Xiangyu Ren

CMPSC464

Collaborator: None

1. Book 3.2(b)

input: "1#1"

Sequence of configuration:

$q_1 1\#1$

$x q_3 \#1$

$x\# q_5 1$

$x\# q_6 x$

$x q_6 \# x$

$q_7 x\# x$

$x q_1 \# x$

$x\# q_8 x$

$x\# x q_8 \sqcup$

$x\# x \sqcup q_{accept}$

2. Give an implementation-level description of a TM that decides language

we decribe a Turing Machine M_1 that decides $A = \{w \mid w$ contains twice as many 0s as 1s$\}$

M_1 = "On input string s   ex. 010, 100010, ε

(a) repeat the following from left to right until there are no more 1s in tape

(i) Scan right until we find a 1

(ii) Mark the leftmost 1 with X

(iii) Return the head to the left hand end of the tape

(iv) Scan right until we find a 0, if no 0 found, Reject

(v) Mark the leftmost 0 with X and jump to step (iv)

(vi) Return the head to the left hand end of the tape and go to step (i)

(b) Scan the tape to check

if no 0 or 1 in the tape, accept

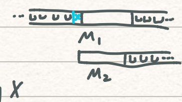
else, reject

3. Show that Turing Machine with doubly infinite tape recognize the class of Turing-recognizable languages.

idea: show equivalence between TM with doubly infinite tape (M_1) and ordinary TM (M_2)

first, simulate ordinary TM with doubly infinite tape TM

as we can see in the graph,



we can simply simulate M_2 with M_1 by

mark the rightmost cell on the left of input in M_1 by X

thus we can get the location of head just like the left hand end in M_2

Second, simulate doubly finite tape TM with ordinary TM

since every multitape TM has an equiv single tape TM

let 2-tape TM (M_3) be equivalent to M_2

To M_3, let tape ① contains input string and blanks to the right

let tape ② stay in blank

thus, if we concatenate reverse of tape ② and tape ①

we can get the same tape like M_1

Thus, we can simulate M_2 by M_1 and simulate M_1 by M_3, M_2 is equiv to M_3

$\Rightarrow$ L that can be recognized by M_1 can be recognized by M_2

L' that can be recognized by M_2 can be recognized by M_1

Therefore, we can say Turing machine with doubly infinite tape recognizes the class of Turing-recognizable language.

4.

a) concatenation

let L_1 and L_2 be the Turing decidable languages and M_1 and M_2 be their respective TMs.

to show L_1L_2 is decidable,

construct a TM M that decides L_1L_2

idea: we try to simulate u and v where $u \in L_1$, $v \in L_2$, and partition the input w into u and v

then construct M based on M_1 and M_2. input u into M_1, input v into M_2, if both M_1 and M_2 accept, M accept

else, M rejects.

M = On input w

(i) Simulate L_1 on input w, if doesn't exist, reject

(ii) Simulate L_2 on input w, if doesn't exist, reject

(iii) accept

Therefore, since M can halt. and M decides L_1L_2, we say the class is closed under concatenation

b) Intersection

let L_1 and L_2 be the Turing decidable languages and M_1 and M_2 be their respective TMs.

to show $L_1 \cap L_2$ is decidable,

construct a TM M that decides $L_1 \cap L_2$

idea: since it's intersection so both M_1 and M_2 should decides $L_1 \cap L_2$.

M = On input w

(i) if M_1 reject w, reject

(ii) if M_2 reject w, reject

(iii) accept

Therefore since M decides $L_1 \cap L_2$ can it can halt. the class is closed under intersection

5.

a) Star

Let M_1 be the TM of Turing recognizable language L

to show L^* is recognizable

construct a TM M that recognize L^*

idea: if $w = \varepsilon$, accept.

loop in input w, try to split w into $u_1 u_2 u_3 \ldots$, if $\forall i, u_i \in L$ and M_1 recognize u_i, accept. else, reject.

M = On input w

(i) Scan right and simulate L from input w. if no more simulation found and no other symbol, goto step (iv), else, reject

(ii) Run M_1 on the simulation, if M_1 reject, M reject

(iii) Go to step (i)

(iv) accept.

Therefore, since M recognize L^*, we say the class is closed under star.

b) Concatenation

Let M_1 M_2 be the TM of Turing recognizable language L_1 and L_2 respectively

to show L_1L_2 is recognizable

construct a TM M that recognize L_1L_2

idea: partition w into uv where $u \in L_1$, $v \in L_2$. if both M_1 M_2 accept, M accept. else, reject

M = On input w

(i) Simulate L_1 on input w, if doesn't exist, reject

(ii) Simulate L_2 on input w, if doesn't exist, reject

(iii) accept

Therefore, since M recognize L_1L_2, we say the class is closed under concatenation.