Contents

Section outline

1. **Turing machines**
   - The language $a^n b^n c^n$
   - Turing machine
   - The language $w\$w$
   - TM practice problems
   - TM recognised languages

The language $a^n b^n c^n$

- Only recognising $a^n b^n$ is easy with a PDA
- $a^n b^n c^n$ is not a CFL
- For $a^n b^n c^n$, each matched pair of a and b, a special symbol $\tilde{c}$ is needed to match a c
- However, the remaining b's need to be skipped over to permit matching of the a's and b's

The language $a^n b^n c^n$

- Only recognising $a^n b^n$ is easy with a PDA
- $a^n b^n c^n$ is not a CFL
- For $a^n b^n c^n$, each matched pair of a and b, a special symbol $\tilde{c}$ is needed to match a c
- However, the remaining b's need to be skipped over to permit matching of the a's and b's
- Context sensitive grammar:
  $S \rightarrow aSBC \mid \epsilon \quad CB \rightarrow BC \quad B \rightarrow b \quad C \rightarrow c$

The language $a^n b^n c^n$

- Only recognising $a^n b^n$ is easy with a PDA
- $a^n b^n c^n$ is not a CFL
- For $a^n b^n c^n$, each matched pair of a and b, a special symbol $\tilde{c}$ is needed to match a c
- However, the remaining b's need to be skipped over to permit matching of the a's and b's
- Context sensitive grammar:
  $S \to aSBC \mid \epsilon \quad CB \to BC \quad B \to b \quad C \to c$
- Workout for $a^3 b^3 c^3$ ...

The language $a^n b^n c^n$

- Only recognising $a^n b^n$ is easy with a PDA
- $a^n b^n c^n$ is not a CFL
- For $a^n b^n c^n$, each matched pair of a and b, a special symbol $\tilde{c}$ is needed to match a c
- However, the remaining b's need to be skipped over to permit matching of the a's and b's
- Context sensitive grammar:
  $S \rightarrow aSBC \,|\, \epsilon \quad CB \rightarrow BC \quad B \rightarrow b \quad C \rightarrow c$
- Workout for $a^3 b^3 c^3$ ...
- A m/c for recognising a palindrome without a mid-marker ... ?

The language $a^n b^n c^n$

- Only recognising $a^n b^n$ is easy with a PDA
- $a^n b^n c^n$ is not a CFL
- For $a^n b^n c^n$, each matched pair of a and b, a special symbol $\tilde{c}$ is needed to match a c
- However, the remaining b's need to be skipped over to permit matching of the a's and b's
- Context sensitive grammar:
  $S \to aSBC \mid \epsilon \quad CB \to BC \quad B \to b \quad C \to c$
- Workout for $a^3 b^3 c^3$ ...
- A m/c for recognising a palindrome without a mid-marker ... ?
- PDA cannot accommodate above requirements ... what extra features are needed?

The language $a^n b^n c^n$

- Only recognising $a^n b^n$ is easy with a PDA
- $a^n b^n c^n$ is not a CFL
- For $a^n b^n c^n$, each matched pair of a and b, a special symbol $\tilde{c}$ is needed to match a c
- However, the remaining b's need to be skipped over to permit matching of the a's and b's
- Context sensitive grammar:
  $S \rightarrow aSBC \mid \epsilon \quad CB \rightarrow BC \quad B \rightarrow b \quad C \rightarrow c$
- Workout for $a^3 b^3 c^3$ ...
- A m/c for recognising a palindrome without a mid-marker ... ?
- PDA cannot accommodate above requirements ... what extra features are needed?
  - Ability to move back and forth over inputs
  - Having inputs available in advance
  - Ability to access stored data in any order

The language $a^n b^n c^n$

- Only recognising $a^n b^n$ is easy with a PDA
- $a^n b^n c^n$ is not a CFL
- For $a^n b^n c^n$, each matched pair of a and b, a special symbol $\tilde{c}$ is needed to match a c
- However, the remaining b's need to be skipped over to permit matching of the a's and b's
- Context sensitive grammar:
  $S \rightarrow aSBC \mid \epsilon \quad CB \rightarrow BC \quad B \rightarrow b \quad C \rightarrow c$
- Workout for $a^3 b^3 c^3$ ...
- A m/c for recognising a palindrome without a mid-marker ... ?
- PDA cannot accommodate above requirements ... what extra features are needed?
  - Ability to move back and forth over inputs
  - Having inputs available in advance
  - Ability to access stored data in any order
- Another kind of m/c needed with above capabilities

Turing machine

A Turing machine is a 7-tuple $\langle Q, \Sigma, \Gamma, \delta, q_i, q_a, q_r \rangle$, where

Q is a finite set of states

- $q_{\mathcal{I}}$ is the initial state, $q_{\mathcal{I}} \in Q$
- q_a is the *accepting* or *final* state, also written q_{acc}
- q_r is the *rejecting* state, also written q_{rej}

Σ is a finite input alphabet

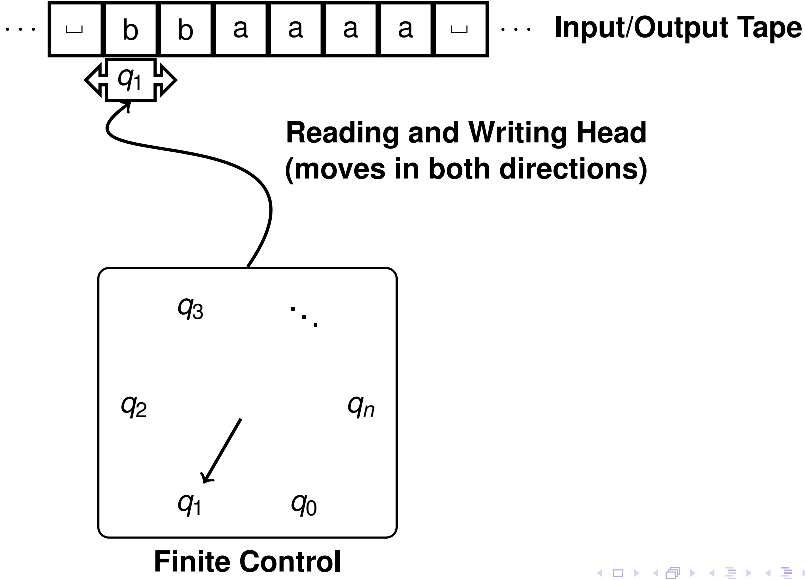
Γ is a finite tape alphabet; $\Sigma \subseteq \Gamma$

Γ has some extra symbols for convenience, such as $\sqcup$ a special blank character, useful for marking the end of the input

δ $(Q \setminus \{q_a, q_r\}) \times \Gamma \to Q \times \Gamma \times \{L, R\}$ is the transition function

For example, if $\delta \langle q, c \rangle = \langle q', c', L \rangle$ means that, if the TM is at state q and the head on the tape reads the character c, then it should move to state q', replace c on the tape by c' and move the head on the tape to the left

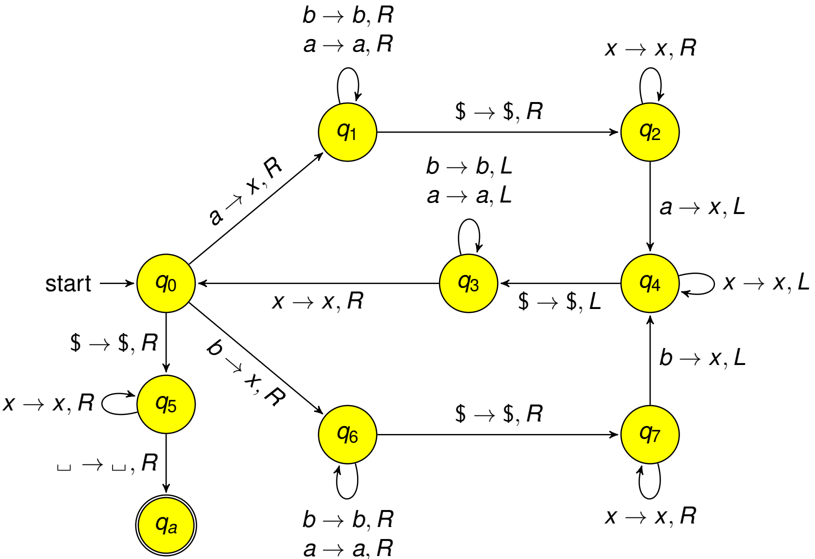
Schematic diagram of TM



The language $w\$w$

- For $\Sigma = \{a, b, \$\}$, consider the non-CFL $L = \left\{ w\$w \,\middle|\, w \in \Sigma^* \right\}$
- A TM algorithm (but possibly not strictly conforming) for reognizing L works as follows, starting at the first character:
  1. Read the character (a or b), call it be u, and replace it with with x (some special character) and remember what character was crossed off by transitioning to a different state
  2. Move right until a \$ is seen
  3. Read aross the sequence of 0 or more x's following the \$
  4. Read the character (not x) on the tape
  5. Depending on the current state, if it does not match with u, immediately reject
  6. Otherwise, replace it with x
  7. Move left and keep going until x is seen on the tape
  8. Move one position right, if the character is \$ skip over next step
  9. Otherwise, continue from the first step
  10. Skip over the run of x's
  11. If ␣ found, accept

TM for $w\$w$


$b \to b, R$
$a \to a, R$
$x \to x, R$
q_1
$\$ \to \$, R$
q_2
$a \to x, R$
$b \to b, L$
$a \to a, L$
$a \to x, L$
start
q_0
$x \to x, R$
q_3
$\$ \to \$, L$
q_4
$x \to x, L$
$\$ \to \$, R$
$b \to x, R$
$b \to x, L$
$x \to x, R$
q_5
q_6
$\$ \to \$, R$
q_7
$\sqcup \to \sqcup, R$
q_a
$b \to b, R$
$a \to a, R$
$x \to x, R$

TM for $w\$w$ (contd.)

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_a, q_r\}$
- $\Sigma = \{a, b, \$\}$
- $\Gamma = \Sigma \cup \{\sqcup, x\}$
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$

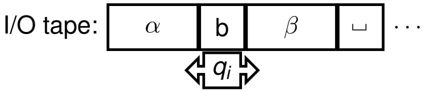
	a	b	$\$$	$\sqcup$	x
q_0	$\langle q_1, x, R\rangle$	$\langle q_6, x, R\rangle$	$\langle q_5, x, R\rangle$	reject	reject
q_1	$\langle q_1, a, R\rangle$	$\langle q_1, b, R\rangle$	$\langle q_2, \$, R\rangle$	reject	reject
q_2	$\langle q_4, x, L\rangle$	reject	reject	reject	$\langle q_2, x, R\rangle$
q_3	$\langle q_3, a, L\rangle$	$\langle q_3, b, L\rangle$	reject	reject	$\langle q_0, x, R\rangle$
q_4	reject	reject	$\langle q_3, \$, L\rangle$	reject	$\langle q_4, x, L\rangle$
q_5	reject	reject	reject	$\langle q_a, \sqcup, R\rangle$	$\langle q_5, x, R\rangle$
q_6	$\langle q_6, a, R\rangle$	$\langle q_6, b, R\rangle$	$\langle q_7, \$, R\rangle$	reject	reject
q_7	reject	$\langle q_4, x, L\rangle$	reject	reject	$\langle q_7, x, R\rangle$
q_a	No need to define				
q_r	No need to define				

TM practice problems

- Construct a TM to recognise 01^*0
- Construct a TM to recognise $a^n b^n$
- Construct a TM to recognise $a^n b^n c^n$
- Construct a TM to duplicate the string w to $w\#w$
- Construct a TM to recognise $w\#w$
- Construct a TM to recognise ww
- Construct a TM that takes its input on the tape, shifts it to the right by one position, and put a \$ on the leftmost position on the tape when $\Sigma = \{a, b\}$
- Construct a TM that takes its input on the tape, shifts it to the right by one position, and put a \$ on the leftmost position on the tape when $\Sigma = \{a, b, \ldots, z\}$

TM configurations

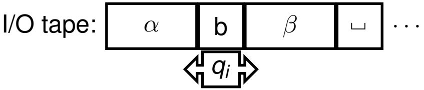
- A configuration of a TM is $\langle x, q, k \rangle \in \Sigma^* \times K \times N$, where x denotes the string on the tape, q denotes the current state of the TM, and k denotes the position of the machine on the tape
- The string x should be well demarcated so that it may start with a $\triangleright$ and end with $\sqcup$.
- The position k is required to satisfy $0 \leq k < |x|$
- Simpler variations of this definition may be used



- Configuration (simpler): $\langle \alpha, q_i, \mathrm{b}\beta \rangle$
- Initial configuration: $\langle \epsilon, q_{\mathcal{I}}, w \rangle$
- Accepting configuration: $\langle \alpha, q_a, \beta \rangle$
- Rejecting configuration: $\langle \alpha, q_r, \beta \rangle$

TM configurations

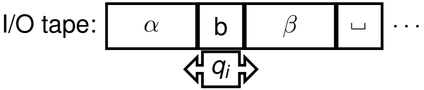
- A configuration of a TM is $\langle x, q, k \rangle \in \Sigma^* \times K \times N$, where x denotes the string on the tape, q denotes the current state of the TM, and k denotes the position of the machine on the tape
- The string x should be well demarcated so that it may start with a $\triangleright$ and end with $\sqcup$.
- The position k is required to satisfy $0 \leq k < |x|$
- Simpler variations of this definition may be used



- Configuration (simpler): $\langle \alpha, q_i, \mathrm{b}\beta \rangle$
- Initial configuration: $\langle \epsilon, q_{\mathcal{I}}, w \rangle$
- Accepting configuration: $\langle \alpha, q_a, \beta \rangle$
- Rejecting configuration: $\langle \alpha, q_r, \beta \rangle$
- Let TM config be: $c_1 = \langle \alpha, q_i, \mathrm{a}\beta \rangle$
- Let $\delta(\langle q_i, \mathrm{a} \rangle) = \langle q_j, \mathrm{c}, \mathrm{R} \rangle$
- Resulting transition: $\langle \alpha, q_i, \mathrm{a}\beta \rangle \Rightarrow \langle \alpha\mathrm{c}, q_j, \beta \rangle = c_2$
- We say c_1 yields c_2; $c_1 \mapsto c_2$

TM configurations

- A configuration of a TM is $\langle x, q, k\rangle \in \Sigma^* \times K \times N$, where x denotes the string on the tape, q denotes the current state of the TM, and k denotes the position of the machine on the tape
- The string x should be well demarcated so that it may start with a $\triangleright$ and end with $\sqcup$.
- The position k is required to satisfy $0 \leq k < |x|$
- Simpler variations of this definition may be used



- Configuration (simpler): $\langle \alpha, q_i, \mathrm{b}\beta\rangle$
- Initial configuration: $\langle \epsilon, q_{\mathcal{I}}, w\rangle$
- Accepting configuration: $\langle \alpha, q_a, \beta\rangle$
- Rejecting configuration: $\langle \alpha, q_r, \beta\rangle$
- Let TM config be: $c_1 = \langle \alpha, q_i, \mathrm{a}\beta\rangle$
- Let $\delta(\langle q_i, \mathrm{a}\rangle) = \langle q_j, \mathrm{c}, \mathrm{R}\rangle$
- Resulting transition: $\langle \alpha, q_i, \mathrm{a}\beta\rangle \Rightarrow \langle \alpha\mathrm{c}, q_j, \beta\rangle = c_2$
- We say c_1 yields c_2; $c_1 \mapsto c_2$
- For what transition? $\langle \gamma\mathrm{d}, q_k, \mathrm{e}\tau\rangle \Rightarrow \langle \gamma, q_m, \mathrm{df}\tau\rangle$

TM recognised languages

Definition (TM acceptance)

For a TM M and a string w, the Turing machine M accepts w if there is a sequence of configurations, $c_1, c_2, \ldots, c_k$ such that:

- $c_1 = \langle \epsilon, q_{\mathcal{I}}, w \rangle$, $q_{\mathcal{I}}$ being the start state of M
- for al i, $1 \leq i < k$, $c_i \mapsto c_{i+1}$
- c_k is the accepting configuration

Definition (TM language)

The language of a TM M is $L(M) = \{w \mid M \text{ accepts } w\}$; such a language L is called Turing recognisable

TM recognised languages

Definition (TM acceptance)

For a TM M and a string w, the Turing machine M accepts w if there is a sequence of configurations, $c_1, c_2, \ldots, c_k$ such that:

- $c_1 = \langle \epsilon, q_{\mathcal{I}}, w \rangle$, $q_{\mathcal{I}}$ being the start state of M
- for al i, $1 \leq i < k$, $c_i \mapsto c_{i+1}$
- c_k is the accepting configuration

Definition (TM language)

The language of a TM M is $L(M) = \{w \mid M \text{ accepts } w\}$; such a language L is called Turing recognisable

Outcomes of running a TM on w

- accepts w (halts)
- rejects w (halts)
- loops indefinitely

- A TM halting on all inputs is a decider
- A language is TM decidable if there is a decider TM M such that $L(M) = L$

Some aliases

Aliases of Turing recognisable languages:

- recursively enumerable
- partially decidable
- semidecidable
- Turing-acceptable

Aliases of Turing decidable languages:

- recursive