Computational Phonology - Class 8

Jeffrey Heinz (Instructor)
Jon Rawski (TA)

LSA Summer Institute
UC Davis
July 18, 2019

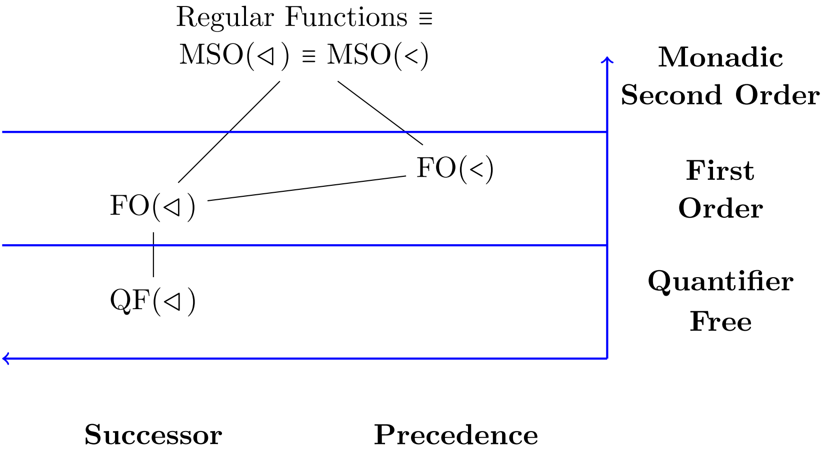
Today

1. Hierarchies of Transformations
2. Computational Theories of Learning
3. Course Summary, Questions and Discussion

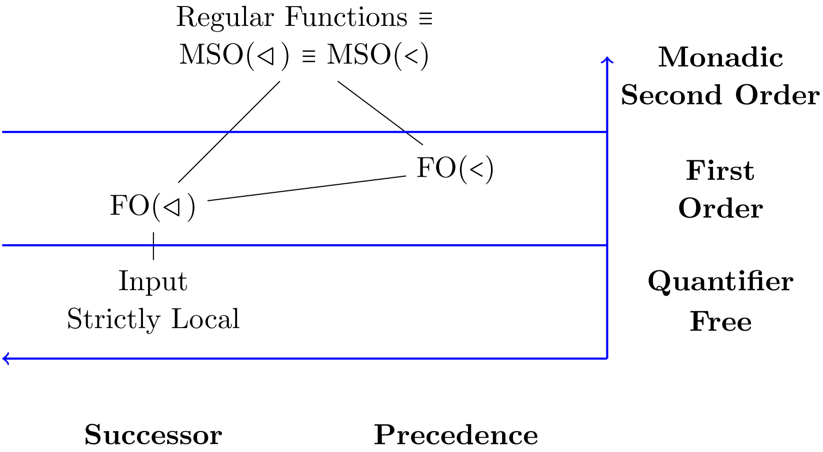
Part I

Hierarchies of Transformations

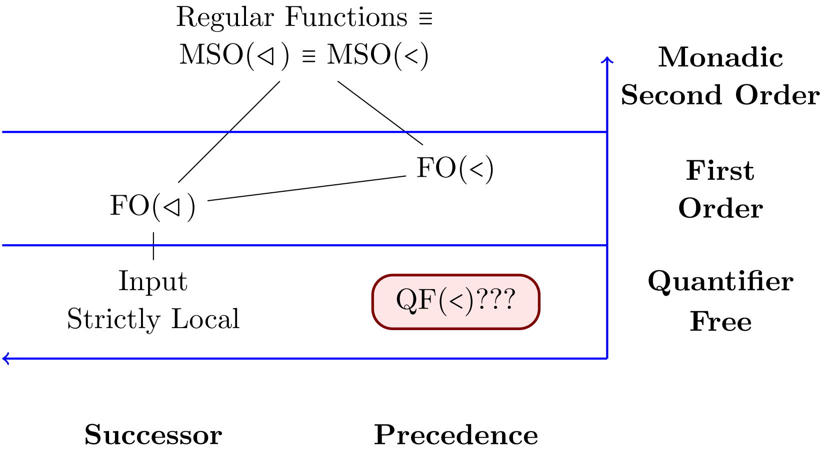
Hierarchies of Transformations



Hierarchies of Transformations



Hierarchies of Transformations



The successor relation is a function.

1. Each element *has at most one* successor.

The successor relation is a function.

1. Each element *has at most one* successor.

$$\lhd \stackrel{\text{def}}{=} \{(1,2),(2,3),(3,4)\}$$

The successor relation is a function.

1. Each element *has at most one* successor.

$$\lhd \stackrel{\text{def}}{=} \{(1,2),(2,3),(3,4)\}$$

2. Therefore, we can use the successor *function* instead of the successor *relation* in the signature of the model.

The successor relation is a function.

1. Each element *has at most one* successor.

$$\triangleleft \overset{\text{def}}{=} \{(1,2),(2,3),(3,4)\}$$

2. Therefore, we can use the successor *function* instead of the successor *relation* in the signature of the model.

The payoff

Instead of

$$C_x C \overset{\text{def}}{=} \texttt{cons}(x) \wedge \exists y[x \triangleleft y \wedge \texttt{cons}(y)]$$

The successor relation is a function.

1. Each element *has at most one* successor.

$$\triangleleft \stackrel{\text{def}}{=} \{(1,2),(2,3),(3,4)\}$$

2. Therefore, we can use the successor *function* instead of the successor *relation* in the signature of the model.

The payoff

Instead of

$$C_xC \stackrel{\text{def}}{=} \texttt{cons}(x) \wedge \exists y[x \triangleleft y \wedge \texttt{cons}(y)]$$

We can write

$$C_xC \stackrel{\text{def}}{=} \texttt{cons}(x) \wedge \texttt{cons}(\texttt{succ}(x))]$$

The successor relation is a function.

1. Each element *has at most one* successor.

$$\triangleleft \stackrel{\text{def}}{=} \{(1,2),(2,3),(3,4)\}$$

2. Therefore, we can use the successor *function* instead of the successor *relation* in the signature of the model.

The payoff

Instead of

$$C_xC \stackrel{\text{def}}{=} \texttt{cons}(x) \wedge \exists y[x \triangleleft\ y \wedge \texttt{cons}(y)]$$

We can write

$$C_xC \stackrel{\text{def}}{=} \texttt{cons}(x) \wedge \texttt{cons}(\texttt{succ}(x))]$$

No quantification needed to introduce elements local to x!

The precedence relation is NOT a function.

1. Each element *may precede more than one* element.

The precedence relation is NOT a function.

1. Each element *may precede more than one* element.

$$\triangleleft \stackrel{\text{def}}{=} \{(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)\}$$

The precedence relation is NOT a function.

1. Each element *may precede more than one* element.

$$\triangleleft \stackrel{\text{def}}{=} \{(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)\}$$

2. Therefore, we cannot use the precedence *function* in the signature of the model – there is literally no such thing!

The precedence relation is NOT a function.

1. Each element *may precede more than one* element.

$$\triangleleft \stackrel{\text{def}}{=} \{(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)\}$$

2. Therefore, we cannot use the precedence *function* in the signature of the model – there is literally no such thing!

Open Questions

- How can the notion of precedence be employed to describe transformations?

The precedence relation is NOT a function.

1. Each element *may precede more than one* element.

$$\lhd \stackrel{\text{def}}{=} \{(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)\}$$

2. Therefore, we cannot use the precedence *function* in the signature of the model – there is literally no such thing!

Open Questions

- How can the notion of precedence be employed to describe transformations?

1. Tiers: Chandlee and McMullin (2018), Burness and McMullin (2019)
2. Least fixed point logics: Chandlee and Jardine (2019)

The precedence relation is NOT a function.

1. Each element *may precede more than one* element.

$$\triangleleft \stackrel{\text{def}}{=} \{(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)\}$$

2. Therefore, we cannot use the precedence *function* in the signature of the model – there is literally no such thing!

Open Questions

- How can the notion of precedence be employed to describe transformations?

1. Tiers: Chandlee and McMullin (2018), Burness and McMullin (2019)
2. Least fixed point logics: Chandlee and Jardine (2019)
3. Plenty of room for new ideas here!

Summary

1. Using model signatures with functions instead of relations is a key element to obtaining Quantifier-Free transformations.
2. Characterizing long-distance transformations is a challenging frontier.
3. The constraint hierarchies have two levels below FO: Prop and CNL. The transformation hierarchies only have one level QF, which appears to correspond to CNL. What fragment of FO in the transformation hierarchy corresponds to the Prop level in the constraint hierarchy?

Part II

Computational Theories of Learning

Formal Learning Theory

1. What does it mean "to learn"?
2. How can we define "learning"?
3. Under the definition, what can be learned and how?

Formal Learning Theory

1. What does it mean "to learn"?
2. How can we define "learning"?
3. Under the definition, what can be learned and how?

Learning requires a structured hypothesis space, which excludes at least some finite-list hypotheses.

Gleitman 1990, p. 12:

> *'The trouble is that an observer who notices* ***everything*** *can learn* ***nothing*** *for there is no end of categories known and constructable to describe a situation [emphasis in original].'*

Theoretical and Empirical Results

On the one hand, a shift in focus from the analysis of properties that define various learnable classes of languages to the behavior of humans is undoubtedly appealing to any who find that the results of learnability theory are too abstract and remote from real-world learning problems.

Heinz and Riggle 2011, p. 67-68

Theoretical and Empirical Results

On the other hand, having observed that an algorithm $\mathcal{A}$ and human subject $\mathcal{H}$ give similar responses for a particular set of test items $\mathcal{T}$ after being exposed to a set of training data $\mathcal{D}$, it is not clear what we can conclude about the relationship between $\mathcal{A}$ and $\mathcal{H}$ because they might wildly diverge for some other data $\mathcal{T}'$ and $\mathcal{D}'$.

Heinz and Riggle 2011, p. 67-68

Theoretical and Empirical Results

The goal of determining which properties of the data critically underlie learnability—or in this case the correlation between $\mathcal{A}$ and $\mathcal{H}$ is precisely why learning theory focuses mainly on the ***properties of classes of languages*** *or the* ***general behavior of specific algorithms****, as opposed to the specific behavior of specific algorithms. [emphasis in original]*

Heinz and Riggle 2011, p. 67-68

Not just words: Weiss et al. 2018

1. Weiss et al. 2018 (ICML) study how well Recurrent Neural Networks (RNNs) learn to recognize acceptable email addresses.

Not just words: Weiss et al. 2018

1. Weiss et al. 2018 (ICML) study how well Recurrent Neural Networks (RNNs) learn to recognize acceptable email addresses.
2. The language of valid email addresses is a regular language, easily expressed with a DFA.

Not just words: Weiss et al. 2018

1. Weiss et al. 2018 (ICML) study how well Recurrent Neural Networks (RNNs) learn to recognize acceptable email addresses.
2. The language of valid email addresses is a regular language, easily expressed with a DFA.
3. One example from their paper: They trained an RNN to 100% accuracy on a 40,000 sample training set and a 2,000 sample test set.

Not just words: Weiss et al. 2018

1. Weiss et al. 2018 (ICML) study how well Recurrent Neural Networks (RNNs) learn to recognize acceptable email addresses.
2. The language of valid email addresses is a regular language, easily expressed with a DFA.
3. One example from their paper: They trained an RNN to 100% accuracy on a 40,000 sample training set and a 2,000 sample test set.
4. They refined a method to extract, from the learned RNN, a DFA approximation of it.

Not just words: Weiss et al. 2018

1. Weiss et al. 2018 (ICML) study how well Recurrent Neural Networks (RNNs) learn to recognize acceptable email addresses.
2. The language of valid email addresses is a regular language, easily expressed with a DFA.
3. One example from their paper: They trained an RNN to 100% accuracy on a 40,000 sample training set and a 2,000 sample test set.
4. They refined a method to extract, from the learned RNN, a DFA approximation of it.
5. Comparing the original and extracted DFA, they could find possible counterexamples.

Not just words: Weiss et al. 2018

1. Weiss et al. 2018 (ICML) study how well Recurrent Neural Networks (RNNs) learn to recognize acceptable email addresses.
2. The language of valid email addresses is a regular language, easily expressed with a DFA.
3. One example from their paper: They trained an RNN to 100% accuracy on a 40,000 sample training set and a 2,000 sample test set.
4. They refined a method to extract, from the learned RNN, a DFA approximation of it.
5. Comparing the original and extracted DFA, they could find possible counterexamples.
6. They find the RNN actually makes very stupid errors! (Cf. Gorman and Sproat 2016)

Not just words: Weiss et al. 2018

Table 4. Counterexamples generated during extraction from an LSTM email network with 100% train and test accuracy. Examples of the network deviating from its target language are shown in bold.

Counter-example	Time (s)	Network Classification	Target Classification
0@m.com	provided	$\surd$	$\surd$
@@y.net	2.93	$\times$	$\times$
25.net	1.60	$\surd$	$\times$
5x.nem	2.34	$\surd$	$\times$
0ch.nom	8.01	$\times$	$\times$
9s.not	3.29	$\times$	$\times$
2hs.net	3.56	$\surd$	$\times$
@cp.net	4.43	$\times$	$\times$

Not just words: Weiss et al. 2018

> *[They] note such cases are "annoyingly frequent: for many RNN-acceptors with 100% train and test accuracy on large test sets, our method was able to find many simple misclassified examples." They state this reveals the "brittleness in generalization" of trained RNNs, and they suggest that evidence based on test-set performance "should be interpreted with extreme caution."*

(Rawski and Heinz, 2019)

Niyogi 2006

Mathematical models with their equations and proofs and computational models with their programs and simulations provide different and important windows of insight into the phenomena at hand.

Niyogi 2006

In the first, one constructs idealized and simplified models but one can now reason precisely about the behavior of such models and therefore be sure of one's conclusions. In the second, one constructs more realistic models but because of the complexity, one will need to resort to heuristic arguments and simulations.

Niyogi 2006

In summary, for mathematical models the assumptions are more questionable but the conclusions are more reliable — for computational models, the assumptions are more believable but the conclusions more suspect.

Formal Learning Theory

1. What does it mean “to learn”?
2. How can we define “learning”?
3. Under the definition, what can be learned and how?

Formal Language Theory

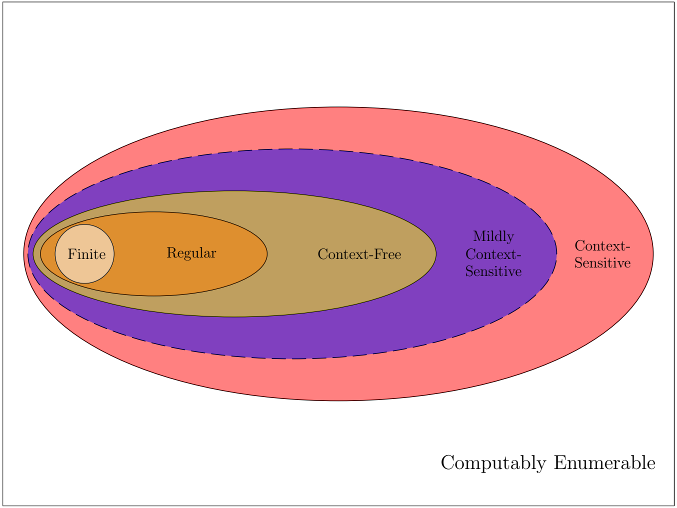


Figure: The Chomsky hierarchy classifies logically possible patterns.

Chomsky 1956, 1959, Harrison 1978

Formal Language Theory

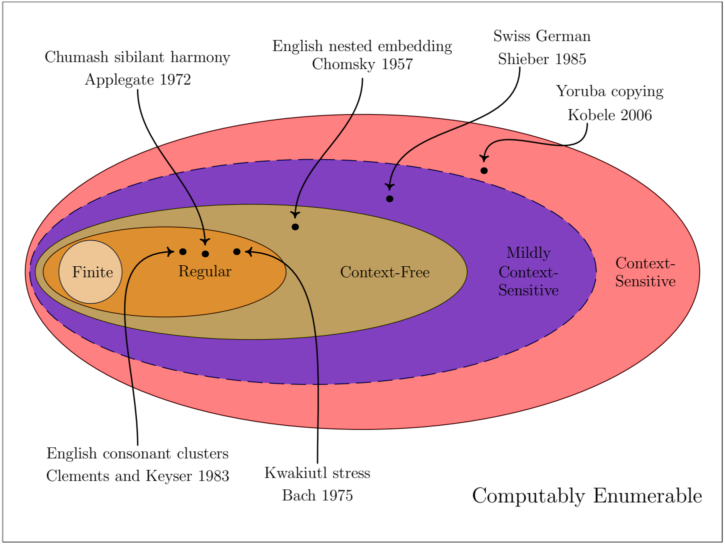


Figure: Natural language patterns in the Chomsky hierarchy.

Formal Language Theory

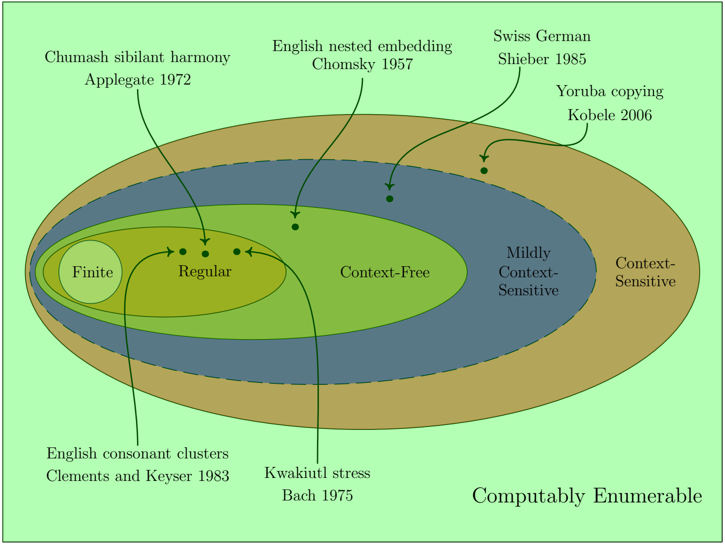


Figure: Possible theories of natural language.

Formal Language Theory

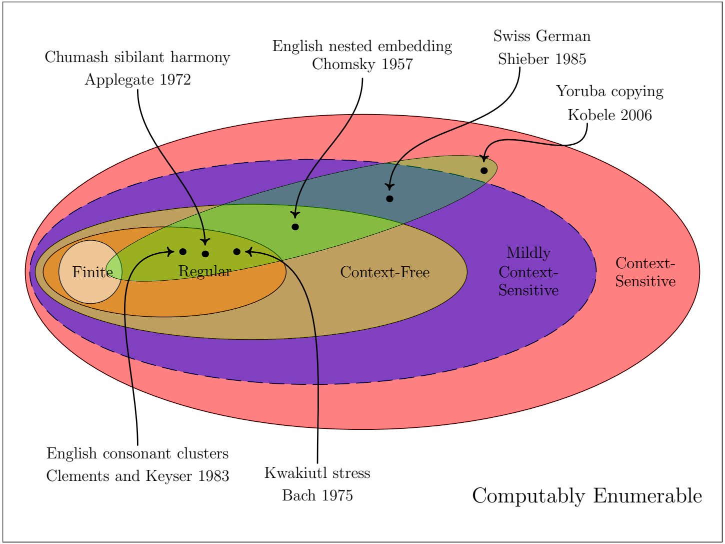


Figure: Possible theories of natural language.

Formal Language Theory

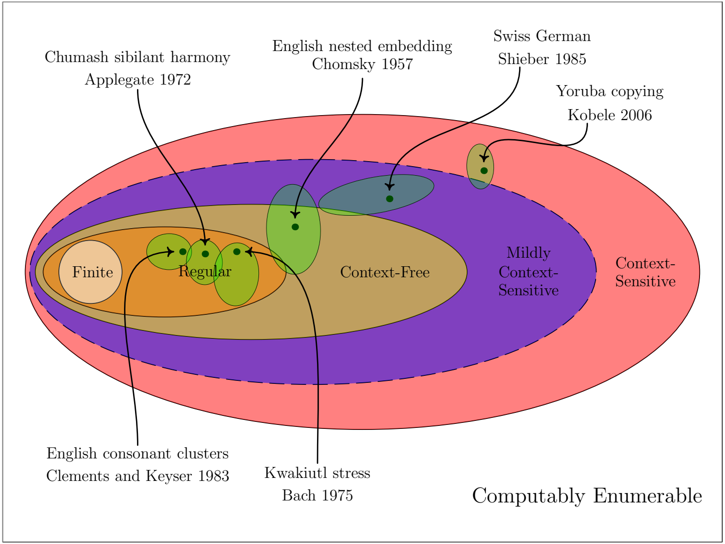
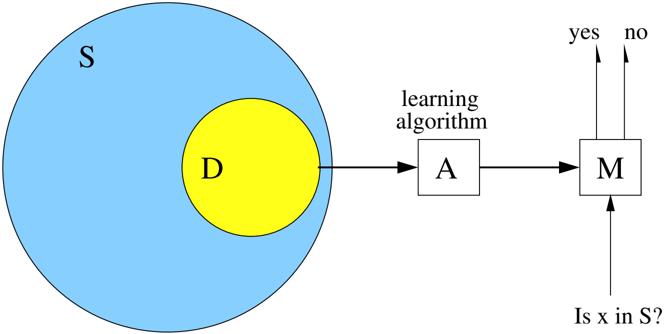


Figure: Possible theories of natural language.

What is Learning?

An abstraction



What counts as success?

$\boxed{\text{We are interested in learners of \textit{classes of languages} and not just a single language.}}$

Why?

What counts as success?

We are interested in learners of *classes of languages* and not just a single language.

Why?

Because every language can be learned by a constant function! Learning algorithms should learn any one of many languages just like humans.

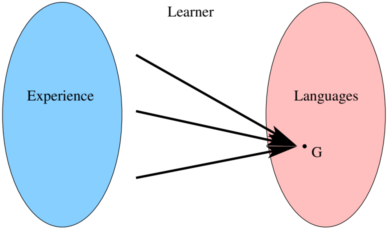


Figure: Learners are functions ϕ from experience to grammars.

Computational Theory: Three Important Questions

Regarding Learning Algorithm A

1. Does it exist?
2. Is it computable?
3. Is it feasibly computable?

Formal Learning Theories

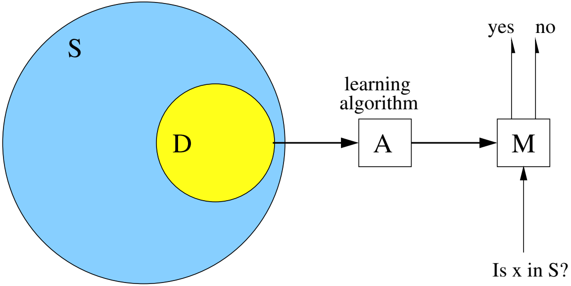


Figure: Learners are functions ϕ from experience to grammars.

(Gold 1967, Horning 1969, Angluin 1980, Osherson et al. 1984, Angluin 1988, Anthony and Biggs 1991, Kearns and Vazirani 1994, Vapnik 1994, 1998, Jain et al. 1999, Niyogi 2006, de la Higuera 2010, Mohri et al. 2012)

The Experience

1. It is a sequence.
2. It is finite.

$$w_0$$
$$w_1$$
$$w_2 \qquad\qquad \downarrow \text{time}$$
$$\ldots$$
$$w_n$$

Types of Experience

Positive evidence

$$\begin{array}{c} w_0 \in L \\ w_1 \in L \\ w_2 \in L \\ \ldots \\ w_n \in L \end{array} \qquad \downarrow \text{time}$$

Types of Experience

Positive and negative evidence

$$\begin{array}{c} w_0 \in L \\ w_1 \notin L \\ w_2 \notin L \\ \dots \\ w_n \in L \end{array} \qquad \Big\downarrow \text{time}$$

Types of Experience

Noisy evidence

$$\begin{gathered} w_0 \in L \\ w_1 \notin L \\ w_2 \in L\ (\textbf{but in fact } w_2 \notin L) \\ \ldots \\ w_n \in L \end{gathered}$$

$\downarrow$ time

Types of Experience

Queried Evidence

$$w_0 \in L$$
$$w_1 \notin L$$
$w_2 \in L$ (because learner
specifically asked about w_2)
$$\dots$$
$$w_n \in L$$

$\downarrow$ time

The Languages

1. They can be sets of words or distributions over words.
2. They are computable.

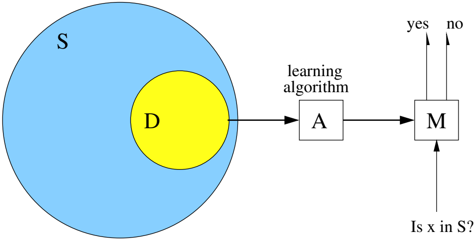


Figure: Learners are functions from experience to grammars.

Learning Criteria

1. What does it mean to learn a language?
2. What kind of experience is required for success?
3. What counts as success?

What does it mean to learn a language?

1. Convergence.
2. Imagine an infinite sequence. Is there some point n after which the learner's hypothesis doesn't change (much)?

datum	Learner's Hypothesis
w_0	$\phi(\langle w_0 \rangle) = G_0$

$\downarrow$ time

What does it mean to learn a language?

1. Convergence.
2. Imagine an infinite sequence. Is there some point n after which the learner's hypothesis doesn't change (much)?

datum	Learner's Hypothesis
w_0	$\phi(\langle w_0 \rangle) = G_0$
w_1	$\phi(\langle w_0, w_1 \rangle) = G_1$

$\downarrow$ time

What does it mean to learn a language?

1. Convergence.
2. Imagine an infinite sequence. Is there some point n after which the learner's hypothesis doesn't change (much)?

datum	Learner's Hypothesis
w_0	$\phi(\langle w_0 \rangle) = G_0$
w_1	$\phi(\langle w_0, w_1 \rangle) = G_1$
w_2	$\phi(\langle w_0, w_1, w_2 \rangle) = G_2$

$\downarrow$ time

What does it mean to learn a language?

1. Convergence.
2. Imagine an infinite sequence. Is there some point n after which the learner's hypothesis doesn't change (much)?

datum	Learner's Hypothesis
w_0	$\phi(\langle w_0 \rangle) = G_0$
w_1	$\phi(\langle w_0, w_1 \rangle) = G_1$
w_2	$\phi(\langle w_0, w_1, w_2 \rangle) = G_2$
$\ldots$	

$\downarrow$ time

What does it mean to learn a language?

1. Convergence.
2. Imagine an infinite sequence. Is there some point n after which the learner's hypothesis doesn't change (much)?

datum	Learner's Hypothesis
w_0	$\phi(\langle w_0 \rangle) = G_0$
w_1	$\phi(\langle w_0, w_1 \rangle) = G_1$
w_2	$\phi(\langle w_0, w_1, w_2 \rangle) = G_2$
$\dots$	
w_n	$\phi(\langle w_0, w_1, w_2, \dots, w_n \rangle) = G_n$

$\downarrow$ time

What does it mean to learn a language?

1. Convergence.
2. Imagine an infinite sequence. Is there some point n after which the learner's hypothesis doesn't change (much)?

datum	Learner's Hypothesis
w_0	$\phi(\langle w_0 \rangle) = G_0$
w_1	$\phi(\langle w_0, w_1 \rangle) = G_1$
w_2	$\phi(\langle w_0, w_1, w_2 \rangle) = G_2$
$\ldots$	
w_n	$\phi(\langle w_0, w_1, w_2, \ldots, w_n \rangle) = G_n$
$\ldots$	

$\downarrow$ time

What does it mean to learn a language?

1. Convergence.
2. Imagine an infinite sequence. Is there some point n after which the learner's hypothesis doesn't change (much)?

datum	Learner's Hypothesis
w_0	$\phi(\langle w_0 \rangle) = G_0$
w_1	$\phi(\langle w_0, w_1 \rangle) = G_1$
w_2	$\phi(\langle w_0, w_1, w_2 \rangle) = G_2$
$\ldots$	
w_n	$\phi(\langle w_0, w_1, w_2, \ldots, w_n \rangle) = G_n$
$\ldots$	
w_m	$\phi(\langle w_0, w_1, w_2, \ldots, w_m \rangle) = G_m$

$\downarrow$ time

Does $G_m \simeq G_n$?

What kind of experience is required for success?

Types of Experience

1. Positive-only or positive and negative evidence.
2. Noiseless or noisy evidence.
3. Queries allowed or not?

Which infinite sequences require convergence?

1. only **complete** ones? I.e. where every piece of information occurs at some finite point
2. only **computable** ones? I.e. the infinite sequence itself is describable by some grammar

What kind of experience is required for success?

Makes learning easier	Makes learning harder
positive and negative evidence	positive evidence only
noiseless evidence	noisy evidence
queries permitted	queries not permitted
approximate convergence	exact convergence
complete infinite sequences	any infinite sequence
computable infinite sequences	any infinite sequence

What kind of experience is required for success?

Makes learning easier	Makes learning harder
positive and negative evidence	positive evidence only
noiseless evidence	noisy evidence
queries permitted	queries not permitted
approximate convergence	exact convergence
complete infinite sequences	any infinite sequence
computable infinite sequences	any infinite sequence

1. Identification in the limit from positive data (Gold 1967)

What kind of experience is required for success?

Makes learning easier	Makes learning harder
positive and negative evidence	positive evidence only
noiseless evidence	noisy evidence
queries permitted	queries not permitted
approximate convergence	exact convergence
complete infinite sequences	any infinite sequence
computable infinite sequences	any infinite sequence

2. Identification in the limit from positive and negative data (Gold 1967)

What kind of experience is required for success?

Makes learning easier	Makes learning harder
positive and negative evidence	positive evidence only
noiseless evidence	noisy evidence
queries permitted	queries not permitted
approximate convergence	exact convergence
complete infinite sequences	any infinite sequence
computable infinite sequences	any infinite sequence

3. Identification in the limit from positive data from c.e. texts (Gold 1967)
4. Learning context-free and c.e. distributions (Horning 1969, Angluin 1988)

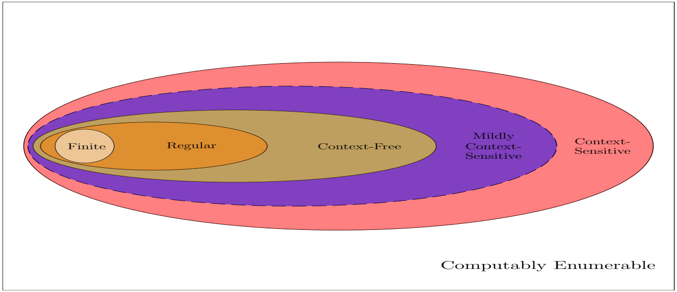
What kind of experience is required for success?

Makes learning easier	Makes learning harder
positive and negative evidence	positive evidence only
noiseless evidence	noisy evidence
queries permitted	queries not permitted
approximate convergence	exact convergence
complete infinite sequences	any infinite sequence
computable infinite sequences	any infinite sequence

5. Probably Approximately Correct learning
   (Valiant 1984, Anthony and Biggs 1991, Kearns and Vazirani 1994

Results of Formal Learning Theories: Existence

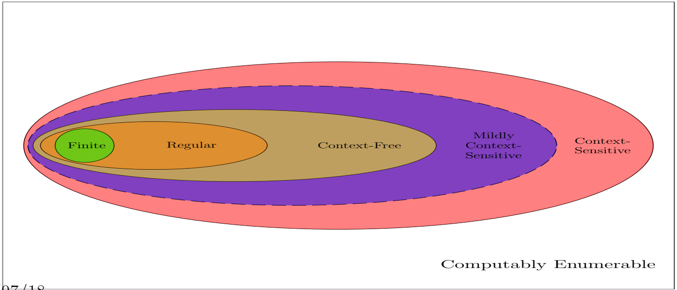
Makes learning easier	Makes learning harder
positive and negative evidence	positive evidence only
noiseless evidence	noisy evidence
queries permitted	queries not permitted
approximate convergence	exact convergence
complete infinite sequences	any infinite sequence
computable infinite sequences	any infinite sequence



Results of Formal Learning Theories: Existence

Makes learning easier	Makes learning harder
positive and negative evidence	positive evidence only
noiseless evidence	noisy evidence
queries permitted	queries not permitted
approximate convergence	exact convergence
complete infinite sequences	any infinite sequence
computable infinite sequences	any infinite sequence

1. Identification in the limit from positive data (Gold 1967)

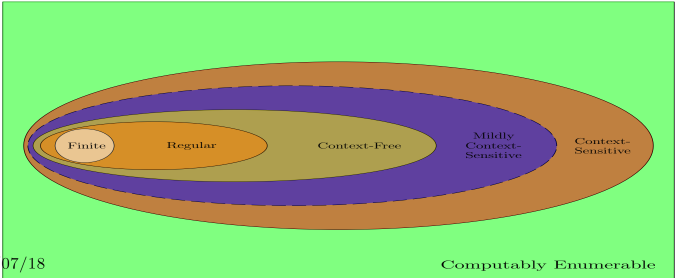


Results of Formal Learning Theories: Existence

Makes learning easier	Makes learning harder
positive and negative evidence	positive evidence only
noiseless evidence	noisy evidence
queries permitted	queries not permitted
approximate convergence	exact convergence
complete infinite sequences	any infinite sequence
computable infinite sequences	any infinite sequence

2. Identification in the limit from positive and negative data

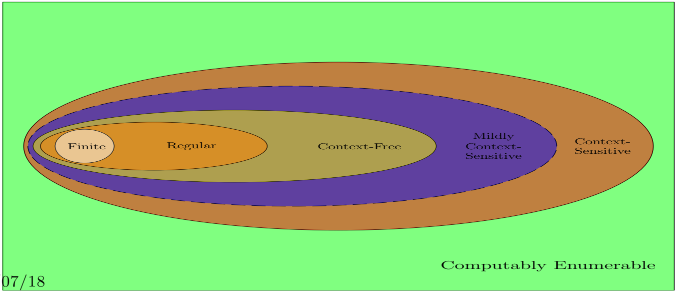
(Gold 1967)



Results of Formal Learning Theories: Existence

Makes learning easier	Makes learning harder
positive and negative evidence	positive evidence only
noiseless evidence	noisy evidence
queries permitted	queries not permitted
approximate convergence	exact convergence
complete infinite sequences	any infinite sequence
computable infinite sequences	any infinite sequence

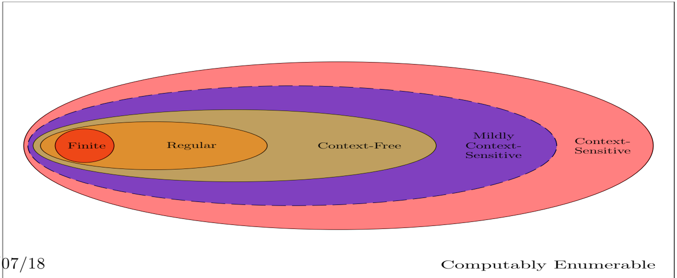
3. Identification in the limit from positive data from c.e. texts (Gold 1967)
4. Learning context-free and c.e. distributions (Horning 1969, Angluin 1988)



Results of Formal Learning Theories: Existence

Makes learning easier	Makes learning harder
positive and negative evidence	positive evidence only
noiseless evidence	noisy evidence
queries permitted	queries not permitted
approximate convergence	exact convergence
complete infinite sequences	any infinite sequence
computable infinite sequences	any infinite sequence

5. Probably Approximately Correct learning
   (Valiant 1984, Anthony and Biggs 1991, Kearns and Vazirani 1994



Results of Formal Learning Theory: Feasibility

Makes learning easier	Makes learning harder
positive and negative evidence	positive evidence only
noiseless evidence	noisy evidence
queries permitted	queries not permitted
approximate convergence	exact convergence
complete infinite sequences	any infinite sequence
computable infinite sequences	any infinite sequence

Results of Formal Learning Theory: Feasibility

Makes learning easier	Makes learning harder
positive and negative evidence	positive evidence only
noiseless evidence	noisy evidence
queries permitted	queries not permitted
approximate convergence	exact convergence
complete infinite sequences	any infinite sequence
computable infinite sequences	any infinite sequence

1. Identification in the limit from positive data (Gold 1967)

   No superfinite class is learnable.
   The finite class is feasibly learnable.

Results of Formal Learning Theory: Feasibility

Makes learning easier	Makes learning harder
positive and negative evidence	positive evidence only
noiseless evidence	noisy evidence
queries permitted	queries not permitted
approximate convergence	exact convergence
complete infinite sequences	any infinite sequence
computable infinite sequences	any infinite sequence

2. Identification in the limit from positive and negative data (Gold 1967, 1978)

   The c.e. class is learnable but finding the minimal DFA consistent with any data sample is NOT feasible.

   *Oncina and Garcia 1992 solve a related, but different problem in cubic time (RPNI).

Results of Formal Learning Theory: Feasibility

Makes learning easier	Makes learning harder
positive and negative evidence	positive evidence only
noiseless evidence	noisy evidence
queries permitted	queries not permitted
approximate convergence	exact convergence
complete infinite sequences	any infinite sequence
computable infinite sequences	any infinite sequence

3. Identification in the limit from positive data from c.e. texts (Gold 1967)

4. Learning context-free and c.e. distributions (Horning 1969, Angluin 1988)

   The c.e. class of languages and distributions is learnable but NOT even the class of PNFAs is feasibly learnable.

   *Clark and Thollard (2004) solve a related, but different learning problem for PDFAs in polynomial time.

Results of Formal Learning Theory: Feasibility

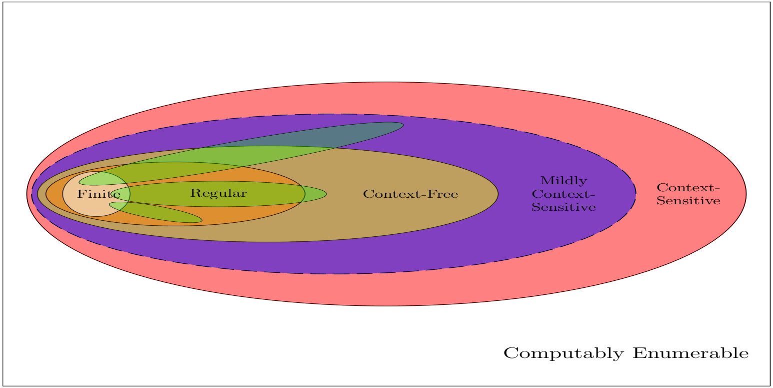
Makes learning easier	Makes learning harder
positive and negative evidence	positive evidence only
noiseless evidence	noisy evidence
queries permitted	queries not permitted
approximate convergence	exact convergence
complete infinite sequences	any infinite sequence
computable infinite sequences	any infinite sequence

5. Probably Approximately Correct learning (Valiant 1984, Anthony and Biggs 1991, Kearns and Vazirani 1994)

   Not even the finite class of languages is learnable.

Formal Learning Theory: Positive Results

Many classes which cross-cut the Chomsky hierarchy and exclude some finite languages are feasibly learnable in the senses discussed.



(Angluin 1980, 1982, Garcia et al. 1990, Muggleton 1990, Denis et al. 2002, Fernau 2003, Yokomori 2003, Oates et al. 2006, Niyogi 2006, Clark and Eryaud 2007, Heinz 2008, 2010, Yoshinaka 2008, Case et al. 2009, de la Higuera 2010, Clark and Lappin 2011, Heinz et al. 2015, Heinz and Sempere 2016)

Summary

1. The larger and less structured the class, the more data and time are needed to distiguish the target from other members of the class, irregardless of statistics.
2. On the other hand, structured, restricted hypothesis spaces can be feasibly learned.
3. The positive learning results are proven results, and the proofs are often constructive.
4. The claim that "statistical learning" is more powerful than "symbolic learning" mischaracterizes the learning issues.
5. The issue is whether or not success ought to be defined only with respect to data sequences generated by computable means.

Putting it all together

1. I am not claiming the following learners are the full story.
2. I am claiming that they are good approximations to the full story and that the full story will incorporate their key elements.
3. The role of phonological features, similarity, sonority, etc. is ongoing and will refine the present proposals.

Local sound patterns

Distinctions are made on the basis of contiguous subsequences.

possible English words	impossible English words
thole	**pt**ak
plast	**hl**ad
flitch	**sr**am
	mgla
	vlas
	dnom
	rtut

Local sound patterns and formal language theory

1. The formal languages which make distinctions on the basis of k-long contiguous subsequences are called Strictly k-Local (McNaughton and Papert 1971, Rogers and Pullum 2007)
2. They are subregular and exclude some finite languages.
3. If every k-long contiguous subsequence is licensed by the grammar, the word belongs to the language.

Local sound patterns and formal language theory

1. The formal languages which make distinctions on the basis of k-long contiguous subsequences are called Strictly k-Local (McNaughton and Papert 1971, Rogers and Pullum 2007)
2. They are subregular and exclude some finite languages.
3. If every k-long contiguous subsequence is licensed by the grammar, the word belongs to the language.

stip

ptip

Local sound patterns and formal language theory

1. The formal languages which make distinctions on the basis of k-long contiguous subsequences are called Strictly k-Local (McNaughton and Papert 1971, Rogers and Pullum 2007)
2. They are subregular and exclude some finite languages.
3. If every k-long contiguous subsequence is licensed by the grammar, the word belongs to the language.

stip

ptip

Local sound patterns and formal language theory

1. The formal languages which make distinctions on the basis of k-long contiguous subsequences are called Strictly k-Local (McNaughton and Papert 1971, Rogers and Pullum 2007)
2. They are subregular and exclude some finite languages.
3. If every k-long contiguous subsequence is licensed by the grammar, the word belongs to the language.

s**ti**p

ptip

Local sound patterns and formal language theory

1. The formal languages which make distinctions on the basis of k-long contiguous subsequences are called Strictly k-Local (McNaughton and Papert 1971, Rogers and Pullum 2007)
2. They are subregular and exclude some finite languages.
3. If every k-long contiguous subsequence is licensed by the grammar, the word belongs to the language.

st**ip**

ptip

Local sound patterns and formal language theory

1. The formal languages which make distinctions on the basis of k-long contiguous subsequences are called Strictly k-Local (McNaughton and Papert 1971, Rogers and Pullum 2007)
2. They are subregular and exclude some finite languages.
3. If every k-long contiguous subsequence is licensed by the grammar, the word belongs to the language.

stip ✓

ptip

Local sound patterns and formal language theory

1. The formal languages which make distinctions on the basis of k-long contiguous subsequences are called Strictly k-Local (McNaughton and Papert 1971, Rogers and Pullum 2007)
2. They are subregular and exclude some finite languages.
3. If every k-long contiguous subsequence is licensed by the grammar, the word belongs to the language.

stip ✓

ptip

Local sound patterns and formal language theory

1. The formal languages which make distinctions on the basis of k-long contiguous subsequences are called Strictly k-Local (McNaughton and Papert 1971, Rogers and Pullum 2007)
2. They are subregular and exclude some finite languages.
3. If every k-long contiguous subsequence is licensed by the grammar, the word belongs to the language.

stip ✓

ptip ×

Learning local sound patterns

1. Strictly k-Local languages are identifiable in the limit from positive data (Garcia et al. 1990).
2. Strictly k-Local distributions can be efficiently estimated (Jurafsky & Martin 2008) (they are n-gram models)
3. **Keep track of the observed k-long contiguous subsequences.**

i	$t(i)$	$SL_2(t(i))$	Grammar G	$L(G)$
-1			$\varnothing$	$\varnothing$
0	$aaaa$	$\{aa\}$	$\{\mathbf{aa}\}$	aaa^*
1	aab	$\{aa, ab\}$	$\{aa, \mathbf{ab}\}$	$aaa^* \cup aaa^*b$
2	ba	$\{ba\}$	$\{aa, ab, \mathbf{ba}\}$	$\Sigma^*/\Sigma^*bb\Sigma^*$
...				

The Strictly 2-Local learner learns *bb

Long-distance sound patterns

Distinctions are made on the basis of potentially discontiguous subsequences.

possible Chumash words	impossible Chumash words
shtoyonowonowash	**s**toyonowonowa**sh**
stoyonowonowas	**sh**toyonowonowa**s**
pisotonosikiwat	pi**s**otono**sh**ikiwat

Long-distance sound patterns and formal language theory

1. The formal languages which make distinctions on the basis of k-long contiguous subsequences are called Strictly k-Piecewise (Heinz 2007, Rogers et al. 2010, Heinz 2010).
2. They are subregular and exclude some finite languages.
3. If every k-long subsequence is licensed by the grammar, the word belongs to the language.

Long-distance sound patterns and formal language theory

1. The formal languages which make distinctions on the basis of k-long contiguous subsequences are called Strictly k-Piecewise (Heinz 2007, Rogers et al. 2010, Heinz 2010).
2. They are subregular and exclude some finite languages.
3. If every k-long subsequence is licensed by the grammar, the word belongs to the language.

sotos

sotosh

Long-distance sound patterns and formal language theory

1. The formal languages which make distinctions on the basis of k-long contiguous subsequences are called Strictly k-Piecewise (Heinz 2007, Rogers et al. 2010, Heinz 2010).
2. They are subregular and exclude some finite languages.
3. If every k-long subsequence is licensed by the grammar, the word belongs to the language.

sotos

sotosh

Long-distance sound patterns and formal language theory

1. The formal languages which make distinctions on the basis of k-long contiguous subsequences are called Strictly k-Piecewise (Heinz 2007, Rogers et al. 2010, Heinz 2010).
2. They are subregular and exclude some finite languages.
3. If every k-long subsequence is licensed by the grammar, the word belongs to the language.

so**t**os

sotosh

Long-distance sound patterns and formal language theory

1. The formal languages which make distinctions on the basis of k-long contiguous subsequences are called Strictly k-Piecewise (Heinz 2007, Rogers et al. 2010, Heinz 2010).
2. They are subregular and exclude some finite languages.
3. If every k-long subsequence is licensed by the grammar, the word belongs to the language.

sot**o**s

sotosh

Long-distance sound patterns and formal language theory

1. The formal languages which make distinctions on the basis of k-long contiguous subsequences are called Strictly k-Piecewise (Heinz 2007, Rogers et al. 2010, Heinz 2010).
2. They are subregular and exclude some finite languages.
3. If every k-long subsequence is licensed by the grammar, the word belongs to the language.

soto**s**

sotosh

Long-distance sound patterns and formal language theory

1. The formal languages which make distinctions on the basis of k-long contiguous subsequences are called Strictly k-Piecewise (Heinz 2007, Rogers et al. 2010, Heinz 2010).
2. They are subregular and exclude some finite languages.
3. If every k-long subsequence is licensed by the grammar, the word belongs to the language.

s**ot**os

sotosh

Long-distance sound patterns and formal language theory

1. The formal languages which make distinctions on the basis of k-long contiguous subsequences are called Strictly k-Piecewise (Heinz 2007, Rogers et al. 2010, Heinz 2010).
2. They are subregular and exclude some finite languages.
3. If every k-long subsequence is licensed by the grammar, the word belongs to the language.

sotos

sotosh

Long-distance sound patterns and formal language theory

1. The formal languages which make distinctions on the basis of k-long contiguous subsequences are called Strictly k-Piecewise (Heinz 2007, Rogers et al. 2010, Heinz 2010).
2. They are subregular and exclude some finite languages.
3. If every k-long subsequence is licensed by the grammar, the word belongs to the language.

sotos ✓

sotosh

Long-distance sound patterns and formal language theory

1. The formal languages which make distinctions on the basis of k-long contiguous subsequences are called Strictly k-Piecewise (Heinz 2007, Rogers et al. 2010, Heinz 2010).
2. They are subregular and exclude some finite languages.
3. If every k-long subsequence is licensed by the grammar, the word belongs to the language.

sotos ✓

soto**sh**

Long-distance sound patterns and formal language theory

1. The formal languages which make distinctions on the basis of k-long contiguous subsequences are called Strictly k-Piecewise (Heinz 2007, Rogers et al. 2010, Heinz 2010).
2. They are subregular and exclude some finite languages.
3. If every k-long subsequence is licensed by the grammar, the word belongs to the language.

sotos ✓

sotosh ×

Learning long-distance sound patterns

1. Strictly k-Piecewise languages are identifiable in the limit from positive data (Heinz 2007, 2010).
2. Strictly k-Piecewise distributions can be efficiently estimated (Heinz & Rogers 2013, Shibata and Heinz 2019)
3. **Keep track of the observed k-long subsequences.**

i	$t(i)$	$SP_2(t(i))$	Grammar G	Language of G
-1			$\varnothing$	$\varnothing$
0	$aaaa$	$\{\lambda, a, aa\}$	{**λ, a, aa**}	a^*
1	aab	$\{\lambda, a, b, aa, ab\}$	{λ, a, aa, **b, ab**}	$a^* \cup a^*b$
2	baa	$\{\lambda, a, b, aa, ba\}$	{λ, a, b, aa, ab, **ba**}	$\Sigma^* \backslash (\Sigma^* b \Sigma^* b \Sigma^*)$
3	aba	$\{\lambda, a, b, ab, ba\}$	{λ, a, b, aa, ab, ba}	$\Sigma^* \backslash (\Sigma^* b \Sigma^* b \Sigma^*)$
…				

The learner ϕ_{SP_2} learns *b…b

Further comments

1. Like the regions in the Chomsky hierarchy, the Strictly Local and Strictly Piecewise classes have multiple, independent, converging characterizations from formal language theory, automata theory, and logic.
2. They are incomparable.
3. Consequently, Strictly Local learners cannot learn Strictly Piecewise patterns and vice versa.
4. Strictly Piecewise learners cannot learn:
    - blocking patterns, e.g. *s…sh unless [z] intervenes.
    - harmony patterns which apply only to the first and last sounds.

Summary

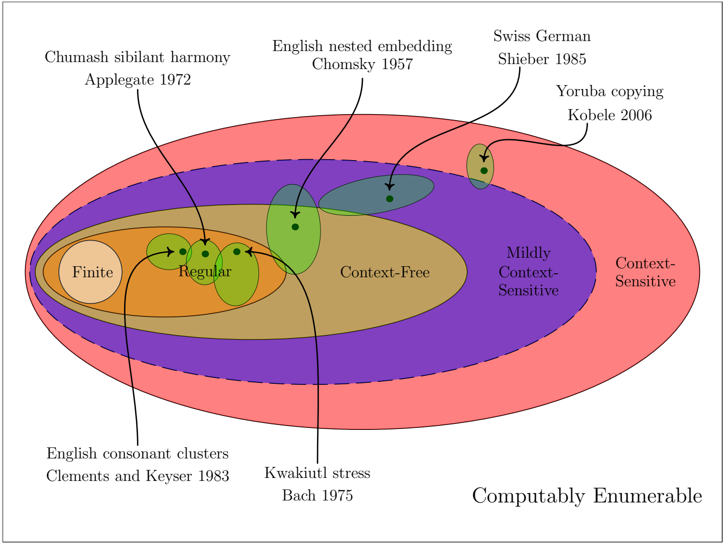


Figure: SL, SP, and SPL classes.

Summary

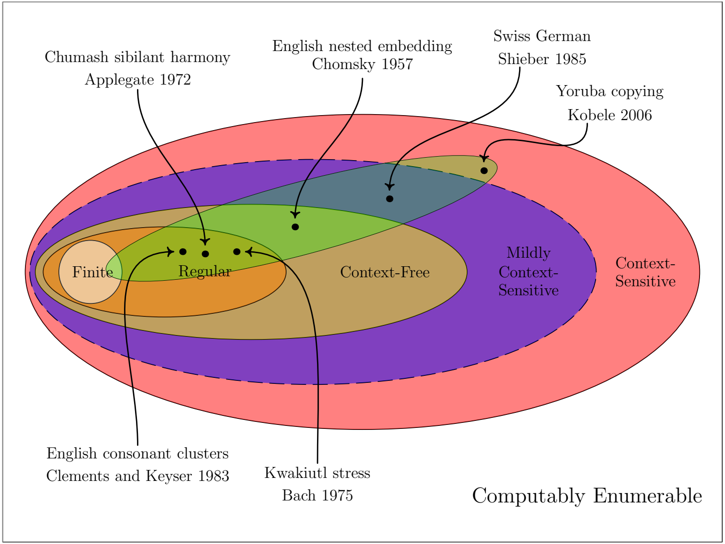


Figure: Where is the feasible learner of this class?

Summary

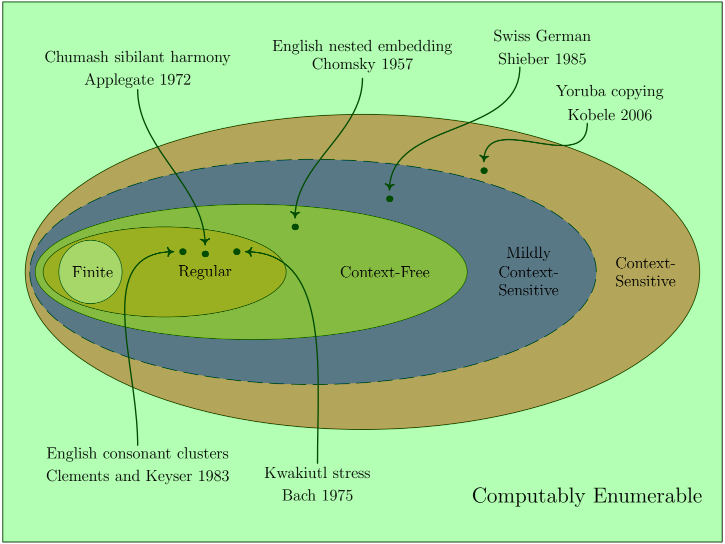


Figure: Where is the feasible learner of this class?

Modular Learning and Biology

Adaptive specialization of mechanism is so ubiquitous and so obvious in biology, at every level of analysis, and for every kind of function, that no one thinks it necessary to call attention to it as a general principle about biological mechanisms…

From a biological perspective, the idea of a general-learning mechanism is equivalent to assuming that there is a general-purpose sensory organ, which solves the problem of sensing.

(Gallistel and King 2009:218)

Conclusion

1. Linguistic patterns are not arbitrary.

Conclusion

1. Linguistic patterns are not arbitrary.
2. Only structured classes of patterns can be learned.

Conclusion

1. Linguistic patterns are not arbitrary.
2. Only structured classes of patterns can be learned.
3. Distinct, feasible learning models for distinct phonological patterns exist.

Conclusion

1. Linguistic patterns are not arbitrary.
2. Only structured classes of patterns can be learned.
3. Distinct, feasible learning models for distinct phonological patterns exist.
4. These help explain the character of the typology.

Conclusion

1. Linguistic patterns are not arbitrary.
2. Only structured classes of patterns can be learned.
3. Distinct, feasible learning models for distinct phonological patterns exist.
4. These help explain the character of the typology.
5. A single, feasible learning model for these distinct phonological patterns does not exist (yet, ever?).

Conclusion

1. Linguistic patterns are not arbitrary.
2. Only structured classes of patterns can be learned.
3. Distinct, feasible learning models for distinct phonological patterns exist.
4. These help explain the character of the typology.
5. A single, feasible learning model for these distinct phonological patterns does not exist (yet, ever?).
6. Such a model is likely to have to attribute the character of the typology to something else.

Conclusion

1. Linguistic patterns are not arbitrary.
2. Only structured classes of patterns can be learned.
3. Distinct, feasible learning models for distinct phonological patterns exist.
4. These help explain the character of the typology.
5. A single, feasible learning model for these distinct phonological patterns does not exist (yet, ever?).
6. Such a model is likely to have to attribute the character of the typology to something else.
7. Artificial language learning experiments can help.

Conclusion

1. Linguistic patterns are not arbitrary.
2. Only structured classes of patterns can be learned.
3. Distinct, feasible learning models for distinct phonological patterns exist.
4. These help explain the character of the typology.
5. A single, feasible learning model for these distinct phonological patterns does not exist (yet, ever?).
6. Such a model is likely to have to attribute the character of the typology to something else.
7. Artificial language learning experiments can help.

The hypothesis that phonological learning is modular currently offers the best explanation not only for how phonological patterns are learned but also for the character of the typology.

Part III

Course Review

Course Review

Advantages of Logic and Model-theoretic Representations

1. Logic provides a flexible description language to precisely describe phonological (linguistic) generalizations, both constraints and transformations.

Course Review

Advantages of Logic and Model-theoretic Representations

1. Logic provides a flexible description language to precisely describe phonological (linguistic) generalizations, both constraints and transformations.
2. Model-theoretic representations let linguists specify the representations they want.

Course Review

Advantages of Logic and Model-theoretic Representations

1. Logic provides a flexible description language to precisely describe phonological (linguistic) generalizations, both constraints and transformations.
2. Model-theoretic representations let linguists specify the representations they want.
3. MSO logic with word models based on successor or precedence is sufficient to describe phonological generalizations.

Course Review

Advantages of Logic and Model-theoretic Representations

1. Logic provides a flexible description language to precisely describe phonological (linguistic) generalizations, both constraints and transformations.
2. Model-theoretic representations let linguists specify the representations they want.
3. MSO logic with word models based on successor or precedence is sufficient to describe phonological generalizations.
4. This is an advantageous for grammarians and field researchers who want to document a language for the ages.

Course Review

Advantages of Logic and Model-theoretic Representations

1. Logic provides a flexible description language to precisely describe phonological (linguistic) generalizations, both constraints and transformations.
2. Model-theoretic representations let linguists specify the representations they want.
3. MSO logic with word models based on successor or precedence is sufficient to describe phonological generalizations.
4. This is an advantageous for grammarians and field researchers who want to document a language for the ages.
5. It is also useful for theorists, who want to understand what the grammarian wrote down.

Course Review

Advantages of Logic and Model-theoretic Representations

1. Logic provides a flexible description language to precisely describe phonological (linguistic) generalizations, both constraints and transformations.
2. Model-theoretic representations let linguists specify the representations they want.
3. MSO logic with word models based on successor or precedence is sufficient to describe phonological generalizations.
4. This is an advantageous for grammarians and field researchers who want to document a language for the ages.
5. It is also useful for theorists, who want to understand what the grammarian wrote down.
6. The transformations not only can provide descriptions, we can use them to translate between analyses!

Advantages of Logic and Model-theoretic Representations

7. The logic (MSO, FO, PROP, CNL/QF) and representations ($\triangleleft$, $<$, …) factor generalizations along two axes.

Advantages of Logic and Model-theoretic Representations

7. The logic (MSO, FO, PROP, CNL/QF) and representations ($\triangleleft, <, \ldots$) factor generalizations along two axes.
8. The classes they demarcate have remarkable **grammar-independent** properties.

Advantages of Logic and Model-theoretic Representations

7. The logic (MSO, FO, PROP, CNL/QF) and representations ($\triangleleft, <, \ldots$) factor generalizations along two axes.
8. The classes they demarcate have remarkable **grammar-independent** properties.
9. So **any grammar** which describes an extensionally equivalent generalization has these properties.

Advantages of Logic and Model-theoretic Representations

7. The logic (MSO, FO, PROP, CNL/QF) and representations ($\triangleleft, <, \ldots$) factor generalizations along two axes.
8. The classes they demarcate have remarkable **grammar-independent** properties.
9. So **any grammar** which describes an extensionally equivalent generalization has these properties.
10. These properties have implications for typology, learning, and psychology.

Course Review

Phonology

1. This line of inquiry has revealed that phonological generalizations cluster at the bottom.

Course Review

Phonology

1. This line of inquiry has revealed that phonological generalizations cluster at the bottom.
2. A theory of phonology ought to explain this fact.

Phonology

1. This line of inquiry has revealed that phonological generalizations cluster at the bottom.
2. A theory of phonology ought to explain this fact.
3. OT, with its emphasis on global optimization, does not.

Course Review

Phonology

1. This line of inquiry has revealed that phonological generalizations cluster at the bottom.
2. A theory of phonology ought to explain this fact.
3. OT, with its emphasis on global optimization, does not.
4. If humans are wired to generalize in the way suggested by those grammar-independent properties, then it accounts for both the typology and the learnability.

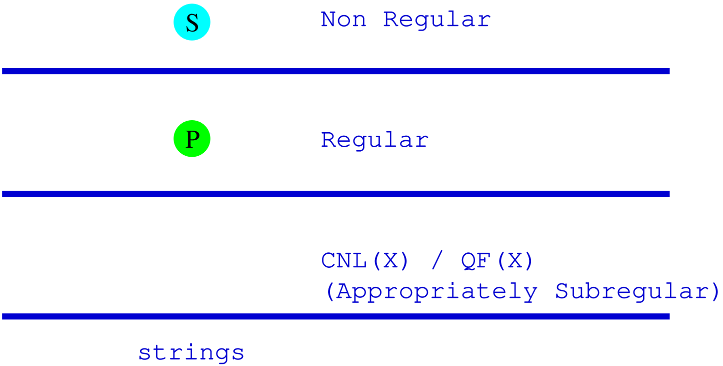
Course Review

Issues

1. The trade-off between representation and computational power needs to be better understood.
2. There is wiggle-room, but not anything goes.
3. Identifying principles here will help.

Course Review

Beyond Phonology



(Chomsky 1957, Johnson 1972, Kaplan and Kay 1994, Roark and Sproat 2007, Heinz and Idsardi 2011, and many others)

Beyond Phonology

S Non Regular

Regular

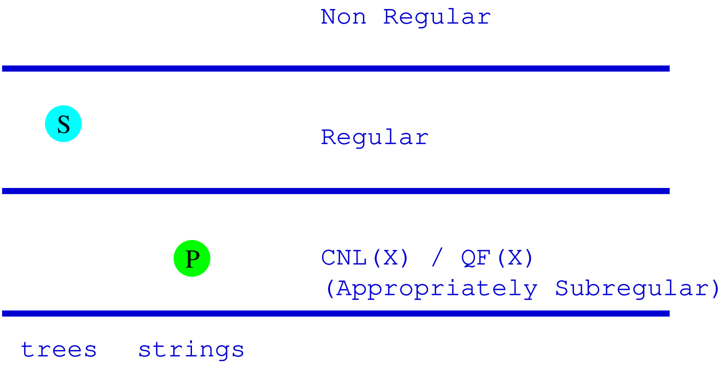
P CNL(X) / QF(X)
(Appropriately Subregular)

strings

(Potts and Pullum 2002, Heinz 2007 et seq., Graf 2010 and many others)

Course Review

Beyond Phonology



(Rogers 1994, 1998, Knight and Graehl 2005, Pullum 2007, Kobele 2011, Graf 2011 and many others)

Beyond Phonology

Non Regular

Regular

S P CNL(X) / QF(X)
(Appropriately Subregular)

trees strings

(Graf 2013, Graf and Heinz 2015, Graf 2017 and others)

Thank you all!

This has been a lot of fun for me!