1 Closure Properties

1.1 Decidable Languages

Boolean Operators

Proposition 1. *Decidable languages are closed under union, intersection, and complementation.*

Proof. Given TMs M_1, M_2 that decide languages L_1, and L_2

- A TM that decides $L_1 \cup L_2$: on input x, run M_1 and M_2 on x, and accept iff either accepts. (Similarly for intersection.)
- A TM that decides $\overline{L_1}$: On input x, run M_1 on x, and accept if M_1 rejects, and reject if M_1 accepts. □

Regular Operators

Proposition 2. *Decidable languages are closed under concatenation and Kleene Closure.*

Proof. Given TMs M_1 and M_2 that decide languages L_1 and L_2.

- A TM to decide L_1L_2: On input x, for each of the $|x|+1$ ways to divide x as yz: run M_1 on y and M_2 on z, and accept if both accept. Else reject.
- A TM to decide L_1^*: On input x, if $x = \epsilon$ accept. Else, for each of the $2^{|x|-1}$ ways to divide x as $w_1 \dots w_k$ ($w_i \neq \epsilon$): run M_1 on each w_i and accept if M_1 accepts all. Else reject. □

Inverse Homomorphisms

Proposition 3. *Decidable languages are closed under inverse homomorphisms.*

Proof. Given TM M_1 that decides L_1, a TM to decide $h^{-1}(L_1)$ is: On input x, compute $h(x)$ and run M_1 on $h(x)$; accept iff M_1 accepts. □

Homomorphisms

Proposition 4. *Decidable languages are not closed under homomorphism*

Proof. We will show a decidable language L and a homomorphism h such that $h(L)$ is undecidable

- Let $L = \{xy \mid x \in \{0,1\}^*, y \in \{a,b\}^*, x = \langle M, w \rangle$, and y encodes an integer n such that the TM M on input w will halt in n steps $\}$
- L is decidable: can simply simulate M on input w for n steps
- Consider homomorphism h: $h(0) = 0$, $h(1) = 1$, $h(a) = h(b) = \epsilon$.
- $h(L) =$ HALT which is undecidable. □

1.2 Recursively Enumerable Languages

Boolean Operators

Proposition 5. *R.E. languages are closed under union, and intersection.*

Proof. Given TMs M_1, M_2 that recognize languages L_1, L_2

- A TM that recognizes $L_1 \cup L_2$: on input x, run M_1 and M_2 on x *in parallel*, and accept iff either accepts. (Similarly for intersection; but no need for parallel simulation) □

Complementation

Proposition 6. *R.E. languages are not closed under complementation.*

Proof. A_{TM} is r.e. but $\overline{A_{\text{TM}}}$ is not. □

Regular Operations

Proposition 7. *R.E languages are closed under concatenation and Kleene closure.*

Proof. Given TMs M_1 and M_2 recognizing L_1 and L_2

- A TM to recognize L_1L_2: On input x, do *in parallel*, for each of the $|x|+1$ ways to divide x as yz: run M_1 on y and M_2 on z, and accept if both accept. Else reject.
- A TM to recognize L_1^*: On input x, if $x = \epsilon$ accept. Else, do *in parallel*, for each of the $2^{|x|-1}$ ways to divide x as $w_1 \dots w_k$ ($w_i \neq \epsilon$): run M_1 on each w_i and accept if M_1 accepts all. Else reject. □

Homomorphisms

Proposition 8. *R.E. languages are closed under both inverse homomorphisms and homomorphisms.*

Proof. Let TM M_1 recognize L_1.

- A TM to recognize $h^{-1}(L_1)$:On input x, compute $h(x)$ and run M_1 on $h(x)$; accept iff M_1 accepts.
- A TM to recognize $h(L_1)$: On input x, start going through all strings w, and if $h(w) = x$, start executing M_1 on w, using *dovetailing* to interleave with other executions of M_1. Accept if any of the executions accepts. □
