Final #1

Mark all correct answers for each of the following questions.

Σ denotes an arbitrary alphabet and L an arbitrary language over Σ, unless otherwise specified.

1. Given a language L over Σ, denote:

$$L^{\leftarrow} = \{w_2 w_1 : w_1, w_2 \in \Sigma^*, w_1 w_2 \in L\}.$$

(a) If $L \subseteq \Sigma^*$, then $(L^{\leftarrow})^{\leftarrow} = L$.

(b) If $L_1, L_2 \subseteq \Sigma^*$, then $(L_1 L_2)^{\leftarrow} = L_2^{\leftarrow} L_1^{\leftarrow}$.

(c) If $L^{\leftarrow}$ is regular, then so is L.

(d) There exists a language $L \subseteq \Sigma^*$ such that, defining the sequence of languages $(L_n)_{n=0}^{\infty}$ inductively by

$$\begin{aligned} L_0 &= L, \\ L_n &= L_{n-1}^{\leftarrow}, \qquad n \geq 1, \end{aligned}$$

we have

$$L_0 \subsetneqq L_1 \subsetneqq L_2 \subsetneqq \ldots .$$

(e) If $L = \{a^n b^n : n \geq 0\}$, then $L^{\leftarrow}$ is context-free.

2. (a) If $M = (Q, \Sigma, \Gamma, \Delta, s, A)$ is a pushdown automaton and $\alpha_0 \in \Gamma^*$, then the language

$$L = \{w \in \Sigma^* : (s, w, \varepsilon) \vdash_M^* (q, \varepsilon, \alpha_0) \text{ for some } q \in A\}$$

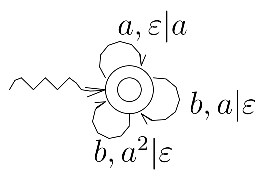
is context-free.

(b) If M and α_0 are as in the preceding part, then the language

$$L = \{w \in \Sigma^* : (s, w, \alpha_0) \underset{M}{\overset{*}{\vdash}} (q, \varepsilon, \varepsilon) \text{ for some } q \in A\}$$

is context-free.

(c) Let $M = (Q, \{a, b\}, \{a\}, \Delta, s, \{s\})$ be the pushdown automaton depicted in the following diagram:



Then $L(M)$ is the collection of all words w over the alphabet $\{a, b\}$, satisfying:

i. $|w|_a \leq 2|w|_b$.

ii. For every prefix u of w we have $|u|_b \leq |u|_a$.

(d) Let $G = (N, \Sigma, R, S)$ be the context-free grammar satisfying $L(G) = L(M)$ for the pushdown automaton M from the preceding part, constructed according to the algorithm presented in class. Then $|N| = 9$.

(e) Let G be the grammar from the preceding part. Then $|R| = 8$.

3. Let $G = (N, \Sigma, R, S)$ be a context-free grammar.

(a) If the rules in R are

- $S \to SSS \mid w_1 \mid w_2 \mid \ldots \mid w_k$,

where $w_1, w_2, \ldots, w_k \in \Sigma^*$, then there exists a regular grammar equivalent to G.

(b) Let $G' = (N, \Sigma, R', S)$ be another context-free grammar, such that $R' \supsetneq R$. If G, G' are arbitrary grammars, then it may be the case that $L(G') = L(G)$. However, if both of them are regular, then $L(G') \supsetneq L(G)$.

(c) Let $G' = (N, \Sigma, R', S)$ be a context-free grammar with $R \cap R' = \emptyset$. Then $L(G) \cap L(G') = \emptyset$.

(d) Let $G' = (N, \Sigma, R', S)$ be a context-free grammar with $R \cap R' \neq \emptyset$. Then $L(G) \cap L(G') \neq \emptyset$.

(e) If $(S, SS) \in R$, then $L(G)^+ = L(G)$.

(f) Let $M = (Q, \Sigma, \Gamma, \Delta, s, \{s\})$ be a pushdown automaton. Then $L(M)^+ = L(M)$.

4. (a) We have seen in class that the language K_0 is Turing-accepted but not Turing-decidable. K_0 is not the only language over the alphabet $\{I, c\}$ having these two properties. In fact, there are uncountably many languages over this alphabet with these properties.

(b) Let $\mathcal{L}_{\text{TD}}$ denote the set of all Turing-decidable languages over Σ and let $\mathcal{L}_{\text{TD}}^C$ denote the complementary set of languages. Put:

$$L_1 = \bigcup_{L \in \mathcal{L}_{\text{TD}}} L, \qquad L_2 = \bigcup_{L \in \mathcal{L}_{\text{TD}}^C} L.$$

Then both languages L_1 and L_2 are Turing-decidable.

(c) Let $f : \mathbf{Z}_+ \to \mathbf{Z}_+$ be a function having the property that for each $x \in \mathbf{Z}_+$ we have either $f(x) = x$ or $f(x) = x + 1$. Then there exists a Turing machine computing f.

(d) If M is a Turing machine computing the function $f : \{0,1\}^* \to \{0,1\}^*$ given by

$$f(w) = ww^R w, \qquad w \in \{0,1\}^*,$$

then M computes a function $g_{1,3} : \mathbf{Z}_+ \to \mathbf{Z}_+^3$ which is injective but not surjective.

Solutions

1. If $L = \{ab\}$, then $L^{\leftarrow} = \{ab, ba\}$ and $(L^{\leftarrow})^{\leftarrow} = \{ab, ba\} \neq L$.

   If $L_1 = \{a\}, L_2 = \{b\}$, then $L_1 L_2 = \{ab\}$, and therefore $(L_1 L_2)^{\leftarrow} = \{ab, ba\}$, whereas $L_2^{\leftarrow} L_1^{\leftarrow} = \{ba\}$.

   Let L be a language over $\{a, b\}$, with the property that, for each $n \geq 2$, it contains all words of length n except for exactly one word, which is neither a^n nor b^n. (To be specific, you may suppose the missing word is either ab^{n-1} or $a^{n-1}b$.) Moreover suppose L contains all words of length at most 1. Clearly, L may be selected in uncountably many distinct ways, and in particular can be chosen not to be regular. However, in any case $L^{\leftarrow} = \{a, b\}^*$.

   Obviously, the set of words belonging to $L^{\leftarrow}$ due to any word w in L contains w itself, and hence $L^{\leftarrow} \supseteq L$. However, the words belonging to $(L^{\leftarrow})^{\leftarrow}$ due to each of these words belongs already to $L^{\leftarrow}$. Thus, $(L^{\leftarrow})^{\leftarrow} = L^{\leftarrow}$. It follows that, for the construction in (d), all sets L_n, with the possible exception of L_0, are identical.

   If $L = \{a^n b^n : n \geq 0\}$, then $L^{\leftarrow} = \{a^k b^n a^{n-k} : n \geq k \geq 0\} \cup \{b^k a^n b^{n-k} : n \geq k \geq 0\}$. We may express $L^{\leftarrow}$ alternatively in the form

   $$\begin{aligned} L^{\leftarrow} &= \{a^k b^k b^l a^l : k, l \geq 0\} \cup \{b^k a^k a^l b^l : k, l \geq 0\} \\ &= LL' \cup L'L, \end{aligned}$$

   where: $L' = \{b^n a^n : n \geq 0\}$. Since L, and similarly L', are context-free, so is $L^{\leftarrow}$.

   Thus, only (e) is true.

2. To show that the language L in (a) is context-free, consider the push-down automaton $M' = (Q \cup \{f\}, \Sigma, \Gamma, \Delta', s, \{f\})$, where $f \notin Q$ and

   $$\Delta' = \Delta \cup \{((q, \varepsilon, \alpha_0), (f, \varepsilon)) : q \in A\}.$$

   A word w is accepted by M' if and only if $(s, w, \varepsilon) \vdash_{M'}^{*} (f, \varepsilon, \varepsilon)$, which holds if and only if $(s, w, \varepsilon) \vdash_{M}^{*} (q, \varepsilon, \alpha_0)$ for some $q \in A$. Hence $L = L(M')$.

Similarly, the language L in (b) is the language accepted by the pushdown automaton $M' = (Q \cup \{s'\}, \Sigma, \Gamma, \Delta', s', A)$, where $s' \notin Q$ and $\Delta' = \Delta \cup \{((s', \varepsilon, \varepsilon), (s, \alpha_0))\}$.

Non-empty words accepted by the automaton M in (c) must clearly end with the letter b. In particular, the word $aba \notin L(M)$. However, this word does satisfy both properties (c).i and (c).ii.

To transform M into a simple pushdown automaton, we first have to add to Δ the transition $((s, a, a), (s, a^2))$. Also, we have to remove the transition $((s, b, a^2), (s, \varepsilon))$, and instead add a state p to Q, along with the transitions $((s, \varepsilon, a), (p, \varepsilon, \varepsilon))$ and $((p, b, a), (s, \varepsilon, \varepsilon))$.

Denote by M' the simple pushdown automaton obtained by these changes Since it has two states, and operates with a stack alphabet of size 1, we have $|N| = 1 + 2 \cdot (1 + 1) \cdot 2 = 9$.

Since M' contains a single accepting state, R contains a single rule of type A. Each transition in M', in which a non-empty word of length n is added to the stack, contributes $|Q'|^n$ rules of type B to R (where Q' is the set of states of M'). In our case, we have one such rule, with $n = 1$, and $|Q'| = 2$, so that we have two rules of type B. Each transition in M', in which nothing is added to the stack, contributes $|Q'|$ rules of type C to R. In our case, this provides $3 \cdot 2 = 6$ rules of type C. Finally, each state in A contributes one rule of type D, and hence we have one such rule in our case. Altogether, $|R| = 10$.

Thus, (a), (b) and (d) are true.

3. For the grammar in (a) we have $L(G) = \{w_1, w_2, \ldots, w_k\}^*$. In fact, each word belonging to the language on the right-hand side can clearly be derived from S, so that $L(G) \supseteq \{w_1, w_2, \ldots, w_k\}^*$. On the other hand, by induction on the number of derivations, we easily show that any word in $(N \cup \Sigma)^*$ which can be derived (in any number of steps) from S belongs to $\{S, w_1, w_2, \ldots, w_k\}^*$, and in particular $L(G) \subseteq \{w_1, w_2, \ldots, w_k\}^*$. Since $L(G)$ is regular, there exists a regular grammar equivalent to G.

   If R in (b) consists of the rules

   - $S \to aS \mid \varepsilon$,

and R' consists of

- $S \rightarrow aS \mid a^2S \mid \varepsilon$,

then $R' \supsetneq R$, yet $L(G') = L(G) = \{a\}^*$.

If R in (c) consists of the rules

- $S \rightarrow aS \mid \varepsilon$,

and R' consists of

- $S \rightarrow Sa \mid a$,

then $R \cap R' = \emptyset$, yet $L(G) = \{a\}^*$ and $L(G') = \{a\}^+$, so that $L(G') \cap L(G) = \{a\}^+$.

If R in (d) consists of the rules

- $S \rightarrow SS \mid a$,

and R' consists of

- $S \rightarrow SS \mid b$,

then $R \cap R' \neq \emptyset$, yet clearly $L(G) \cap L(G') = \emptyset$.

Let $(S, SS) \in R$ and $w_1, w_2, \ldots, w_k \in L(G)$. Then

$$S \underset{G}{\Longrightarrow} SS \underset{G}{\Longrightarrow} SSS \underset{G}{\Longrightarrow} \ldots \underset{G}{\Longrightarrow} S^k \overset{*}{\underset{G}{\Longrightarrow}} w_1 w_2 \ldots w_k.$$

Hence $L(G) \supseteq L(G)^+$. The inverse inclusion is trivial, and consequently $L(G)^+ = L(G)$.

Let M be a pushdown automaton as in (f), and let $w_1, w_2, \ldots, w_k \in L(M)$. Then $(s, w_i, \varepsilon) \vdash_M^* (s, \varepsilon, \varepsilon)$ for $1 \leq i \leq k$. Therefore

$$(s, w_1 w_2 \ldots w_k, \varepsilon) \vdash_M^* (s, w_2 \ldots w_k, \varepsilon) \vdash_M^* (s, w_3 \ldots w_k, \varepsilon) \vdash_M^* \ldots \vdash_M^* (s, w_k, \varepsilon) \vdash_M^* (s, \varepsilon, \varepsilon),$$

so that $w_1 w_2 \ldots w_k \in L(M)$. Hence $L(M) \supseteq L(M)^+$, which implies $L(M) = L(M)^+$. Thus, (a), (e) and (f) are true.

4. There exist only countably many describable languages, and hence there exist only countably many Turing-accepted languages (whether Turing-decidable or not).

   Since Σ^* is Turing-decidable, one of the sets in the union defining L_1 in (b) is Σ^*, and therefore $L_1 = \Sigma^*$. On the other hand, let L_0 be any Turing-undecidable language. Then L_0^C is also Turing-undecidable. The union defining L_2 contains therefore both L_0 and L_0^C, so that $L_2 = \Sigma^*$.

   The collection of functions with the property in (c) is clearly uncountable, and hence not every such function has a Turing machine computing it.

   The number $n \in \mathbf{Z}_+$ is represented by the word 1^n0. For the function f in (d) we have

$$f(1^n0) = 1^n0(1^n0)^R1^n0 = 1^n0^21^{2n}0.$$

   The right-hand side represents the point $(n, 0, 2n) \in \mathbf{Z}_+^3$, so that M computes the function given by

$$g_{1,3}(n) = (n, 0, 2n), \qquad n \in \mathbf{Z}_+.$$

   Obviously, $g_{1,3}$ is injective but not surjective.

   Thus, (b) and (d) are true.