B501, Fall 2023

Assignment 8: Decidability and reductions

Solutions

This assignment contains solved practice problems, numbered in red.
The assigned problems and sub-problems are numbered in green.

1. (5%)

i. Prove that the concatenation of decidable languages is decidable.

Solution. If L and K have decision algorithms, then decide whether $w \in L \cdot K$ by cycling through all partitions $w = u \cdot v$, and checking, using the given decision algorithms for L and K, whether $u \in L$ and $v \in K$. If both are true for some u, v then stop and accept. If not, reject.

(a) Prove that the star of a decidable language is decidable.

Solution. Given a decision algorithm for a language $L \subseteq \Sigma^*$, here is a decision algorithm for L^*. For input $w \in \Sigma^*$ consider successively all partitions $u_1 \cdot \cdots \cdot u_k$ of w into non-empty strings, and accept w if for some such partition we have $u_i \in L$ for all i, which can be decided using the given decision algorithm for L.

Prove that every regular language is decidable. [Hint: Use the closure of the collection of decidable languages under set operations. Alternatively, explain how a DFA can be construed as an algorithm.]

Solution. Approach 1: If M is a DFA recognizing L then it can adjusted to become Turing decider, with $+$ as the only action. A new initial state would recognize the gate and move to the initial input symbol. By virtue of not having any transition for the blank symbol, the machine would stop as the blank is encountered.

Approach 2: Having proved that all finite languages are decidable, and that the collection of decidable languages is closed under union, concatenation and star, it follows by induction on regular expressions that all regular languages are decidable.

2. (5+10+10%)

(a) Exhibit two disjoint undecidable languages whose union is decidable.

Solution. Let $L \subseteq \Sigma^*$ be an undecidable language. Then $\bar{L}$ is also undecidable, or else L would decidable. But the union $L \cup \bar{L} = \Sigma^*$ is decidable.

(b) Show that if $D, N \subseteq \Sigma^*$ are disjoint, where D is decidable and N undecidable, then the union $D \cup N$ is undecidable.

Solution. D and N are disjoint, so $N = (D \cup N) \cap \bar{D}$. Since D is decidable, so is $\bar{D}$, and if $D \cup N$ were also decidable then N would be decidable, as the intersection of decidable languages.

(c) Exhibit two undecidable languages who intersection is infinite but decidable. [Hint: Consider $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, undecidable Σ-languages L, K and decidable Σ-language D. What about $\mathtt{a} \cdot D \cup \mathtt{b} \cdot L$ and $\mathtt{a} \cdot D \cup \mathtt{c} \cdot K$?]

3. (10+5+5%) Let $L \subseteq \{\mathtt{a}, \mathtt{b}\}^*$, and define $X = \mathtt{a} \cdot L \cup \mathtt{b} \cdot \bar{L}$

(a) Prove that if L is decidable then so is X.

(b) Prove that $\bar{L} \leqslant_c X$.

Solution. Define $\rho(w) = \mathtt{b} \cdot w$. ρ is trivially computable, and it is a reduction: $w \in L$ iff $\rho(w) = \mathtt{b} \cdot w \in X$.

(c) Conclude that if X is decidable then so is $\bar{L}$.

4. (10%) We showed that the problem ACCEPTANCE is SD but not decidable. Show that its complement is not even SD.

Solution. If the complement were SD, then ACCEPTANCe would be both SD and co-SD, and therefore decidable, which it is not.

5. (15%) Show that every infinite SD language has an infinite decidable sub-language. [Hint: An infinite SD language is computably enumerated, and from its computable enumeration we can extract an orderly enumeration of a sub-language.]

Solution. Let $L \subseteq \Sigma^*$ be an infinite SD language. We thus have L as the image of a computable function $f: \mathbb{N} \to \Sigma^*$. Let $g: \mathbb{N} \to \Sigma^*$ be defined by: $g(n) =$ the first string $f(k)$ longer than $g(i)$ for all $i < n$. Then g is computable, is an injection by its definition, and is total since L is infinite. So the image of g is a decidable sub-language of L.

i. Show that there is an infinite language L without any infinite decidable sub-language. [Hint: Explain why there is a listing $L_1, L_2, \ldots$ of all infinite decidable languages $L \subseteq \{0,1\}^*$. Now define $L = \{w_0, w_1, w_2 \ldots\}$ as follows. Let $w_0 = \varepsilon$; and given w_i, let u be the first string in L_i longer than w_i, and take w_{i+1} to be a string longer than u. Why is L infinite? Why can't we have $L_i \subseteq L$ for any i?]

Solution. The collection of Turing deciders is countable (when no computability condition is required). So, by elementary Set Theory, there is a listing $L_1, L_2, \ldots$ of all infinite decidable languages $L \subseteq \{0,1\}^*$.

Define $L = \{w_0, w_1, w_2 \ldots\}$ as follows. Let $w_0 = \varepsilon$; and given w_i let u_i be the first string in L_i longer than w_i. Such a u_i must exist, since L_i is infinite. Take w_{i+1} to be any string longer than u_i. So $|w_{i+1}| > |u_i| > |w_i|$. By definition, $u_i \in L_i$. But $u_i \notin L$, because the longest string in L of length $\leqslant |u_i|$ is w_i, which is shorter than u_i. Since $u_i \in L_i - L$ it follows that $L_i \nsubseteq L$ for all $i \geqslant 1$.

6. (15%) Let $L \subseteq \Sigma^*$ and define $D =_{\text{df}} \{ww \mid w \in L\}$.

(a) Show that $L \leqslant_c D$.

Solution. Define $\rho : \Sigma^* \to \Sigma^*$ by $\rho(w) = ww$.
ρ is a reduction of L to D, since we have $w \in L$ iff $\rho(w) = ww \in D$ from the definition of ρ.
ρ is clearly computable.

i. Assume that $\varepsilon \notin L$. Show that $D \leqslant_c L$.
Solution. Define

$$\rho(x) = \textbf{if } x \text{ is of the form } ww \textbf{ then } w \textbf{ else } \varepsilon.$$

Clearly, ρ is computable.
If $x \in D$ then, by the definition of D, $x = ww$ for some $w \in L$. By the definition of ρ this implies that $\rho(x) = w$, so $\rho(x) \in L$.
Conversely, if $\rho(x) \in L$ then, since $\varepsilon \notin L$, x is the form $w\,w$ and $\rho(x) = w$. But we have $w\,w \in D$ only if $w \in L$, so $\rho(x) = w \in L$.

7. (15%) Let E be the set of acceptors that accept every even-length string, and D the set of accepters over $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ that accept every odd-length string. Construct a computable reduction $\rho : D \leqslant_c E$.

Solution. Define ρ to map an acceptor M to the acceptor M' that appends $\mathtt{a}$ to the front of its input and then invokes M. That is $\mathcal{L}(M') = \{\mathtt{a}\} \cdot \mathcal{L}(M)$.
Then M accepts all even-length strings iff M' accepts all odd-length strings.
ρ is trivially computable.