~~Why do we start with linguistics?~~ lecture #4

a story of computation - and, in particular, the story of computation "in the wild" - begins with linguistics.

CSSS 2011
S. DeDeo / simonsdotch
Tuesday, June 27
10.45 AM

You have a language - this is, briefly, just a list of ~~words~~ sentences.

Composed of a finite number of words

The crinkle: infinitely long list! (Or so.)

The good: can you figure out the underlying grammar of the language? ~~What will tell us~~

Grammars tell us why

"I ate the sheep"

is a valid sentence, but

"I the ate sheep"

is it.

"Infinite use of finite means"

The problem: how to describe a grammar? What does it look like? What are the minimal resources you need?

"infinite productivity" - "finite means"

(don't just memorize the good sentences, but ~~learn~~ have a Universal Grammar)

The answer (in 1965) —

the Chomsky hierarchy

decreasing number ⇒ increasing expressive power.

~~three more levels~~

"Derivations"

Grammars are described in terms of derivation

rewriting, or "production" rules.

$V_N = \{ A, B, C, D, \ldots \}$

$V_T = \{ a, b, c, d, \ldots \}$

an example of a production rule:

$$aAb \rightarrow aBb$$

"wherever A is sandwiched between a & b, you can flip it"

$$N \rightarrow AN$$

$$A \rightarrow a$$

aaaaAbbb → aaaaBbbb.

rewriting rules for the levels

level 3 : $A \rightarrow aB$

(non-terminal symbol rewritten as combination).

level 2 : $A \rightarrow$ (any word made from combination of non & terminal symbols).

level 1 : $Q_1 A Q_2 \rightarrow Q_1 ($any word made from...$) Q_2$

$Q_1, Q_2 \in (V_N \cup V_T)^*$.

level 0 : no restrictions.

examples

[NP] → [the][N] level 3

[NP] → [AP][NP] level 2

[SUB][VERB][OBJ] → [OBJ][was][VERB]-ed by [SUB]

level 0

[the] [NP] [VP sing] → [NP sig] [VP sing] level 1

aggreement.

⟹ [level 2 "context free"
(really: context ignorant!)
common in definitions of computer languages.

level 1 - capturing notions such as verb agreement; "context"

level 0 - long-range dependence. "transformational grammar" (& the linguistics wars.)

PUMPING LEMMAS (two of them)

1. prove that you have "at least" a regular language.

XYZ
$\rightarrow XY^nZ$

we'll come back to this and see why the pumping lemma applies

2. prove that it must actually be level 2.

$(((\quad)))$

can "pump" up the parentheses.

$XYZ \rightarrow X^{\overset{\uparrow}{n}} Y Z^n$

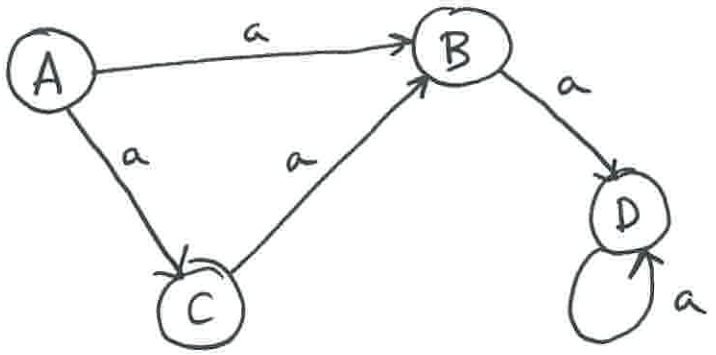
repeated n times

Finite State Automata.

[environmental inputs.]

"input letters"
{a, b, c, ...}

"internal states"
{A, B, C, ...}

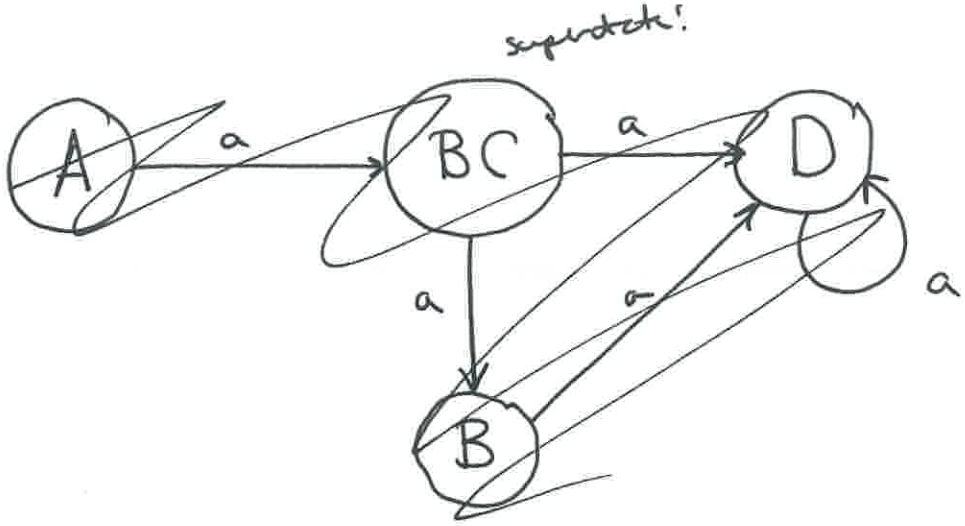


includes:

$A \to aB$
$A \to aC$
$B \to aD$
&c.

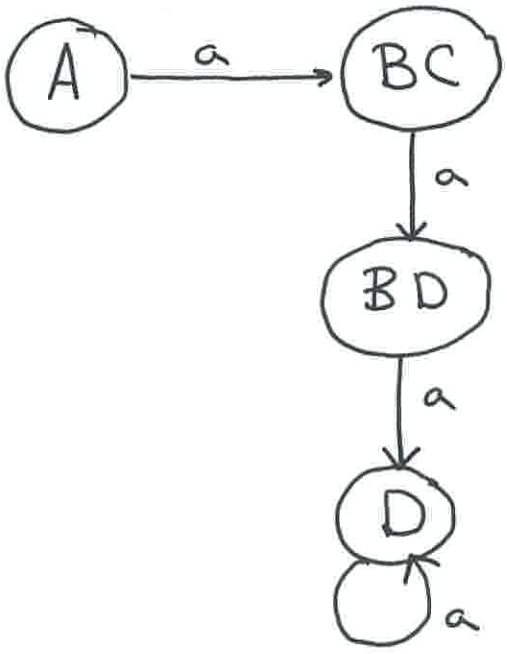
} all of these are level 3 production rules!

"non-deterministic" — when you are in state A, what do you do when you receive "a"?



now it is deterministic!

This is nice, for many reasons — not least of all, you know what "state" the system is in.



You will notice that ~~some of the~~ these automata tend to drag you in to a subspace of states from which you can not escape.

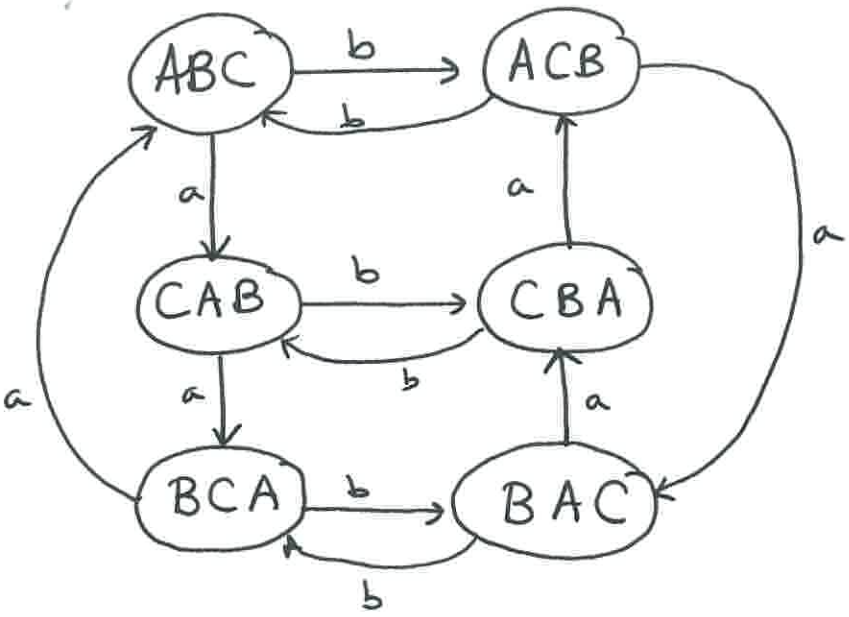
e.g. — in our friend, keep hitting the system with "b" — what happens?

It is not ob like hitting "control-C" on your computer when you've done something wrong. It is, in other words — a reset.

~~In fact,~~ all machines amount to a collection of permutations and of resets.

What is a machine that always has an undo?

... A. a Mac.
B. a group!



a = "shift right"

b = "swap second two"

GROUPS are a kind of finite state automata!

Wait... what does this mean?

for every element,

$g \in G$

there is an inverse

g^{-1}

that takes you back.

No resets! (No "undo" command always available).

JORDAN - HOLDER DECOMPOSITION

a group is either a <u>simple</u> group,

OR has a hierarchical decomposition into
it has "normal subgroups"!

What is a normal subgroup?

G

N is normal in G

N if

gNg^{-1} is also in N.

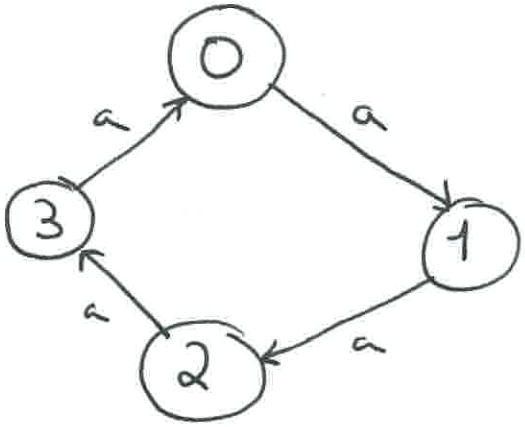
is Z_3 a <u>normal</u> subgroup of S_3?

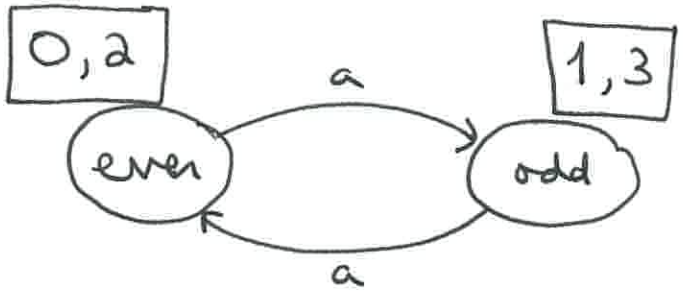
$a \in N$

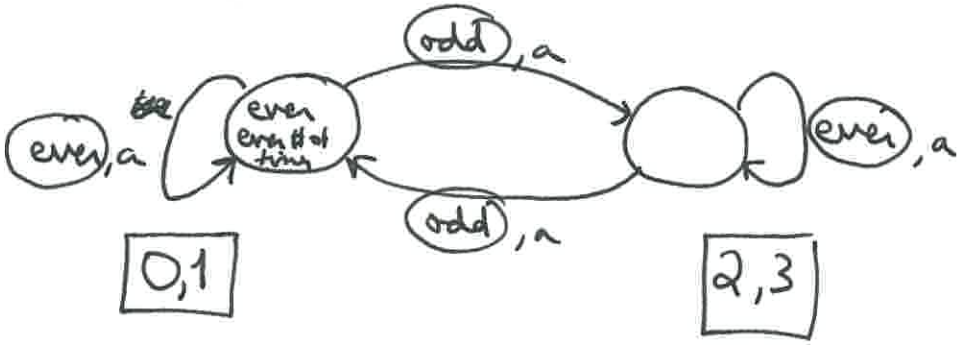
$bab^{-1} \in N$?

$bab \in N$?

} answer: yes!







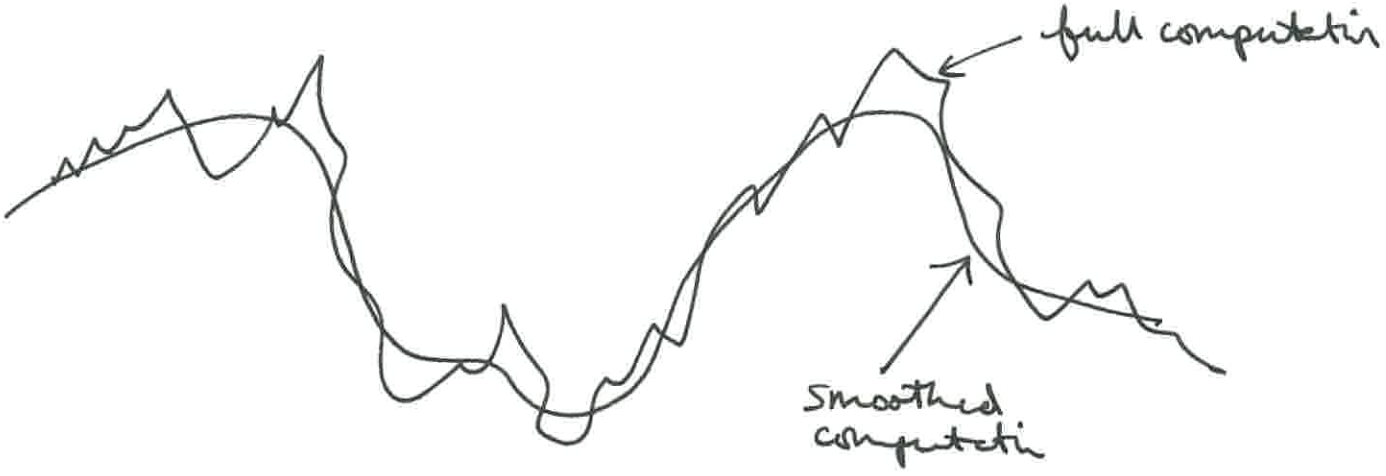
each state in the original machine has a "coordinate" representation in the decomposition.

- talk about SEMIGROUPS.

HERE :- talk about the Krohn-Rhodes theorem.

what is this top level?

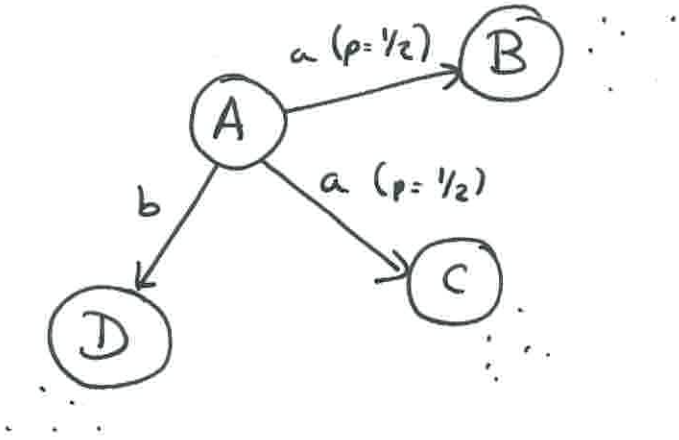
— to know what's going on (how I'm swapping, moving around in it) I don't have to know anything about the lower levels.



residual

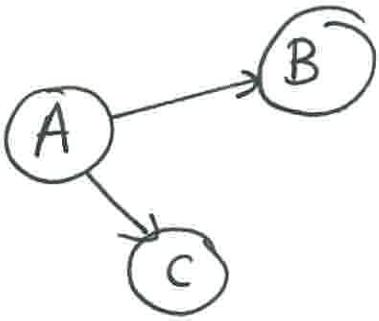
(need to know the smoothed component to

How to handle stochasticity?



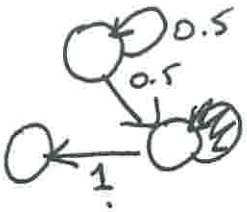
you may think this is a little like the non-deterministic case - & you'd be right!

We'll consider the singleton alphabet only.

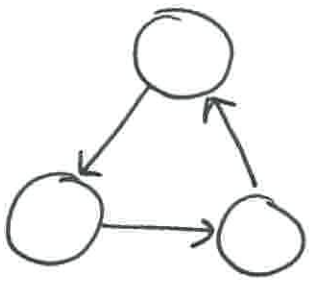


a Markov process!

Basic idea:

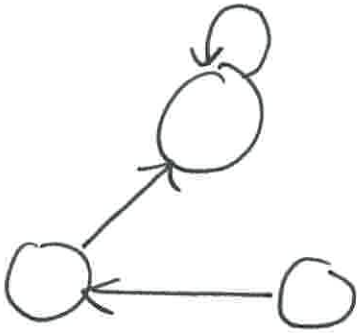


Consider all semigroup actions on the states.



permutation.

has zero probability



partial reset.

⋮

&c.

each of these gets a probability given by the original Markov process.

Now —

each transition has a probability.

So: roll the dice, pick a word of the semigroup, & let's go!

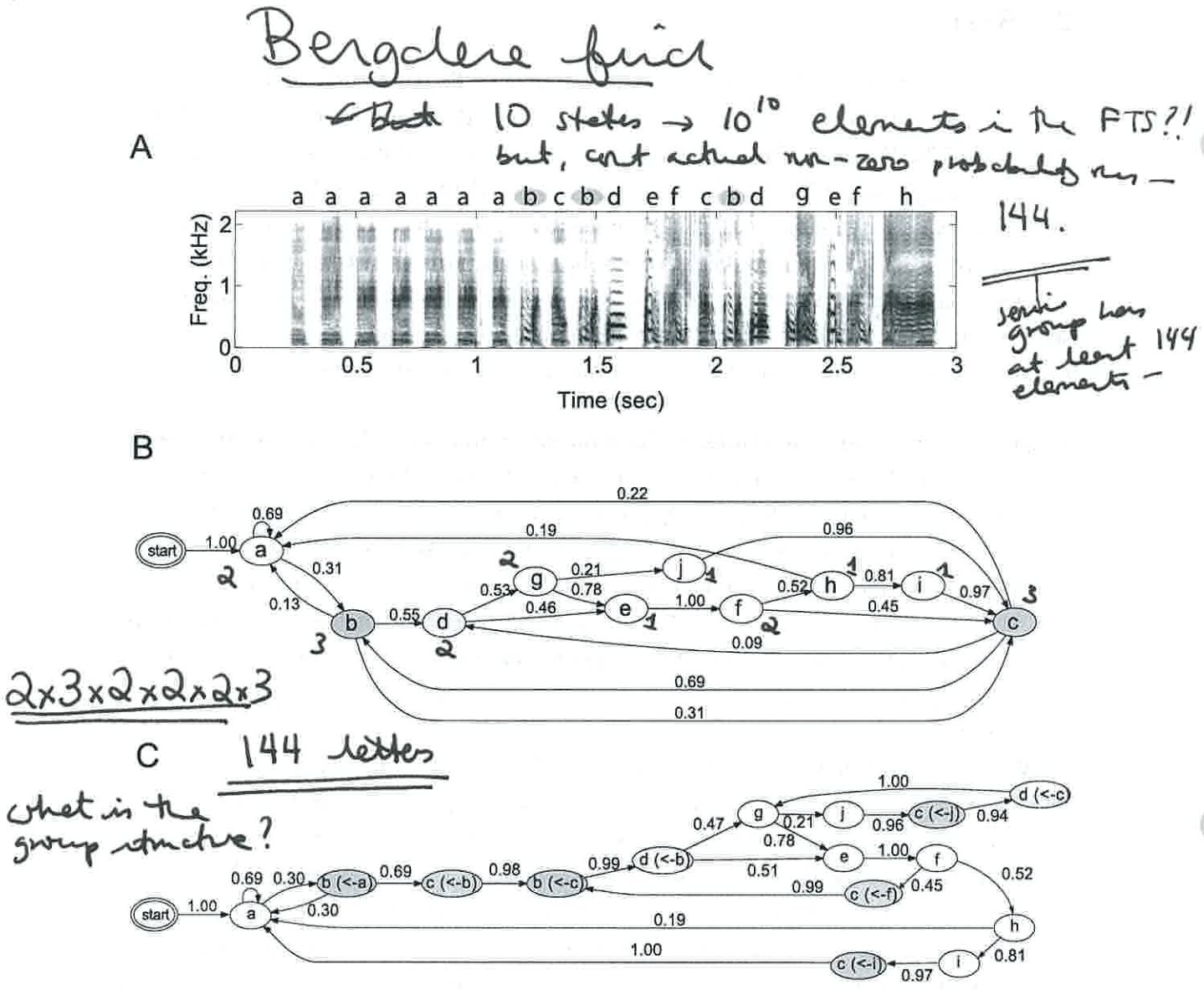


Figure 1: Example of sonogram of Bengalese finch song and its syllable label sequence. (A) Sonogram of Bengalese finch (BF09) with syllable labels annotated by three human experts. Labeling was done based on visual inspection of sonogram and syllables with similar spectrogram given same syllable. (B) Bigram automaton representation (transition diagram) of syllable sequences obtained from same song set as (A). Ellipses represent one syllable and arrows with values represent transitional probabilities. Rare transitions with probabilities < 0.01 are omitted. (C) POMM representation of same sequences as (B). Syllables that have significant higher-order dependency on preceding syllables (colored states in (B)) are divided into distinct states depending on preceding syllables (context).