A logical characterization of (input) strictly local functions

Jane Chandlee & Steven Lindell
Haverford College, PA (USA)

Factors

Definition: Consider words over a finite alphabet. A *k-factor* of a word *w* is a substring of length *k* (including end markers for prefixes and suffixes)

Example: The 2-factors of #*aba*# are {#*a*, *ab*, *ba*, *a*#}.

Two words are *k*-equivalent if they have the same *k* factors. E.g. $aba \sim_2 ababa \nsim_2 abab$ (b/c of *b*#)

Ordering *k*-equivalence classes by inclusion of their *k*-factors is a finite partial order. E.g. $a \lesssim_2 aba$.

Local languages

Definition: A language *L* is *strictly k-local* if it is closed downward under $\lesssim_k$. I.e. $u \lesssim_k v \in L \Rightarrow u \in L$.

Fact: A strictly local language is determined by the (finite) set of *k*-factors it permits / omits, for some *k*.

Example: $L = a(ba)^*$ is 2-local, allowing {#*a*, *ab*, *ba*, *a*#} and forbidding {#*b*, *aa*, *bb*, *b*#, ##}.

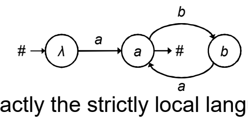
Suffix-prefix substitution closure [Rogers, Pullum '11]: If $v \in \Sigma^{k-1}$ and both $u_1vw_1, u_2vw_2 \in L$, then $u_1vw_2 \in L$.

Local machines

A finite state control can have long term memory, so instead we insist on short term *finite memory* where recollection goes back at most $k - 1$ symbols.

Definition: A *k*-local machine is an incomplete DFA with states labeled $\Sigma^{<k-1}$ (Markovian), one for each allowable $k - 1$ factor (including prefixes).

Example: ($k = 2$)
$L(M) = a(ba)^*$



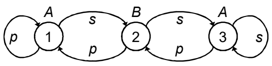
Fact: these accept exactly the strictly local languages

Logical definability

Represent (non-empty) words as structures of the form $\langle D; p, s, A, B\rangle$

Diagram for *aba*:

($D = \{1, 2, 3\}$)



The vocabulary is adjacency with a partition of the domain into letters:

$p(x) = x - 1$ (bottoms out), $s(x) = x + 1$ (tops out), $A \cap B = \emptyset$, $A \cup B = D$.

Formulas in first-order logic are constructed using: →, ∨, ∧, ¬, ∃, ∀

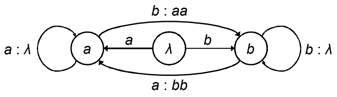
Example: Every local language can be described by a Π_1 sentence. E.g.

$a(ba)^*$: $(\forall x)\ [(p(x) = x \vee s(x) = x) \rightarrow A(x)] \wedge [s(x) \neq x \rightarrow (A(x) \leftrightarrow B(s(x))]$

Local functions (ISL)

Definition: A function is *k*-local if it can be computed by a complete Markovian DFA with outputs on each of its transitions and an output in the last occupied state.

Example: ($k = 2$)



Note: Unlike (sub)-sequential functions, strictly local functions can be computed left-to-right or right-to-left.

Locality preservation

Closure properties for composition: f, g local ⇒ $f \circ g$ is also;
Closed under inverses (reduction) L strictly local ⇒ $f^{-1}(L)$ is.

$|f(x)| \leq m|x|$ (output is bounded in terms of input)
$|x| \leq d|f(x)|$ (input is bounded in terms of output)

Definition: A function f is *non-degenerate* if $|S| = \infty \Rightarrow |f(S)| = \infty$. In other words, f is not infinite-to-one, $|f^{-1}(\{w\})|$ is finite.

Claim: Suppose a function f is computed by minimal finite state machine M. Then f is non-degenerate if and only if M has no null cycles.

Quantifier-free mappings

Use logical formulas to define an output structure in terms of an input structure. E.g. *aba* → *aabbaa*. Allow *copies* (^):

$$\langle \overbrace{\varphi^1(x), \varphi^2(x)}^{D}; \overbrace{\pi^1(x), \pi^2(x)}^{-1}, \overbrace{\sigma^1(x), \sigma^2(x)}^{+1}, \overbrace{\alpha^1(x), \alpha^2(x)}^{A}, \overbrace{\beta^1(x), \beta^2(x)}^{B} \rangle$$

where π and σ are functional definitions by cases. E.g. *doubling*:

σ: $x^1 + 1 = x^2$ and $x^2 + 1 = (x + 1)^1$ unless $x + 1 = x$, $x^2 + 1 = x^2$.
π: $x^2 - 1 = x^1$ and $x^1 - 1 = (x - 1)^2$ unless $x - 1 = x$, $x^1 - 1 = x^1$.

$$1\ 2\ 3 \rightarrow 1^1\ 1^2\ 2^1\ 2^2\ 3^1\ 3^2$$

$\alpha(x^1) \equiv \alpha(x^2) \equiv A(x)$; [$a \rightarrow aa$] $\beta(x^1) \equiv \beta(x^2) \equiv B(x)$; [$b \rightarrow bb$]

Monotonicity

Idea: *output follows input*. I.e. if x precedes y in the input, then x^c cannot succeed $y^{c'}$ in the output.

Example: $1^1\ 1^2\ 2^1\ 2^2\ 3^1\ 3^2$ ...

Counterexample: $1^1\ 2^1\ 3^1\ 1^2\ 2^2\ 3^2$ ...

Equivalent to ±1 monotone: $(x - 1)^{c^-} \leq x^c \leq (x + 1)^{c^+}$

Note: some copies may not exist

Result

Theorem: The non-degenerate input strictly local functions are exactly the monotone quantifier-free interpretations.

Provides an abstract (machine independent) characterization of the input strictly local functions.

Directions:
1. change vocabulary from *adjacency* (± 1) to *precedence* (<)
2. generalize from *strings* to *trees*

Proof ideas

A k-local function can be described by q.f. formulas:
Idea: Inspect the *k*-local neighborhood of each input position using predecessor and successor functions. This tells us which state of the machine we must be in (by locality), and determines the output at that point. Copies are required when multiple symbols come out.

A q.f. interpretation is computable by a local machine:
Idea: Each q.f. formula $\psi(x)$ depends only on a local neighborhood of x, so its behavior can be determined by a k-local machine where k is the maximum nesting depth of the predecessor or successor functions.

References (selected)

1. Rogers, J. and Pullum, G. (2011). Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language and Information*, vol. 20:329-342.

2. Chandlee, J. (2014). Strictly Local Phonological Processes, Ph.D. thesis, University of Delaware.

3. Sakarovitch J. (2009). *Elements of automata theory*, Cambridge University Press.