Subregular Sets, Functions, Relations and their Learnability

Jeffrey Heinz

February 5, 2018

Linguistics Department

@ Google Speech and Language Algorithms

NY, NY

Some Types of Automata

	Boolean	[0,1]	Σ^*
Deterministic	DFA	PDFA	DFT
Non-deterministic	NFA	PNFA	NFT

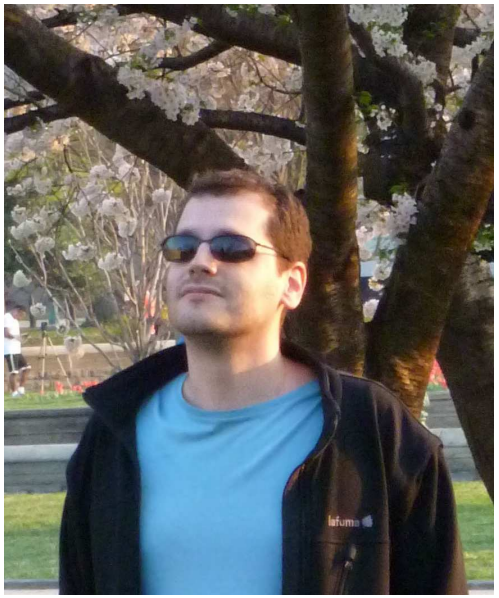
Overview

1. Learning arbitrary DFAs, PDFAs, DFTs
2. Subregular classes of stringsets
3. Learning them from positive examples
4. Strictly Local string-to-string functions
5. Relations!

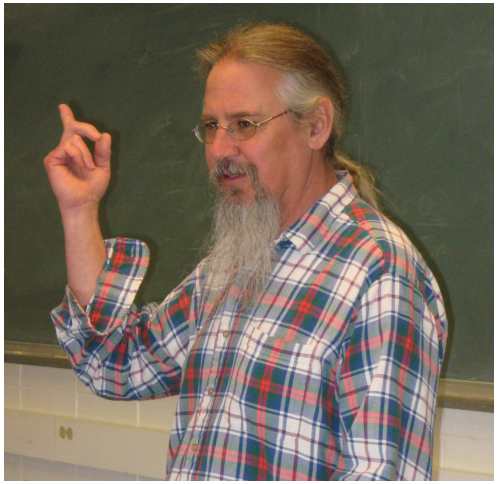
Morales: Determinism is important. Monoids are too.

Collaborators

Jane Chandlee
Haverford College



Rémi Eyraud
U. of Marseilles



Jim Rogers
Earlham College

Classes of stringsets and functions

Theorems

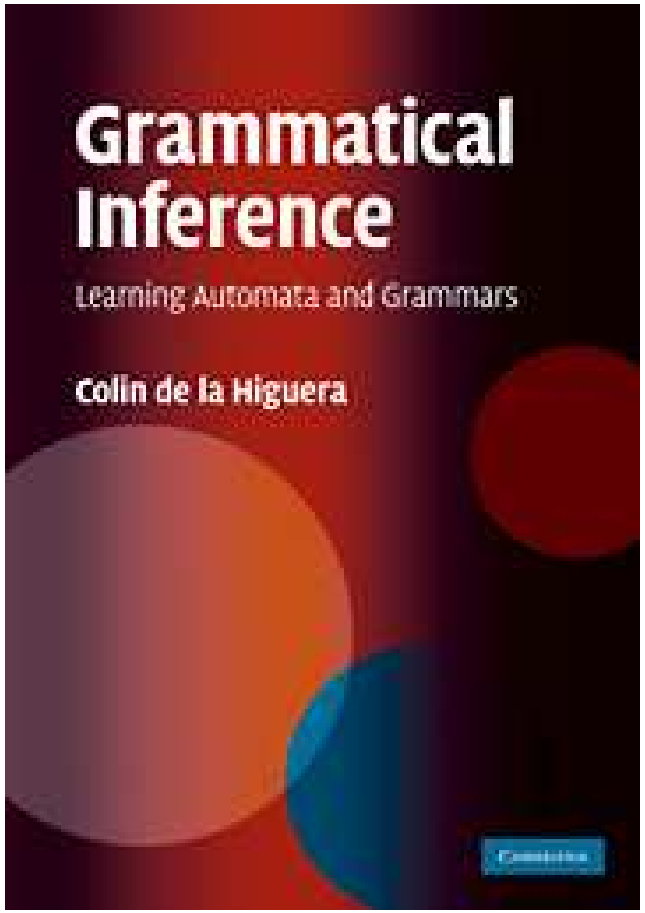
1. No algorithm identifies the regular stringsets in the limit from positive data only.
2. Algorithm RPNI identifies the regular stringsets in the limit from positive and negative data.
3. Algorithm ALERGIA identifies the class of stochastic stringsets expressed with PDFAs in the limit with probability one.
4. Algorithm OSTIA identifies the class of total sequential functions (those expressed with DFTs) in the limit.

- These algorithms are non-parametric: both the automaton's structure and the transitional values are learned.
- Time complexities are generally cubic in the size of the input data.

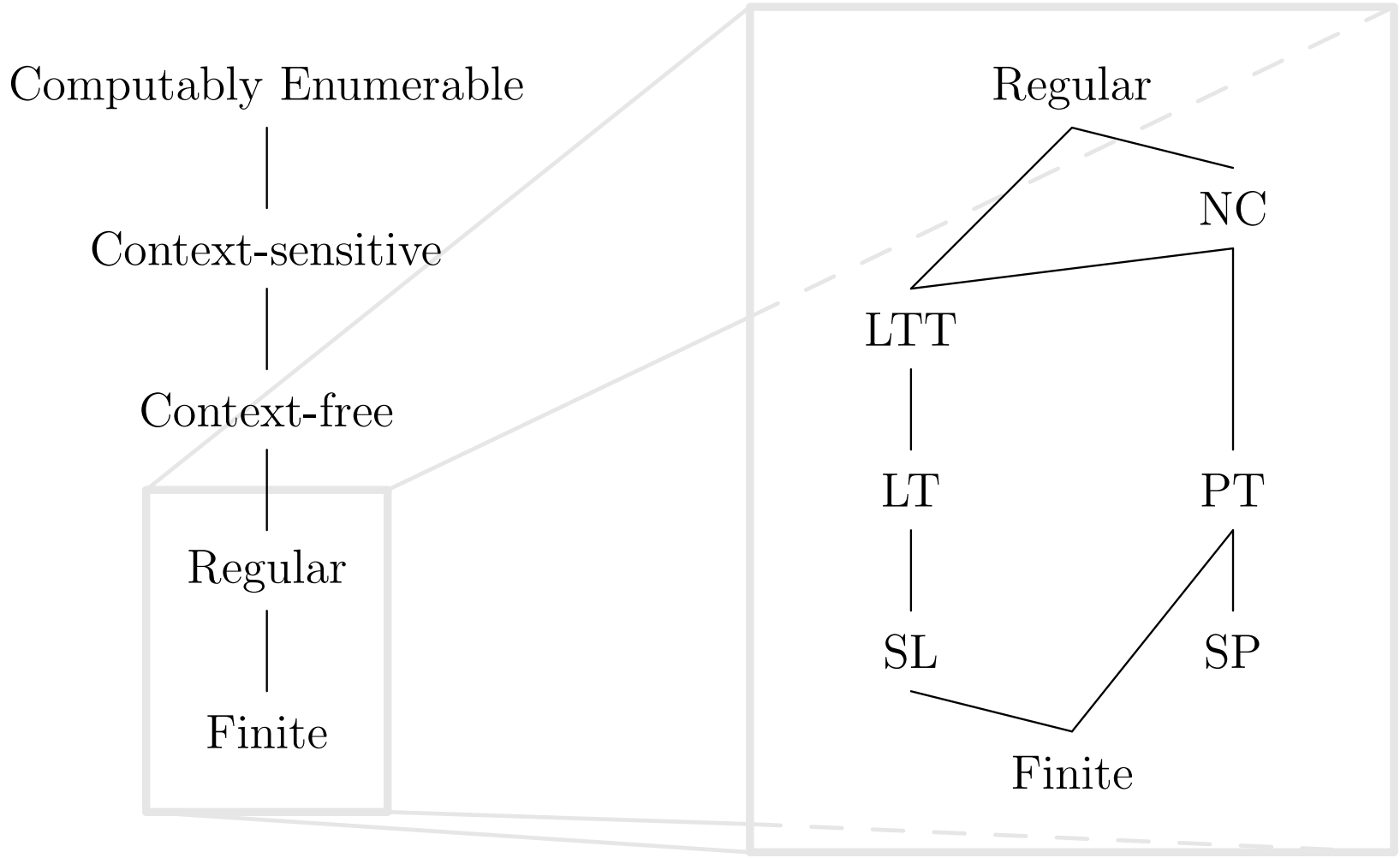
(Oncina and Garcia 1992, Oncina et al. 1993, Carrasco and Oncina 1994, de la Higuera and Thollard 2000, de la Higuera 2010)

de la Higuera 2010

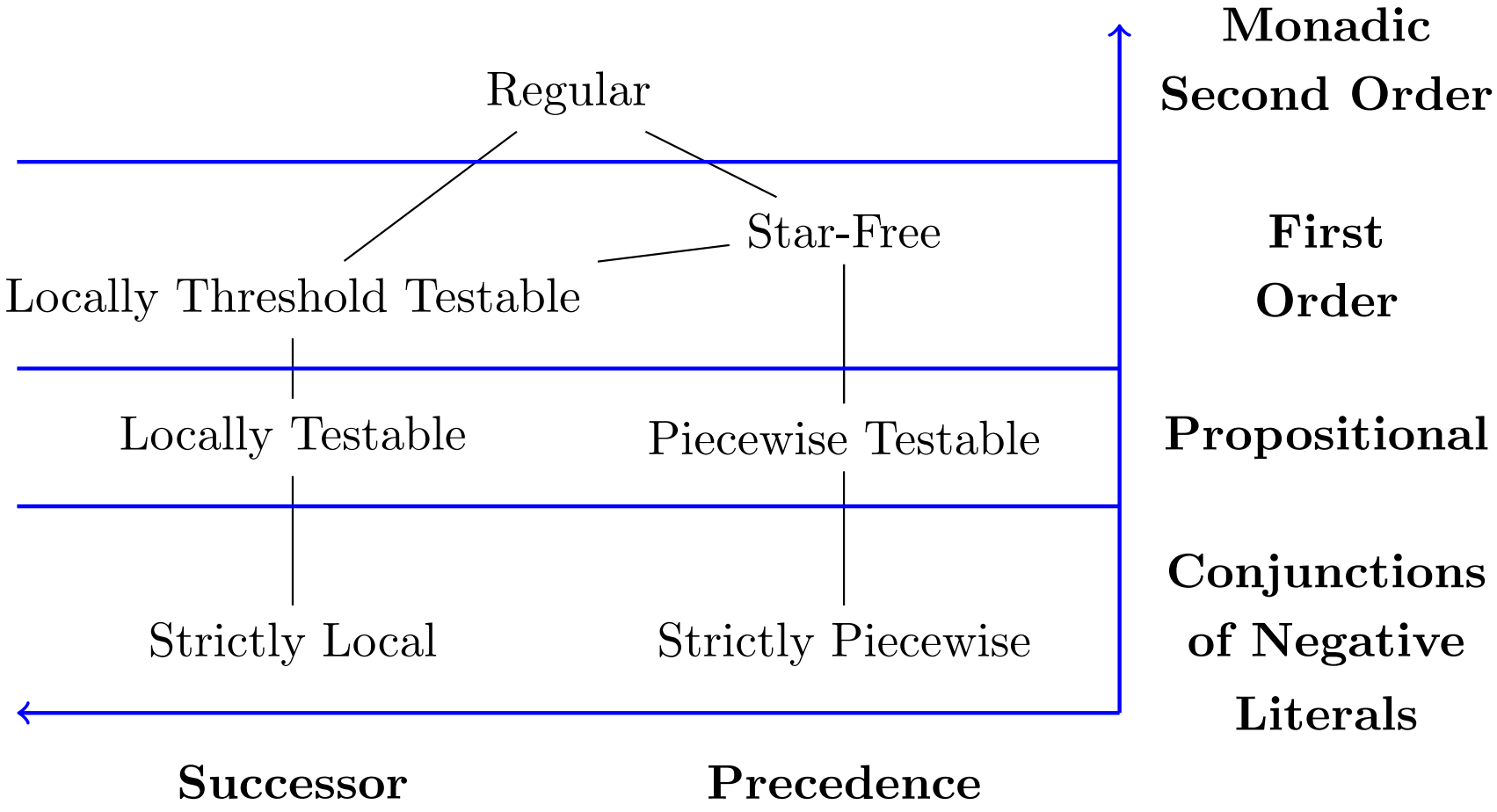
This book explains those algorithms and some variants.



Formal Language Hierarchies



Subregular Classes of Stringsets



(McNaughton and Papert 1971, Rogers and Pullum 2011, Rogers et al. 2013)

Strictly k-Local Stringsets

A stringset L is Strictly k-Local iff there are finitely many strings $\ell_1, \ldots \ell_{n_\ell}, w_1, \ldots w_{n_w}, r_1, \ldots r_{n_r}$, the longest of which is length k, such that

$$\begin{aligned} L \quad = \quad & \overline{\ell_1 \Sigma^*} \cap \ldots \cap \overline{\ell_{n_\ell} \Sigma^*} \\ & \cap \overline{\Sigma^* w_1 \Sigma^*} \cap \ldots \cap \overline{\Sigma^* w_{n_w} \Sigma^*} \\ & \cap \overline{\Sigma^* r_1} \cap \ldots \cap \overline{\Sigma^* r_{n_r}} \end{aligned}$$

Essentially, L forbids certain strings at left edges, right edges, and string-internally.

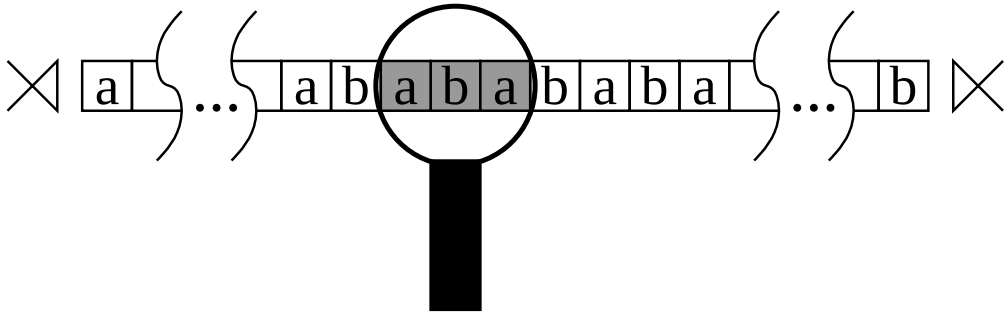
- $\mathrm{SL}_k := \{L \mid L \text{ is Strictly k-Local}\}$
- $\mathrm{SL} := \bigcup_k \mathrm{SL}_k$

Thm. For all k, $\mathrm{SL}_k \subsetneq \mathrm{SL}_{k+1}$.

(McNaughton and Papert 1971)

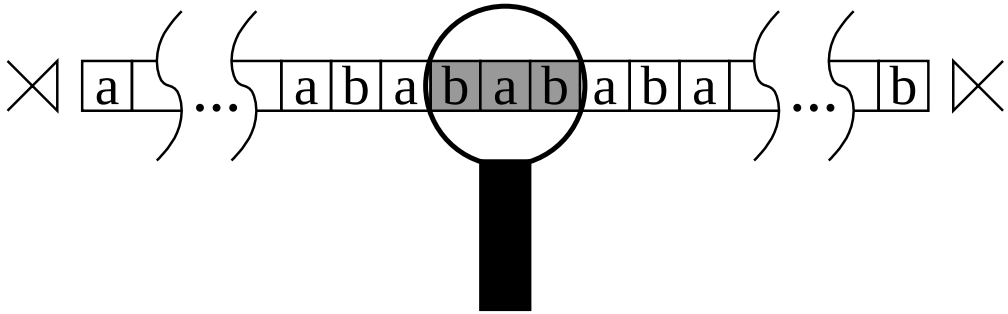
Strictly k-Local Stringsets, Pictorially

- To check whether a string belongs to a given Strictly k-Local stringset, we can just scan a window of size k checking for forbidden strings ℓ_i, w_i, r_i.



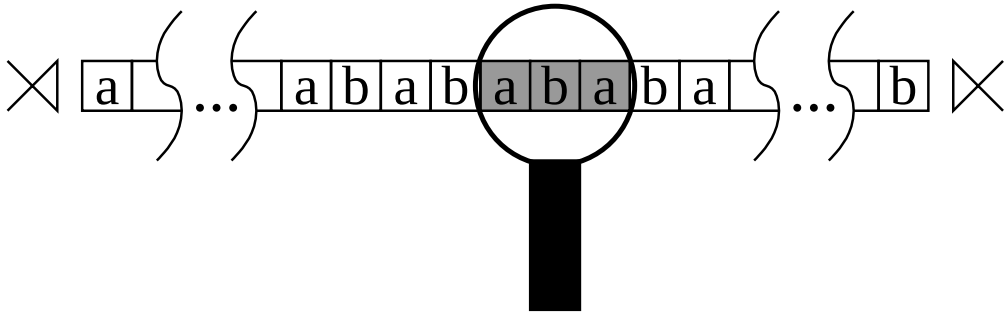
Strictly k-Local Stringsets, Pictorially

- To check whether a string belongs to a given Strictly k-Local stringset, we can just scan a window of size k checking for forbidden strings ℓ_i, w_i, r_i.



Strictly k-Local Stringsets, Pictorially

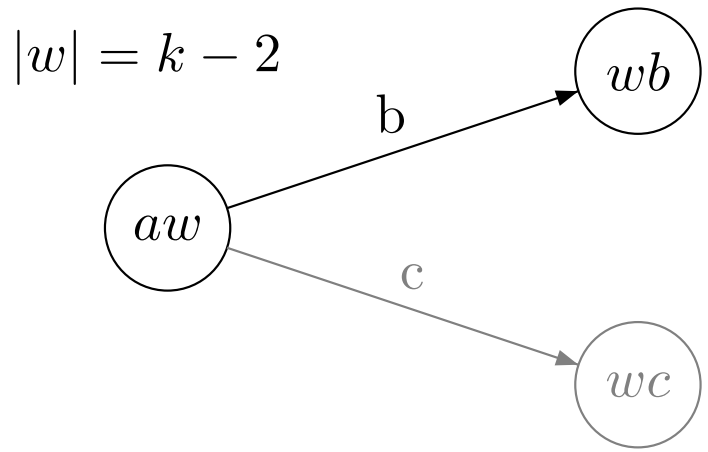
- To check whether a string belongs to a given Strictly k-Local stringset, we can just scan a window of size k checking for forbidden strings ℓ_i, w_i, r_i.



Finite-State Acceptor for Strictly k-Local Stringsets

Thm. There is a deterministic finite-state acceptor (DFA) whose sub-DFAs correspond to every $L \in \mathrm{SL}_k$.

- The states correspond to the last $k-1$ symbols read.
- Formally: $Q = \Sigma^{\leq k-1}$ and $\delta(q, a) = \texttt{Suff}^{k-1}(qa)$.

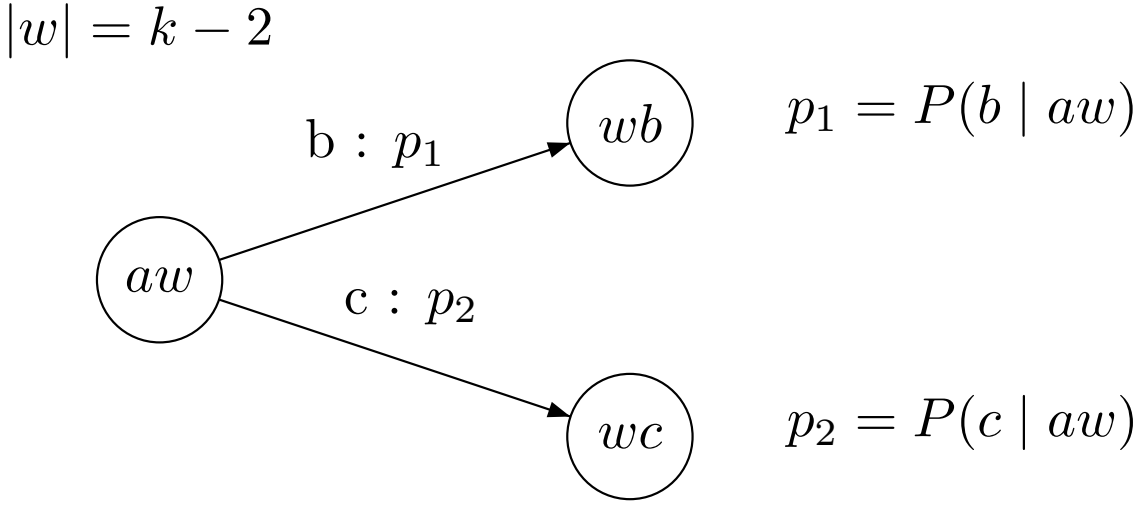


awb is a permissible substring.

awc is a forbidden substring.

N-gram models are stochastic variants of $\mathbf{SL}_n$ stringsets

- The states correspond to the last $k-1$ symbols read.
- Formally: $Q = \Sigma^{\leq k-1}$ and $\delta(q, a) = \texttt{Suff}^{k-1}(qa)$.



$\mathbf{SL}_k$ Theorems on Learning

1. For each k, there is an algorithm which identities SL_k in the limit from positive data.
2. The time complexity is **linear** in the size of the data and its data complexity is **linear** in the size of the automata-theoretic representation.
3. Though the DFA is of size $|\Sigma|^{k-1}$.

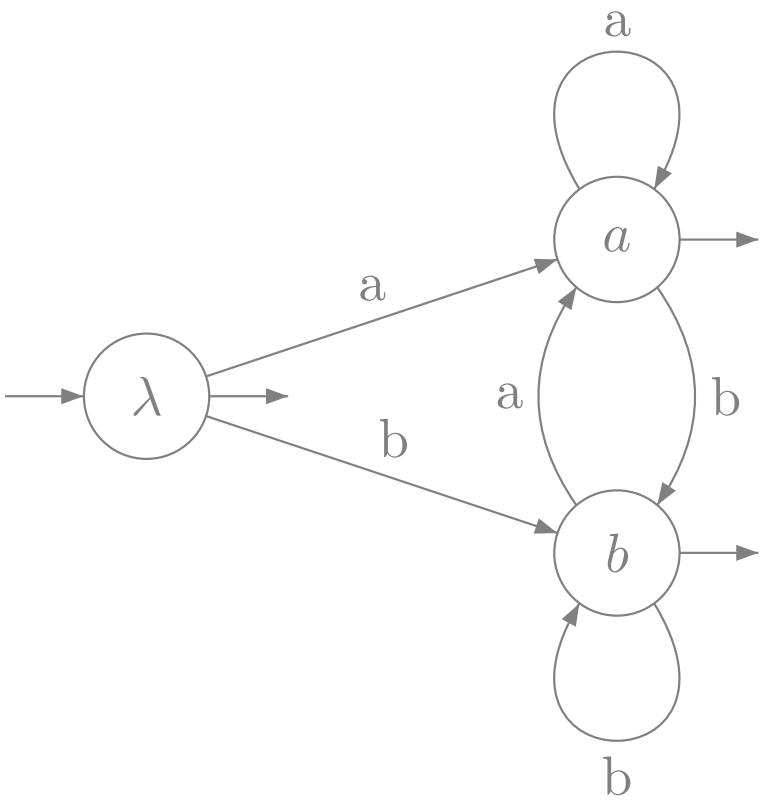
Stochastic $\mathbf{SL}_k$ Theorems on Learning (k-gram model)

1. For each k, there is an algorithm which outputs parameters of the k-gram model which maximize the likelihood of D (MLE).
2. If D was drawn from i.i.d. from a k-gram model then as D gets larger, then the error between learned parameters and true parameters goes to zero (consistency).
3. The time complexity is **linear** in the size of D.

(Garcia et al. 1991, Jurafsky and Martin 2008)

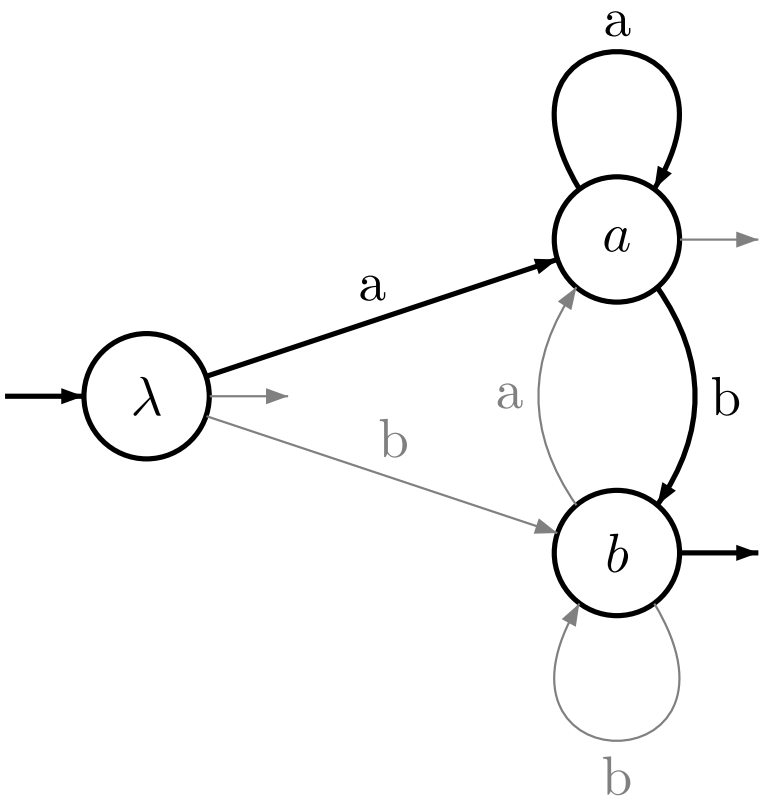
SL Example: $\Sigma = \{a, b\}$, $k = 2$

DFA



SL Example: $\Sigma = \{a, b\}$, $k = 2$

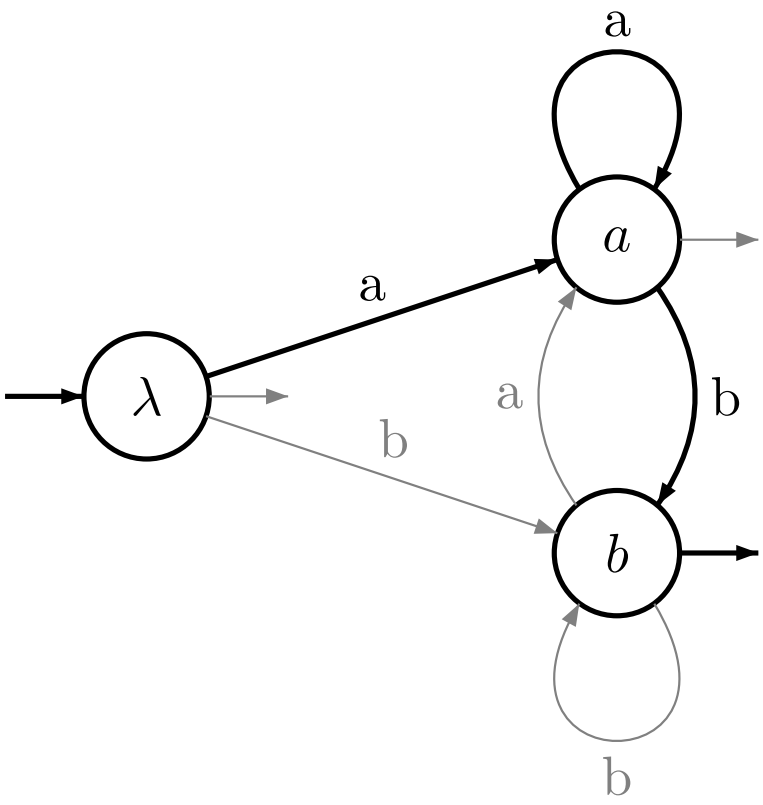
DFA



$$D = \{aab\}$$

SL Example: $\Sigma = \{a, b\}$, $k = 2$

DFA

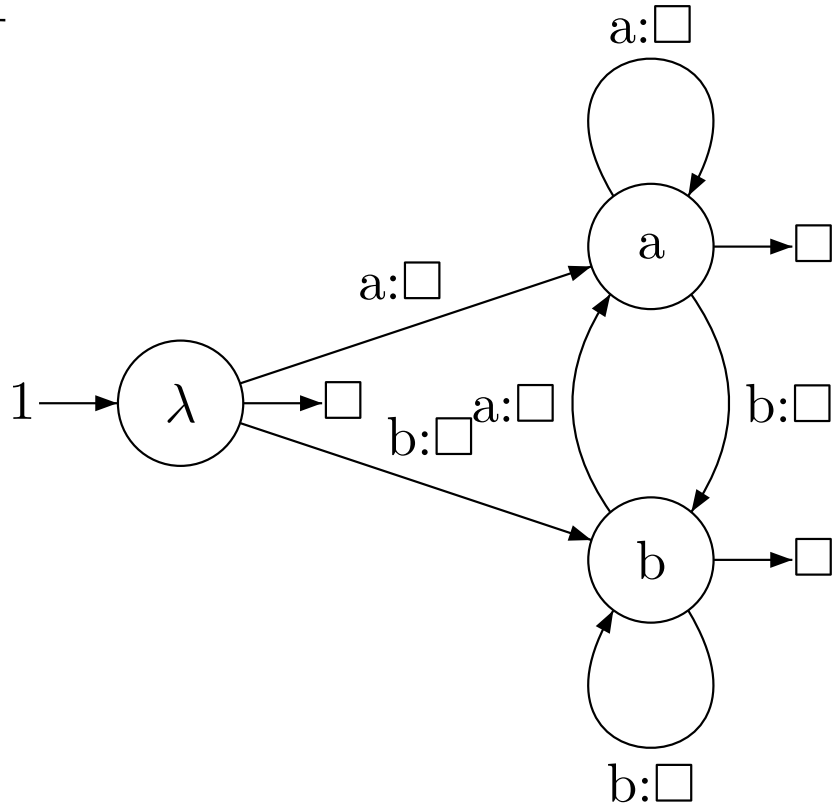


$$D = \{aab\}$$

$$L = aa^*b$$

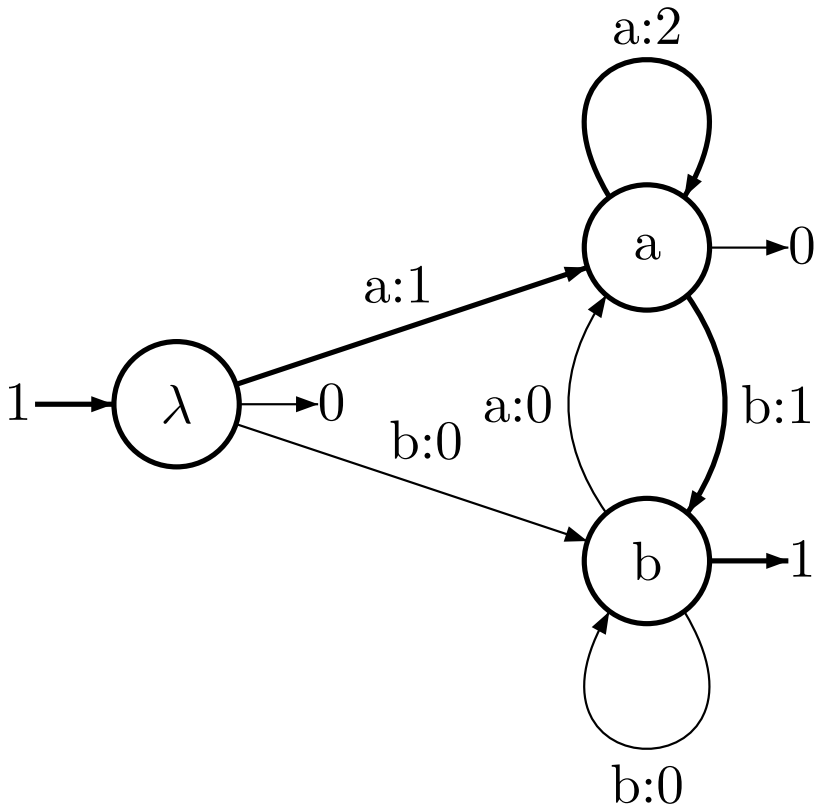
SL Example: $\Sigma = \{a, b\}$, $k = 2$

PDFA



SL Example: $\Sigma = \{a, b\}$, $k = 2$

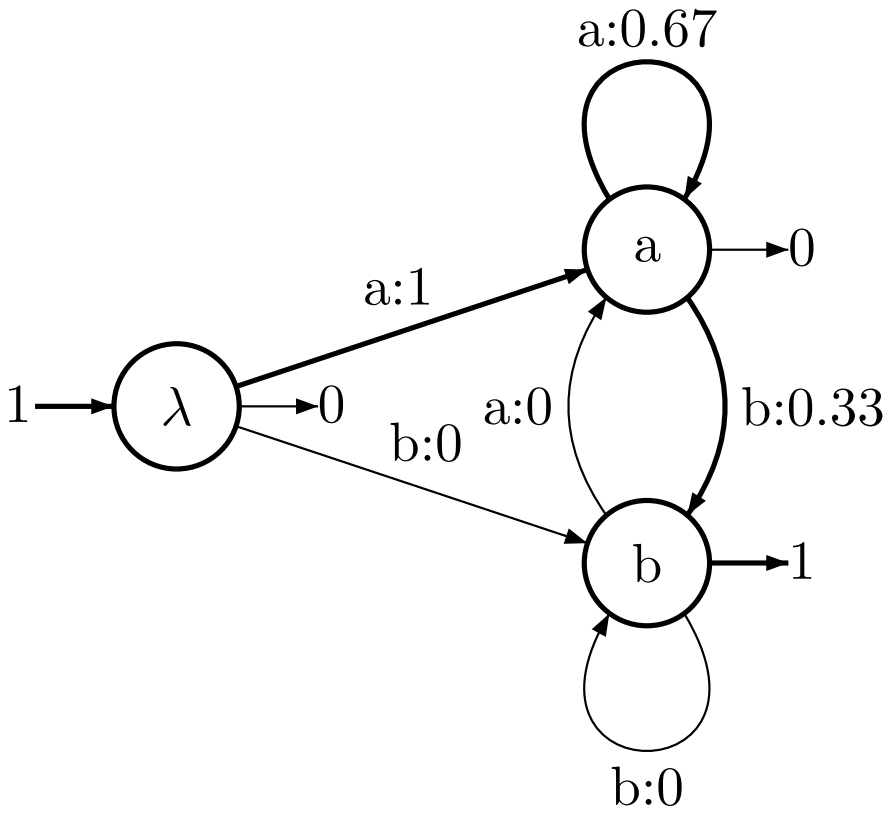
DFA



$$D = \{aaab\}$$

SL Example: $\Sigma = \{a, b\}$, $k = 2$

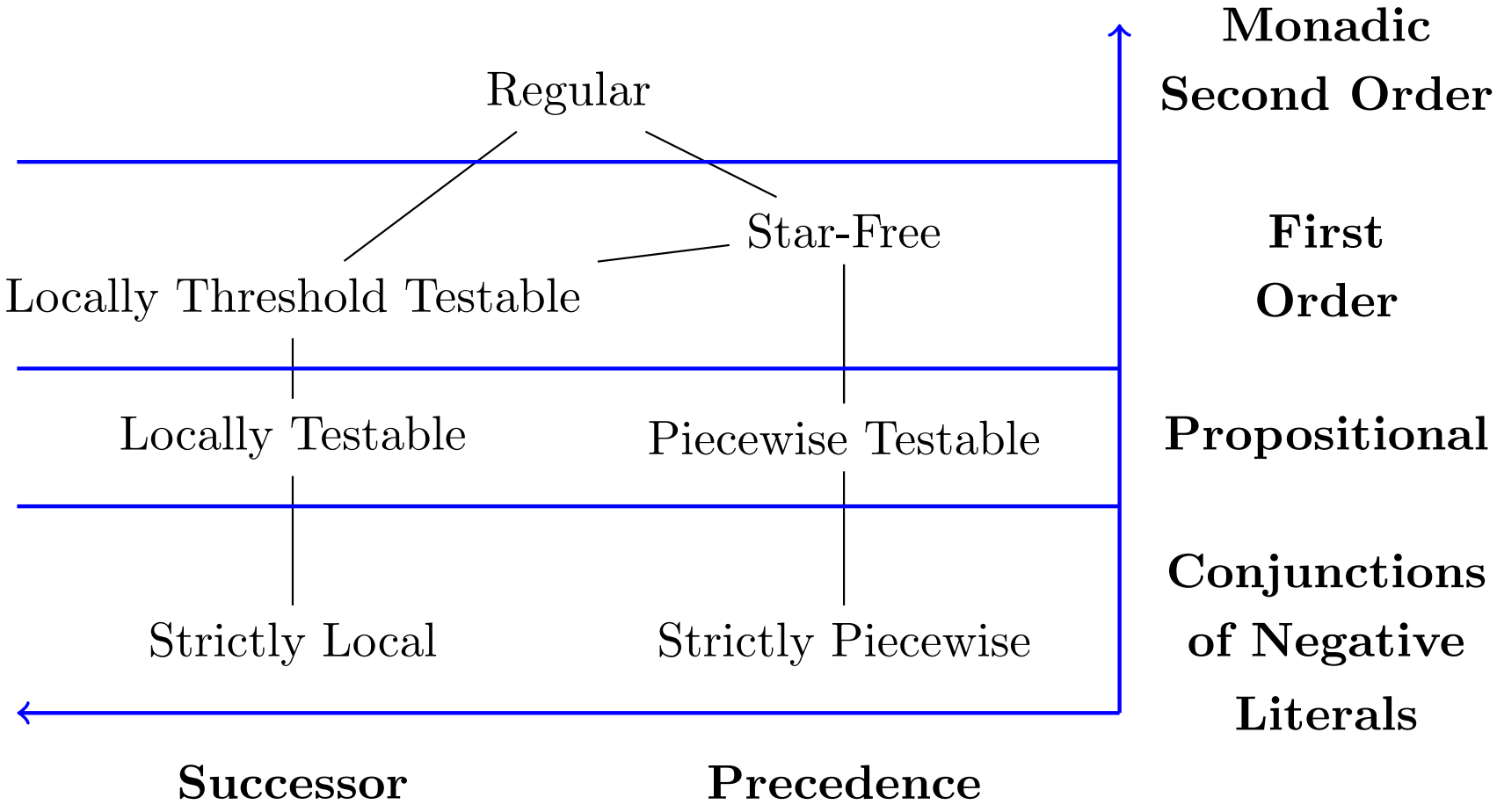
DFA



$$D = \{aaab\}$$

$$L = aa^*b$$

Subregular Classes of Stringsets



(McNaughton and Papert 1971, Rogers and Pullum 2011, Rogers et al. 2013)

Other subregular classes

Locally Testable. Boolean closure of SL. Parameterized by window-size k.

"If aa is a substring then bb must also be a substring."

Locally Threshold Testable. Adds to LT the counting of substrings up to a threshold t. Parameterized by k, t.

"If there are three aa substrings then bb must also be a substring."

(Thomas 1982, McNaughton and Papert 1971, Rogers and Pullum 2011)

Other subregular classes

Strictly Piecewise. Forbids subsequences of size k.

"Words with neither $a \dots b$ nor $b \dots b$ as subsequences."

Piecewise Testable. Boolean closure of SP. Parameterized by k.

"If $a \dots a$ is subsequence then $b \dots b$ is a subsequence."

(Simon 1975, Rogers et al. 2010, 2013)

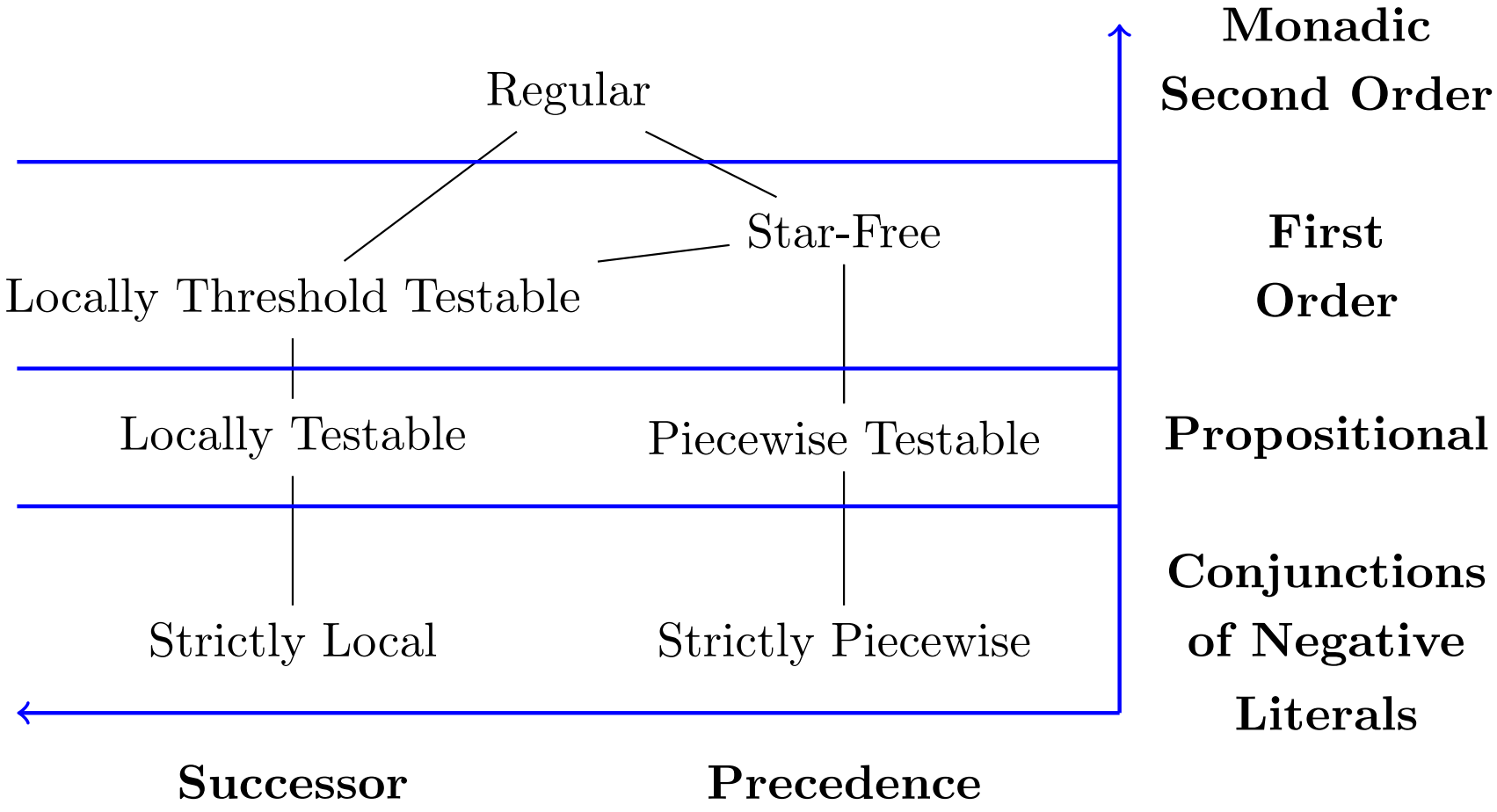
Star-free Stringsets

- Theoretically, regular expressions (RE) have three operations: $\cdot$ (concatenation), $+$ (union), * (Kleene star) and base cases $\emptyset$, λ, $\sigma \in \Sigma$.
- Generalized REs (GRE) add two more operations: $^{-}$ (complementation w.r.t. Σ^*) and $\times$ (intersection)
- Star-Free stringsets are all and only those that can be expressed with a GRE with no Kleene star *.

Thms. Star-Free $\equiv$ closure of LT under concatenation
$\equiv$ First-Order definable stringsets with precedence model

(McNaughton and Papert 1971)

Subregular Classes of Stringsets



(McNaughton and Papert 1971, Rogers and Pullum 2011, Rogers et al. 2013)

Learning results

Thms. For all k, t:

1. For each t, k, SL_k, LT_k, $\mathrm{LTT}_{t,k}$, SP_k, and PT_k are identifiable in the limit from positive data.
2. The time complexity of these algorithms is **linear** in the size of the data and the data complexity is **linear** in the size of the automata-theoretic representation.
3. Though, except for SP_k, the automata-theoretic representation is exponential in $|\Sigma|^{k-1}$.
4. For each class C in

$$\bigcup_{k,t} \{\mathrm{SL}_k, \mathrm{LT}_k, \mathrm{LTT}_{t,k}, \mathrm{SP}_k, \mathrm{PT}_k\} \ ,$$

   there is a deterministic, automata-theoretic representation A such that every stringset in C is a sub-representation of A.

(Heinz and Rogers 2013)

Learning Stochastic Stringsets

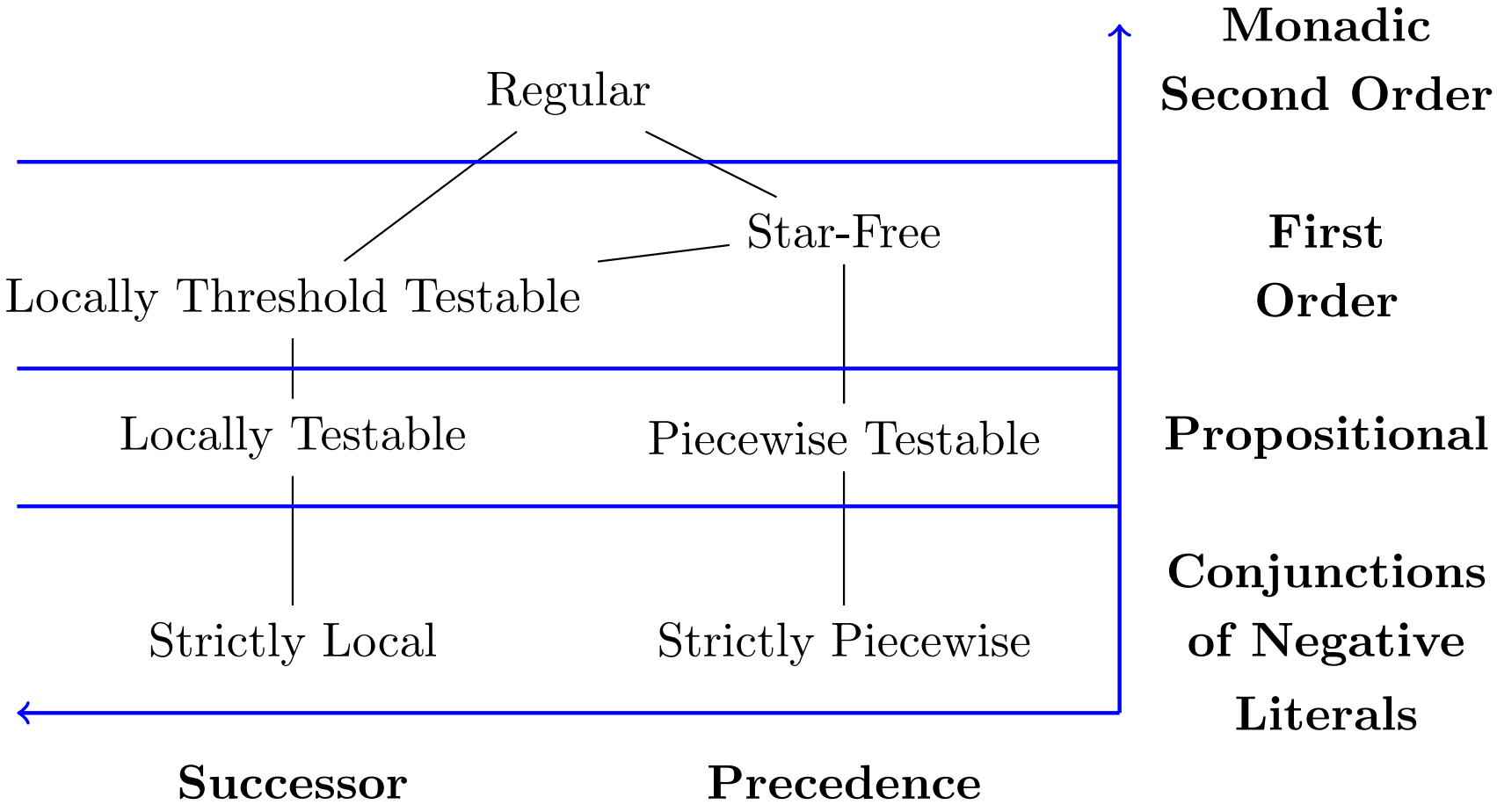
For each k, t, the automata-theoretic representation of each $\mathrm{SL}_k, \mathrm{LT}_k, \mathrm{LTT}_{t,k}, \mathrm{SP}_k, \mathrm{PT}_k$ provides a parametric model M of a family of stochastic stringsets whose parameters are the probabilities of the transitions in the automata.

Thms. For each model M and finite positive sample D:

- There is an algorithm which returns parameters of M which maximize the likelihood of D (MLE).
- If D was drawn from i.i.d from parameters set in M then as D gets larger, then the error between learned parameters and true parameters goes to zero (consistency).
- The time complexity is **linear** in the size of D.

(Vidal et al. 2005, Heinz and Rogers 2010)

Subregular Classes of Stringsets



(McNaughton and Papert 1971, Rogers and Pullum 2011, Rogers et al. 2013)

Linguistic Motivation

Doing typology requires two books:

- "encyclopedia of categories"
- "encyclopedia of types"

Hypothesis: Natural language phonotactics belongs to the conjunction of SL and SP constraints.

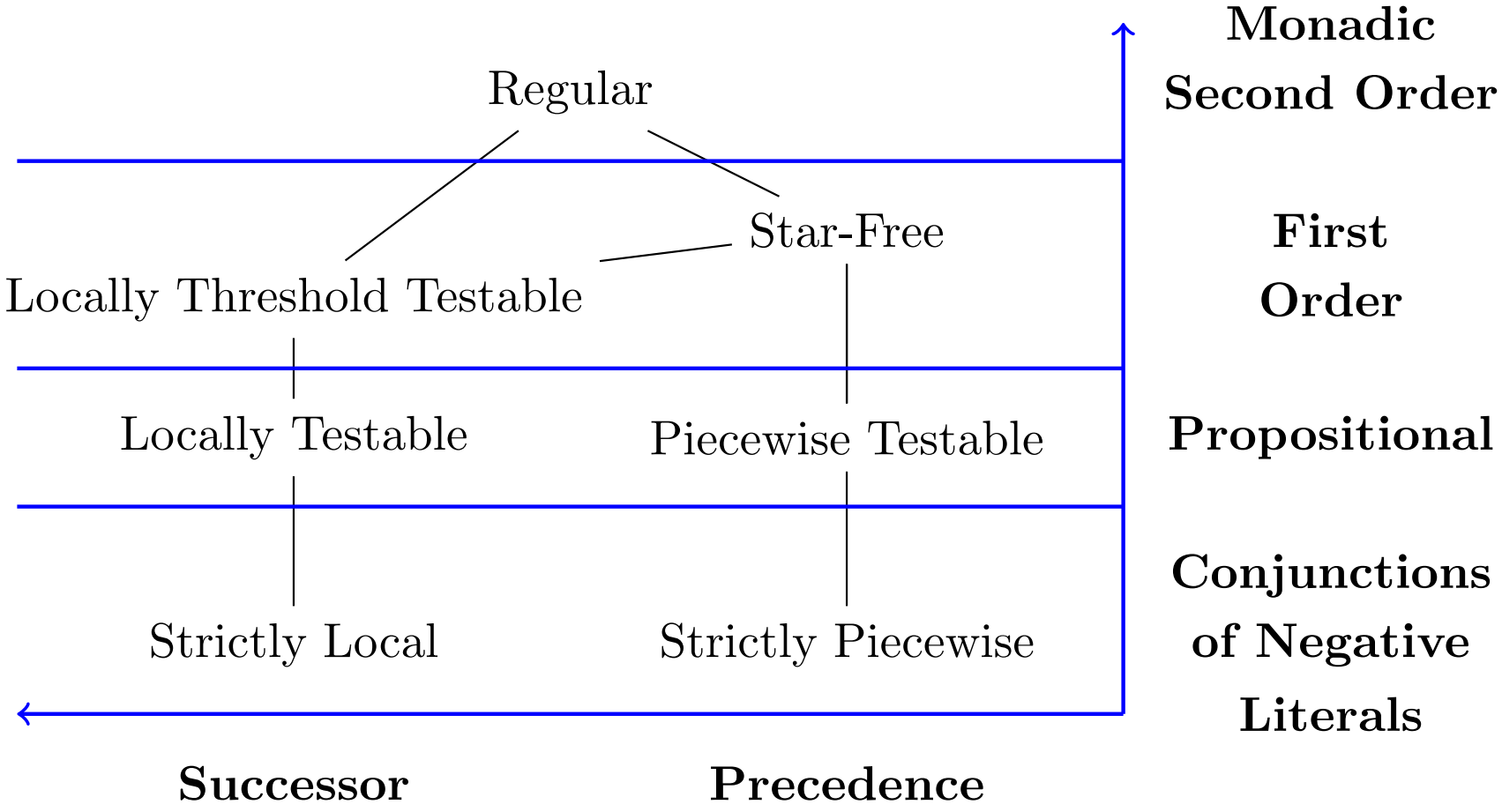
- Heinz 2010, 2014, forthcoming
- Heinz and Idsardi 2011, 2013
- Rogers et al. 2012
- Rogers and Lambert 2017
- cf. Heinz et al. 2011

Wilhelm Von Humboldt

Well-formedness and Transformations

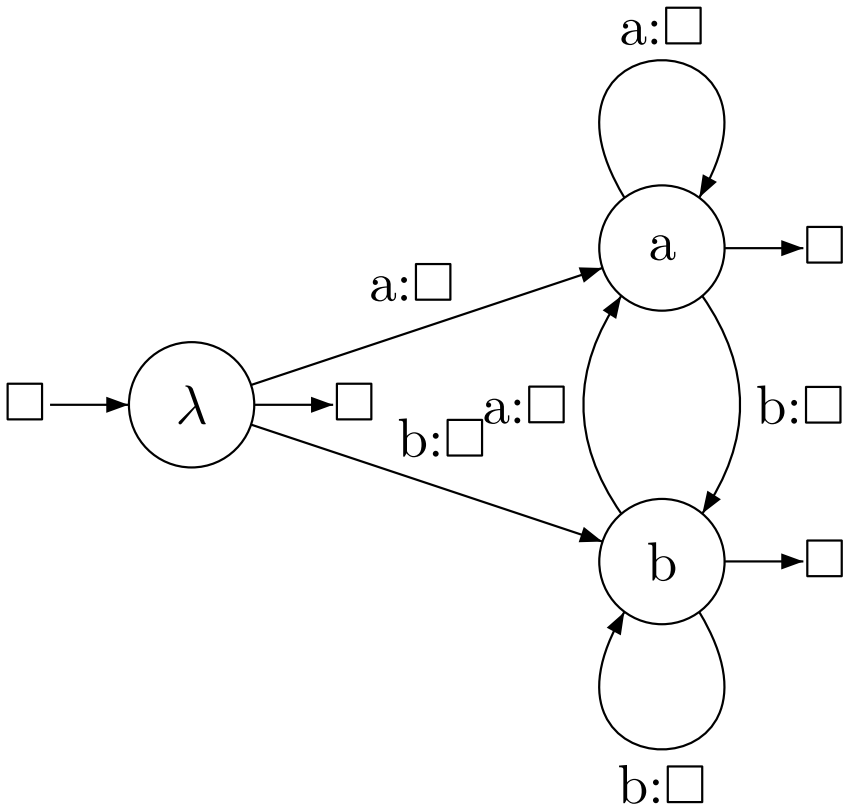
- Linguistic generalizations are not just constraints on well-formedness!
- They also describe transformations from one representation to another!

How do these classes generalize to functions and relations?



Strictly Local Functions

- Keep the structure of the DFA.
- Output strings instead of Boolean values or probabilities.



(Chandlee 2014, 2017, Chandlee et al. 2014, 2015 Chandlee and Heinz 2018)

Input Strictly Local Functions

$$x_0 \; x_1 \; \ldots \; x_n$$

$$\downarrow$$

$$u_0 \; u_1 \; \ldots \; u_n$$

where

1. Each x_i is a single symbol and each u_i is a *string*.
2. There exists a $k \in \mathbb{N}$ such that for all input symbols x_i its output string u_i depends only on x_i and the $k-1$ elements immediately preceding x_i.

Input Strict Locality: Main Idea in a Picture

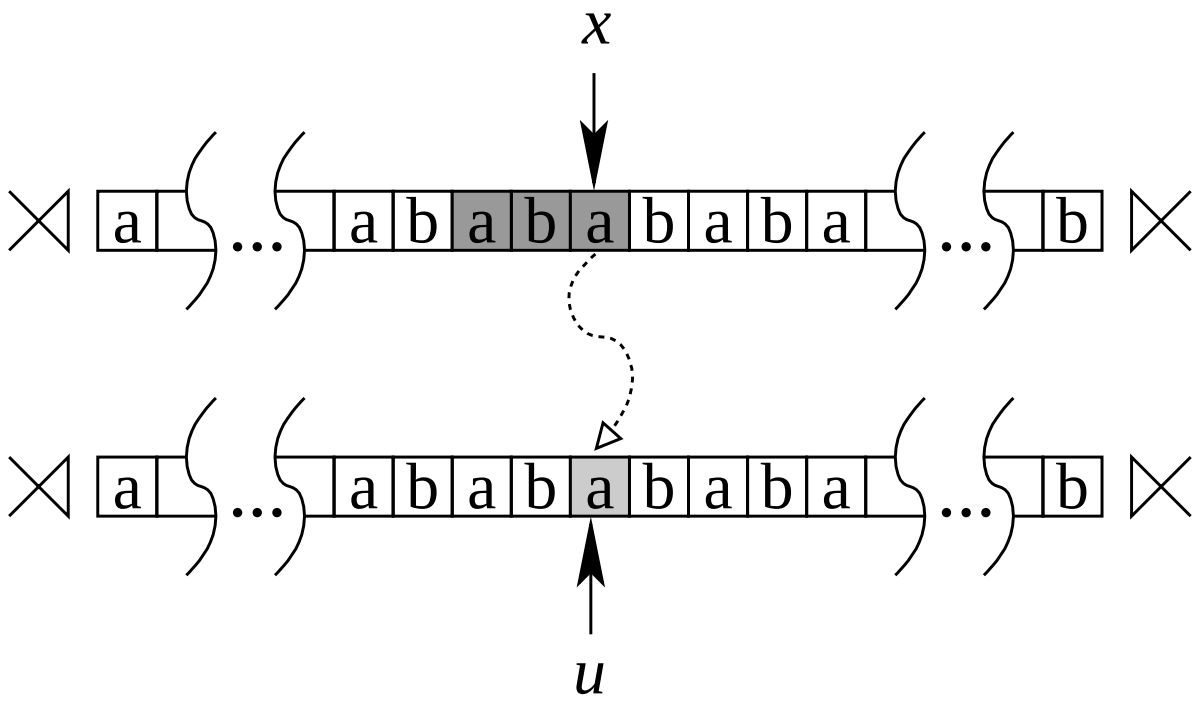


Figure 1: Illustrating an Input Strictly 3-Local function. The output string u of each input element x depends only on x and its two preceding input elements.

What can be modeled with ISL functions?

1. Many individual phonological and morphological processes.
   - local substitution, deletion, epenthesis, metathesis, ...
   - affixation, truncation, much partial reduplication, ...
2. Transformations describable with rules R: A $\longrightarrow$ B / C $\_\_$ D where
   - CAD is a finite set,
   - R applies simultaneously, and
   - contexts, but not targets, can overlap

   are ISL for k equal to the longest string in CAD.

(Chandlee 2014, Chandlee 2017, Chandlee and Heinz 2018)

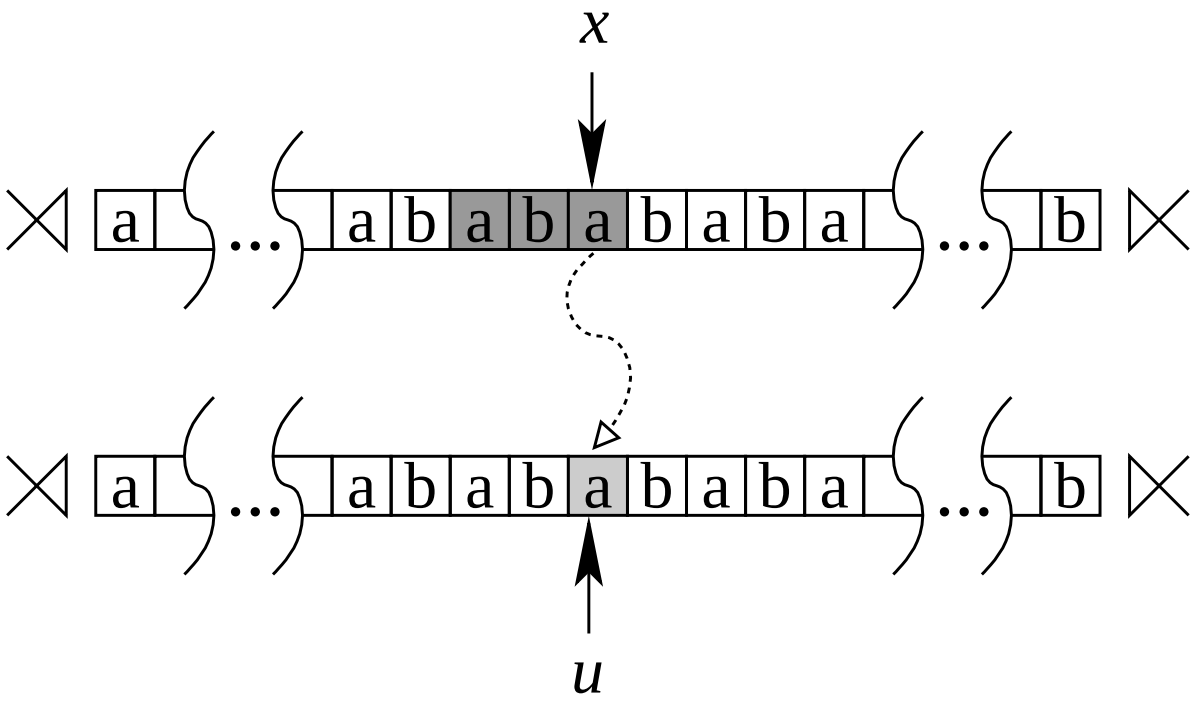
What can be modeled with ISL functions?

3. Approximately 95% of the individual processes in P-Base (v.1.95, Mielke (2008))
4. Many *opaque* transformations without any special modification.

(Chandlee 2014, Chandlee and Heinz 2018, Chandlee et al. to appear)

Interim Summary

Many phonological and morphological transformations have the *necessary information* to decide the output contained within a *window of bounded length* on the *input* side.



What phonological processes CANNOT be modeled with ISL functions

1. Progressive and regressive spreading

   /peŋawasan/ $\mapsto$ [peŋãw̃ãsan] 'supervision'

   (Johore Malay, Onn 1980)

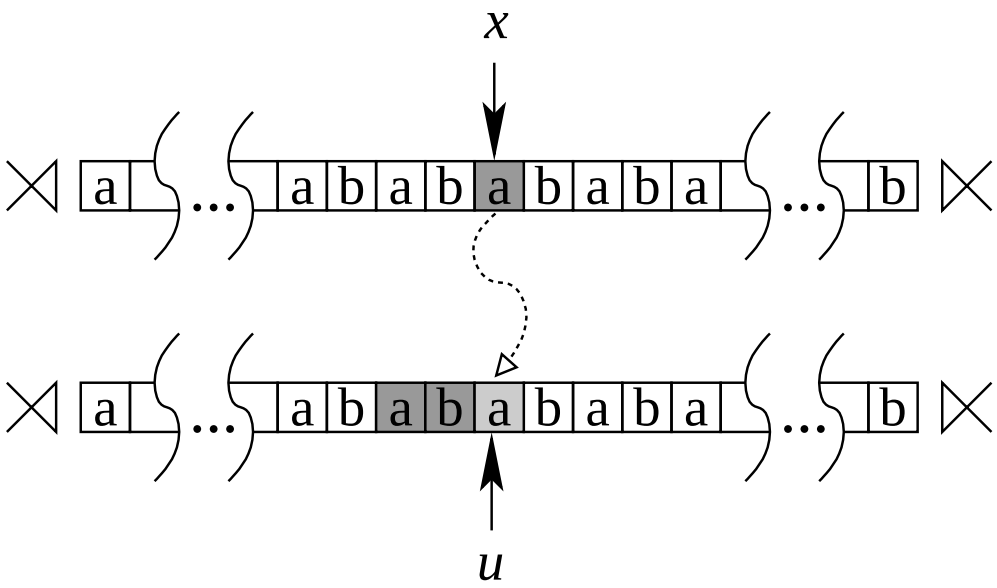
2. Long-distance (unbounded) consonant and vowel harmony

   /ku-ki<u>n</u>is-il-a/ $\mapsto$ [kukinisi**n**a] 'to make dance for'

   (Kikongo, Odden 1994)

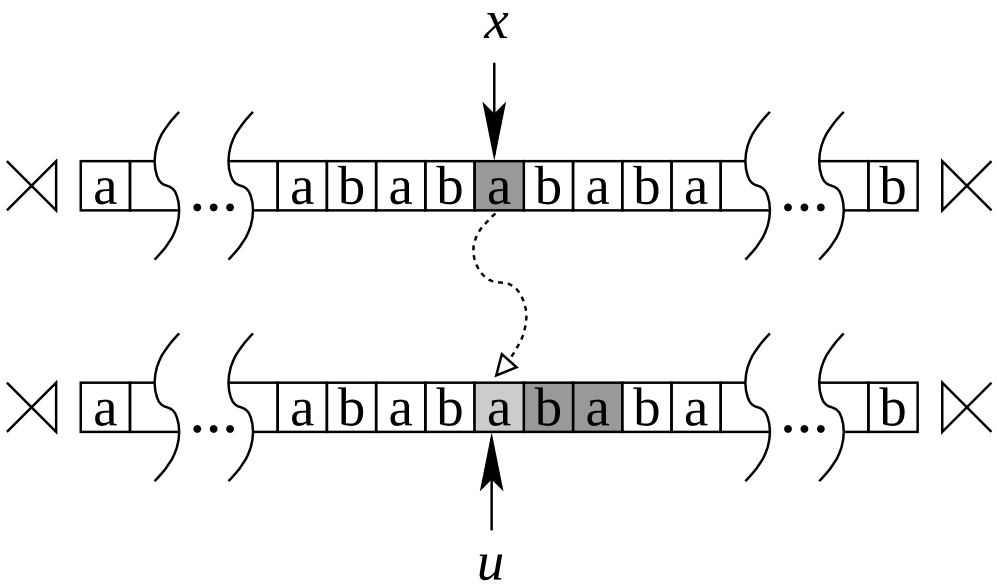
(Chandlee 2014, Chandlee and Heinz, 2018)

Left Output SL Functions



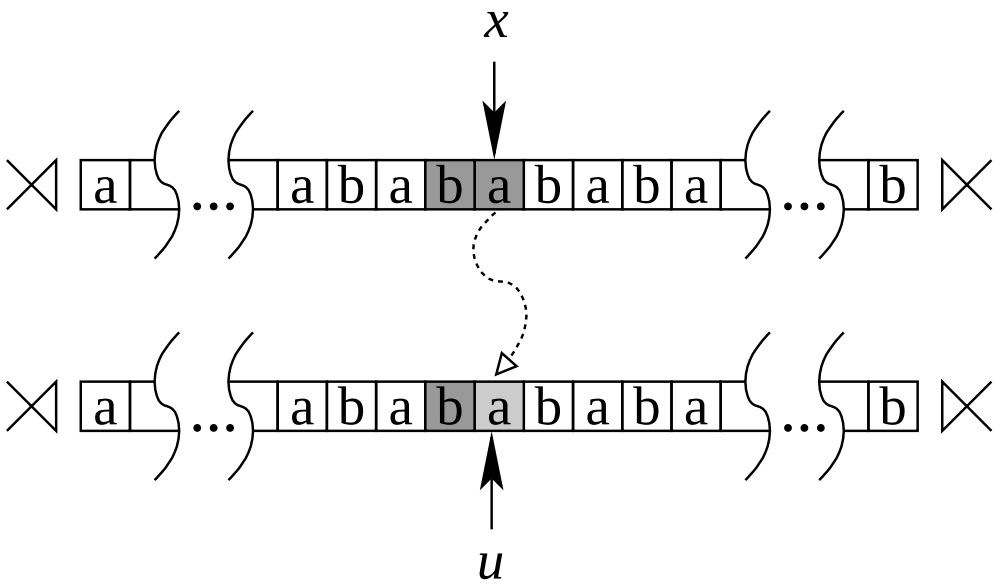
Left OSL definition. The output u written when reading an input symbol x depends only on x and the previous $k-1$ symbols written to the output.

Right Output SL Functions



Right OSL definition. Processing the input and output tapes from right to left, the output u written when reading an input symbol x depends only on x and the previous $k-1$ symbols written to the output.

Left and Right Input-Output SL functions



ISL and OSL functions can be synthesized so that the output depends on a window of size k in the input and a window of size k' in the left or right output.

(Chandlee, Eyraud and Heinz, in progress)

Comparison with Kaplan and Kay 1994

How does "aa→b" apply to *aaa*?

KK94 left-to-right:	ba	OSL-L:	ba
KK94 right-to-left:	ab	OSL-R:	ab
KK94 simultaneous:	{ab,ba}	ISL:	bb

Chomsky and Halle, SPE, p. 344: *"To apply a rule, the entire string is first scanned for segments that satisfy the environmental constraints of the rule. After all such segments have been identified in the string, the changes required by the rule are applied simultaneously."*

Strictly Local Functions

Thms.

1. For all k, $\text{ISL}_k \subsetneq \text{ISL}_{k+1}$.
2. For all k, $\text{L-OSL}_k \subsetneq \text{L-OSL}_{k+1}$.
3. For all k, $\text{R-OSL}_k \subsetneq \text{R-OSL}_{k+1}$.
4. ISL_k, L-OSL_k, and R-OSL_k are incomparable.
5. For each k, algorithm SOSFIA identities ISL_k in the limit from positive data with **linear** time and data complexity.
6. For each k, algorithm OSLFIA identifies (L/R)-OSL_k in the limit from positive data with **quadratic** time and data complexity.

(Jardine et al. 2014, Chandlee et al. 2015)

SOSFIA (and OSLFIA) calculate the *minimal change*

1. The output label for $\delta(q, a)$ is the minimal change between the common outputs of $q\Sigma^*$ and $qa\Sigma^*$ (cf. onwardness).
2. The *common output* of an input prefix w in a sample $S \subset \Sigma^* \times \Delta^*$ for t is the `lcp` of all $t(wv)$ that are in S:

$$\texttt{common\_out}_S(w) = \texttt{lcp}\Big(\{u \in \Delta^* \mid \exists v \text{ s.t. } (wv, u) \in S\}\Big)$$

3. The *minimal change in the output* in $S \subset \Sigma^* \times \Delta^*$ from w to $w\sigma$ is:

$$\texttt{min\_change}_S(w, \sigma) = \begin{cases} \texttt{common\_out}_S(\sigma) & \text{if } w = \lambda \\ \texttt{common\_out}_S(w)^{-1}\texttt{common\_out}_S(w\sigma) & \text{otherwise} \end{cases}$$

(Jardine et al. 2014, Chandlee et al. 2015)

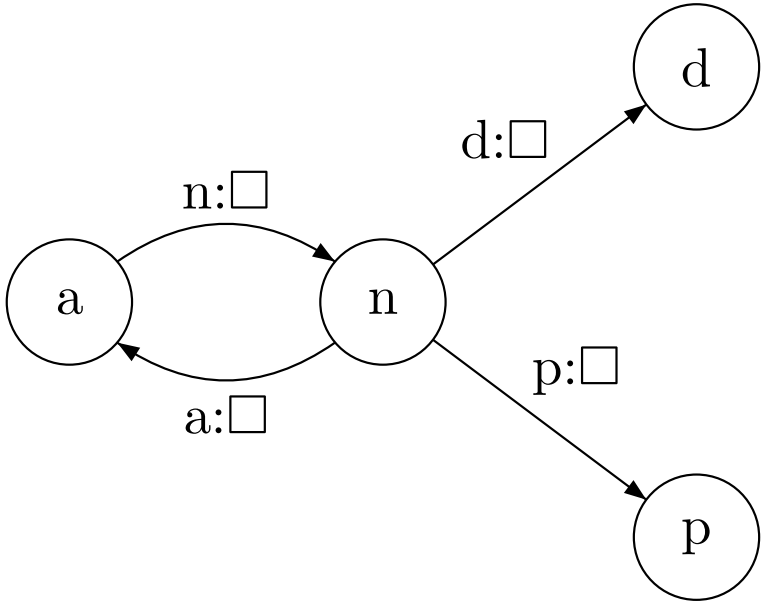
Example illustrating `min_change` with np→m (k=2)

If

$$S = \left\{ \begin{array}{lcllcl} (anpa & , & ama), & & (anpo & , & amo), \\ (ana & , & ana), & & (ano & , & ano), \\ (anda & , & anda), & & (ando & , & ando) \end{array} \right\}$$

Then

1. $\texttt{common\_out}_S(a) = a$
2. $\texttt{common\_out}_S(an) = a$
3. $\texttt{min\_change}_S(a, n) = \lambda$
4. $\texttt{min\_change}(an, p) = m$
5. $\texttt{min\_change}(an, a) = na$
6. $\texttt{min\_change}(an, d) = nd$



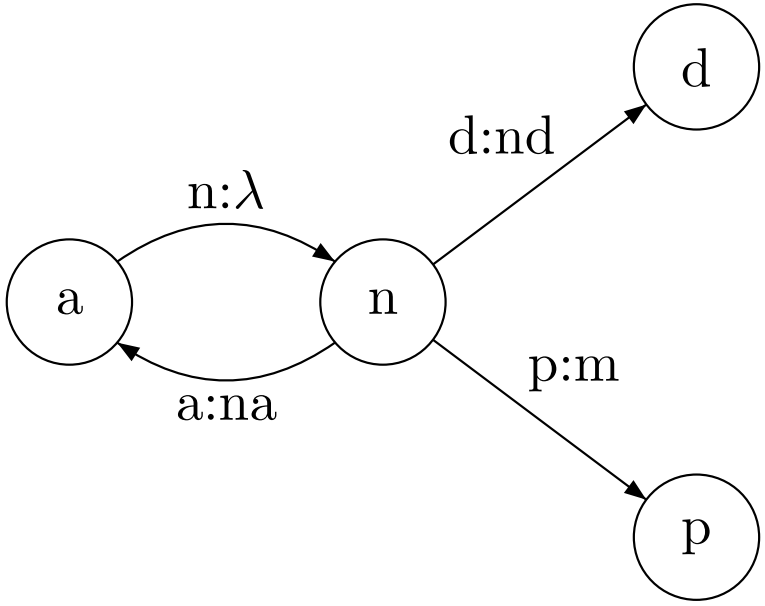
Example illustrating `min_change` with np→m (k=2)

If

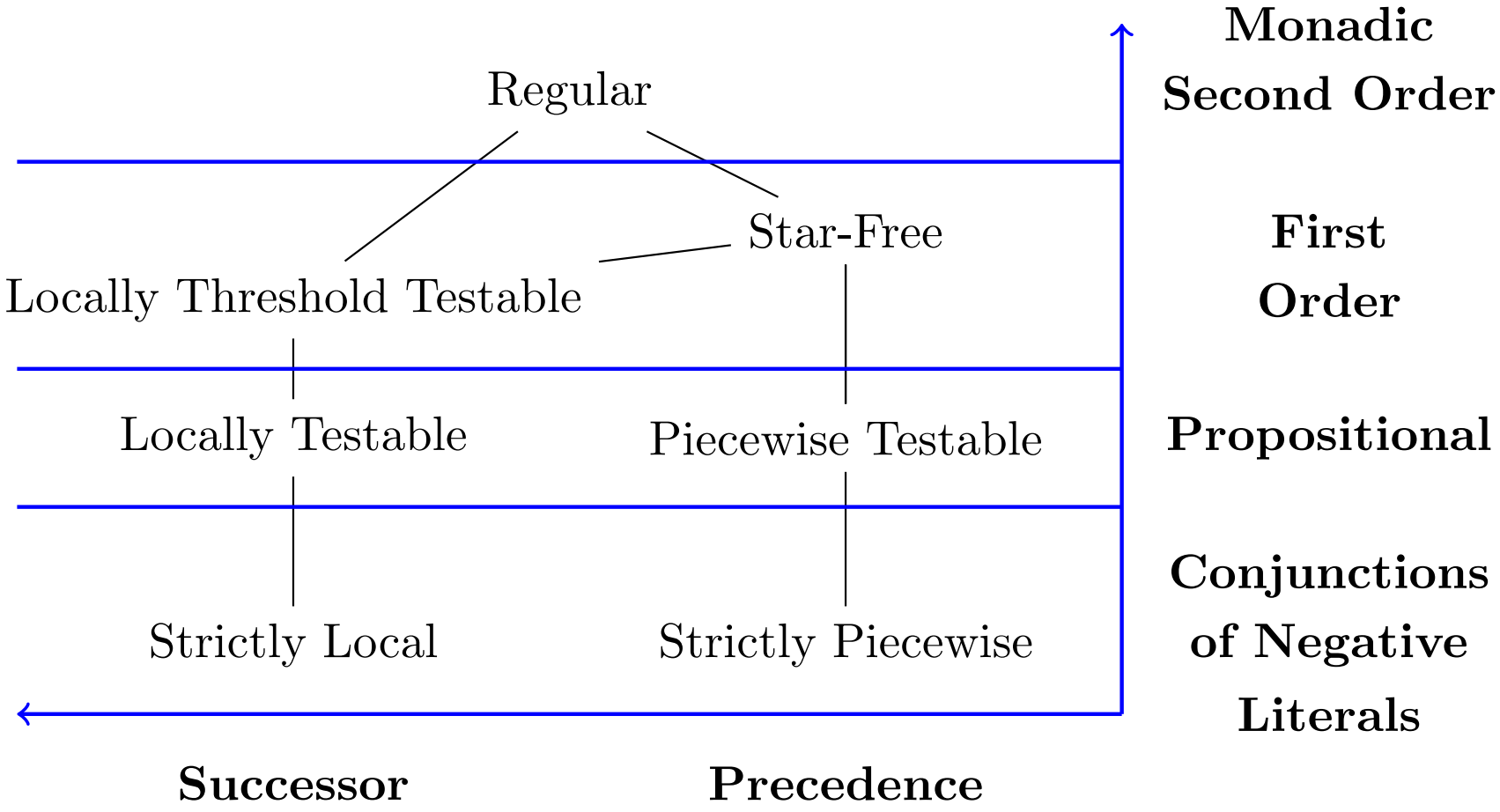
$$S = \left\{ \begin{array}{llll} (anpa & , & ama), & (anpo & , & amo), \\ (ana & , & ana), & (ano & , & ano), \\ (anda & , & anda), & (ando & , & ando) \end{array} \right\}$$

Then

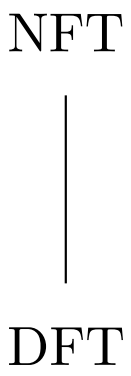
1. $\texttt{common\_out}_S(a) = a$
2. $\texttt{common\_out}_S(an) = a$
3. $\texttt{min\_change}_S(a, n) = \lambda$
4. $\texttt{min\_change}(an, p) = m$
5. $\texttt{min\_change}(an, a) = na$
6. $\texttt{min\_change}(an, d) = nd$



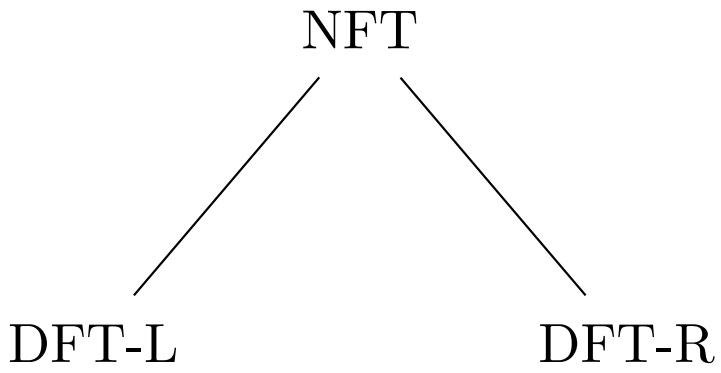
How do these classes generalize to functions and relations?



Subregular Classes of Regular Relations

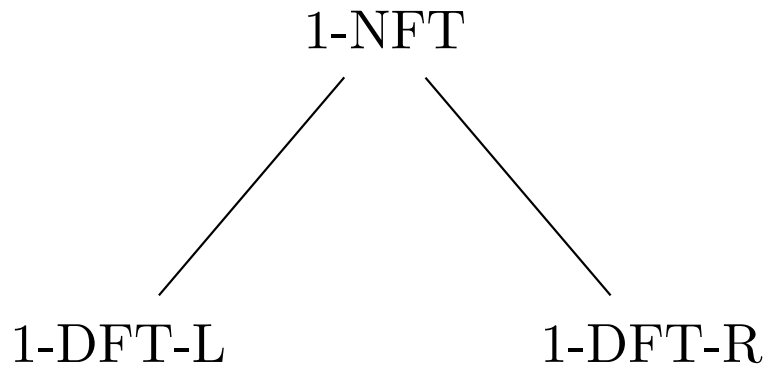


Subregular Classes of Regular Relations



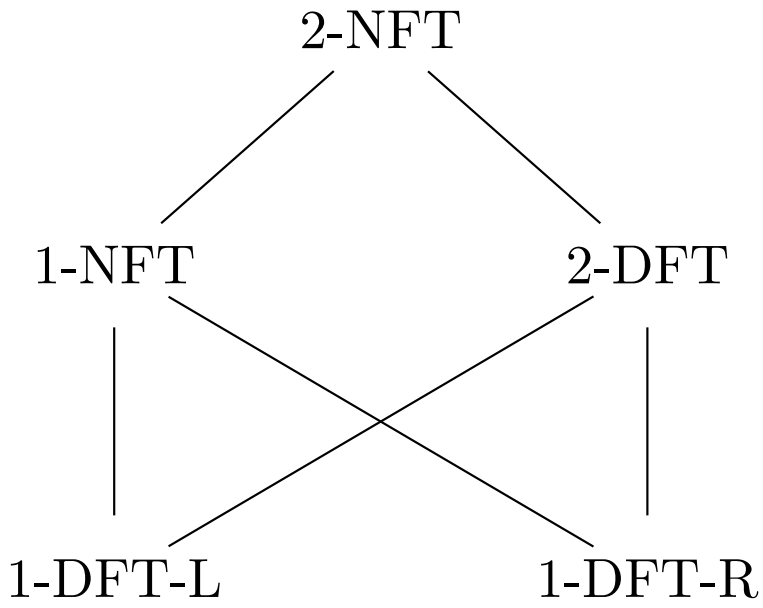
(Elgot and Mezei 1956)

Subregular Classes of Regular Relations



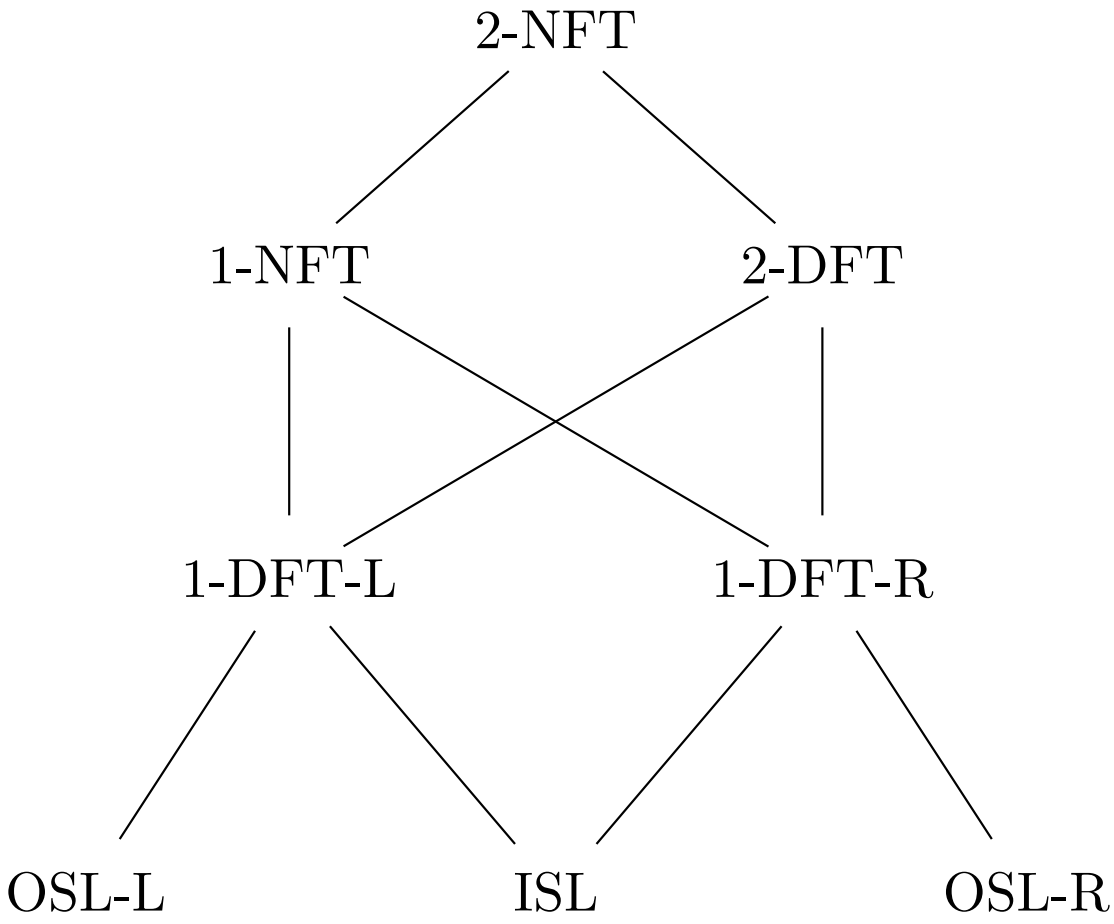
(Elgot and Mezei 1956)

Subregular Classes of Regular Relations

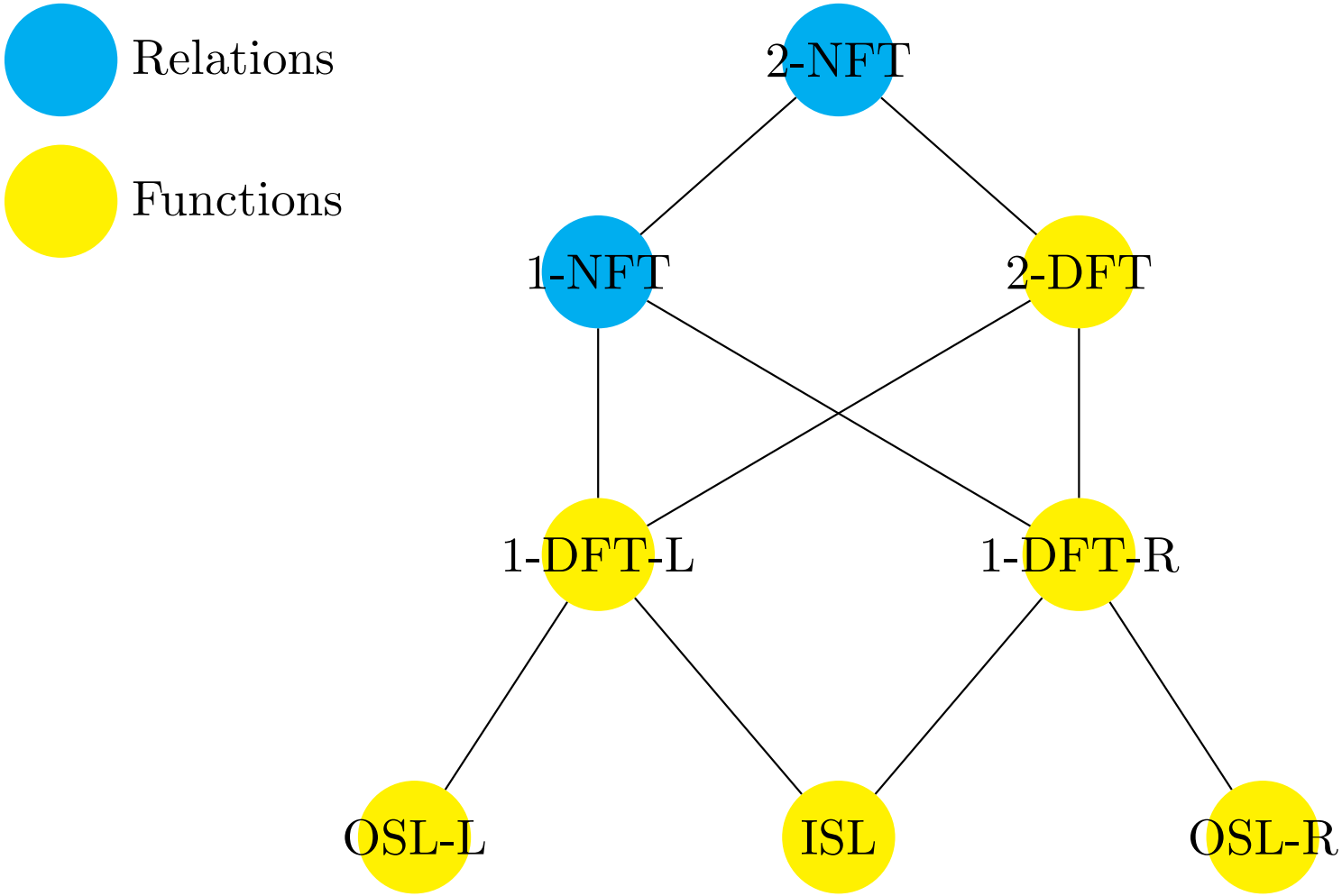


(Elgot and Mezei 1956, Filiot and Reynier 2016)

Subregular Classes of Regular Relations



Subregular Classes of Regular Relations

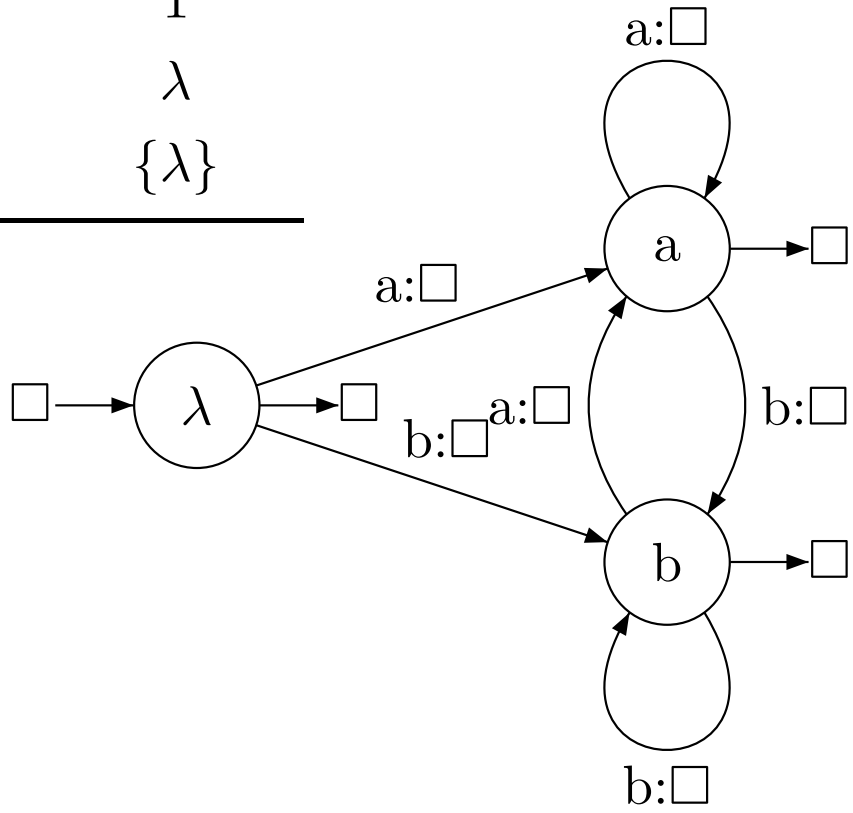


Strictly Local Relations?

	Boolean	[0,1]	Σ^*
Deterministic	DFA	PDFA	DFT
Non-deterministic	NFA	PNFA	NFT

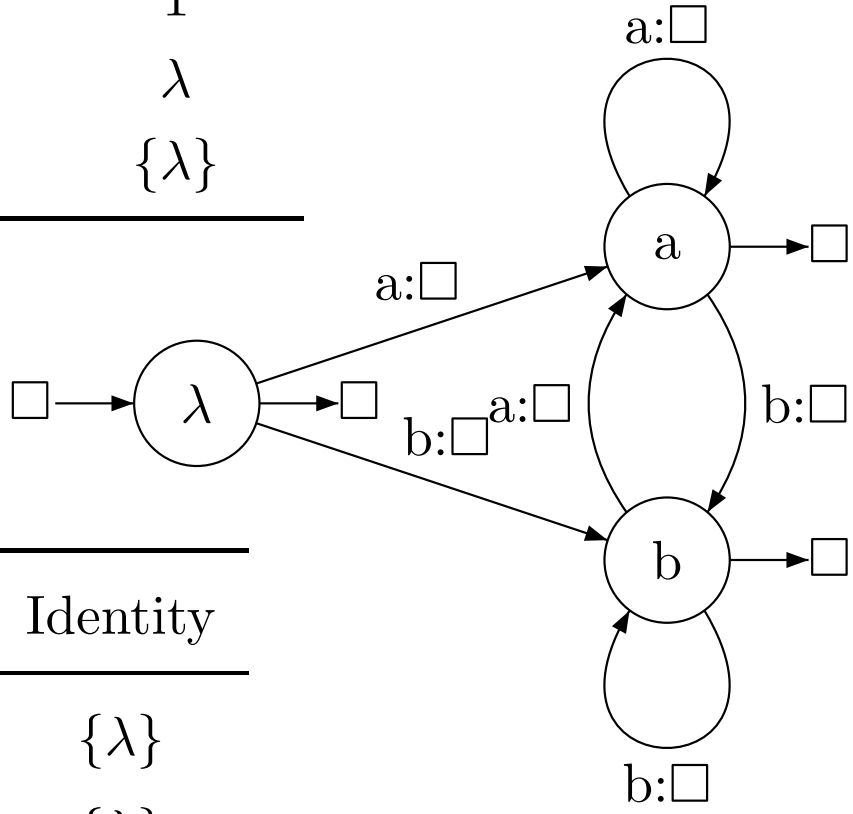
The Weighted Perspective

Monoid	Set	Product	Identity
Boolean	$\{T,F\}$	$\wedge$	T
Real	$[0,1]$	$\times$	1
Strings	Σ^*	$\cdot$	λ
Languages	$\wp(\Sigma^*)$	$\cdot$	$\{\lambda\}$



The Weighted Perspective

Monoid	Set	Product	Identity
Boolean	{T,F}	$\wedge$	T
Real	[0,1]	$\times$	1
Strings	Σ^*	$\cdot$	λ
Languages	$\wp(\Sigma^*)$	$\cdot$	$\{\lambda\}$

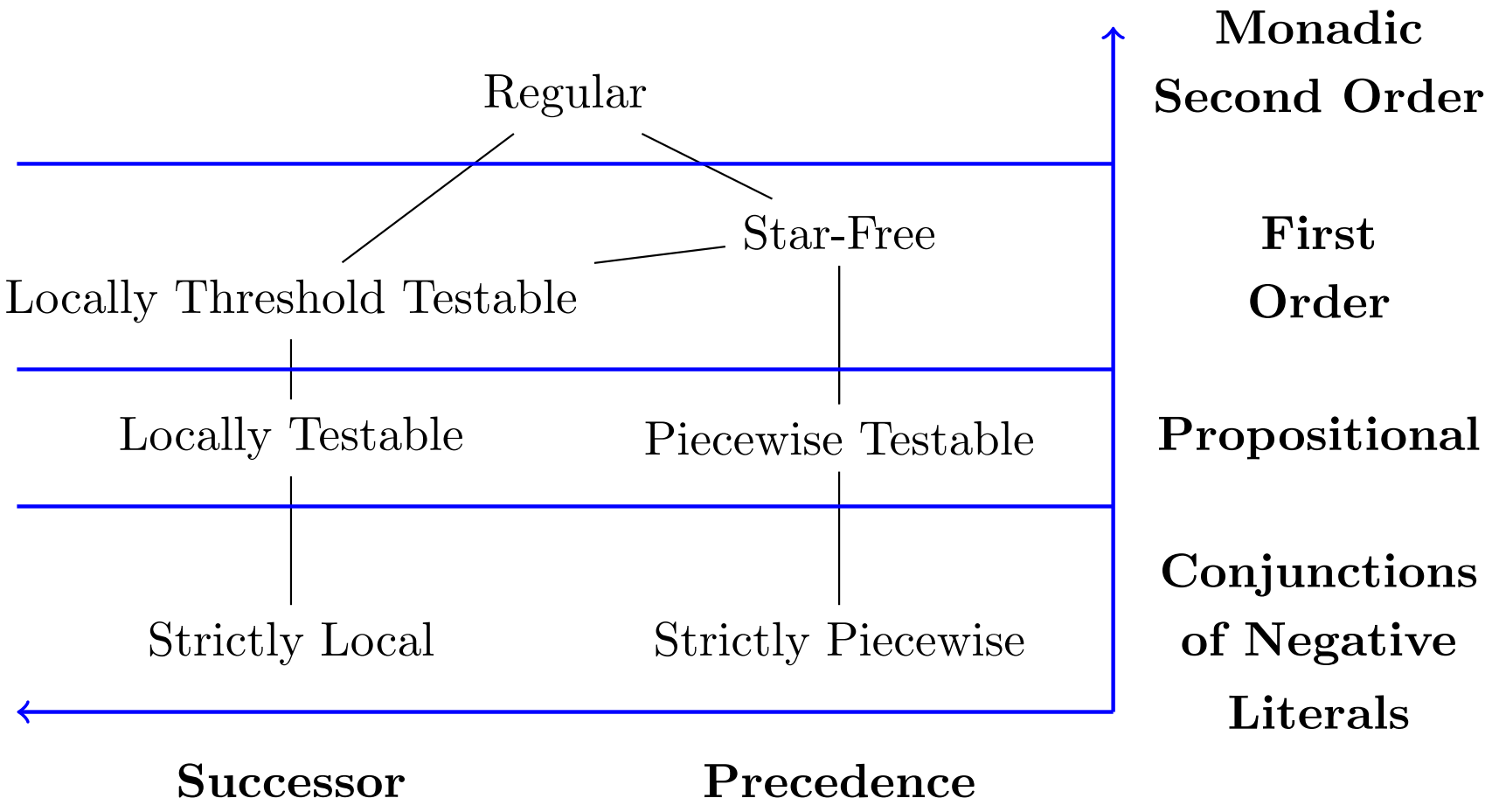


Monoid	Set	Product	Identity
Finite Lgs	FIN	$\cdot$	$\{\lambda\}$
Regular Lgs	DFA	$\cdot$	$\{\lambda\}$

Summary and Conclusion

1. Subregular classes characterize the nature of local and non-local dependencies in patterns.
2. These classes were originally defined to describe sets of strings.
3. They can be generalized to probability distributions over strings and string-to-string transductions.
4. Many of these classes are parameterized which lead to parametric learning models with sound theoretical results.
5. The more general weighted perspective will plausibly lead to parametric learning models for relations.
6. Natural languages are not arbitrary so building in structure to the learning models helps.

Thanks!


Regular
Star-Free
Locally Threshold Testable
Locally Testable
Piecewise Testable
Strictly Local
Strictly Piecewise
Monadic
Second Order
First
Order
Propositional
Conjunctions
of Negative
Literals
Successor
Precedence