Turing Machines

CENG 280, 2019

Extensions of Turing Machines

- Multi-tape TM
- Multi-head TM
- Two way infinite tape TM
- Non-deterministic TM

Multi-tape TM

Standard TM $M = (K, \Sigma, \delta, s, H)$

$\delta : (K \setminus H) \times \Sigma \to K \times (\Sigma \cup \{\leftarrow, \rightarrow\})$
Configuration $\in K \times \triangleright\Sigma^{\star} \times (\Sigma^{\star}(\Sigma \setminus \{\sqcup\}) \cup \{e\})$

Multi-tape TM

Standard TM $M = (K, \Sigma, \delta, s, H)$

$\delta : (K \setminus H) \times \Sigma \to K \times (\Sigma \cup \{\leftarrow, \rightarrow\})$
Configuration $\in K \times \rhd\Sigma^\star \times (\Sigma^\star(\Sigma \setminus \{\sqcup\}) \cup \{e\})$

Definition ($k - tapeTM : M = (K, \Sigma, \delta, s, H)$)

$\delta : (K \setminus H) \times \Sigma^k \to K \times (\Sigma \cup \{\leftarrow, \rightarrow\})^k$
$\delta(q, (a_1, \ldots, a_k)) = (p, (b_1, \ldots, b_k))$
Configuration $\in K \times (\rhd\Sigma^\star \times (\Sigma^\star(\Sigma \setminus \{\sqcup\}) \cup \{e\}))^k$

Multi-tape TM

Standard TM $M = (K, \Sigma, \delta, s, H)$

$\delta : (K \setminus H) \times \Sigma \to K \times (\Sigma \cup \{\leftarrow, \rightarrow\})$
Configuration $\in K \times \rhd\Sigma^\star \times (\Sigma^\star(\Sigma \setminus \{\sqcup\}) \cup \{e\})$

Definition $(k - tapeTM : M = (K, \Sigma, \delta, s, H))$

$\delta : (K \setminus H) \times \Sigma^k \to K \times (\Sigma \cup \{\leftarrow, \rightarrow\})^k$
$\delta(q, (a_1, \ldots, a_k)) = (p, (b_1, \ldots, b_k))$
Configuration $\in K \times (\rhd\Sigma^\star \times (\Sigma^\star(\Sigma \setminus \{\sqcup\}) \cup \{e\}))^k$

Example

Define a two tape copy machine.

Multi-tape TM

Standard TM $M = (K, \Sigma, \delta, s, H)$

$\delta : (K \setminus H) \times \Sigma \to K \times (\Sigma \cup \{\leftarrow, \rightarrow\})$
Configuration $\in K \times \rhd\Sigma^\star \times (\Sigma^\star(\Sigma \setminus \{\sqcup\}) \cup \{e\})$

Definition ($k - tapeTM : M = (K, \Sigma, \delta, s, H)$)

$\delta : (K \setminus H) \times \Sigma^k \to K \times (\Sigma \cup \{\leftarrow, \rightarrow\})^k$
$\delta(q, (a_1, \ldots, a_k)) = (p, (b_1, \ldots, b_k))$
Configuration $\in K \times (\rhd\Sigma^\star \times (\Sigma^\star(\Sigma \setminus \{\sqcup\}) \cup \{e\}))^k$

Example

Define a two tape copy machine.

Example

Define a 2-tape machine that adds two binary numbers.

Multi-tape TM

Theorem

Given a k-tape TM $M = (K, \Sigma, \delta, s, H)$, there exists a standard TM $M' = (K', \Sigma', \delta', s', H')$ where $\Sigma \subseteq \Sigma'$ with the same functionality (i.e. compute the same function, or decide/semi-decide the same language).

In particular, for any input string $x \in \Sigma^\star$ in the first tape of M, M halts with output y on its first tape if and only if M' halts on x with output y.

If M halts on input x after t steps, then M' halts on it after a number of steps in the order of $O(t(|x| + t))$.

Multi-tape TM

Theorem

Given a k-tape TM $M = (K, \Sigma, \delta, s, H)$, there exists a standard TM $M' = (K', \Sigma', \delta', s', H')$ where $\Sigma \subseteq \Sigma'$ with the same functionality (i.e. compute the same function, or decide/semi-decide the same language).

In particular, for any input string $x \in \Sigma^\star$ in the first tape of M, M halts with output y on its first tape if and only if M' halts on x with output y.

If M halts on input x after t steps, then M' halts on it after a number of steps in the order of $O(t(|x| + t))$.

Corollary

Any function that is computed or any language that is decided or semidecided by a k-tape TM is also computed, decided or semi-decided by a single tape TM, respectively.

Multi-tape TM

Theorem

Given a k-tape TM $M = (K, \Sigma, \delta, s, H)$, there exists a standard TM $M' = (K', \Sigma', \delta', s', H')$ where $\Sigma \subseteq \Sigma$ with the same functionality (i.e. compute the same function, or decide/semi-decide the same language).

...

- In multi-tape TM, all heads over the tapes can write, move left or right.
- To simulate this machine, divide the tape into tracks (2k).
- (figure), $\Sigma' = \Sigma \cup (\Sigma \cup \{0, 1\})^k$.
- For each tape, 2 tracks: the first one is for the contents of the track, the second one is to record the position of the head over the track ($(a_1, b_1, ..., a_k, b_k)$ with $a_i \in \Sigma$, $b_i \in \{0, 1\}$ is a symbol of Σ')
- For each transition of M, M' scans its tape twice:
  - Find the symbols under the heads (update the state to remember them)
  - Change them, or move heads

Multi-tape TM

Theorem

Given a k-tape TM $M = (K, \Sigma, \delta, s, H)$, there exists a standard TM $M' = (K', \Sigma', \delta', s', H')$ where $\Sigma \subseteq \Sigma$ with the same functionality (i.e. compute the same function, or decide/semi-decide the same language).

...

Alternative approach

- Show all tapes on the single tape with separators (e.g. $| \notin \Sigma$)
- Mark the symbol under the reading head, $\Sigma' = \Sigma \cup \{|\} \cup \{\dot{a} \mid a \in \Sigma\}$
- For each transition of M, M' scans the tape twice:
  - Find the symbols under the heads
  - Change them, or move heads

Two-way Infinite Tape TM

- Suppose that the machine has a tape that is infinite in both directions.
- All squares are initially blank except the input.
- No $\triangleright$ to mark the left end.

Two-way Infinite Tape TM

- Suppose that the machine has a tape that is infinite in both directions.
- All squares are initially blank except the input.
- No $\triangleright$ to mark the left end.
- Is this machine more powerful?

Two-way Infinite Tape TM

- Suppose that the machine has a tape that is infinite in both directions.
- All squares are initially blank except the input.
- No $\triangleright$ to mark the left end.
- Is this machine more powerful?
- Two-way Infinite Tape can be simulated by a 2-tape TM, that can be simulated by a standard TM.

Single tape - multi head TM

- Multiple heads on a single tape
- In a single transition, all heads can move, read/write.
- Special attention if two heads tries to write on the same cell.

Single tape - multi head TM

- Multiple heads on a single tape
- In a single transition, all heads can move, read/write.
- Special attention if two heads tries to write on the same cell.
- Can we simulate it with a standard TM ?

Single tape - multi head TM

- Multiple heads on a single tape
- In a single transition, all heads can move, read/write.
- Special attention if two heads tries to write on the same cell.
- Can we simulate it with a standard TM ?
- Yes, the construction would be similar to k-tape machine.
- Additional *k* tracks to record the head positions, or mark the cells under the reading heads (a special symbol for each head combination)

Two dimensional tape TM

- The tape is a two dimensional grid.
- The single reading head can move left, right, up and down.
- Mark the top end the left end (similar to $\triangleright$)

Two dimensional tape TM

- The tape is a two dimensional grid.
- The single reading head can move left, right, up and down.
- Mark the top end the left end (similar to $\triangleright$)
- Can we simulate it with a standard TM ?

Two dimensional tape TM

- The tape is a two dimensional grid.
- The single reading head can move left, right, up and down.
- Mark the top end the left end (similar to $\triangleright$)
- Can we simulate it with a standard TM ?
- Yes, (i, j, a) location and symbol
- Simulate it with 3-tape TM, first tape contains the input (in list format)
- The second tape i (location)
- The third tape j (location)

Extensions of Turing Machine

Theorem

Any language decided or semidecided, and any function computed by Turing machines with several tapes, several heads, two-way infinite tapes, or multi-dimensional tapes can be decided, semidecided or computed by a standard Turing machine.

Non-deterministic Turing Machine

Definition (Non-deterministic TM)

A non-deterministic Turing machine is a quintuple $M = (K, \Sigma, \Delta, s, H)$ where K, Σ, s and H are as in the standard TM definition and $\Delta \subseteq (K \setminus H) \times \Sigma \times K \times (\Sigma \cup \{\leftarrow, \rightarrow\})$ is a transition relation. At state q when the input symbol is a, the machine non-deterministically chooses a state and an action (move or write) from the set $\{(p, b) \mid (q, a, p, b) \in \Delta\}$.

Non-deterministic Turing Machine

Definition (Non-deterministic TM)

A non-deterministic Turing machine is a quintuple $M = (K, \Sigma, \Delta, s, H)$ where K, Σ, s and H are as in the standard TM definition and $\Delta \subseteq (K \setminus H) \times \Sigma \times K \times (\Sigma \cup \{\leftarrow, \rightarrow\})$ is a transition relation. At state q when the input symbol is a, the machine non-deterministically chooses a state and an action (move or write) from the set $\{(p, b) \mid (q, a, p, b) \in \Delta\}$.

The yields relation $\vdash_M$ for the configurations of non-deterministic TM is defined w.r. to Δ.

Non-deterministic Turing Machine

Definition (Non-deterministic TM)

A non-deterministic Turing machine is a quintuple $M = (K, \Sigma, \Delta, s, H)$ where K, Σ, s and H are as in the standard TM definition and $\Delta \subseteq (K \setminus H) \times \Sigma \times K \times (\Sigma \cup \{\leftarrow, \rightarrow\})$ is a transition relation. At state q when the input symbol is a, the machine non-deterministically chooses a state and an action (move or write) from the set $\{(p, b) \mid (q, a, p, b) \in \Delta\}$.

The yields relation $\vdash_M$ for the configurations of non-deterministic TM is defined w.r. to Δ.

Definition

Let $M = (K, \Sigma, \Delta, s, H)$ be a non-deterministic TM. M **accepts** an input $w \in \Sigma^\star$, if $(s, \triangleright\underline{\sqcup}w) \vdash_M^\star (h, \triangleright u\underline{a}v)$ for some $h \in H$ (one-halting configuration is sufficient). M **semi-decides** a language $L \subseteq (\Sigma \setminus \{\triangleright, \sqcup\})^\star$ if the following holds for all $w \in (\Sigma \setminus \{\triangleright, \sqcup\})^\star$: $w \in L$ iff M accepts w.

Non-deterministic Turing Machine

Definition

Let $M = (K, \Sigma, \Delta, s, H)$ be a non-deterministic TM. M **accepts** an input $w \in \Sigma^\star$, if $(s, \triangleright\underline{\sqcup}w) \vdash_M^\star (h, \triangleright u\underline{a}v)$ for some $h \in H$ (one-halting configuration is sufficient). M **semi-decides** a language $L \subseteq (\Sigma \setminus \{\triangleright, \sqcup\})^\star$ if the following holds for all $w \in (\Sigma \setminus \{\triangleright, \sqcup\})^\star$: $w \in L$ iff M accepts w.

Example

$L = \{1^n \mid n \text{ is composite}\}$. Design a non-deterministic TM that semidecides L.

Non-deterministic Turing Machine

Definition

Let $M = (K, \Sigma, \Delta, s, H)$ be a non-deterministic TM. M **accepts** an input $w \in \Sigma^\star$, if $(s, \triangleright\underline{\sqcup}w) \vdash_M^\star (h, \triangleright u\underline{a}v)$ for some $h \in H$ (one-halting configuration is sufficient). M **semi-decides** a language $L \subseteq (\Sigma \setminus \{\triangleright, \sqcup\})^\star$ if the following holds for all $w \in (\Sigma \setminus \{\triangleright, \sqcup\})^\star$: $w \in L$ iff M accepts w.

Example

$L = \{1^n \mid n \text{ is composite}\}$. Design a non-deterministic TM that semidecides L.

A deterministic TM that decides L?

Non-deterministic Turing Machine

Example

$L = \{1^n \mid n \text{ is composite}\}$. Design a non-deterministic TM that semidecides L. A deterministic TM that decides L?

Definition

Let $M = (K, \Sigma, \Delta, s, \{y, n\})$ be a non-deterministic TM. M **decides** a language $L \subseteq (\Sigma \setminus \{\rhd, \sqcup\})^\star$ if the following two conditions hold for all $w \in (\Sigma \setminus \{\rhd, \sqcup\})^\star$:

- There is a natural number N depending on M and w such that the computation of M on w always halts in less than N steps
- $w \in L$ iff $(s, \rhd\underline{\sqcup}w) \vdash_M^\star (y, \rhd u\underline{a}v)$.

Non-deterministic Turing Machine

Definition

Let $M = (K, \Sigma, \Delta, s, \{h\})$ be a non-deterministic TM. M **computes** a function $f : (\Sigma \setminus \{\rhd, \sqcup\})^\star \to (\Sigma \setminus \{\rhd, \sqcup\})^\star$ if the following two conditions hold for all $w \in (\Sigma \setminus \{\rhd, \sqcup\})^\star$:

- There is a natural number N depending on M and w such that the computation of M on w always halts in less than N steps
- $(s, \rhd\underline{\sqcup}w) \vdash_M^\star (h, \rhd u\underline{a}v)$ iff $ua = \sqcup$ and $f(w) = v$

Non-deterministic Turing Machine

Theorem

If a non-deterministic TM decides or semidecides a language or computes a function, then there is a standard TM that deciding or semideciding the same language or computing the same function.

Non-deterministic Turing Machine

Theorem

If a non-deterministic TM decides or semidecides a language or computes a function, then there is a standard TM that deciding or semideciding the same language or computing the same function.

- The proof uses a 3-tape deterministic TM.

Non-deterministic Turing Machine

Theorem

If a non-deterministic TM decides or semidecides a language or computes a function, then there is a standard TM that deciding or semideciding the same language or computing the same function.

- The proof uses a 3-tape deterministic TM.
- The first tape always contains the input word.

Non-deterministic Turing Machine

Theorem

If a non-deterministic TM decides or semidecides a language or computes a function, then there is a standard TM that deciding or semideciding the same language or computing the same function.

- The proof uses a 3-tape deterministic TM.
- The first tape always contains the input word.
- A deterministic version of M operates on the second tape according to the string on the third tape.

Non-deterministic Turing Machine

Theorem

If a non-deterministic TM decides or semidecides a language or computes a function, then there is a standard TM that deciding or semideciding the same language or computing the same function.

- The proof uses a 3-tape deterministic TM.
- The first tape always contains the input word.
- A deterministic version of M operates on the second tape according to the string on the third tape.
- The ith symbol on the third tape solves the non-determinism on the i-th transition.

Non-deterministic Turing Machine

Theorem

If a non-deterministic TM decides or semidecides a language or computes a function, then there is a standard TM that deciding or semideciding the same language or computing the same function.

- The proof uses a 3-tape deterministic TM.
- The first tape always contains the input word.
- A deterministic version of M operates on the second tape according to the string on the third tape.
- The ith symbol on the third tape solves the non-determinism on the i-th transition.
- If the end of the string on the third tape is reached or the ith symbol is higher than all possible transitions to take, end the computation, copy the input word again, find the next string (lexicographically), and restart the computation on the second tape.