♢ **Best before:** November 18 (term test 2).

1. Consider these languages over alphabet $\Sigma = \{0, 1\}$.

   - *Exact.* $L_{1e} = \{x \in \Sigma^* : x \text{ is a palindrome (i.e., } x = x^R)\}$.
   - *Double.* $L_{1d} = \{x : \text{for some } y \in \Sigma^*,\ x = y{\cdot}\text{twice}(y)^R \text{ or } x = \text{twice}(y)^R \cdot y\}$,
     where twice(y) is the string obtained from y by "doubling" each of its symbols.
     I.e., each 0 is doubled to become 00 and each 1 is doubled to become 11.
     E.g., twice(1101) = 11110011. So $110111001111 \in L_{1d}$ and $110011111101 \in L_{1d}$.
   - *Toggle.* $L_{1t} = \{x \in \Sigma^* : \text{for some } y, z \in \Sigma^*,\ x = yz \text{ and } y{\cdot}\text{toggle}(z) \text{ is a palindrome}\}$,
     where toggle(z) is the string obtained from z by toggling each of its symbols.
     E.g., toggle(10100) = 01011, so $110110100 \in L_{1t}$, with $y = 1101$ and $z = 10100$.

   For each of the above languages, do the following.

   (a) Give a CFG that generates it and briefly explain why your CFG is correct.
   Try to "optimize" your CFG by (i) minimizing the number of variables and/or (ii) minimizing the total number of productions.
   Is your CFG ambiguous? If so, then is there an unambiguous CFG that generates the language? Explain your answer.

   (b) Give a PDA that accepts it and briefly explain why your PDA is correct.
   Try to "optimize" your PDA by (i) minimizing the number of states and/or (ii) minimizing the size of the tape alphabet.
   Is your PDA deterministic? If not, then is there a deterministic PDA that accepts the language? Explain your answer.

2. For arbitrary strings x and y, we formally define $\#_y(x)$ to be $|\{(u, v) : u, v \text{ are strings and } x = uyv\}|$.
   Informally, $\#_y(x)$ is the number of places in x where y appears as a substring.
   Here are some examples.

   - $\#_{010}(01011) = \#_{101}(01011) = \#_{011}(01011) = 1$. (010, 101 and 011 each appears once in 01011.)
   - For any string x, $\#_0(x)$ is the number of 0s in x.
   - For any string x, $\#_\epsilon(x) = |x| + 1$. (Recall that $|x|$ is the length of x.)
   - For any $j, k \in \mathbb{N}$, $\#_{1^j}(1^k) = k - j + 1$ if $j \leq k$, and $\#_{1^j}(1^k) = 0$ if $j > k$.
     (A string of j 1s appears in $k - j + 1$ places in a string of k 1s if $j \leq k$.)

   Repeat question 1 with these languages (also over alphabet $\Sigma = \{0, 1\}$).

   - *Exact.* $L_{2e} = \{x \in \Sigma^* : \#_{010}(x) = \#_{011}(x)\}$.
   - *Double.* $L_{2d} = \{x \in \Sigma^* : \#_{010}(x) = 2\,\#_{011}(x) \text{ or } \#_{011}(x) = 2\,\#_{010}(x)\}$.
   - *Almost.* $L_{2a} = \{x \in \Sigma^* : \#_{010}(x) = \#_{011}(x) + 1 \text{ or } \#_{011}(x) = \#_{010}(x) + 1\}$.
   - *Mono.* $L_{2m} = \{x \in \Sigma^* : \text{for some } k \in \mathbb{N} \text{ and } y \in \Sigma^*,$
     $x = 0^k \cdot y,\ \#_1(y) = k \text{ and } y \text{ does not start with } 0\}$.
   - *Stereo.* $L_{2s} = \{x \in \Sigma^* : x \in L_{2m} \text{ or } x^R \in L_{2m}\}$.

   Beware: The CFGs for these languages are not easy to find.

3. Repeat question 1 with each of the following languages.

   (a) $L_{3a} = \{0^p 1^q 0^r 1^s : p, q, r, s \in \mathbb{N} \text{ and } p + q = r + s\}$.

   (b) $L_{3b} = \{0^p 1^q 0^r 1^s : p, q, r, s \in \mathbb{N} \text{ and } p + r = q + s\}$.

   (c) $L_{3c} = \{0^p 1^q 0^r 1^s : p, q, r, s \in \mathbb{N} \text{ and } p + s = q + r\}$.

   (d) Create your own language of the form

   $$L = \{0^p 1^q 0^r 1^s : p, q, r, s \in \mathbb{N} \text{ and } \textit{some condition on } p, q, r, s\},$$

   and try to find a CFG and a PDA for it.
   Beware: Some conditions make this task impossible.

4. Take your favourite regular language and find a right-linear CFG for it.
   Briefly explain why your CFG is correct. Is your CFG also strict right-linear?

5. Do exercise 4 on page 267 of the course notes (about the "shuffle" operation).

6. For each CFG G_i in the list below, determine $\mathcal{L}(G_i)$. Find all pairs that generate the same language.

   G_1: $S \to S0S,\ 0$

   G_2: $S \to SS,\ \epsilon$

   G_3: $S \to SS,\ 0$

   G_4: $S \to SS,\ 00$

   G_5: $S \to S0,\ 0$

   G_6: $S \to 0S0,\ 00,\ 0$

   G_7: $S \to SAS,\ \epsilon$