# Theory of Computation Notes: Turing Machine Closure Properties

Arjun Chandrasekhar

## 1 Closure Properties of Decidable Languages

**Proposition 1.1.** Decidable languages are closed under union.

*Proof.* To prove this, we must prove that if $A$ and $B$ are decidable, then $A \cup B$ is decidable. We know that there exist machines $M_A$ and $M_B$ that decide $A, B$ respectively. We design a machine $M$ to decide $A \cup B$. On input $w$, $M$ does the following:

1. Run $M_A$ on $w$
2. Run $M_B$ on $w$
3. If either machine accepts, $M$ accepts. Otherwise, $M$ rejects.

Because $M_A$ and $M_B$ are deciders, both will halt, so $M$ will also halt. $M$ halts and accepts $w$ if and only if $w \in A$ or $w \in B$, and it rejects otherwise. $\square$

**Definition 1.1.** Let $L$ be a formal language. Let $\#(L) = \{w = w_1\#w_2\#\ldots\#w_n | w_1, w_2, \ldots w_n \in L\}$, i.e. a set of strings that are all in $L$, with # signs separating all the strings.

**Proposition 1.2.** Decidable languages are closed under #

*Proof.* We will show that if $L$ is decidable then $\#(L)$ is decidable. We know there is a machine $M$ to decide $L$. We will design a machine $\#M$ to decide $\#(L)$. On input $w$, $\#M$ does the following:

1. Check that $w$ is in the format $w = w_1\#w_2\ldots w_n$. If not, reject
2. For each $w_i$, run $M$ on $w_i$
3. If $M$ accepts each $w_i$, accept. Otherwise reject

Because $M$ decides $L$, $M$ will always halt so $\#M$ will halt. $\#M$ will halt and accept $w$ if and only if $w \in \#(L)$. $\square$

**Proposition 1.3.** Decidable languages are closed under Kleene star.

*Proof.* We will show that if $L$ is regular then $L^*$ is regular. We know there is a machine $M$ that decides $L$. We will construct a machine $M^*$ that recognizes $L^*$. On input $w$, $M$ does the following:

1. Try all possible ways of splitting $w$ into $w = w_1\#w_2\#\ldots\#w_n$, i.e. try all possible ways of inserting several # signs.
   (a) For each way of splitting it up, run $M$ on $w_1, w_2, \ldots, w_n$. If $M$ accepts on all of them, accept.
2. If any way of splitting up $w$ worked, then accept; otherwise reject.

Because $M$ is a decider for $L$, it will always halt. There are only a finite number of ways to split up $w$ into substrings, so $M^*$ will halt. If $w \in L^*$ there is some way to split up $w$ into substrings such that $M$ accepts every substring; otherwise, every method of splitting it up will fail. Thus, $M^*$ decides $L^*$ $\square$

## 2 Closure Properties of RE Languages

**Proposition 2.1.** RE languages are closed under union.

*Proof.* We must show that if $A$ and $B$ are RE, then $A \cup B$ is RE. Let $M_A, M_B$ be the machines recognize $A$ and $B$, respectively. We will design a machine $M$ that recognizes $A \cup B$.

A naive approach would be to have $M$ do the following on input $w$:

1. Run $M_A$ on $w$
2. Run $M_B$ on $w$
3. If either machine accepts, then accept. Otherwise reject.

The problem is that $M_A$ and $M_B$ are *recognizers*, not *deciders*. So if $w \notin A$, then $M_A$ is not guaranteed to halt.

If $w \notin A, w \in B$, then $w \in A \cup B$, so $M$ *should* accept $w$. But it may not. It is possible that $M_A$ loops on $w$ and $M_B$ never gets the chance to accept $w$.

To resolve this, we introduce the technique of **running two machines in parallel**. To run $M_A$ and $M_B$ in parallel, we let each the two machines take turns running one step at a time. Specifically, do the following:

1. Run $M_A$ for one step
2. Run $M_B$ for one step
3. Run $M_A$ for one more step
4. Run $M_B$ for one more step
5. Run $M_A$ for one more step
6. Run $M_B$ for one more step
7. ...

On input $w$, $M$ does the following:

1. Run both $M_A$ and $M_B$ in parallel
2. If either $M_A$ or $M_B$ reaches an accept state, $M$ accepts
3. If neither machine accepts then $M$ will never accept

$M$ accepts $w$ if and only if $M_A$ or $M_B$ accepts $w$, which happens if and only if $w \in A \cup B$ □

**Proposition 2.2.** RE languages are closed under #

*Proof.* We will show that if $L$ is RE then $\#(L)$ is RE. We know there is a machine $M$ to recognize $L$. We will design a machine $\#M$ to recognize $\#L$. On input $w$, $\#M$ does the following:

1. Check that $w$ is in the format $w = w_1\#w_2 \dots w_n$. If not, reject
2. For each $w_i$, run $M$ on $w_i$ in parallel
3. If $M$ accepts each $w_i$, accept.

Because $M$ recognizes $L$, $M$ will always halt and accept if $w_i \in L$. This means $\#M$ will halt and accept if *every* $w_i \in L$. If any $w_i \notin L$ then $M$ may loop, so $\#M$ may halt, but it won't accept (which is good enough). □

**Proposition 2.3.** Decidable languages are closed under Kleene star.

*Proof.* We will show that if $L$ is regular then $L^*$ is regular. We know there is a machine $M$ that decides $L$. We will construct a machine $M^*$ that recognizes $L^*$. On input $w$, $M$ does the following:

1. Try all possible ways of splitting $w$ into $w = w_1 \# w_2 \# \dots \# w_n$, i.e. try all possible ways of inserting several # signs. We do this part in parallel

   (a) For each way of splitting it up, run $M$ on $w_1, w_2, \dots, w_n$. We also do this part in parallel. If $M$ accepts on all of them, accept.

2. If any way of splitting up $w$ worked, then $M^*$ accepts.

Because $M$ is a recognizer for $L$, it will always halt and accept if $w \in L$. There are only a finite number of ways to split up $w$ into substrings, so $M^*$ can try all of these ways in parallel. If we split up $w = w_1 \# w_2 \# \dots \# w_n$ correctly, each $w_i$ will be accepted by $M$, so $M^*$ will accept this way of splitting up $w$. If there is no valid way to split up $w$, then $M^*$ may loop, but that's ok. □

## 2.1 Exercises

**1.** Prove that decidable languages are closed under complement.

**2.** Prove that decidable languages are closed under intersection.

**3.** Prove that RE languages are closed under complement.

**4.** Prove that RE languages are closed under intersection.