COURSE: THEORY OF AUTOMATA COMPUTATION

TOPICS TO BE COVERED

- Closure Properties of the Class of Recursively Enumerable Languages

CLOSURE PROPERTY UNDER UNION

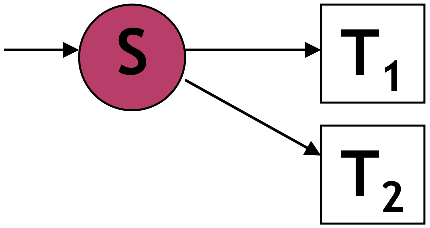
Theorem: Let L_1 and L_2 be recursively enumerable languages over Σ. Then, $L_1 \cup L_2$ is also recursively enumerable.

Proof:

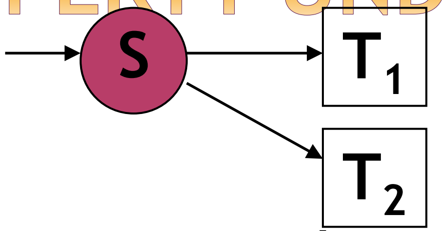
Let L_1 and L_2 be recursively enumerable languages over Σ.

Then, there exist TM's T_1 and T_2 accepting L_1 and L_2, respectively.

Construct an NTM M as follows.



CLOSURE PROPERTY UNDER UNION



If w is in L_1, but not in L_2, then T_1 in M runs and halts.

If w is in not L_1, but in L_2, then T_2 in M runs and halts.

If w is in both L_1 and L_2, then either T_1 or T_2 runs and halts.

For these 3 cases, M halts.

If w is neither in L_1 nor in L_2, then either T_1 or T_2 runs but both never halt. Then, M does not halt.

Thus, M accepts $L_1 \cup L_2$. That is, $L_1 \cup L_2$ is recursively enumerable.

Closure Property Under Intersection

Theorem: Let L_1 and L_2 be recursively enumerable languages over Σ. Then, $L_1 \cap L_2$ is also recursively enumerable.

Proof:

Let L_1 and L_2 be recursively enumerable languages over Σ.

Then, there exist TM's T_1 and T_2 accepting L_1 and L_2, respectively.

Construct an NTM M as follows.

$\rightarrow T_{copyTape1ToTape2} \rightarrow T_1 \rightarrow T_{moveRight} \xrightarrow{1} T_{copyTape2ToTape1} \rightarrow T_2$

CLOSURE PROPERTY UNDER INTERSECTION

$\rightarrow T_{copyTape1ToTape2} \rightarrow T_1 \rightarrow T_{moveRight} \xrightarrow{1} T_{copyTape2ToTape1} \rightarrow T_2$

If w is in not L_1, then T_1 in M does not halt. Then, M does not halt.

If w is in L_1, but not in L_2, then T_1 in M halts and T_2 can finally start, but does not halt. Then, M does not halt.

If w is in both L_1 and L_2, then T_1 in M halts and T_2 can finally start, and finally halt. Then, M halts.

Thus, M accepts $L_1 \cap L_2$. That is, $L_1 \cap L_2$ is recursively enumerable.

Closure Property Under Union (II)

Theorem: Let L_1 and L_2 be recursively enumerable languages over Σ. Then, $L_1 \cup L_2$ is also recursively enumerable.

Proof:

Let L_1 and L_2 be recursively enumerable languages over Σ.

Then, there exist DTM's $T_1 = (Q_1, \Sigma, \Gamma, \delta_1, s_1)$ and $T_2 = (Q_2, \Sigma, \Gamma, \delta_2, s_2)$ accepting L_1 and L_2, respectively.

Construct a 2-tape TM M which simulates T_1 and T_2 simultaneously. Tape 1 represents T_1's tape and Tape 2 represents T_2's tape.

CLOSURE PROPERTY UNDER UNION (II)

Let $M = (Q_1 \times Q_2, \Sigma, \Gamma, \delta, (s_1, s_2))$ where

- $\delta((p_1,p_2),a_1,a_2) = ((q_1,q_2),b_1,b_2,d_1,d_2)$ for $\delta_1(p_1,a_1)=(q_1,b_1,d_1)$ and $\delta_2(p_2,a_2)=(q_2,b_2,d_2)$
- $\delta((p_1,p_2),a_1,a_2) = (h,b_1,b_2,d_1,d_2)$ for $\delta_1(p_1,a_1)=(h,b_1,d_1)$ or $\delta_2(p_2,a_2)=(h,b_2,d_2)$

If either T_1 or T_2 halt, M finally gets to the state h.

If neither T_1 nor T_2 halt, M never gets to the state h.

Closure Property under Intersection (II)

Theorem: Let L_1 and L_2 be recursively enumerable languages over Σ. Then, $L_1 \cap L_2$ is also recursively enumerable.

Proof:

Let L_1 and L_2 be recursively enumerable languages over Σ.

Then, there exist DTM's $T_1 = (Q_1, \Sigma, \Gamma, \delta_1, s_1)$ and $T_2 = (Q_2, \Sigma, \Gamma, \delta_2, s_2)$ accepting L_1 and L_2, respectively.

Construct a 2-tape TM M which simulates T_1 and T_2 simultaneously. Tape 1 represents T_1's tape and Tape 2 represents T_2's tape.

CLOSURE PROPERTY UNDER INTERSECTION (II)

Let $M = ((Q_1 \cup \{h\}) \times (Q_2 \cup \{h\}), \Sigma, \Gamma, \delta, (s_1,s_2))$ where

- $\delta((p_1,p_2),a_1,a_2) = ((q_1,q_2),b_1,b_2,d_1,d_2)$ for $\delta_1(p_1,a_1)=(q_1,b_1,d_1)$ and $\delta_2(p_2,a_2)=(q_2,b_2,d_2)$
- $\delta((h,p_2),a_1,a_2) = ((h,q_2),a_1,b_2,S,d_2)$ for all p_2,a_1,a_2 and $\delta_2(p_2,a_2)=(q_2,b_2,d_2)$
- $\delta((p_1,h),a_1,a_2) = ((q_1,h),b_1,a_2,d_1,S)$ for all p_1,a_1,a_2 and $\delta_1(p_1,a_1)=(q_1,b_1,d_1)$
- $\delta((h,h),a_1,a_2) = (h,a_1,a_2,S,S)$ for all a_1,a_2

If neither T1 nor T2 halt, M never gets to the state h.

If T_1 halts and T_2 does not halt, M gets to the state (h,p).

If T_2 halts and T_1 does not halt, M gets to the state (p,h).

If both T_1 and T_2 halt, M finally gets to the state h.