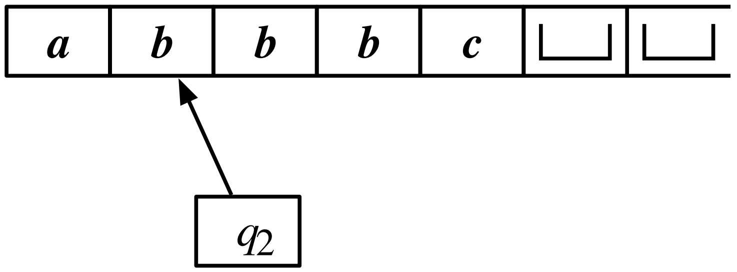
The infinite stretch of $\sqcup$'s to the right of the head position is omitted from configuration. However, at least one symbol must appear after the state symbol.

Thus a configuration is a word in $\Gamma^* Q \Gamma (\Gamma - \{\sqcup\})^*$.

Examples



The configuration is abq_1abc.



The configuration is aq_2bbbc.

Configurations (cont'd)

The action of a TM can be viewed as rewriting of the configuration.

A configuration C_1 **yields** C_2 if the Turing machine can go from C_1 to C_2 in a single step.

- uaq_ibv yields uq_jacv if $\delta(q_i, b) = (q_j, c, L)$.
- q_ibv yields q_jcv if $\delta(q_i, b) = (q_j, c, L)$ (the set cannot move to the left).
- uq_ibv yields ucq_jv' if $\delta(q_i, b) = (q_j, c, R)$.

  Here $v' = v$ if $v \neq \epsilon$ and $v = \sqcup$ if $v = \epsilon$.

Special Configurations

An accepting configuration (**A rejecting configuration**) is one in which the state is $q_{\rm accept}$ ($q_{\rm reject}$).

Both accepting configuration and rejecting configuration are **halting configurations**.