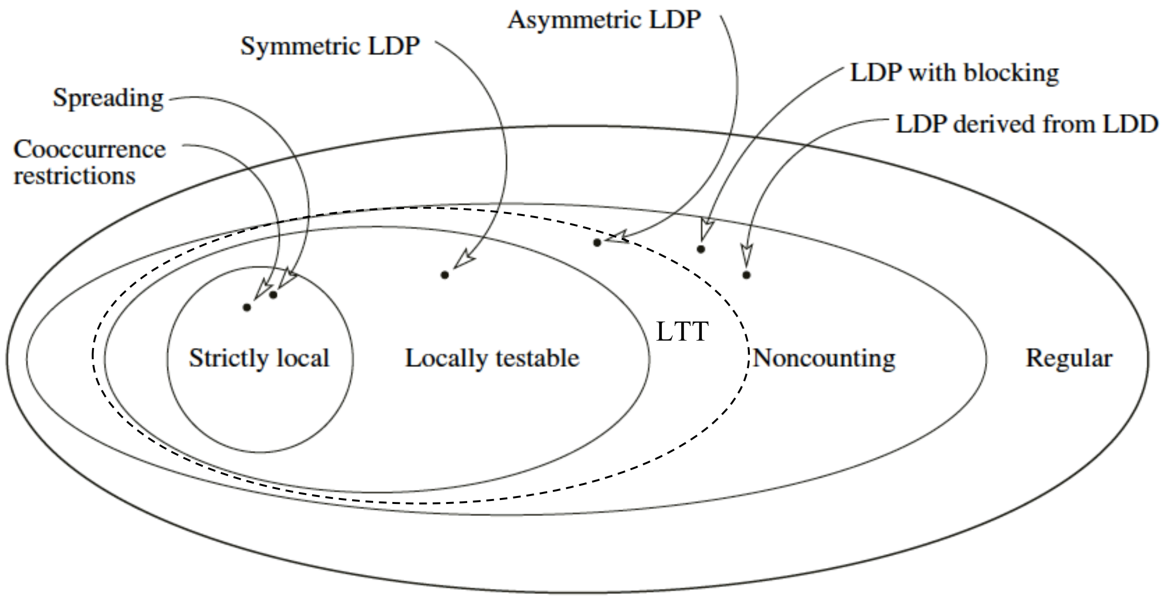
Aural Pattern Recognition Experiments and the Subregular Hierarchy
James Rogers and Geoffrey K. Pullum
Presented by: Nazila Shafiei
December 5th, 2017



The Subregular Hierarchy (Heinz 2010, Figure 2)

The Subregular Hierarchy- Four main classes:

1. Strictly Local Stringsets
2. Locally Testable Stringsets
3. Locally Threshold Testable
4. Star-Free Stringsets

1. Strictly Local Stringsets

No need for repetition, we have seen enough of this ☺

2. Locally (*k*-) Testable Stringsets (LT$_k$)

A stringset L is Locally Testable iff there is some k such that, for all strings x and y: if F_k (⋊.x.⋉) = F_k (⋊.y.⋉) then $x \in L \Leftrightarrow y \in L$ (or $x \notin L \Leftrightarrow y \notin L$).
In plain English: If the set of the *k*-factors of one string equals the set of the *k*-factors of another string, either both strings belong to L, or neither belongs to L.

In other words, a pattern is locally k-testable iff it is possible to decide whether the set of k-factors making up the word is allowable. So, any locally 2-testable pattern either includes both *fifizt* and *fififizt* or excludes both (since they have the same set of 2-factors*: fi, if, iz, zt*) (Heinz 2010).

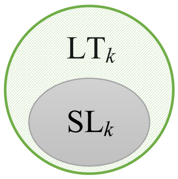
Example 1: Consider the following two stringsets:

Some-B= $\{w \in \{A,B\}^* \mid |w|_B \geq 1\}$ (the set of strings of A's and B's with at least one B)
One-B= $\{w \in \{A,B\}^* \mid |w|_B = 1\}$ (the set of strings of A's and B's with exactly one B)

Some-B is Locally Testable, but One-B is not. The language of Some-B is A^k BA^kB A^k, while the language of One-B is A^kB A^k. These two strings have the same *k*-factors (eg. 1-factors={⋊, A, B,

$\ltimes$}), but Some-B is learnable whereas One-B is not. The reason is because it is not possible to keep track of the number of B's occurring in the string.

Difference between SL_k and LT_k: An LT_k pattern may include a word like *rakt* but exclude a word like *rak* since the two words have different sets of *k*-factors (2-factors={*ra, ak, kt*} versus {*ra, ak*}). On the other hand, a SL_k includes both *rakt* and *rak* because the *k*-factors for the first one is a superset for the *k*-factors of the second one.
For each k, the class SL_k is a proper subset of LT_k.



But, SL_{k+1} is not a subset of LT_k nor is LT_k a subset of SL_{k+1}.
In fact, LT_2 includes stringsets that are not SL for any *k*.

Similarities between SL_k and LT_k: As with SL, the LT_k stringsets are learnable in the limit if k is fixed.

3. Locally Threshold Testable

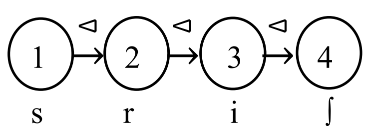
It would be better to keep track of *how many* times a *k*-factor occurs. We can set a threshold for this and still have a finite-state. This way we can recognize any n-B string for any n smaller than the threshold. These are the languages definable in First-Order Logic with the successor relation (but without the order (Place et al. 2014)).

FO(+1): An $\lhd$-model of a string w is a structure

$$\langle \mathcal{D}, \lhd, P_\sigma \rangle \;\; \sigma \in \Sigma$$

where the domain $\mathcal{D} \overset{\text{def}}{=} \{i \in \mathbb{N} \mid 0 \le i < |w|\}$ is the set of positions in w, $\lhd$ is the successor relation on these positions ($x \lhd y \overset{def}{\Longleftrightarrow} y = x+1$) and, for each $\sigma \in \Sigma$, the predicate P_σ picks out the set of positions at which σ occurs in w.

A hypothetical [sriʃ]



For instance, $P_s = 1$, $P_r = 2$, etc.

For instance, the language $a^+b^+a^+b^+$ is locally threshold testable. This is because in a string *abab*, which has an *a* as a prefix, we have *ab* as an infix exactly two times, and *ba* as an infix exactly one time (Bojańczyk 2007).

We show these languages in this format: $LTT_{[k,t]}$, where *k* means *k*-factor and *t* is our threshold.

So, One-B above is $LTT_{[1,2]}$. But, the following stringset is not:

B-before-C $\overset{\text{def}}{=} \{w \in \{A, B, C\}^* \mid$ at least one B precedes any C$\}$

Reason: The set of 1-factor for ABACA is the same for ACABA.

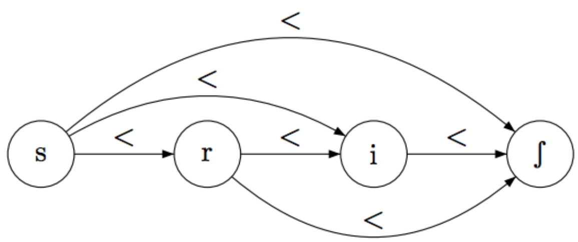
LT_k is a special case of $LTT_{[k,t]}$ when t=1 (Place et al. 2014).

4. Star-Free Stringsets

The next step is to extend the FO signature to include the order ("precedes" or "less-than"). This class is called FO(<), which coincides with the Star-Free sets (SF).

B-before-C, for example, is the set of strings over {A, B, C} which satisfy:

$$(\forall x)[C(x) \rightarrow (\exists y)\ [B(y) \wedge y < x]].$$



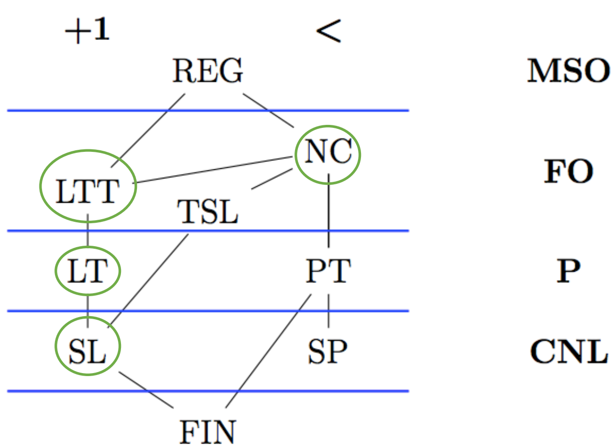
A set of strings is First-Order definable in FO(<), i.e. , relative to the class of finite $\langle \mathcal{D}, \triangleleft, <, P_\sigma \rangle\ \sigma \in \Sigma$ models, iff it is non-counting.

A stringset L is SF iff it is Non-Counting (NC). This means iff there exists some $n > 0$ such that, for all strings u, v, w over Σ, if $uv^n w$ occurs in L, then $uv^{n+1}w$, for all $i \geq 1$, occurs in L as well.

An example of a **<u>not</u>** NC stringset, which requires modular counting is the set of strings of A's and B's in which the number of B's is even:

$$\text{Even-B} \stackrel{\text{def}}{=} \{w \in \{A, B\}^* \mid |w|_B \bmod 2 = 0\}$$ [1]

Using LT strategies cannot recognize this pattern because it cannot distinguish $(A^*BA^*)^{2n}$ from $(A^*BA^*)^{2n+1}$.



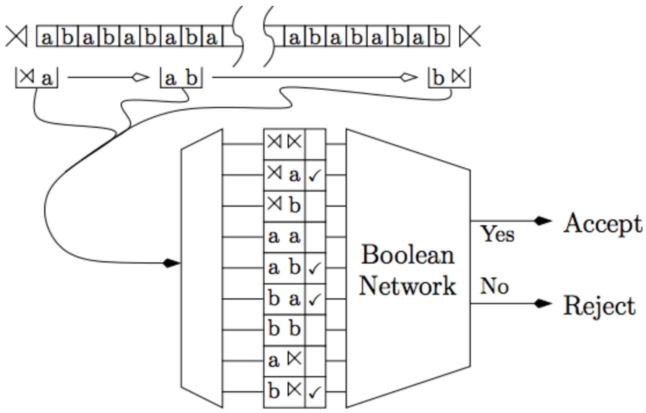
Subregular Hierarchies of Stringsets (from Heinz 2015)

Classes	Learnable	Counts Occurrence	Tracks Precedence	Example
SL_k	if k is fixed	✗	✗	*CC
LT_k	if k is fixed	✗	✗	Some-B
$LTT_{[k,t]}$	if k and t are fixed	✓	✗	One-B
SF	??	✗	✓	B-before-C

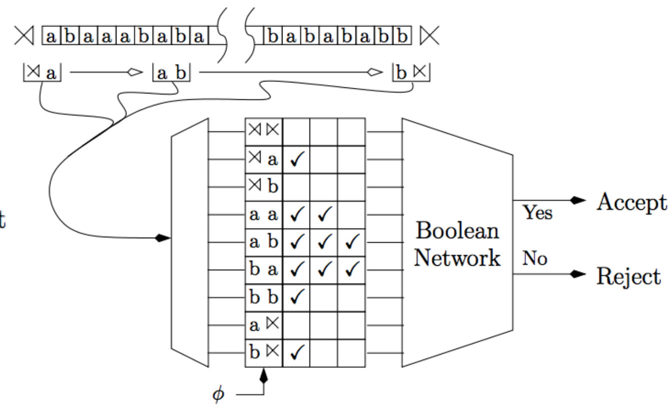
[1] Mod 2= 0 means the number of B's divided by 2 should have the remainder of 0.

LT_k versus LTT_k :

LT Automata:



LTT Automata:



(Rogers and Heinz 2014)

References:

Bojańczyk, Mikołaj. 2007. A new algorithm for testing if a regular language is locally threshold testable. *Information Processing Letters* 104(3): 91-94.

Heinz, Jeffrey. 2010. Learning long-distance phonotactics. *Linguistic Inquiry* 41(4): 623-661.

Heinz, Jeffrey. 2015. The computational nature of phonological generalizations. *Ms., University of Delaware* (2015).

Place, Thomas, Lorijn Van Rooijen, and Marc Zeitoun. 2013. On separation by locally testable and locally threshold testable languages, *Logical Methods in Computer Science* 10 (3:24): 1-28.

Rogers, James, and Geoffrey K. Pullum. 2011. Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language and Information* 20(3): 329-342.

Rogers, James, and Jeffrey Heinz. 2014. Model Theoretic Phonology. Workshop slides in *the 26th European Summer School in Logic, Language and Information*.