Cognition, Language & Communication'14

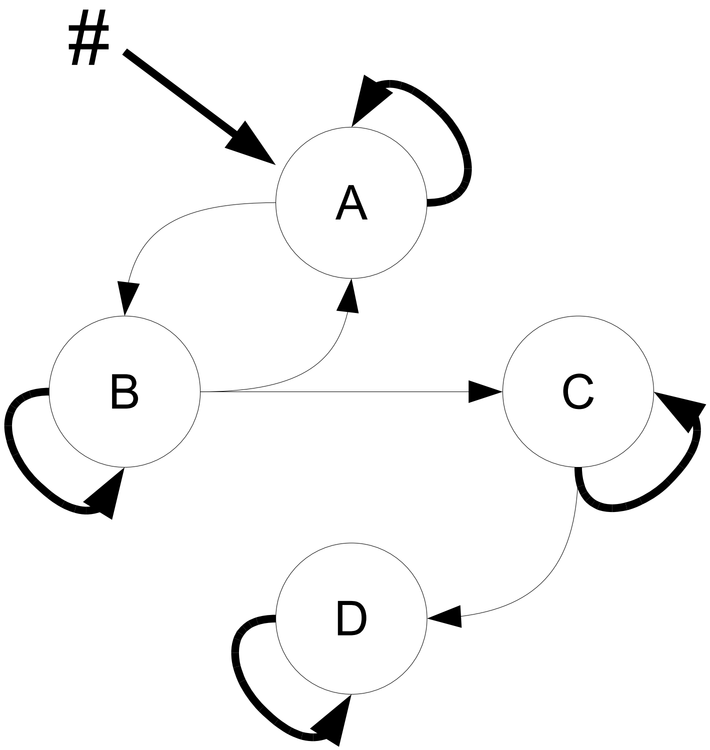
MSc Brain & Cognitive Science, UvA
track Cognitive Science

Jelle Zuidema
Institute for Logic, Language & Computation
zuidema@uva.nl

week 4: Artificial Language Learning

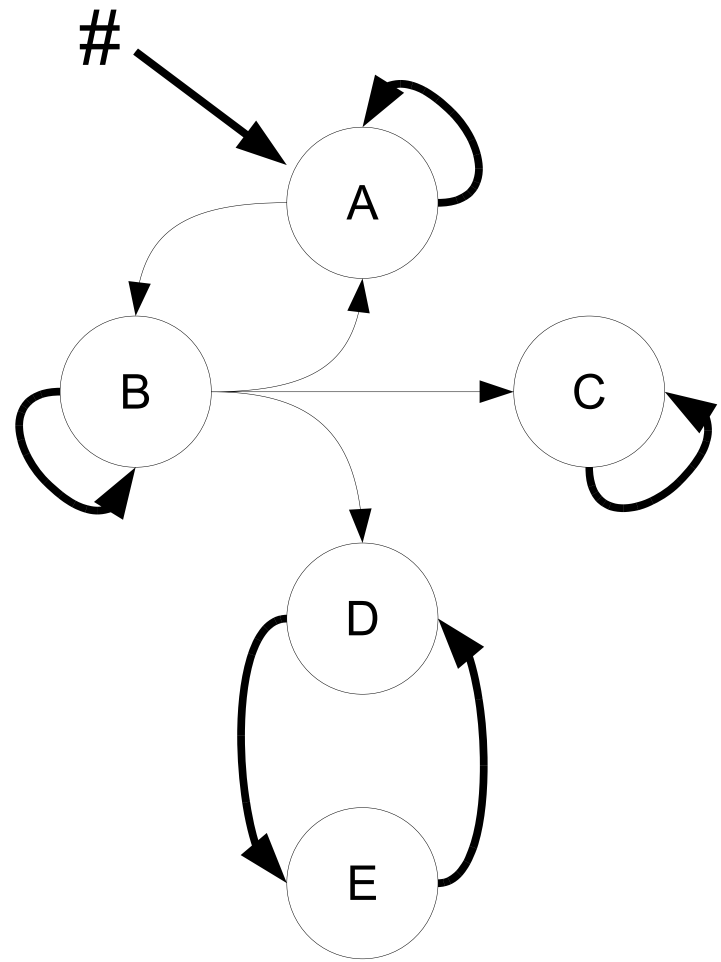
Recap: Transitional Probabilities

	A	B	C	D
#	1	0	0	0
A	0.8	0.2	0	0
B	0.1	0.8	0.1	0
C	0	0	0.8	0.2
D	0	0	0	1



D is a “sink” (point attractor)

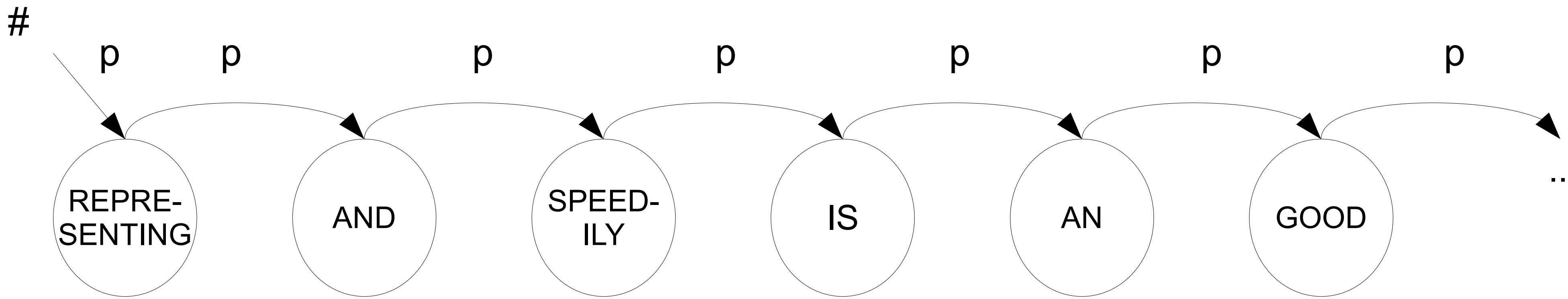
Recap: Transitional Probabilities

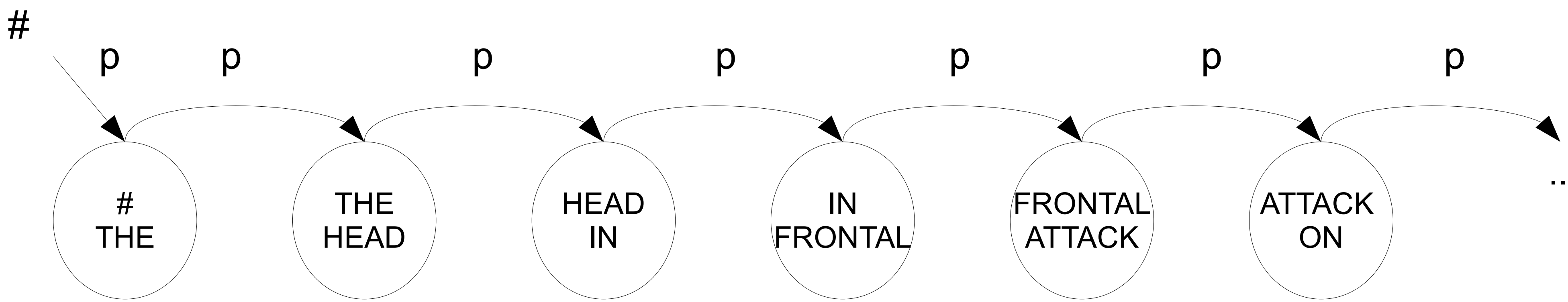


This system has multiple attractors

C is a “sink” (point attractor)

D-E is a “limit cycle”



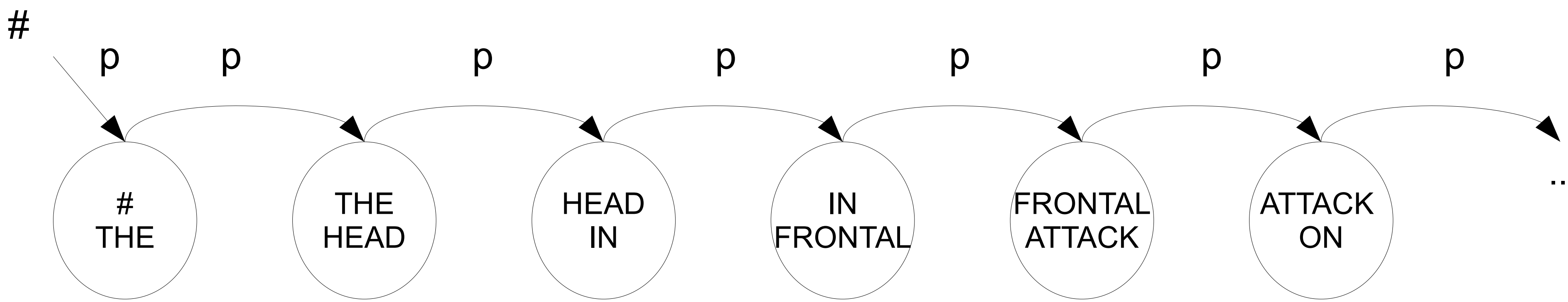


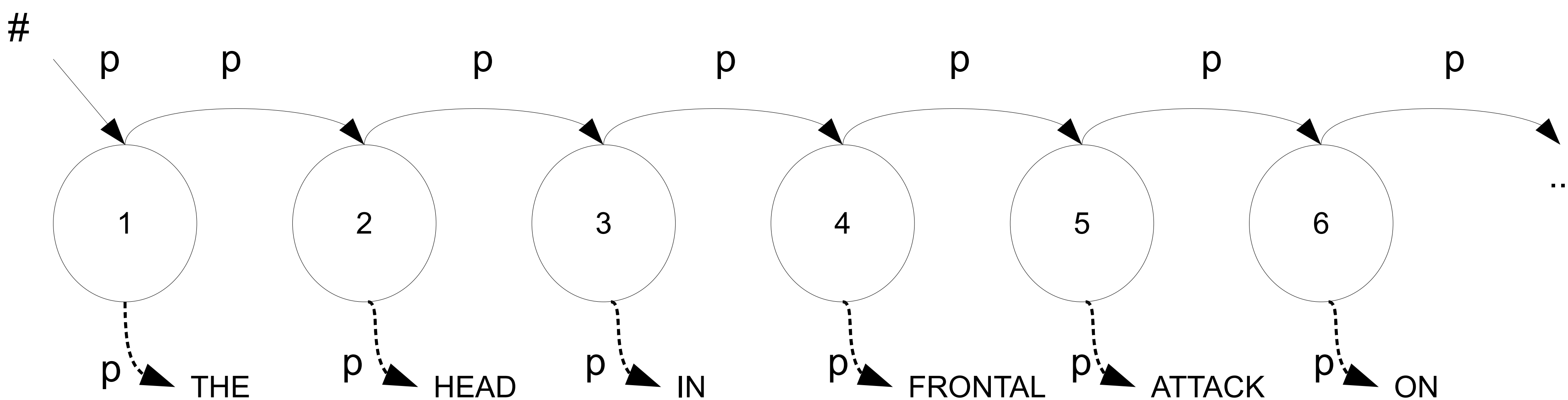
- Markov order 1: the probability of the next state depends only on the current state
- Markov order 0: the probability of the next state is independent of the current state
- Markov order n: the probability of the next state depends on the current state and the previous (n-1) states
- Equivalently: the previous (n-1) states are incorporated in the current state description!
- In the language domain, (n+1)-th order Markov models are also called ngrams!

Recap: Markov models

- Markov property: the probability of the next event is only dependent on the current state
- Terms to know:
  - (In)dependence of current state
  - Transitional probabilities, transition matrix
  - Sink / point attractor, Limit cycle
  - Markov order

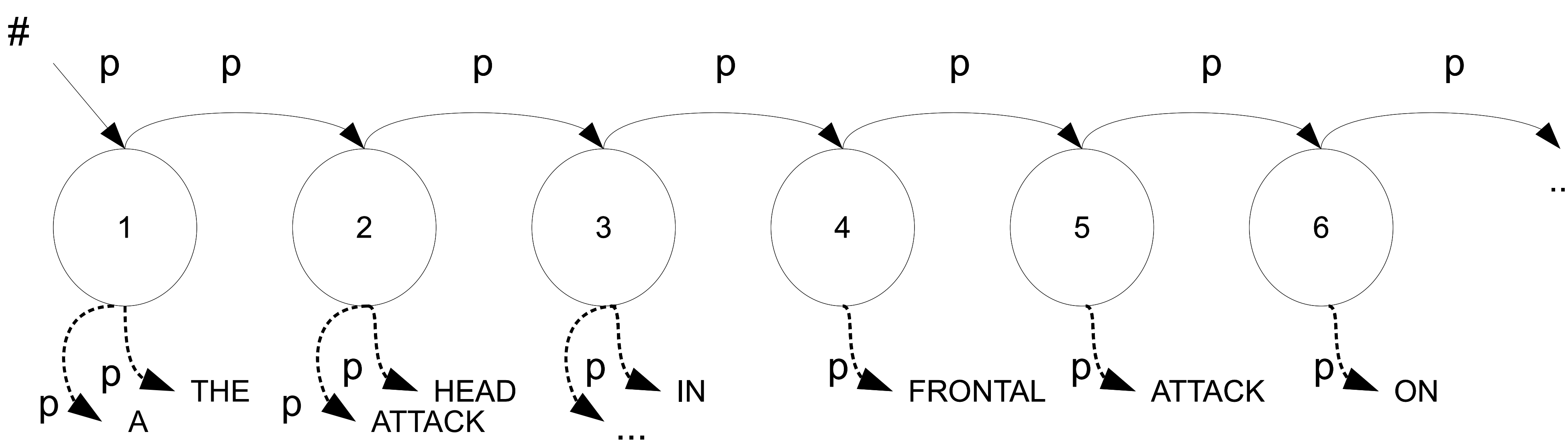
Generalizing over states

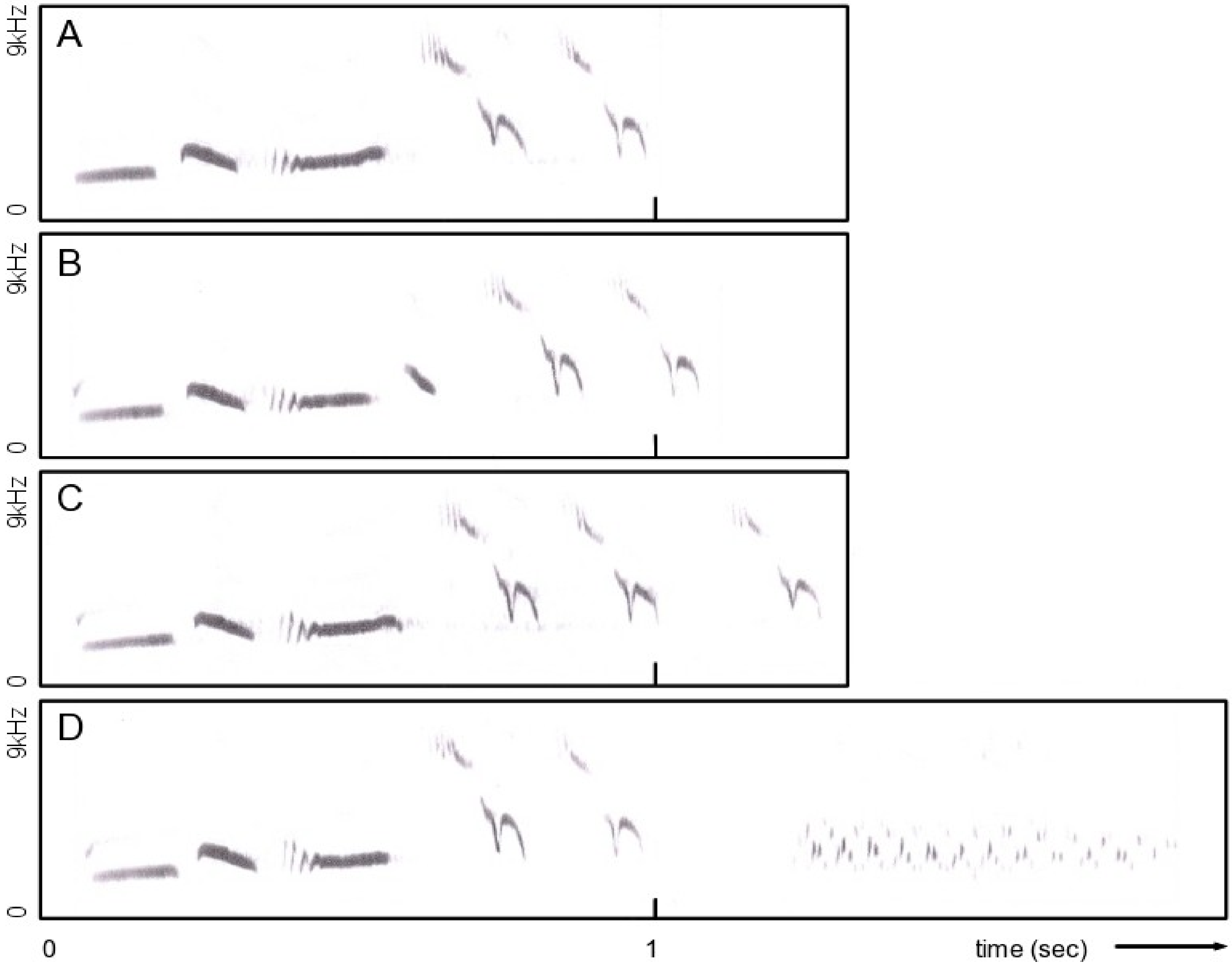




Recap: Hidden Markov Model

- Finite number of hidden states
- “Transition probabilities” from state tot state
- Finite number of observable symbols
- “Emission probabilities” from hidden states to observable symbols




A
B
C
D
9kHz
0
0
1
time (sec)

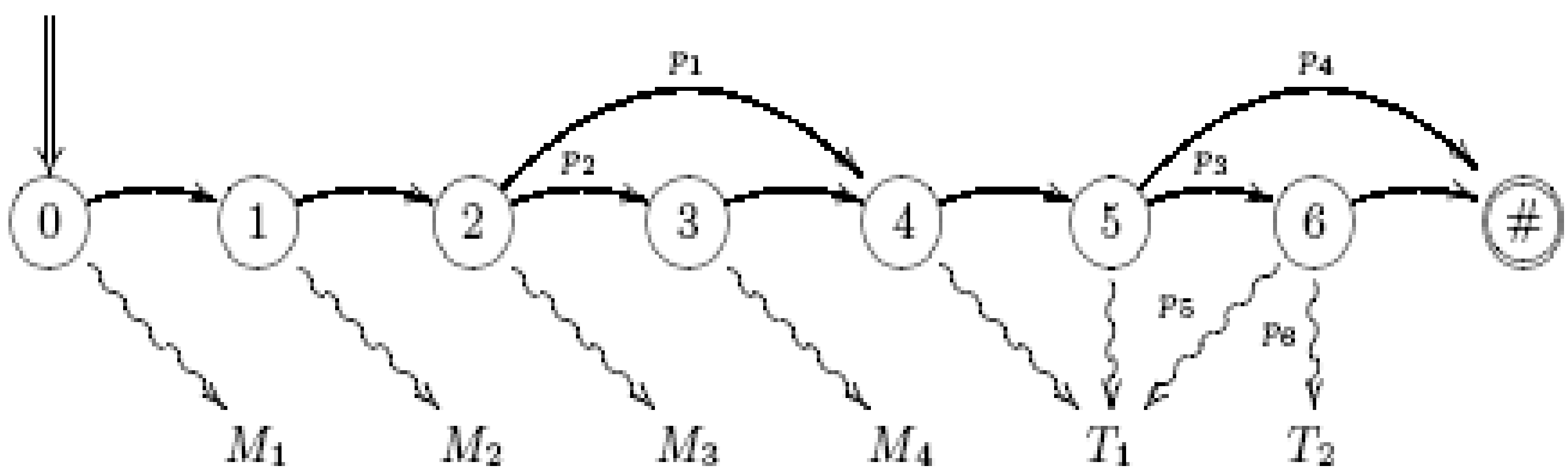
(a) Transition probabilities

	next element						
	M_1	M_2	M_3	M_4	T_1	T_2	#
0	1	0	0	0	0	0	0
M_1	0	1	0	0	0	0	0
M_2	0	0	1	0	0	0	0
M_3	0	0	0	p_2	p_1	0	0
M_4	0	0	0	0	1	0	0
T_1	0	0	0	0	p_3	p_4	p_5
T_2	0	0	0	0	0	0	1

(b) Bigram analysis

State	$\leadsto$	*Sound*	*Probability*
0	$\leadsto$	M_1	1
M_1	$\leadsto$	M_2	1
M_2	$\leadsto$	M_3	1
M_3	$\leadsto$	T_1	p_1
M_3	$\leadsto$	M_4	p_2
M_4	$\leadsto$	T_1	1
T_1	$\leadsto$	T_1	p_3
T_1	$\leadsto$	T_2	p_4
T_1	$\leadsto$	#	p_5
T_2	$\leadsto$	#	1

(c) HMM



Terms to know:

- finite-state automaton (FSA)
- hidden markov model (HMM)
- Forward algorithm:

  P(**o|**HMM)

- Viterbi algorithm:

  argmax_**h** P(**o**|**h,**HMM)

- Baum-Welch algorithm:

  argmax_HMM P(**o**|HMM)

Recap: Chomsky'57 vs. the FSA

Let S1, S2, S3, S4, S5 be simple declarative sentences in English. Then also

(2) If S1, then S2.
(3) Either S3 or S4.
(4) The man who said that S5, is arriving today

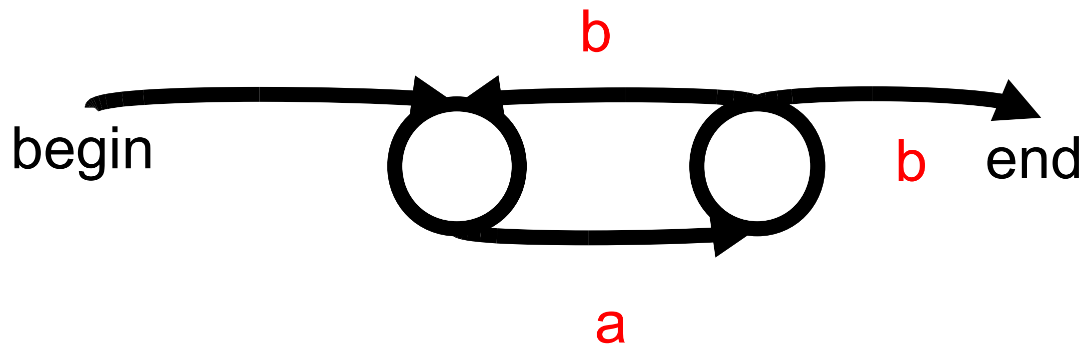
are sentences of English.

E.g., *if either you are with us or you are against us applies here, then there is nothing more to discuss*.

Simplest example of a "finite-state language":

$(ab)^n$

E.g. ab, abab, ababab, abababab



Simplest example of a "context-free language":

$a^n b^n$

E.g. ab, aabb, aaabbb, aaaabbbb, ...

push-down automaton!

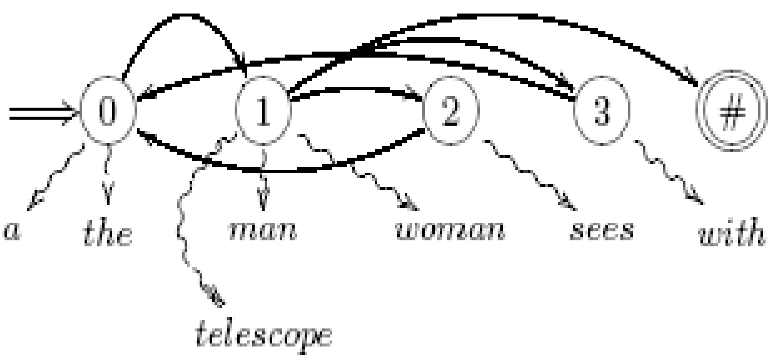
a man sees the woman with the telescope

- bigram, hmm & cfg models & derivations

(a) Bigram

0	⇝	a
a	⇝	man
man	⇝	sees
sees	⇝	the
the	⇝	woman
woman	⇝	with
with	⇝	the
the	⇝	telescope
telescope	⇝	#

(b) HMM



(c) Context-free grammar

S	→	NP	V P	
NP	→	DET	N	
NP	→	DET	N	PP
VP	→	V	NP	
PP	→	PREP	NP	
DET	⇝	a		
DET	⇝	the		
N	⇝	man		
N	⇝	woman		
V	⇝	sees		
PREP	⇝	with		

Table 2: Three models for the production of a sentence (probabilities omited for simplicity)

(a) Bigram

Step	*State*	*Sound*
1	0	a
2	a	man
3	man	sees
4	sees	the
5	the	woman
6	woman	with
7	with	the
8	the	telescope
9	telescope	-
10	#	-

(b) HMM

Step	*State*	*Sound*
1	0	a
2	1	man
3	2	sees
4	0	the
5	1	woman
6	3	with
7	0	the
8	1	telescope
9	#	-

(c) Context-free grammar

Step	*State*	*Sound*
1	S	-
2	NP VP	-
3	DET N VP	-
4	N VP	a
5	VP	man
6	V NP	-
7	NP	sees
8	DET N PP	-
9	N PP	the
10	PP	woman
11	PREP NP	-
12	NP	with
13	DET N	-
14	N	the
15	#	telescope

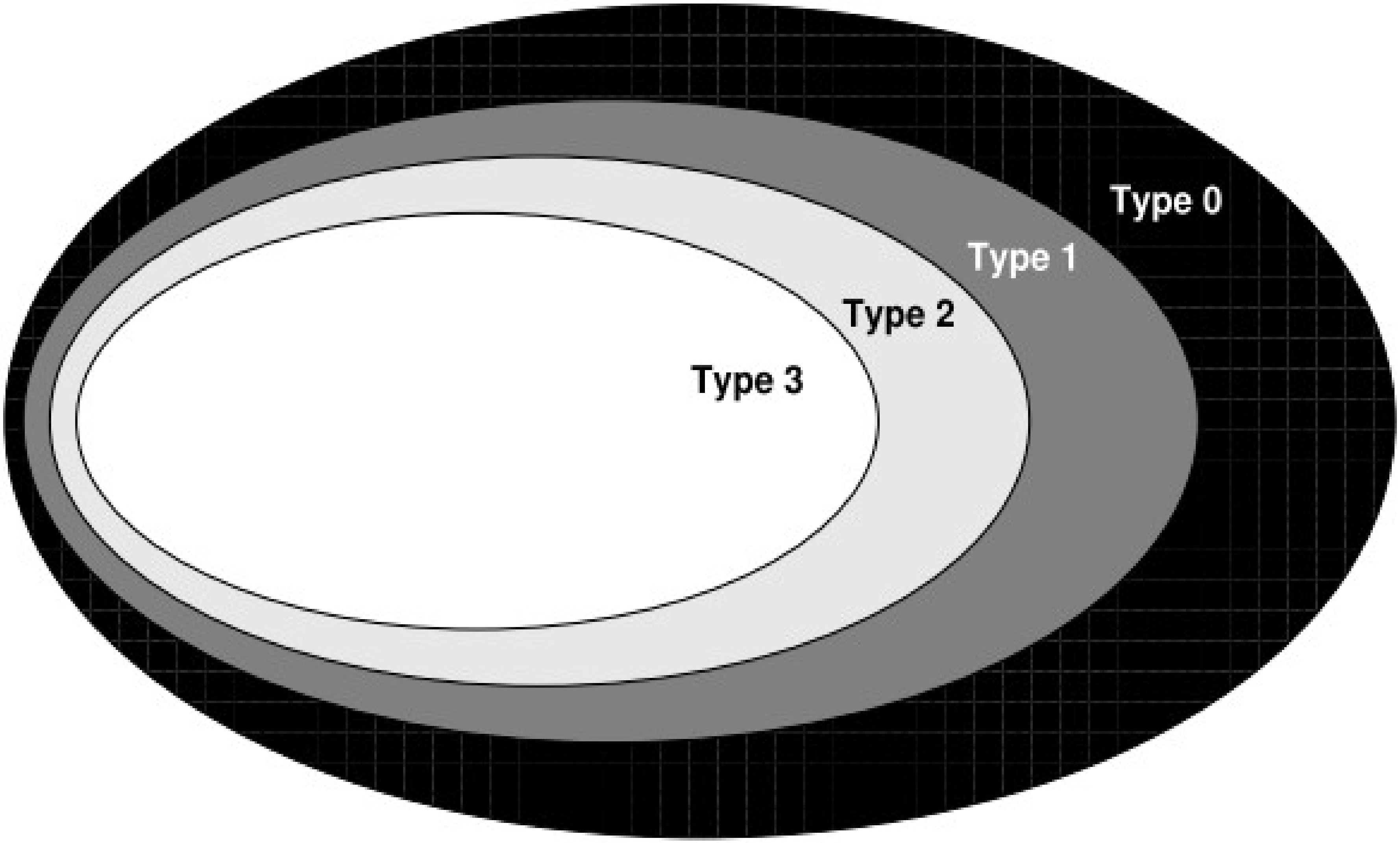
Table 3: Three corresponding derivation sequences in the production of a sentence

discrete infinity!

Chomsky Hierarchy

3. Finite state grammars	$A \rightarrow a$, $A \rightarrow aB$	$(ab)^n$, $a^n b^m$
2. Context-free grammars	$A \rightarrow \gamma$	$a^n b^n$
1. Context-sensitive grammars	$\alpha A \beta \rightarrow \alpha \gamma \beta$	$a^n b^n c^n$
0. Unrestricted grammars	$\alpha \rightarrow \gamma$	$\{a^n b^m c^l \mid l = n * m\}$

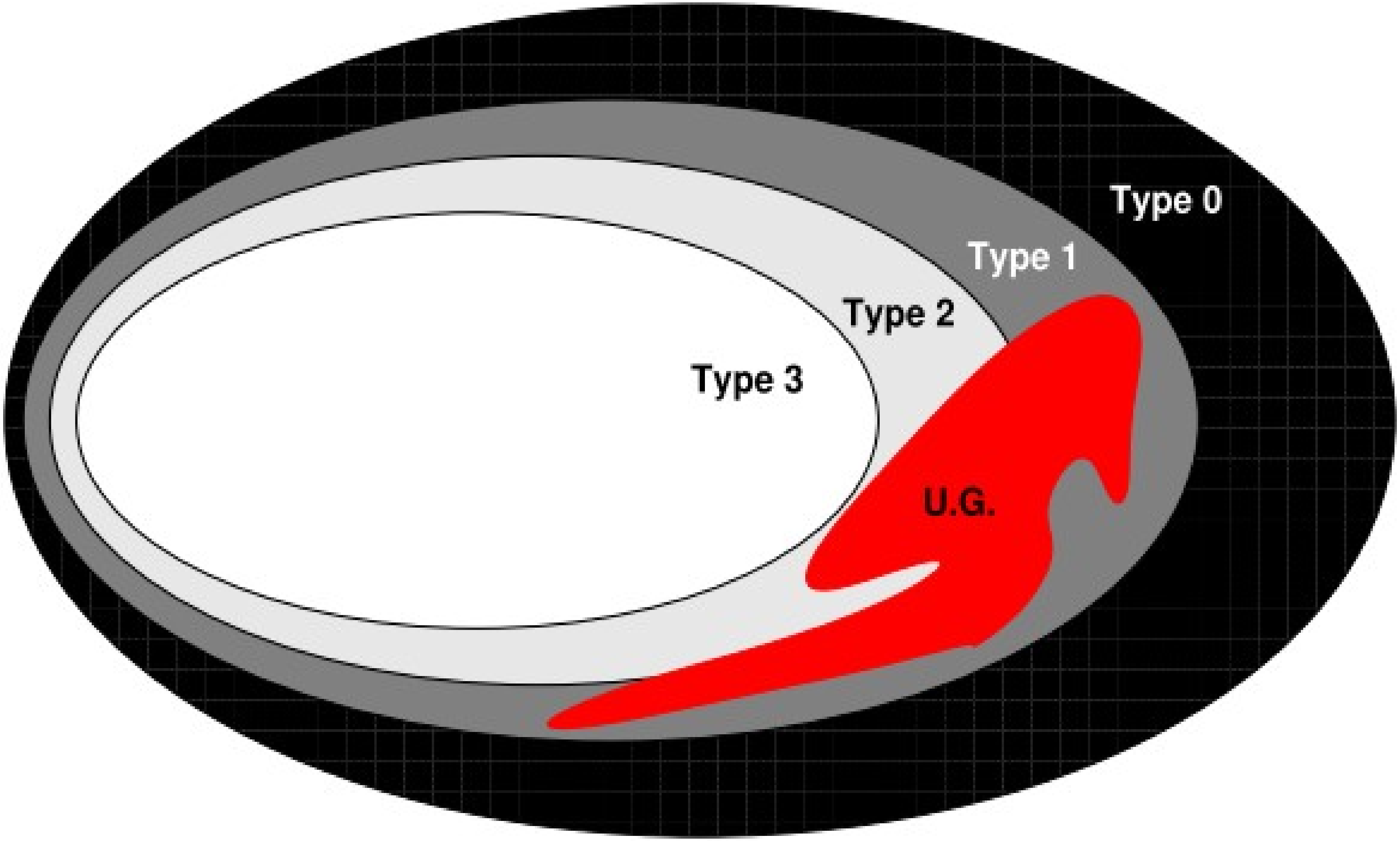
The Chomsky Hierarchy



(1) a. Gilligan claims that Blair deceived the public.
b. Gilligan claims that Campbell helped Blair deceive the public.
c. Gilligan claims that Kelly saw Campbell help Blair deceive the public. (tail recursion)

(2) a. Gilligan behaupte dass Kelly Campbell Blair das Publikum belügen helfen sah. (center embedding)
b. Gilligan beweert dat Kelly Campbell Blair het publiek zag helpen bedriegen. (crossing dependencies)

The Chomsky Hierarchy

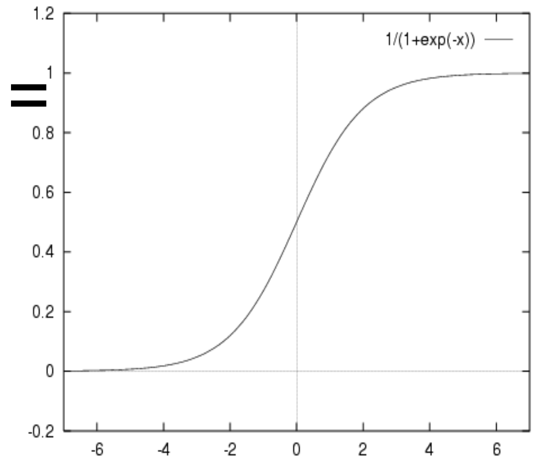


Terms to know

- Rewrite grammars, rewrite operation
  - Production rules
  - Terminal alphabet / observable symbols
  - Nonterminal alphabet / hidden states
  - Start symbol
  - Derivation
  - Phrase-structure
- Contextfree grammars, contextfree constraint
- Push-down automaton
- Discrete infinity

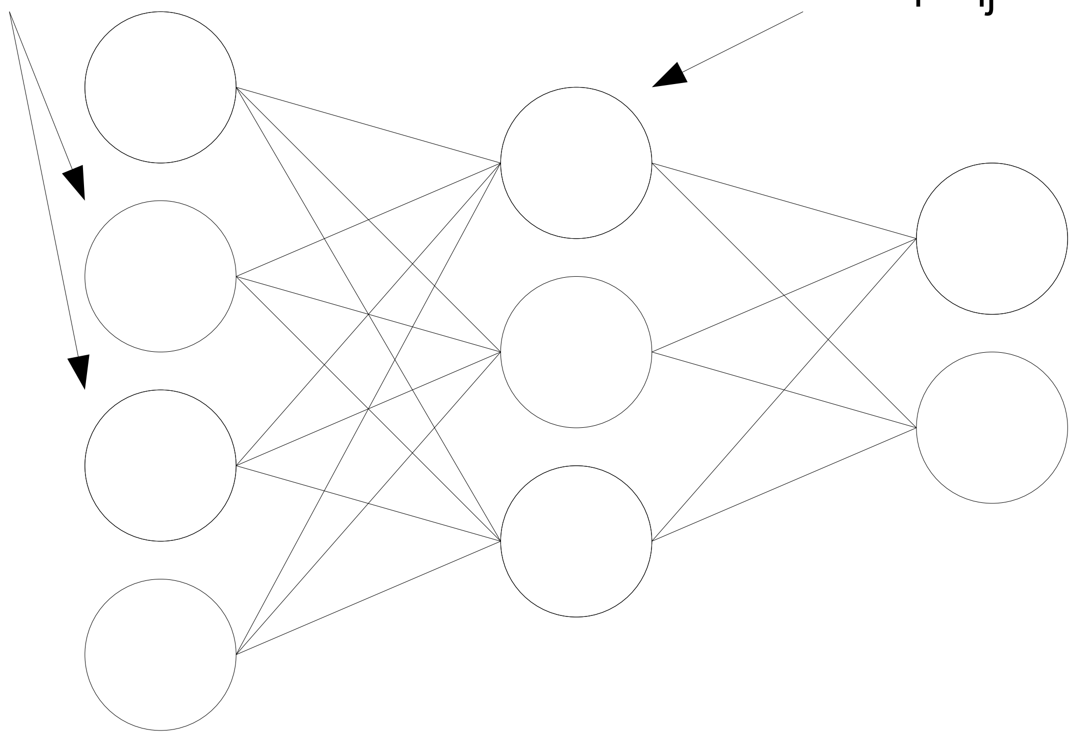
Neural Network

$f() =$

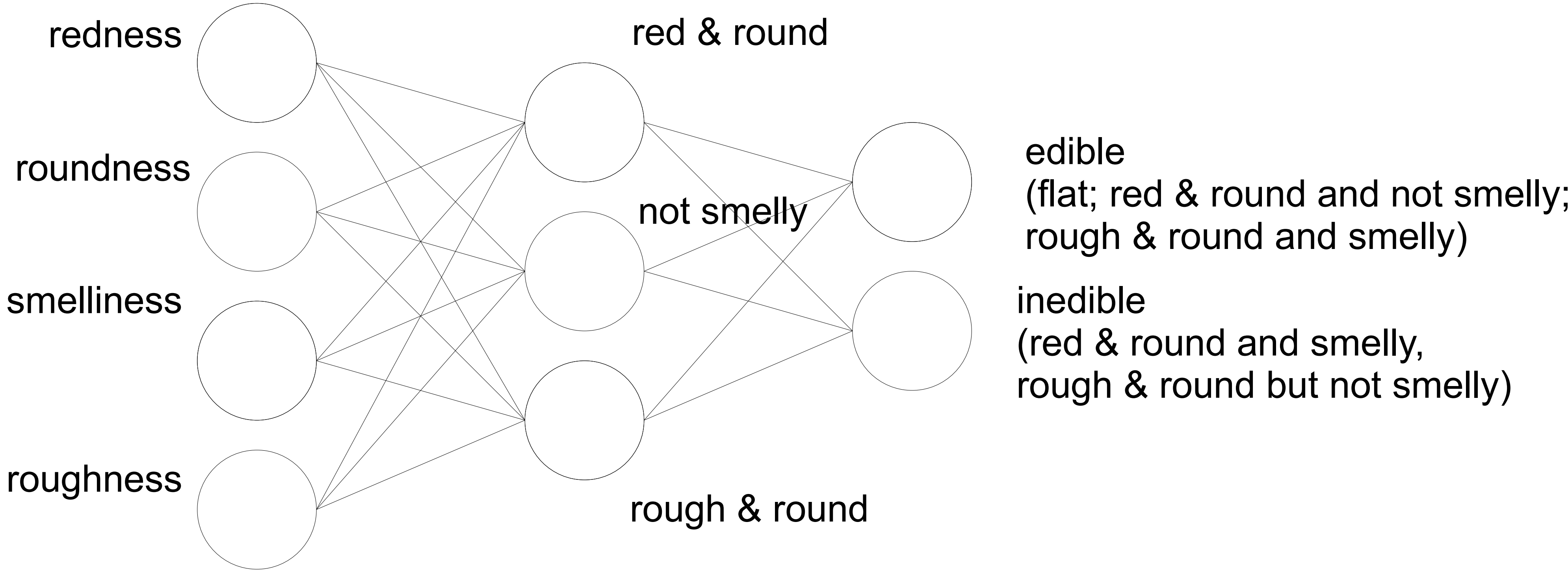


x_i

$f(\Sigma x_i w_{ij} + bias_j)$



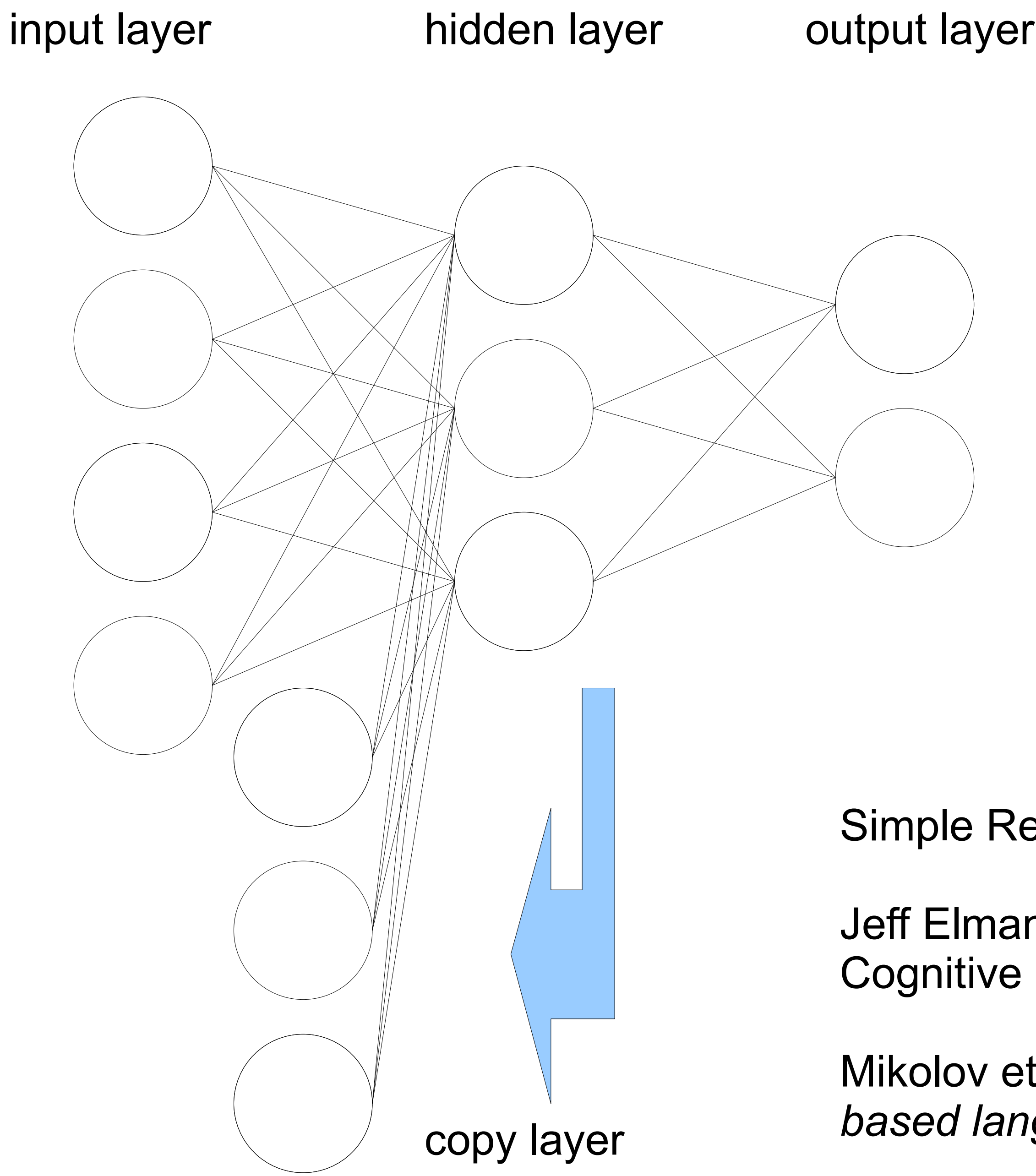
Neural Network



Fictional example: distinguish edible mushrooms from poisonous ones

Suppose: red & round and smelly and rough & round but not smelly mushrooms are poisonous

Recurrent Neural Network



Simple Recurrent Neural Network

Jeff Elman, 1990, *Finding Structure in Time,* Cognitive Science;

Mikolov et al. 2010, *Recurrent neural network based language model, Interspeech 2010*

Simple Recurrent Neural Network

- Processes input sequentially
- Input items represented by a continuous vector
- Computes new internal state (hidden layer) based on input and previous internal state
  - like transition probabilities in HMM
  - but: infinity of possible states (*not* discrete infinity)
- Computes current output based on current internal state
  - like emission probabilities in HMM

Marcus et al. 1999 *Science*

le di di

Marcus et al. 1999 *Science*

fi je je

Marcus et al. 1999 *Science*

je je di

Marcus et al. 1999 *Science*

di le le

- The 16 sentences w/ ABA pattern:
  - ga ti ga, ga na ga,
  - ga gi ga, ga la ga,
  - li na li, li ti li,
  - li gi li, li la li,
  - ni gi ni, ni ti ni,
  - ni na ni, ni la ni,
  - ta la ta, ta ti ta,
  - ta na ta, ta gi ta.
- The 16 sentences w/ ABB pattern:
  - ga ti ti, ga na na,
  - ga gi gi, ga la la,
  - li na na, li ti ti,
  - li gi gi, li la la,
  - ni gi gi, ni ti ti,
  - ni na na, ni la la,
  - ta la la, ta ti ti,
  - ta na na, ta gi gi

Human-specific 'algebraic' reasoning?

- Marcus et al. 1999 *Science*
  - 7.5 month-old infants generalize ABB and AAB patterns to novel stimuli, e.g. "wo fe wo","wo fe fe"
    - I.e., infants significantly preferred the *other* patterns
  - Simple Recurrent Neural Networks cannot learn the pattern
- Hauser et al. '02: monkeys can also do this.

RETRACTED!

Issues

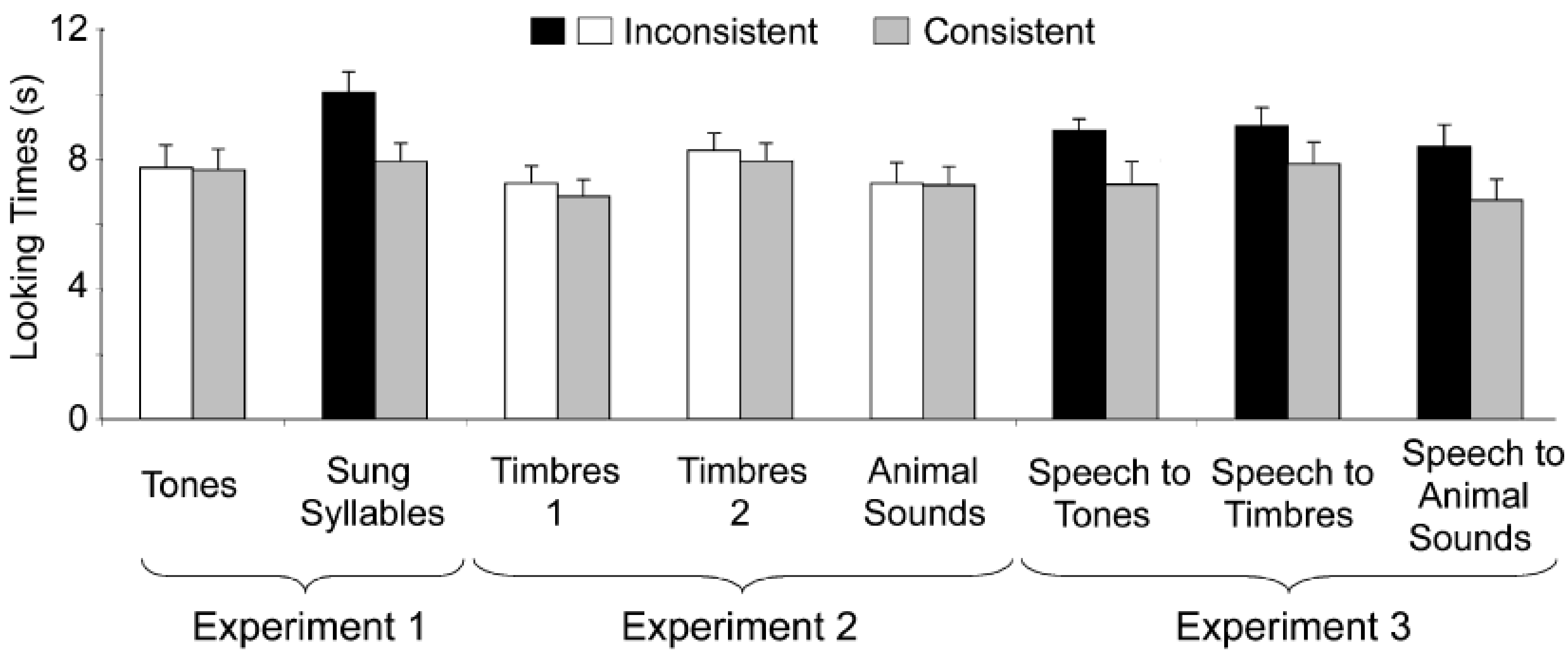
- something-same-different pattern
- Marcus claims that SRN cannot learn such patterns – we need algebraic rules
- Interestingly, this pattern cannot be represented by contextfree grammars either!
- Repetition detector as a cognitive primitive?
- Crucial issue: what makes us generalize?

	Syllable B			
Syllable A	di	je	li	we
le	leledi	leleje	leleli	lelewe
wi	wiwidi	wiwije	wiwi li	wiwiwe
ji	ji ji di	ji ji je	ji ji li	ji ji we
de	dededi	dedeje	dedeli	dedewe

Fig. 3. The design of Marcus, Vijayan, Bandi Rao, and Vishton (1999). The two sets of four words used by Gerken (2006) are highlighted in red and blue.

Language-specific 'algebraic' reasoning?

- Marcus et al. 2007, *PsychSci*



Language-specific 'algebraic' reasoning?

- Marcus et al. 2007, *PsychSci*
  - 7.5 month old children can do this only for speech stimuli; they fail on tones, pictures, timbres, animal sounds
  - Older children can do it in any domain
  - 7.5 month old succeed when first familiarized with speech stimuli

Starlings

- Gentner et al (*Nature*, 2006) showed that starlings can learn to discriminate between songs with and without ‘recursion’

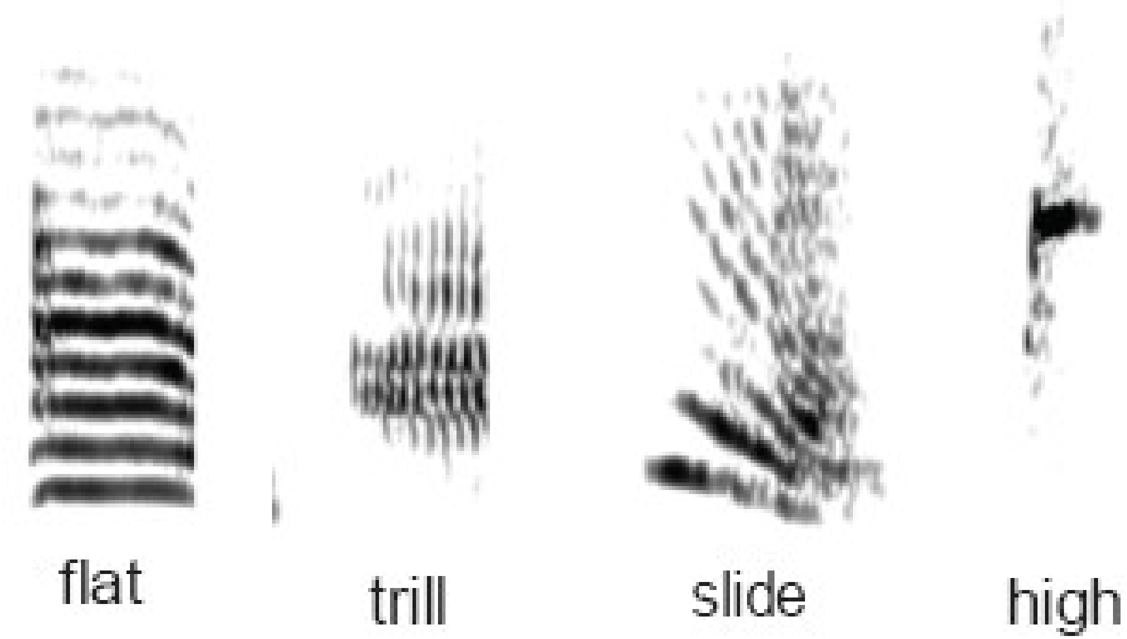
Is it really center-embedded recursion that they use?

In Leiden, we replicated the experiment with zebra finches (van Heijningen, de Visser, ten Cate, Zuidema)

(Van Heijningen, de Visser, Zuidema & ten Cate, PNAS 2009)

Can song birds learn to recognize patterns in sequences characterized by a context-free grammar?

Element types



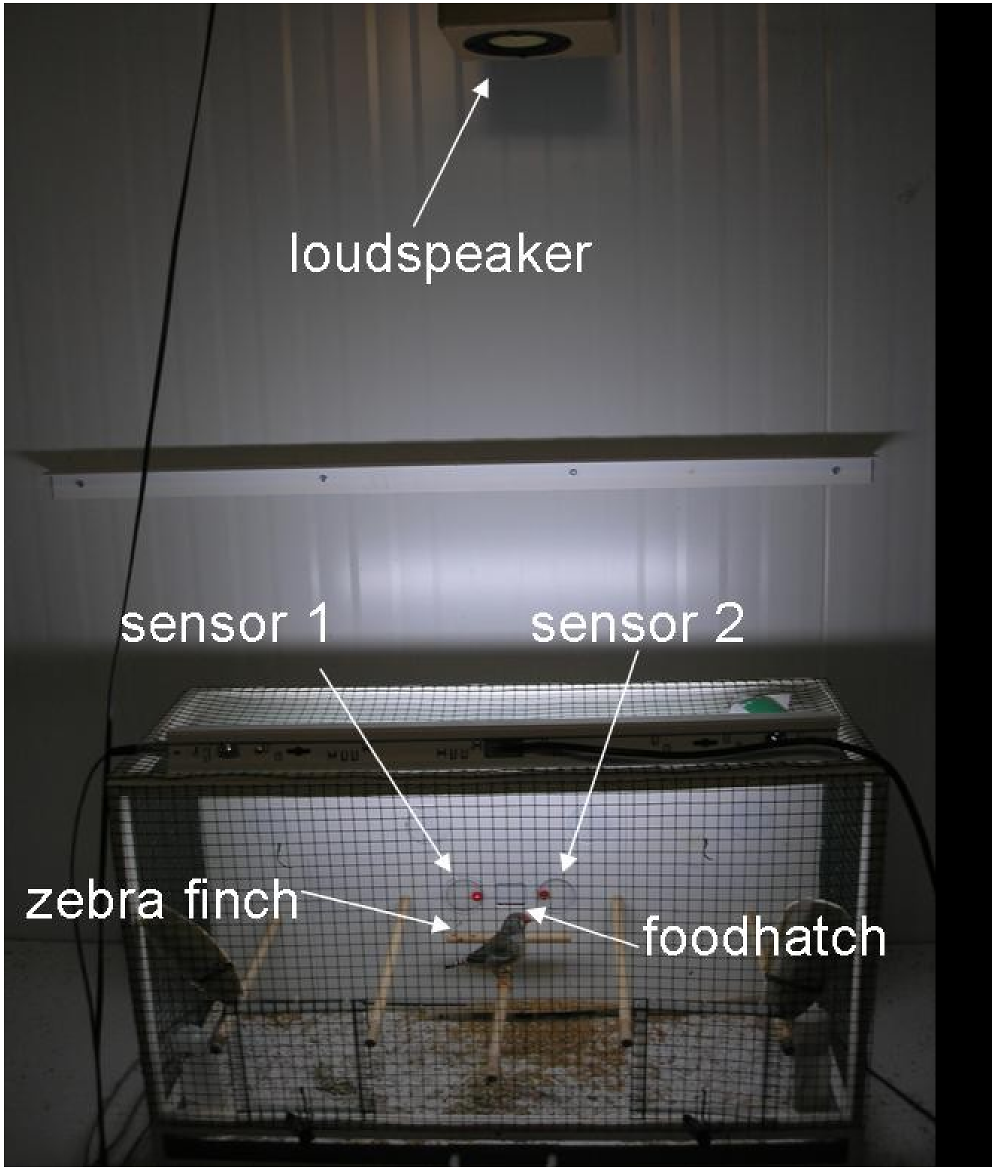
A B C D

- 4 element types
- Of each element type 10 examples
- A_1-A_{10}
- 40 elements

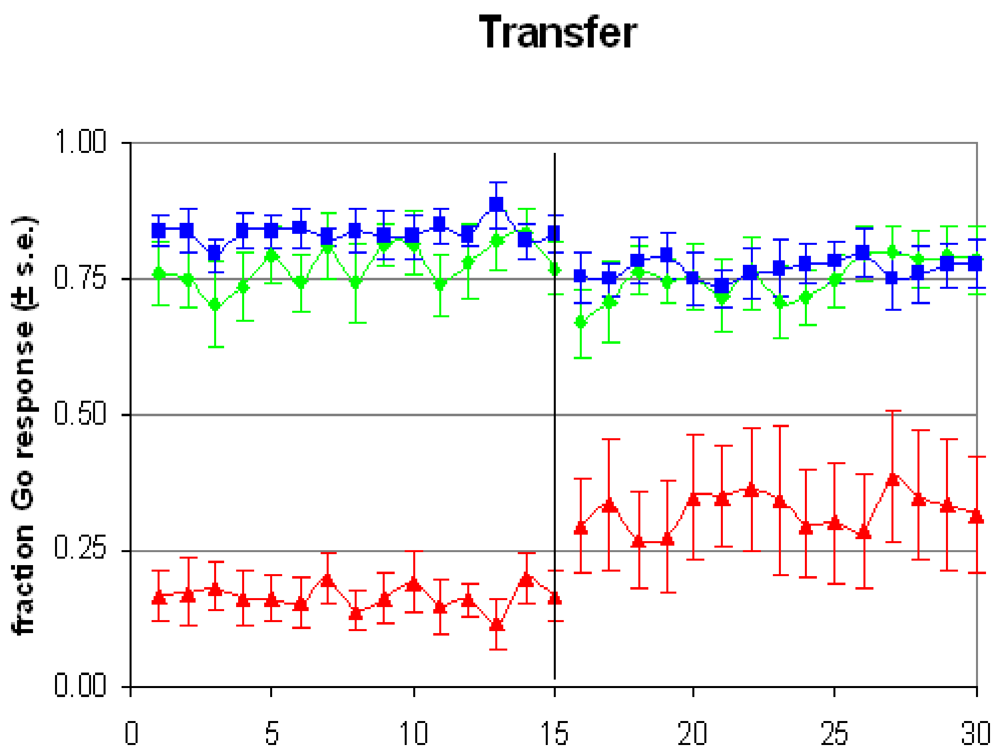
Method: Stimuli

- Finite State Grammar: ABAB
- Context Free Grammar: AABB

Method: *Skinnerbox*



Results
A and B -> other A and B



[redacted]

Average for 6 zebra finches

New element examples

A_1-A_5 ->A_6-A_{10}

B_1-B_5 ->B_6-B_{10}

Short dip, but still discrimination

video

Controls

- It is possible to distinguish between the two stimuli sets using simple strategies, e.g.:
  - Presence/absence *bigrams* AA, BB and BA
  - *Primacy rule: AB or AB at beginning, or not*
  - *Recency rule: AB or BB at end, or not*
- *Previous studies did not or not properly control for these*

Probes

- Are alternative strings (same alphabet) treated as positive or negative stimuli?

  - BAAB
  - ABBA
  - AAAA
  - BBBB
  - ABABAB
  - AAABBB

Probes

- Are alternative strings (same alphabet) treated as positive or negative stimuli?

  - BAAB -
  - ABBA -
  - AAAA -
  - BBBB +
  - ABABAB -
  - AAABBB +

Probes

- Are alternative strings (same alphabet) treated as positive or negative stimuli?

• BAAB	-	+
• ABBA	-	-
• AAAA	-	+
• BBBB	+	-
• ABABAB	-	-
• AAABBB	+	+

Conclusions

- Humans, starlings and zebra finches successfully distinguish AABB from ABAB
- Results from zebra finches show they can solve it without recourse to recursion
- Future work:
  - How do humans solve this task?
  - Where on the Chomsky Hierarchy should we place natural songs of birds?
    - Automatic identification of elements & rules