LOCALLY TESTABLE LANGUAGES

Definition 1. The set of $k$-factors of a string $w$, $F_k(w)$, is the set of all substrings of $w$ of length $k$. If $|w| < k$, then it is $\{w\}$.

We are most interested in the $k$-factors of strings that have word boundaries attached. So, we would be interested, for example, in the 2-factors of strings like $\langle aabaa \rangle$, namely: $\{\langle a \rangle, \langle aa \rangle, \langle ab \rangle, \langle ba \rangle, \langle a \rangle\}$ (notice that we do not need to say $aa$ twice).

Now, what strings could we build out of these pieces?

Example. The above language is strictly 2-locally testable. The language $aa^*(b+c)$ is strictly 2-local. The language of all strings over $\{a, b\}$ containing no substrings of three $b$'s, and not terminating or beginning with a $b$, is strictly 3-local, but it is not strictly 2-local.

Theorem 3. The strictly $k$-locally testable languages are regular for all $k$.

Proof. You are given a set of factors $\triangleright w, z\triangleleft, wa, \ldots, bz$; now:

1. Add an initial state and a non-accepting state $\triangleright$; add a transition from the initial state to $\triangleright$, upon reading $\triangleright$. Mark the initial state as complete.
2. Add a final state $z\triangleleft$ for each factor $a z \triangleleft$. Mark all these states incomplete.
3. For each state $q = s_1 \ldots s_l$ already added to the automaton which is marked incomplete:
   a. For each symbol $a$ in the alphabet, or $\triangleright$, $\triangleleft$:
      i. If $l < k - 1$:
         A. If there is a factor $qa \ldots$, add a transition from $q$ to $qa$ on reading $a$ (add the state and mark it incomplete if it does not exist).
      ii. If $l = k - 1$:
         A. If there is a factor $qa$, add a transition from $q$ to $s_2 \ldots s_l a$ on reading $a$ (add the state and mark it incomplete if it does not exist). (In the case where $k - 1 = 1$, what we mean by $s_2 \ldots s_l$ is simply $q$.)
   b. Mark the state as complete.
4. Merge the final states.
5. Replace boundaries on transitions with $\varepsilon$.

Theorem 4. The strictly $k$-local languages are strictly $k + 1$-local for all $k$.

Proof. (Construction only.) Suppose we have some set of factors $F$. Construct all factors of length $k + 1$ that can be formed by overlapping the middle $k - 1$ symbols of two factors in $F$. Add in all existing factors of the form $\triangleright \cdots \triangleleft$ (for strings with length less than $k$).

Theorem 5. If $L$ is strictly $k$-local, then there exists some finite set of forbidden factors $G$ such that $L = \bigcap_{f \in G} \{w | f \in F_k(\triangleright w \triangleleft)\}$

Definition 6. A language $L$ is strictly local or locally testable in the strict sense if it is strictly $k$-local for finite $k$.
There are regular languages which are not strictly local for any finite \( k \). For example, the set of all strings over \( \{a,b\} \) containing either \( aa \) or \( bb \) but not both.

**Definition 7.** A language \( L \) is *locally testable* if it is expressible as a boolean combination of strictly local languages.

Again, there are regular languages which are not locally testable: the set of all strings such that exactly one \( c \) precedes exactly one \( b \).