Constraint-driven analysis of formal languages

Dakotah Lambert

16 Feb 2024

Key points

- Classification
  - What relations are available?
  - How are they used?
- **M** operator :: multiple tiers
- Learning
  - What information is available?
  - How do we extrapolate from it?

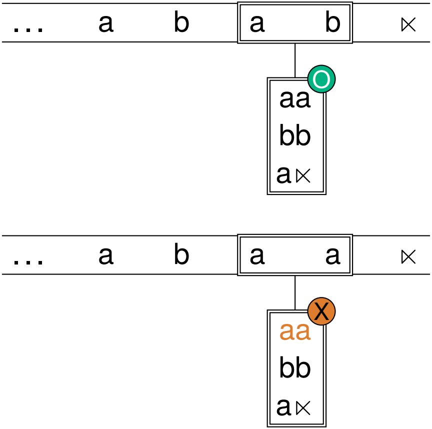
Deriving a class hierarchy
as the linguists would

Alternation

ab
bab
abab
…

(Asmat stress)

Strict locality



Long-distance dependencies: Symmetric harmony

sik’is
ʃitʃedza

Navajo sibilant harmony

Suffix-substitution closure

$aXb \in L$ and $cXd \in L \longrightarrow aXd \in L$
X shared, length at least k

s(it)^{k}is and ʃ(it)^{k}iʃ,
but not s(it)^{k}iʃ

not strictly local

Locally testable

Acceptability based on set S of factors.

Symmetric harmony: $\{\text{s}, \int\} \not\subseteq S$

Long-distance dependencies: Culminativity

Less than two ‘b’:
a, ab, abaa, aaaaab,
but not abab or abba

(basically every stress pattern ever)

Locally threshold testable

$a^k b a^k$ and $a^k b a^k b a^k$ have same k-factor set

not locally testable

use first-order logic instead:
$\neg(\exists x, y)[x \neq y \wedge b(x) \wedge b(y)]$

Logical levels

FO: locally threshold testable

|

Prop: locally testable

|

CNL: strictly local

Long-distance dependencies: Asymmetric harmony

ʃotos but not sotoʃ

(attested in Sarcee)

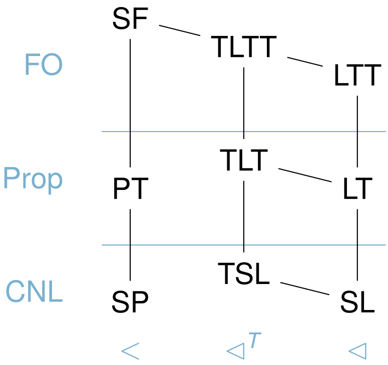
Asymmetric harmony via precedence

Reanalyzing harmony: Tiers

s $\xrightarrow[\text{o t o}]{}$ ʃ

other symbols $\rightarrow$ neutral

Classes

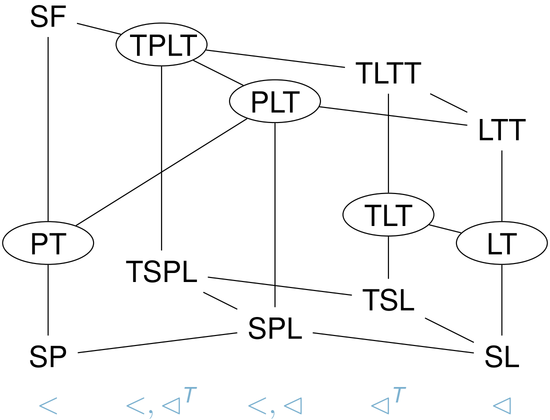


Mixed relations

factors "abc comes before def"
piecewise-locally testable = dot-depth one

propositional level subsumes LTT

Classes


SF
TPLT
TLTT
PLT
LTT
TLT
PT
LT
TSPL
TSL
SPL
SP
SL
$<$
$<, \lhd^T$
$<, \lhd$
$\lhd^T$
$\lhd$

Varieties

Variety

a class $\mathcal{V}$ where for each alphabet Σ,
if $L_1, L_2 \in \Sigma^*\mathcal{V}$:

- $\complement L_1 \in \Sigma^*\mathcal{V}$ and $L_1 \cup L_2 \in \Sigma^*\mathcal{V}$
  Boolean operations*
- $\sigma^{-1}L_1 \in \Sigma^*\mathcal{V}$ and $L_1\sigma^{-1} \in \Sigma^*\mathcal{V}$
  Quotients
- $f : \Gamma \to \Sigma$ homomorphic, $f^{-1}(L_1) \in \Gamma^*\mathcal{V}$
  Inverse homomorphisms

Eilenberg's theorem

$\Sigma^*\mathcal{V} \sim$ variety of monoids
$\Sigma^+\mathcal{V} \sim$ variety of semigroups

collection closed under

- submonoid (subsemigroup)
- quotient
- finitary direct product

Piecewise branch, expanded

$$\mathrm{SF} = \mathrm{A} = [\![x^{\omega}x = x^{\omega}]\!]$$

$$\mathrm{DA} = [\![(xyz)^{\omega}y(xyz)^{\omega} = (xyz)^{\omega}]\!]$$

$$\mathrm{PT} = \mathrm{J} = [\![(xy)^{\omega}x = (xy)^{\omega} = y(xy)^{\omega}]\!]$$

$$\mathrm{LTT}_{1,t} = \mathrm{ACom} = [\![xy = yx; x^{\omega}x = x^{\omega}]\!]$$

$$\mathrm{SP} = [\![1 \leq x]\!]$$

$$\mathrm{PT}_1 = \mathrm{LT}_1 = \mathrm{J}_1 = [\![xy = yx; xx = x]\!]$$

The local branch

convert k-factors to their own individual letters

$$\mathbf{V} \mapsto \mathbf{V} * \mathbf{D}$$

contains corresponding piecewise class

Local branch

$\text{SF} = \text{A} = \text{A} * \text{D}$

$\text{FO}^2[<, \lhd] = \text{DA} * \text{D}$

$\text{PLT} = \text{J} * \text{D}$

$\text{LTT} = \text{ACom} * \text{D}$

$\text{SPL} = \text{SP} * \text{D}$

$\text{LT} = \text{J}_1 * \text{D}$

The M operator

$\mathbf{MV}$ the variety of monoids
generated by $S^{\cdot}$ for $S \in \mathbf{V}$

M = multiple tiers

linguistic “lift onto a tier” = algebraic “$S \mapsto S^{\cdot\cdot}$”

multiple tiers interacting (Boolean combinations):
converts $+$-variety $\mathcal{V}$ to $*$-variety $\mathcal{MV}$

Learning framework

What kinds of data do learners receive?

How do we extrapolate from that back to patterns?

Limit-learnability with positive data

- Only valid words happen
- Every valid word will eventually happen
- Finite samples
- Incrementally: eventually hypothesis stops changing

String extension learning

- Assume nothing is valid
- For each word, extract information
- Add that information into a “grammar”
- Information is never removed from the “grammar”
- How is this interpreted?

Learning SP

Information: “set of subsequences”
Insertion: set-union

Interpretation:
valid iff set of subsequences is subset of grammar

Learning J_1

Information: “set of letters”
Insertion: element-insertion

Interpretation:
valid iff set of letters in grammar

Learning Acom

Information: “thresholding multiset of letters”
Insertion: element-insertion

Interpretation:
valid iff multiset of letters in grammar

Learning PT

Information: “set of subsequences”
Insertion: element-insertion

Interpretation:
valid iff set of subsequences in grammar

Learning locality

choose k
apply factor–letter transformation
learn piecewise base class

Learning multitiers

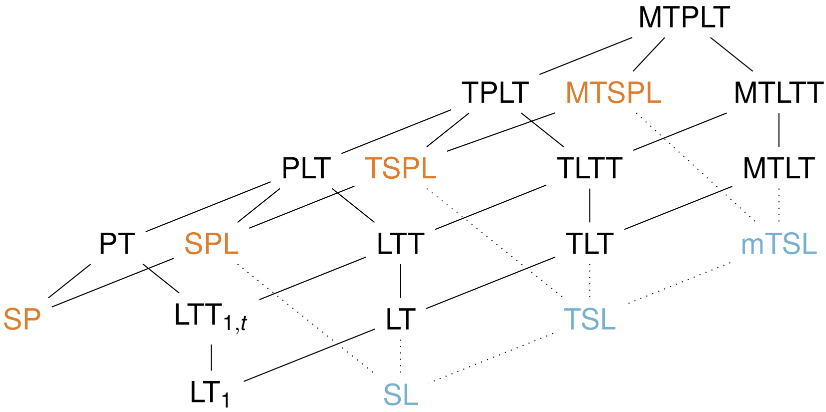
choose *k*
for each subset of the alphabet:
apply erasing transformation then factor–letter transformation
information based on this collection

The system

- Piecewise base class
- Close under inverse factor-collapse (“localize”)
- Close under neutral-letter injection (“tierify”)
- Close under Boolean operations (“multitierify”)
- Result: a new Piecewise base class

$$\mathbf{A} = \mathbf{A} * \mathbf{D} = \mathbf{T}(\mathbf{A} * \mathbf{D}) = \mathbf{M}(\mathbf{A} * \mathbf{D})$$

Classes



and so much more

Future directions

- Parameter-finding from machines
  known for some like D, Acom, …
- Decomposition of machines
- Same structural classes apply to functions
  inferring those? (SOSFIA/++)
- Other bases