Turing Machines

Integers, Strings, and Other Things

- Data types have become very important as a programming tool.
- But at another level, there is only one type, which you may think of as integers or strings.
- Key point: Strings that are programs are just another way to think about the same one data type.

Example: Text

- Strings of ASCII or Unicode characters can be thought of as binary strings, with 8 or 16 bits/character.
- Binary strings can be thought of as integers.

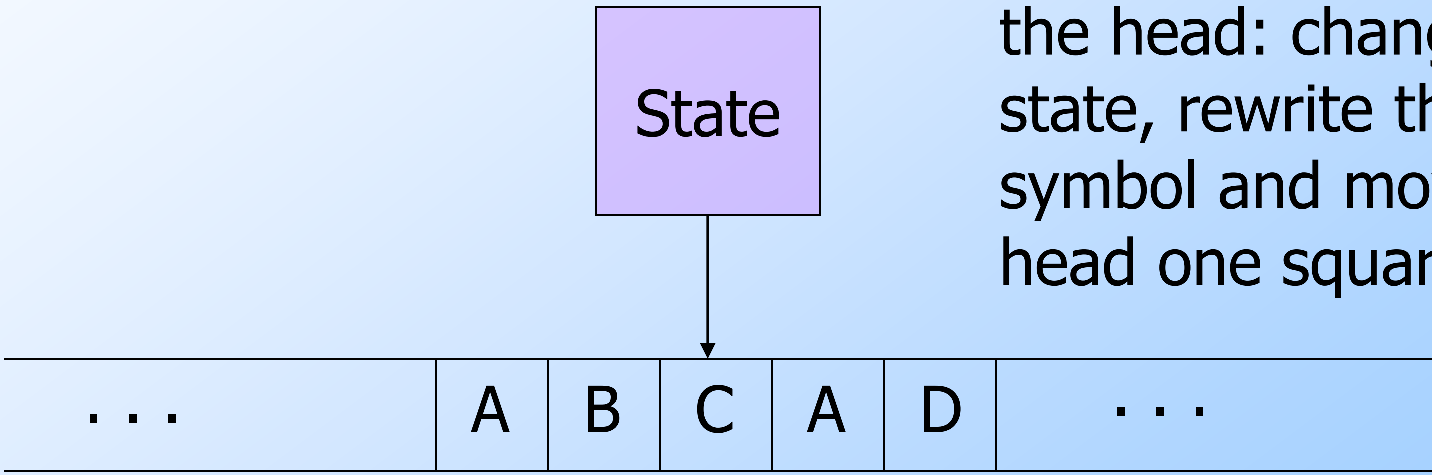
Example: Images

- Represent an image in (say) GIF.
- The GIF file is an ASCII string.
- Convert string to binary.
- Convert binary string to integer.

Example: Programs

- Programs are just another kind of data.
- Represent a program in ASCII.
- Convert to a binary string, then to an integer.

Turing Machine



Action: based on the state and the tape symbol under the head: change state, rewrite the symbol and move the head one square.

Infinite tape with squares containing tape symbols chosen from a finite alphabet

Why Turing Machines?

- Why not deal with C programs or something like that?
- Answer: You can, but it is easier to prove things about TM's, because they are so simple.
  - And yet they are as powerful as any computer.
    - More so, in fact, since they have infinite memory.

Turing-Machine Formalism

- A TM is described by:

1. A finite set of *states* (Q, typically).
2. An *input alphabet* (Σ, typically).
3. A *tape alphabet* (Γ, typically; contains Σ).
4. A *transition function* (δ, typically).
5. A *start state* (q_0, in Q, typically).
6. A *blank symbol* (B, in Γ− Σ, typically).
   - All tape except for the input is blank initially.
7. A set of *final states* (F ⊆ Q, typically).

Conventions

- a, b, … are input symbols.
- …, X, Y, Z are tape symbols.
- …, w, x, y, z are strings of input symbols.
- α, β,… are strings of tape symbols.

The Transition Function

- Takes two arguments:
  1. A state, in Q.
  2. A tape symbol in Γ.
- δ(q, Z) is either undefined or a triple of the form (p, Y, D).
  - p is a state.
  - Y is the new tape symbol.
  - D is a *direction*, L or R.

Actions of the PDA

- If $\delta(q, Z) = (p, Y, D)$ then, in state q, scanning Z under its tape head, the TM:
  1. Changes the state to p.
  2. Replaces Z by Y on the tape.
  3. Moves the head one square in direction D.
     - D = L: move left; D = R; move right.

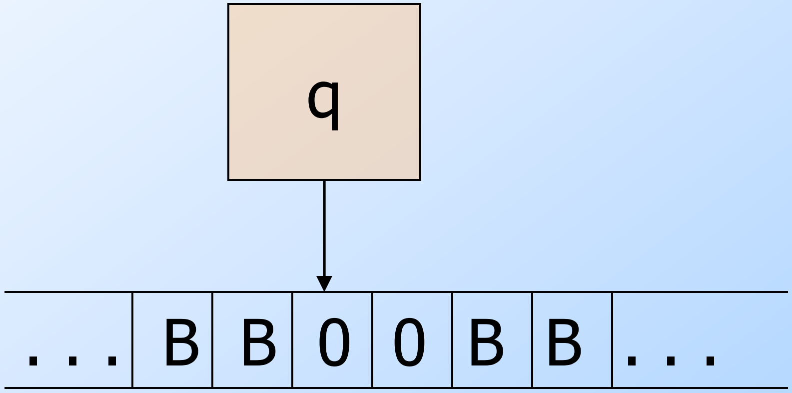
Example: Turing Machine

- This TM scans its input right, looking for a 1.
- If it finds one, it changes it to a 0, goes to final state f, and halts.
- If it reaches a blank, it changes it to a 1 and moves left.

Example: Turing Machine – (2)

- States = {q (start), f (final)}.
- Input symbols = {0, 1}.
- Tape symbols = {0, 1, B}.
- $\delta(q, 0) = (q, 0, R)$.
- $\delta(q, 1) = (f, 0, R)$.
- $\delta(q, B) = (q, 1, L)$.

Simulation of TM



$\delta(q, 0) = (q, 0, R)$

$\delta(q, 1) = (f, 0, R)$

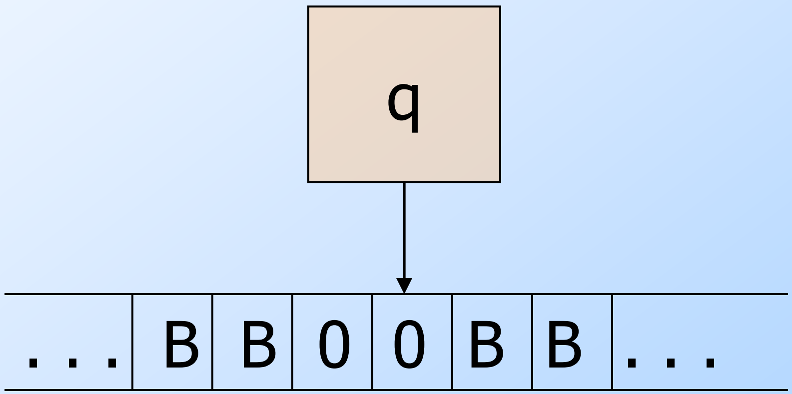
$\delta(q, B) = (q, 1, L)$

Simulation of TM

$\delta(q, 0) = (q, 0, R)$

$\delta(q, 1) = (f, 0, R)$

$\delta(q, B) = (q, 1, L)$

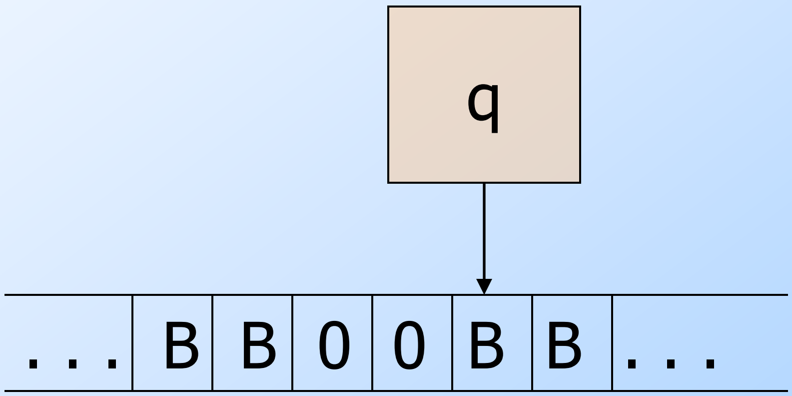


Simulation of TM

δ(q, 0) = (q, 0, R)

δ(q, 1) = (f, 0, R)

δ(q, B) = (q, 1, L)

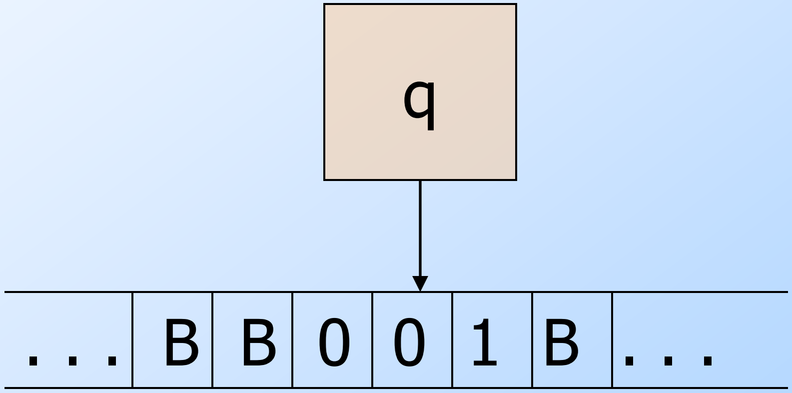


Simulation of TM

$\delta(q, 0) = (q, 0, R)$

$\delta(q, 1) = (f, 0, R)$

$\delta(q, B) = (q, 1, L)$

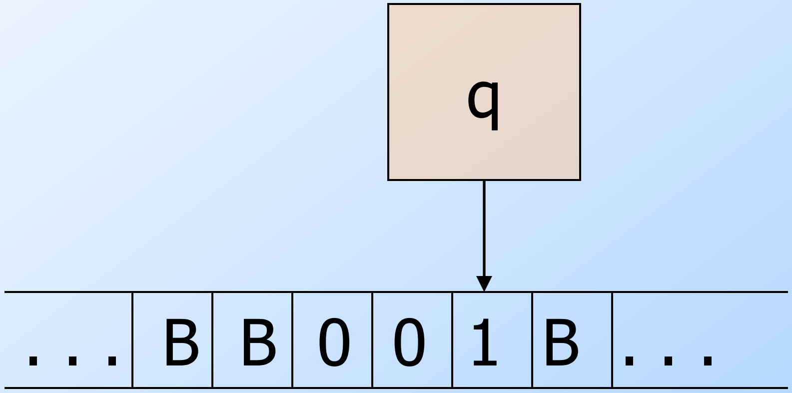


Simulation of TM

$\delta(q, 0) = (q, 0, R)$

$\delta(q, 1) = (f, 0, R)$

$\delta(q, B) = (q, 1, L)$

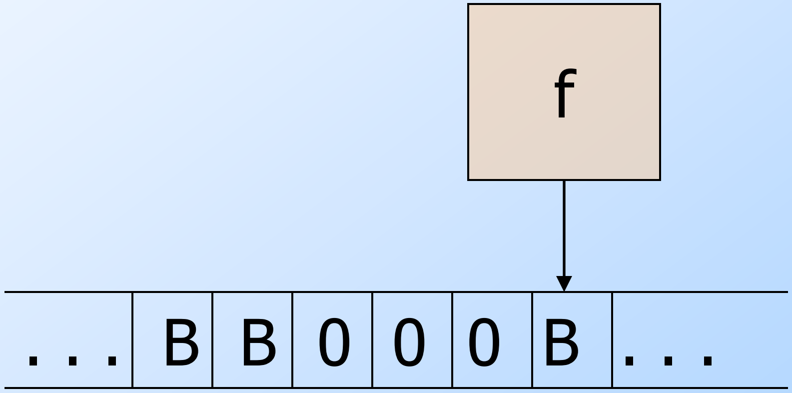


Simulation of TM

$\delta(q, 0) = (q, 0, R)$

$\delta(q, 1) = (f, 0, R)$

$\delta(q, B) = (q, 1, L)$



No move is possible. The TM halts and accepts.

Instantaneous Descriptions of a Turing Machine

- Initially, a TM has a tape consisting of a string of input symbols surrounded by an infinity of blanks in both directions.
- The TM is in the start state, and the head is at the leftmost input symbol.

TM ID's – (2)

- An ID is a string $\alpha q \beta$, where $\alpha\beta$ is the tape between the leftmost and rightmost nonblanks (inclusive).
- The state q is immediately to the left of the tape symbol scanned.
- If q is at the right end, it is scanning B.
  - If q is scanning a B at the left end, then consecutive B's at and to the right of q are part of α.

TM ID's – (3)

- As for PDA's we may use symbols ⊦ and ⊦* to represent "becomes in one move" and "becomes in zero or more moves," respectively, on ID's.
- Example: The moves of the previous TM are q00⊦0q0⊦00q⊦0q01⊦00q1⊦000f

Formal Definition of Moves

1. If $\delta(q, Z) = (p, Y, R)$, then
   - $\alpha qZ\beta \vdash \alpha Yp\beta$
   - If Z is the blank B, then also $\alpha q \vdash \alpha Yp$
2. If $\delta(q, Z) = (p, Y, L)$, then
   - For any X, $\alpha XqZ\beta \vdash \alpha pXY\beta$
   - In addition, $qZ\beta \vdash pBY\beta$

Languages of a TM

- A TM defines a language by final state, as usual.
- $L(M) = \{w \mid q_0w \vdash^* I$, where I is an ID with a final state$\}$.
- Or, a TM can accept a language by halting.
- $H(M) = \{w \mid q_0w \vdash^* I$, and there is no move possible from ID $I\}$.