University of Waterloo
CS 462 — Formal Languages and Parsing
Winter 2018
Problem Set 1

Distributed Monday, January 8 2018.

Due Monday, January 15 2018 by 5 PM. Hand in to LEARN.

All answers should be accompanied by proofs. In all problems the underlying alphabet Σ is assumed to be finite.

1. [10 marks] Recall that x^R is the reversal of the string x.

   Prove that if x is a string and i is an integer ≥ 0, then $(x^i)^R = (x^R)^i$.

   (It may be useful to recall that $(xy)^R = y^R x^R$, a fact that can easily be proved by induction.)

2. [10 marks] Suppose x, y are strings with $x = y^i$ for some $i \geq 1$. Prove that x is a palindrome if and only if y is.

3. [10 marks] Recall that a string x is a subsequence of a string y if we can strike out some symbols of y to get x. Thus `out` is a subsequence of `computer`.

   Call a sequence of strings $x_1, x_2, \ldots$ *good* if it has the property that x_i is never a subsequence of x_j if $i < j$. For example, the sequence of strings

   `abc`, `bc`, `ac`, `a`

   is good. We will see eventually (in Theorem 3.12.1) that every good sequence must be of finite length.

   Call such a sequence *very good* if it is good and satisfies the additional restriction that $|x_i| \leq i$ for all i.

   A natural question is, how long is the longest very good sequence over an alphabet Σ of size k? Although this is easy to answer for small k, it is *much* harder for general k.

   For example, if $k = 1$ and $\Sigma = \{a\}$ then the longest very good sequence is of length 2: $x_1 = a$, $x_2 = \epsilon$.

   (a) How long is the longest very good sequence over an alphabet of size 2?

   (b) How long is the longest very good sequence over an alphabet of size 3? This might be hard. You might have to do some actual computation to answer this one.