CS340: Theory of Computation — Sem I 2017-18

Lecture Notes 15: Closure Properties of Decidable Languages

Raghunath Tewari — IIT Kanpur

We will study the closure properties of decidable and Turing recognizable languages under some of the standard operations on languages.

1 Closure Properties of Decidable and Turing Recognizable Languages

1. **Union**

   Both decidable and Turing recognizable languages are closed under union.

   - For decidable languages the proof is easy. Suppose L_1 and L_2 are two decidable languages accepted by halting TMs M_1 and M_2 respectively. The machine for $L_1 \cup L_2$ is designed as follows:

     Given an input x, simulate M_1 on x. If M_1 accepts then *accept*, else simulate M_2 on x. If M_2 accepts then *accept* else *reject*.
   - Now suppose L_1 and L_2 are two Turing recognizable languages accepted by TMs M_1 and M_2 respectively. Since L_1 and L_2 are Turing recognizable languages, therefore for strings that do not belong to these languages, the corresponding machines may not even halt. The previous strategy will not work because we can have a scenario where M_2 accepts x but M_1 loops forever.

     Here the trick is to simulate both M_1 and M_2 "simultaneously". In other words, we design a machine that executes one step of M_1, followed by one step of M_2, then again one step of M_1 and so on.

2. **Concatenation**

   Both decidable and Turing recognizable languages are closed under concatenation.

   I will give the proof for Turing recognizable languages. The proof for decidable languages is similar. Let L_1 and L_2 be two Turing recognizable languages. Given an input w, use nondeterminism and guess a partition w (say $w = xy$). Now run the respective Turing machines of L_1 and L_2 on x and y respectively. If both accepts then *accept* else *reject*.

3. **Star**

   Both decidable and Turing recognizable languages are closed under star operation.

   This is also similar to concatenation. Nondeterministically first guess a number k, and then guess a k partition of the given input. Now for each string in the partition, check whether it belongs to the original language.

4. **Intersection**

   Both decidable and Turing recognizable languages are closed under intersection.

   Run the TMs of both the languages on the given input. *accept* if and only if both the machines accept. In the case of intersection we can run the TMs of L_1 and L_2 one after the other (as opposed to union).

5. **Complementation**

   - Decidable languages are closed under complementation. To design a machine for the complement of a language L, we can simulate the machine for L on an input. If it accepts then *accept* and vice versa.
   - Turing recognizable languages are not closed under complement. In fact, Theorem 1 better explains the situation.

Theorem 1. *A language L is decidable if and only if both L and $\overline{L}$ are Turing recognizable.*

Proof. If L is decidable then it is Turing recognizable. Moreover since decidable languages are closed under complement, $\overline{L}$ is also Turing recognizable.
Suppose L is Turing recognizable via a TM M and $\overline{L}$ is Turing recognizable via a TM M'. Given an input x, simulate x on both the machines M and M' simultaneously (similar to union). If M accepts then *accept* and if M' accepts then *reject*. Observe that since the string x either belongs to L or $\overline{L}$ therefore one of the two machines must accept x.

□