HW1CSDS343

Collaborators

Compared Turing Machines in Q3 with Carson Whitehouse

1

Let L_1 and L_2 be decidable languages over the same alphabet Σ. Consider language $L = L_1 \oplus L_2$ Prove that L is decidable.

Ans

Assume L_1, L_2 are decidable languages
$\exists A_1$ that decides L_1
$\exists A_2$ that decides L_2
Create A_3
A_3 runs on x:

- Run A_1 on x
- If A_1 accepts:
  - Run A_2 on x
  - If A_2 accepts:
    - Output "no"
  - Else:
    - Output "yes"
- Else:
  - Run A_2 on x
  - If A_2 accepts:
    - Output "yes:
  - Else:
    - Output "no"

Proof

Show A_3 decides L
$L = L_1 \oplus L_2$
If $x \in L$ then $(x \in L_1 \wedge x \notin L_2) \vee (x \notin L_1 \wedge x \in L_2)$

- If $x \in L_1 \land x \notin L_2$
  - A_3 will run A_1 which accepts, then it will run A_2 which rejects. So A_3 accepts.
- If $x \notin L_1 \land x \in L_2$
  - A_3 will run A_1 which rejects, then it will run A_2 which accepts. So A_3 accepts.

If $x \notin L$ then $(x \in L_1 \land x \in L_2) \lor (x \notin L_1 \land x \notin L_2)$

- If $x \in L_1 \land x \in L_2$
  - A_3 will run A_1 which accepts, then it will run A_2 which accepts. So A_3 rejects.
- If $x \notin L_1 \land x \notin L_2$
  - A_3 will run A_1 which rejects, then it will run A_2 which rejects. So A_3 rejects.

2

Let L be a language over alphabet Σ. Prove that if both L and $\bar{L}$ (the complement of L) are recognizable, then L is decidable.

Ans

Assume L_1, L_2 are recognizable languages
$\exists A_1$ that recognizes L_1
$\exists A_2$ that recognizes L_2
Create A_3
A_3 runs on x:

- For $i = 0, 1, 2, \ldots$:
  - Run A_1 on x
  - If A_1 accepts:
    - Ouput "yes"
  - Else:
    - Run A_2 on x
    - If A_2 accepts:
      - Ouput "no"

Proof

Show A_3 decides L
If $x \in L$ then A_3 runs A_1 which accepts in a finite number of steps. So A_3 will output "yes" in a finite number of steps.
If $x \notin L$ then $x \in \bar{L}$ so A_3 runs A_2 which accepts in a finite number of steps. So A_3 will output "no" in a finite number of steps.

3

Let L be the set of all strings over the alphabet $\Sigma = \{a, b, c, d\}$ defined as $L = \{a^n b^m c^{\max\{n-m,0\}} d^{\max\{m-n,0\}}\}$ for n, m non-negative integers. For example, $aaabbc$ and $aabbbd$ are both strings of the language. (This is basically doing the subtraction $n - m$). Write a Turing machine that will accept all strings that are in L and reject all other strings. Explicitly give your machine's alphabet, set of states, and transition function.

Create a TM for L

$L = \{a^n b^m c^{\max\{n-m,0\}} d^{\max\{m-n,0\}}\}$
$\Sigma = \{a, b, c, d\}$
$\Gamma = \{a, b, c, d, \_, a', b', c', d'\}$
$Q = \{q_0, q_{reject}, q_{accept}, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_b, q_c\}$
$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$

q_0 -- init

$\delta(q_0, a) = (q_1, a', R)$
$\delta(q_0, b) = (q_7, b', R)$ -- More b's than a's
$\delta(q_0, c) = (q_{reject}, \_, R)$
$\delta(q_0, d) = (q_{reject}, \_, R)$
$\delta(q_0, \_) = (q_{accept}, \_, R)$
$\delta(q_0, a') = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_0, b') = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_0, c') = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_0, d') = (q_{reject}, \_, R)$ * This should not happen

q_1 -- Match b's to a's (R)

$\delta(q_1, a) = (q_1, a, R)$
$\delta(q_1, b) = (q_2, b', R)$
$\delta(q_1, c) = (q_4, c', L)$ -- More a's than b's
$\delta(q_1, d) = (q_{reject}, \_, R)$
$\delta(q_1, \_) = (q_{reject}, \_, R)$
$\delta(q_1, a') = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_1, b') = (q_1, b', R)$
$\delta(q_1, c') = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_1, d') = (q_{reject}, \_, R)$ * This should not happen

q_2 -- Find next a (a vs b) (L)

$\delta(q_2, a) = (q_2, a, L)$
$\delta(q_2, b) = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_2, c) = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_2, d) = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_2, \_) = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_2, a') = (q_3, a', R)$
$\delta(q_2, b') = (q_2, b', L)$
$\delta(q_2, c') = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_2, d') = (q_{reject}, \_, R)$ * This should not happen

q_3 -- Mark the found a (a vs b) (R)

$\delta(q_3, a) = (q_1, a', R)$
$\delta(q_3, b) = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_3, c) = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_3, d) = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_3, \_) = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_3, a') = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_3, b') = (q_b, b', R)$ -- No more a's, check if b's are greater or equal
$\delta(q_3, c') = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_3, d') = (q_{reject}, \_, R)$ * This should not happen

q_4 -- Find next a (a vs c) (L)

$\delta(q_4, a) = (q_4, a, L)$
$\delta(q_4, b) = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_4, c) = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_4, d) = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_4, \_) = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_4, a') = (q_5, a', R)$
$\delta(q_4, b') = (q_4, b', L)$

$\delta(q_4, c') = (q_4, c', L)$
$\delta(q_4, d') = (q_{reject}, \_, R)$ * This should not happen

q_5 -- Mark the found a (a vs c) (R)

$\delta(q_5, a) = (q_6, a', R)$
$\delta(q_5, b) = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_5, c) = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_5, d) = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_5, \_) = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_5, a') = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_5, b') = (q_c, b', R)$ -- No more a's, check no more c's
$\delta(q_5, c') = (q_c, c', R)$ -- No more a's, check no more c's
$\delta(q_5, d') = (q_{reject}, \_, R)$ * This should not happen

q_6 -- Match c's to a's (R)

$\delta(q_6, a) = (q_6, a, R)$
$\delta(q_6, b) = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_6, c) = (q_4, c', L)$
$\delta(q_6, d) = (q_{reject}, \_, R)$
$\delta(q_6, \_) = (q_{reject}, \_, R)$ -- More a's than c's
$\delta(q_6, a') = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_6, b') = (q_6, b', R)$
$\delta(q_6, c') = (q_6, c', R)$
$\delta(q_6, d') = (q_{reject}, \_, R)$ * This should not happen

q_7 -- Match d's to b's (b vs d) (R)

$\delta(q_7, a) = (q_{reject}, \_, R)$
$\delta(q_7, b) = (q_7, b', R)$
$\delta(q_7, c) = (q_{reject}, \_, R)$
$\delta(q_7, d) = (q_8, d', L)$
$\delta(q_7, \_) = (q_{reject}, \_, R)$ -- More b's than d's
$\delta(q_7, a') = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_7, b') = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_7, c') = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_7, d') = (q_7, d', R)$

q_8 -- Find next b (b vs d) (L)

$\delta(q_8, a) = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_8, b) = (q_8, b, L)$
$\delta(q_8, c) = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_8, d) = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_8, \_) = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_8, a') = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_8, b') = (q_9, b', R)$
$\delta(q_8, c') = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_8, d') = (q_8, d', L)$

q_9 -- Mark the found b (b vs d) (R)

$\delta(q_9, a) = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_9, b) = (q_7, b', R)$
$\delta(q_9, c) = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_9, d) = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_9, \_) = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_9, a') = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_9, b') = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_9, c') = (q_{reject}, \_, R)$ * This should not happen
$\delta(q_9, d') = (q_c, d', R)$ -- No more b's, check no more d's

q_b -- More b's than a's, find next b (OR A==B so check no c's or d's)

$\delta(q_b, a) = (q_{reject}, \_, R)$
$\delta(q_b, b) = (q_7, b'R)$
$\delta(q_b, c) = (q_{reject}, \_, R)$
$\delta(q_b, d) = (q_{reject}, \_, R)$
$\delta(q_b, \_) = (q_{accept}, \_, R)$
$\delta(q_b, a') = (q_{reject}, \_, R)$* This should not happen
$\delta(q_b, b') = (q_b, b', R)$
$\delta(q_b, c') = (q_{reject}, \_, R)$* This should not happen
$\delta(q_b, d') = (q_{reject}, \_, R)$* This should not happen

q_c -- Check if we have went through the whole string (R)

$\delta(q_c, a) = (q_{reject}, \_, R)$
$\delta(q_c, b) = (q_{reject}, \_, R)$
$\delta(q_c, c) = (q_{reject}, \_, R)$
$\delta(q_c, d) = (q_{reject}, \_, R)$
$\delta(q_c, \_) = (q_{accept}, \_, R)$
$\delta(q_c, a') = (q_c, a', R)$
$\delta(q_c, b') = (q_c, b', R)$

$\delta(q_c, c') = (q_c, c', R)$
$\delta(q_c, d') = (q_c, d', R)$