Part IV
Case studies

Fausto Carcassi

Intro & structure

We have seen how pLoT models work

This has been applied to many domains in the literature

Four examples today:

- Kinship
- Numerals
- Visual concepts
- Binary sequences

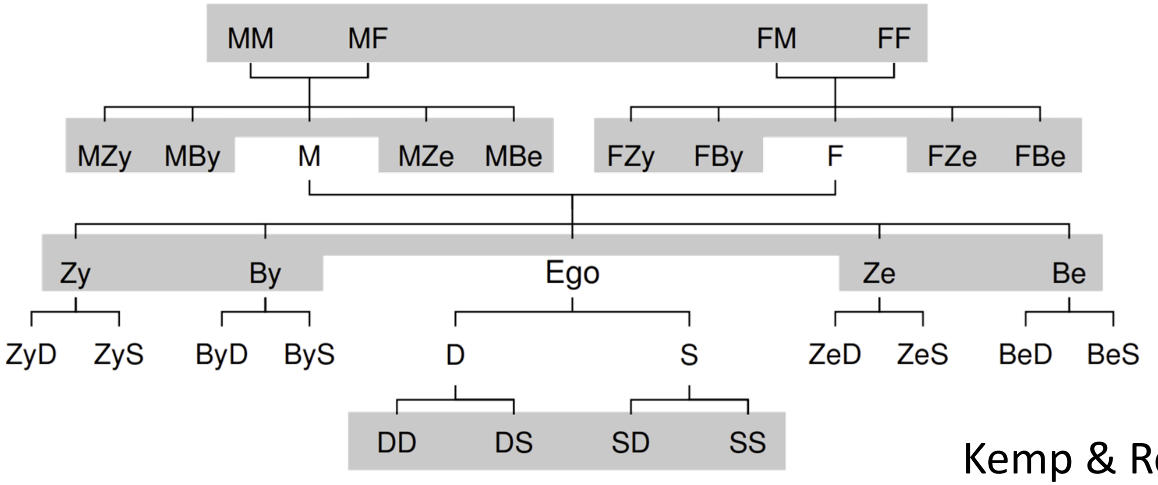
Part I	Introduction: On the very idea of an LoT
Part II	Technical background
Part III	Bayesian program induction (LOTlib3)
Part IV	**Case studies**
Part V	Summary & Future prospects

Kinship - Mollica & Piantadosi (2021)

Kinships terms

Mollica & Piantadosi (2021), *Logical word learning: The case of kinship*.

Kinship terms express relative positioning in a family



Kemp & Regier (2012)

Rich logical structure: they express complex relations.

Kinships terms – Data

A single datapoint is a collection of four objects:

- **Speaker** who uses the kinship word
- **Word** used by the speaker
- **Referent** identified by the word
- **Context** consists of a family tree

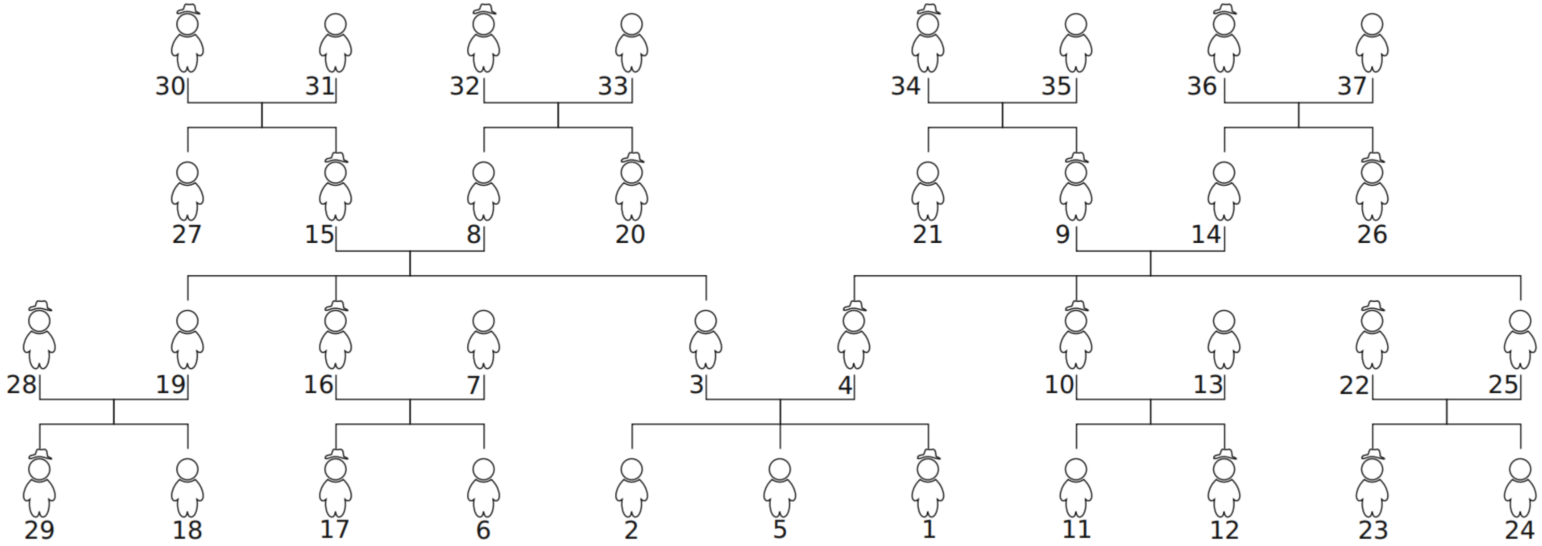
The child sees datapoints & infers the meaning of kinship terms!

Kinships terms – Hypothesis space

Hypothesis: set of people in a family from the point of view of the speaker.

- 37 possible people

Numbered by rank of number of interactions with the speaker:



PCFG & induced prior

The PCFG contains the following primitives:

$\text{SET} \xrightarrow{1} \text{union(SET,SET)}$ $\quad$ $\text{SET} \xrightarrow{1} \text{parent(SET)}$ $\quad$ $\text{SET} \xrightarrow{1} \text{generation0(SET)}$ $\quad$ $\text{SET} \xrightarrow{1} \text{male(SET)}$

$\text{SET} \xrightarrow{1} \text{intersection(SET,SET)}$ $\quad$ $\text{SET} \xrightarrow{1} \text{child(SET)}$ $\quad$ $\text{SET} \xrightarrow{1} \text{generation1(SET)}$ $\quad$ $\text{SET} \xrightarrow{1} \text{female(SET)}$

$\text{SET} \xrightarrow{1} \text{difference(SET,SET)}$ $\quad$ $\text{SET} \xrightarrow{1} \text{lateral(SET)}$ $\quad$ $\text{SET} \xrightarrow{1} \text{generation2(SET)}$ $\quad$ $\text{SET} \xrightarrow{1} \text{sameGender(SET)}$

$\text{SET} \xrightarrow{1} \text{complement(SET)}$ $\quad$ $\text{SET} \xrightarrow{1} \text{coreside(SET)}$ $\quad$ $\text{SET} \xrightarrow{\frac{1}{37}} \text{concreteReferent}$ $\quad$ $\text{SET} \xrightarrow{1} \text{all}$ $\quad$ $\text{SET} \xrightarrow{10} \text{X}$

E.g.,

English	*aunt*	PZ, PGW	female(difference(generation1(X), parent(X)))
	brother	B	male(child(parent(X)))
	cousin	PGC, PGEC	difference(generation0(X), child(parent(X)))
	father	F	male(parent(X))
	grandma	PM	female(parent(parent(X)))
	grandpa	PF	male(parent(parent(X)))
	mother	M	female(parent(X))
	sister	Z	female(child(parent(X)))
	uncle	PB, PGH	male(difference(generation1(X), parent(X)))

Likelihood function

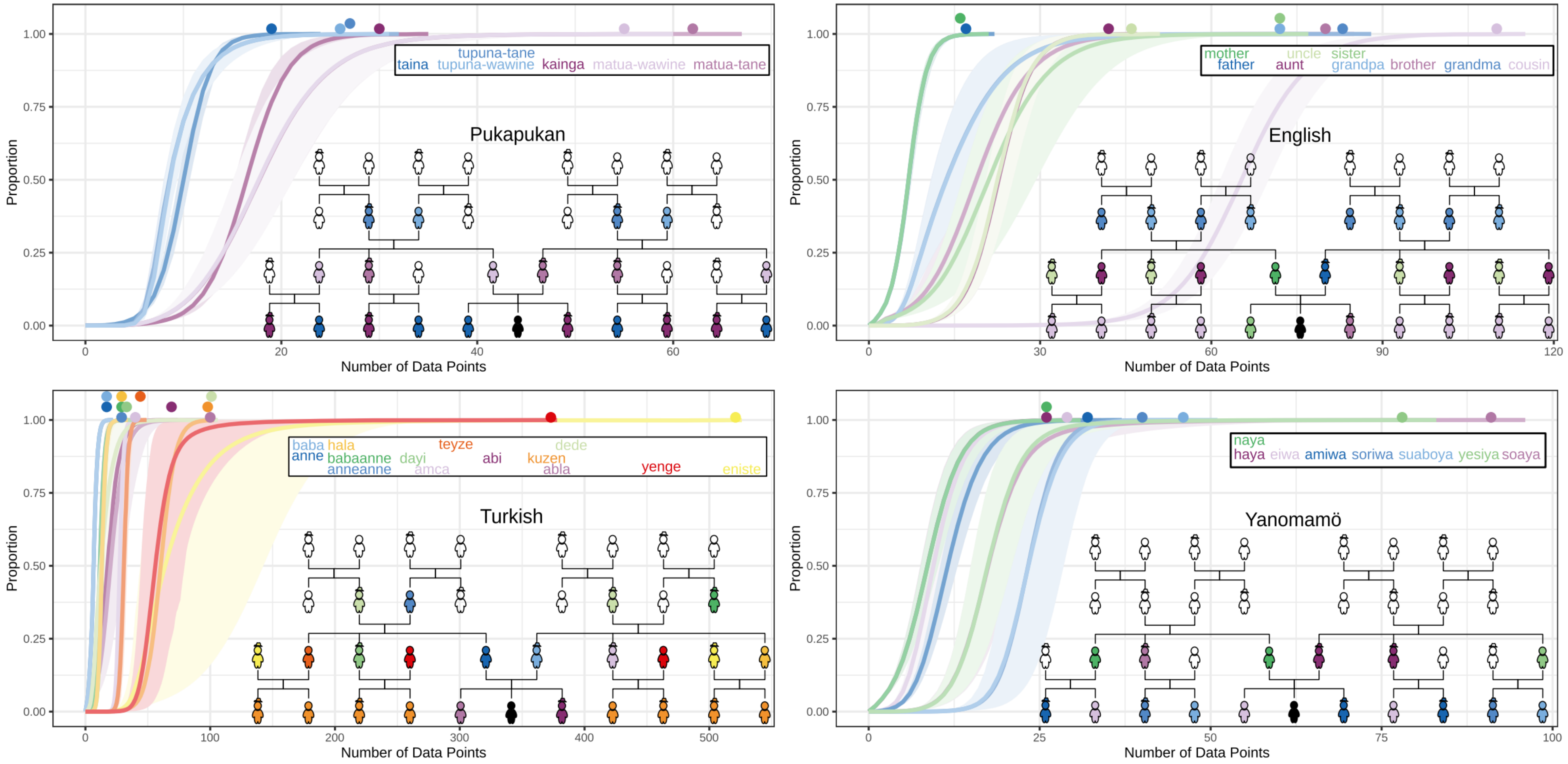
The data is generated in one of two ways:

- With probability α, hypothesis generates the datapoint
  i.e., one of the people is sampled
- With probability $1-\alpha$, data is random

Likelihood function:

$$P(d|h) = \underbrace{\delta_{d\in h}}_{\text{Whether } d \text{ belongs to } h} \cdot \frac{\alpha}{\underbrace{|h|}_{\text{Size of } h}} + \frac{1-\alpha}{\underbrace{|\mathcal{D}|}_{\text{Size of domain}}}$$

Results


Pukapukan
tupuna-tane
taina tupuna-wawine kainga matua-wawine matua-tane
English
mother uncle sister
father aunt grandpa brother grandma cousin
Turkish
baba hala teyze dede
anne babaanne dayi abi kuzen
anneanne amca abla yenge eniste
Yanomamö
naya
haya eiwa amiwa soriwa suaboya yesiya soaya
Proportion
Number of Data Points

Results

For few datapoints, preference for concrete reference (single individuals) over classes of individuals.

- This is consistent with what children do!

Predicts *overextension*

- The phenomenon where children learn a larger category that includes more individuals than the word’s true reference.

Characteristic-to-defining shift

- Young children over-extend with characteristic features (“robbers are mean”) vs defining features (“robbers steal things”).

Results

Order of acquisition in model mostly align with children:

Empirical Order	Word	Original H&C Order & Formalization	Log Prior	CHILDES Freq.
1	*mother*	Level I: [X PARENT Y][FEMALE]	-9.457	6812
1	*father*	Level I: [X PARENT Y][MALE]	-9.457	3605
2	*brother*	Level III: [X CHILD A][A PARENT Y][MALE]	-13.146	41
2	*sister*	Level III: [X CHILD A][A PARENT Y][FEMALE]	-13.146	89
3	*grandma*	Level II: [X PARENT A][A PARENT Y][FEMALE]	-13.146	526
3	*grandpa*	Level II: [X PARENT A][A PARENT Y][MALE]	-13.146	199
4	*aunt*	Level IV: [X SIB A][A PARENT Y][FEMALE]	-19.320	97
4	*uncle*	Level IV: [X SIB A][A PARENT Y][MALE]	-19.320	68
4	*cousin*	Level IV: [X CHILD A][A SIB B][B PARENT Y]	-18.627	14

Much more in paper (e.g. experimental results) but we do not have time!

Numbers - Piantadosi et al (2012)

Learning Numerals with an LoT

Piantadosi et al (2012), *Bootstrapping in a language of thought: A formal model of numerical concept learning*.

Regular patterns in number systems acquisition.

The first learn to recognize small sets of size 1, then 2, then 3, etc.

At about 3;6 they learn the full recursive system

- *Cardinal Principal learners*
- Qualitative jump rather than smooth progress!

Can pLoT reproduce qualitative conceptual jumps?

Learning Numerals with an LoT

Three choices for functions encoding meaning of number words:

- Number word to a predicate on sets n –> "Are there n?"
- Set to a number word S –> "How many are there?"
- Objects in the situation to set n –> "Construct n"

Children can do all these three things; no clear empirical evidence.

They go the second way: from a set to a number word.

Learning Numerals with an LoT

Functions mapping sets to truth values	
(*singleton? X*)	Returns true iff the set *X* has exactly one element
(*doubleton? X*)	Returns true iff the set *X* has exactly two elements
(*tripleton? X*)	Returns true iff the set *X* has exactly three elements
Functions on sets	
(*set-difference X Y*)	Returns the set that results from removing *Y* from *X*
(*union X Y*)	Returns the union of sets *X* and *Y*
(*intersection X Y*)	Returns the intersect of sets *X* and *Y*
(*select X*)	Returns a set containing a single element from *X*
Logical functions	
(*and P Q*)	Returns TRUE if *P and Q* are both true
(*or P Q*)	Returns TRUE if either *P or Q* is true
(*not P*)	Returns TRUE iff *P* is false
(*if P X Y*)	Returns *X* iff *P* is true, *Y* otherwise
Functions on the counting routine	
(*next W*)	Returns the word after *W* in the counting routine
(*prev W*)	Returns the word before *W* in the counting routine
(*equal-word? W V*)	Returns TRUE if *W* and *V* are the same word
Recursion	
(*L S*)	Returns the result of evaluating the entire current lambda expression on set *S*

$$\lambda S \cdot (\mathit{if}\,(\mathit{singleton?}\ S)\ \text{“}\mathit{one}\text{”}\ (\mathit{next}\ (L\ (\mathit{select}\ S)))).$$

Learning Numerals with an LoT

One-knower

```
λ S . (if (singleton? S)
          "one"
          undef)
```

Two-knower

```
λ S . (if (singleton? S)
          "one"
          (if (doubleton? S)
              "two"
              undef))
```

Three-knower

```
λ S . (if (singleton? S)
          "one"
          (if (doubleton? S)
              "two"
              (if (tripleton? S)
                  "three"
                  undef))
```

CP-knower

```
λ S . (if (singleton? S)
          "one"
          (next (L (set-difference S
                                   (select S)))))
```

Singular-Plural

```
λ S . (if (singleton? S)
          "one"
          "two")
```

2-not-1-knower

```
λ S . (if (doubleton? S)
          "two"
          undef)
```

Mod-5

```
λ S . (if (or (singleton? S)
              (equal-word? (L (set-difference S)
                                   (select S))
                           "five"))
          "one"
          (next (L (set-difference S
                                   (select S)))))
```

2N-knower

```
λ S . (if (singleton? S)
          "one"
          (next (next (L (set-difference S (select S))))))
```

Learning Numerals with an LoT

<u>Likelihood</u>

A set of objects is chosen from the universe of object, e.g., ‘cats’

The hypothesis is evaluated on the set.

This can result either in a number word or ‘undef’

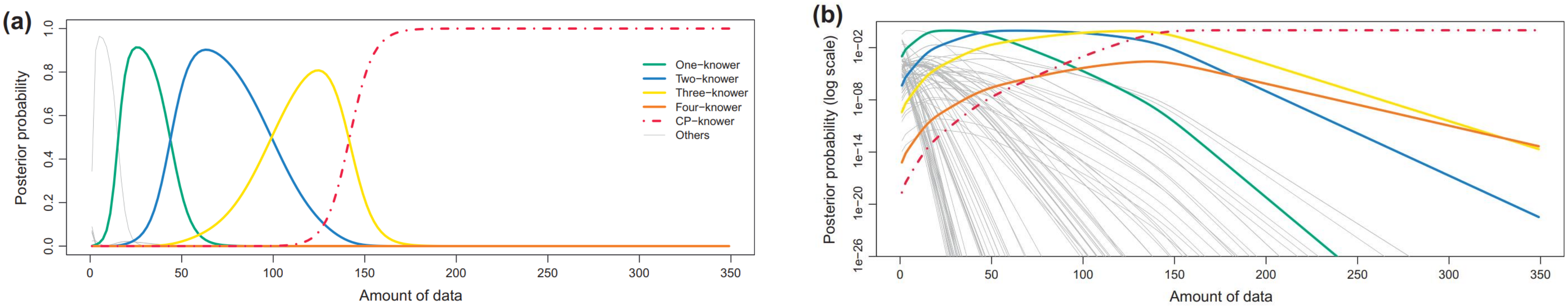
- If the result is ‘undef’, a random number word is produced
- If the result is a number word, the word is produced with probability α and with $1-\alpha$ a random word is picked.

$$P(w_i|t_i, c_i, L) = \begin{cases} \frac{1}{N} & \text{if } L \text{ yields } \textit{undef} \\ \alpha + (1-\alpha)\frac{1}{N} & \text{if } L \text{ yields } w_i \\ (1-\alpha)\frac{1}{N} & \text{if } L \text{ does not yield } w_i \end{cases}$$

Learning Numerals with an LoT

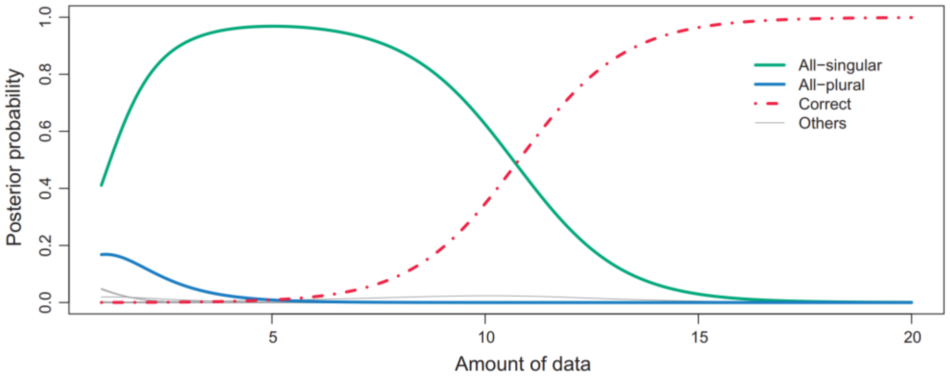
The model penalizes recursive functions with a parameter γ

Results look strikingly like human learning patterns:



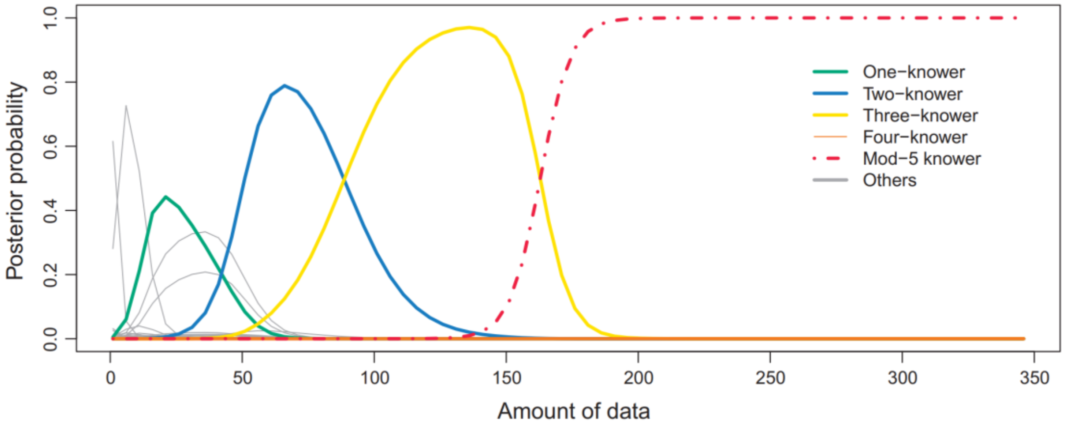
Note: many hypotheses are considered and then disregarded!

Learning Numerals with an LoT



The same LoT
on singular/plural morphology

And mod-n systems
(e.g. days of the week)

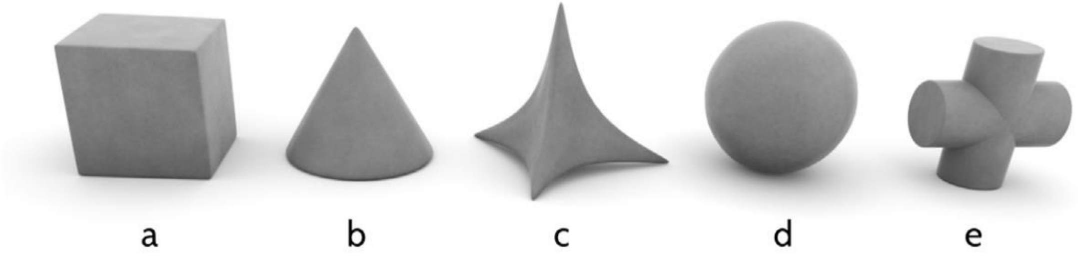


Visual concepts - Overlan et al (2017)

Visual Concept Learning in an LoT

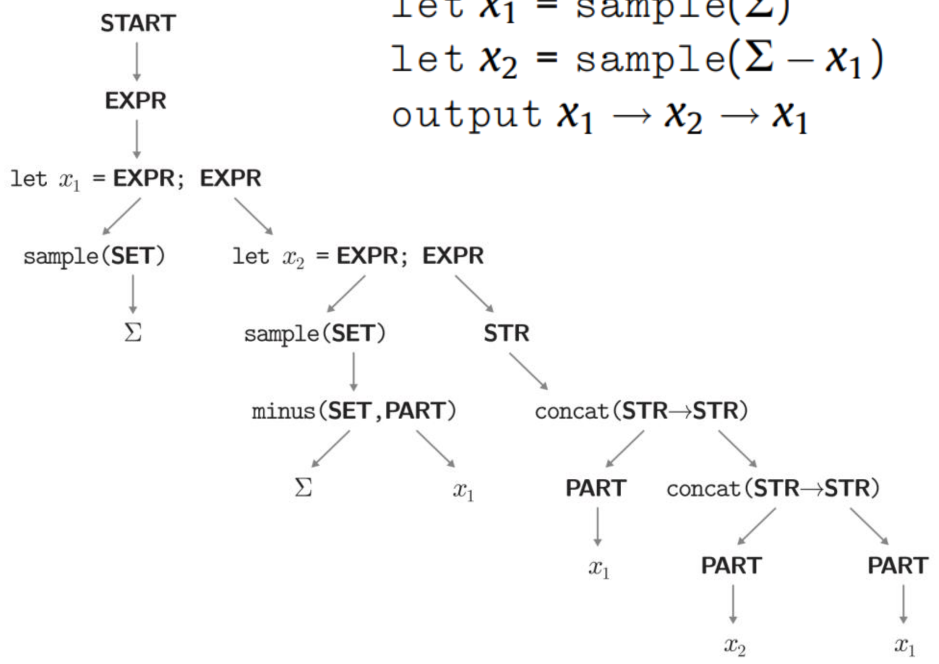
Overlan et al (2017), *Learning abstract visual concepts via probabilistic program induction in a Language of Thought*

pLoT model of 3-d objects



$$
\begin{array}{rll}
\textbf{START} & \rightarrow \texttt{let } <\!\textbf{BV\_PART}\!>: x_1 = \textbf{FIRST\_PART}; \textbf{EXPR} & \\
\textbf{EXPR} & \rightarrow \texttt{let } <\!\textbf{BV\_PART}\!>: x_n = \textbf{PART}; \textbf{EXPR} & \\
 & \rightarrow \textbf{STRING} & \\
\textbf{STRING} & \rightarrow \textbf{BV\_PART} & \\
 & \rightarrow \textbf{STRING CONNECT STRING} & \\
 & \rightarrow \{\textbf{STRING}\} & \\
\textbf{FIRST\_PART} & \rightarrow \texttt{sample}(\textbf{FIRST\_SET}) & 1 - p_{single} \\
 & \rightarrow \textbf{SINGLE} & p_{single} \\
\textbf{PART} & \rightarrow \textbf{BV\_PART} & (1 - p_{single})/2 \\
 & \rightarrow \texttt{sample}(\textbf{SET}) & (1 - p_{single})/2 \\
 & \rightarrow \textbf{SINGLE} & p_{single} \\
\textbf{FIRST\_SET} & \rightarrow \Sigma & 1 - p_{minus} \\
 & \rightarrow \texttt{minus}(\textbf{FIRST\_SET},\ \textbf{FIRST\_PART}) & p_{minus} \\
\textbf{SET} & \rightarrow \Sigma & 1 - p_{minus} \\
 & \rightarrow \texttt{minus}(\textbf{SET},\ \textbf{BV\_PART}) & p_{minus} \\
\textbf{CONNECT} & \rightarrow `\uparrow' \mid `\downarrow' \mid `\leftarrow' \mid `\rightarrow' & \\
\textbf{SINGLE} & \rightarrow `\text{a}' \mid \ldots \mid `\text{e}' &
\end{array}
$$

A derivation & some categories



$\texttt{let } x_1 = \texttt{sample}(\Sigma)$
$\texttt{let } x_2 = \texttt{sample}(\Sigma - x_1)$
$\texttt{output } x_1 \to x_2 \to x_1$

ABA

$\texttt{let } x_1 = \texttt{sample}(\Sigma_R)$
$\texttt{let } x_2 = \texttt{sample}(\Sigma_R - x_1)$
$\texttt{output } x_1 \to x_2 \to x_1$

xBB

$\texttt{let } x_1 = \texttt{sample}(\Sigma)$
$\texttt{let } x_2 = \text{‘}a\text{’}$
$\texttt{output } x_2 \to x_1 \to x_1$

ABC

$\texttt{let } x_1 = \texttt{sample}(\Sigma)$
$\texttt{let } x_2 = \texttt{sample}(\Sigma - x_1)$
$\texttt{let } x_3 = \texttt{sample}(\Sigma - x_2 - x_1)$
$\texttt{output } x_2 \to x_3 \to x_1$

Ring

$\texttt{let } x_1 = \texttt{sample}(\Sigma_R)$
$\texttt{let } x_2 = \texttt{sample}(\Sigma_R - x_1)$
$\texttt{output } x_1 \to ((x_2 \uparrow x_1) \downarrow x_1) \to x_1$

Experimental design

120 participants

<u>Instruction stage</u>

Show all possible objects

<u>Training stage</u>

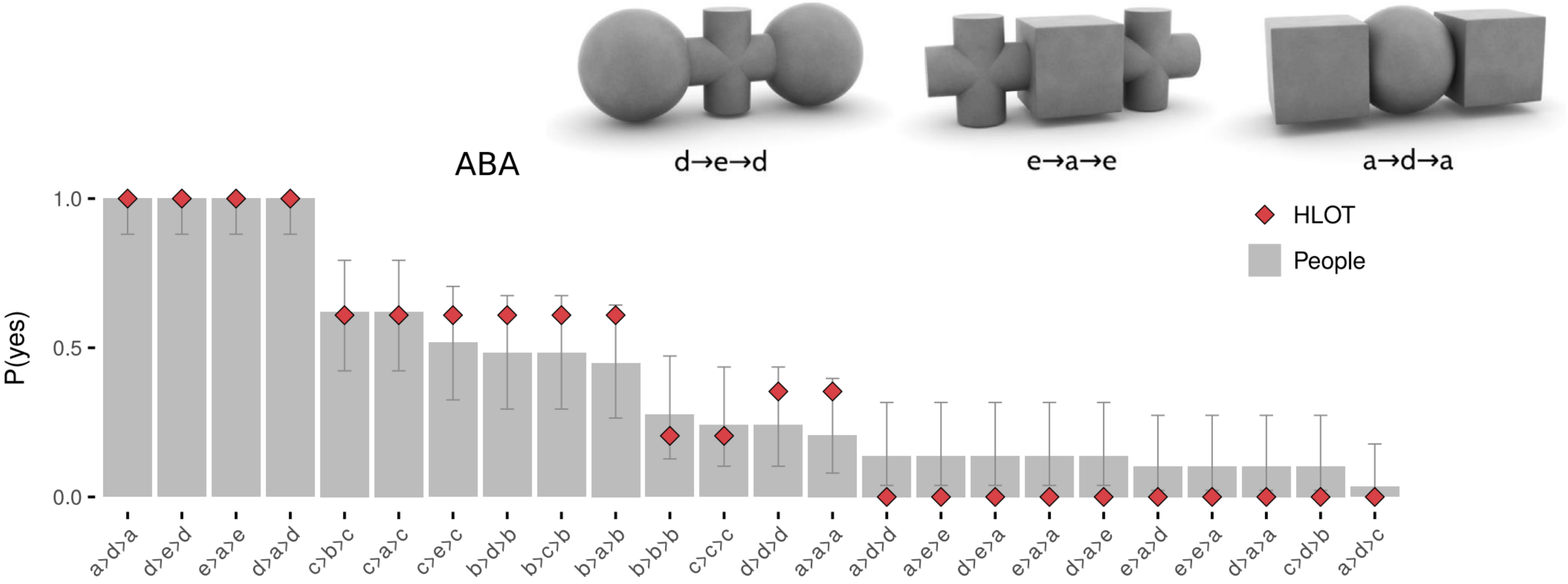
Show participants three *examples* of a concept

<u>Testing stage</u>

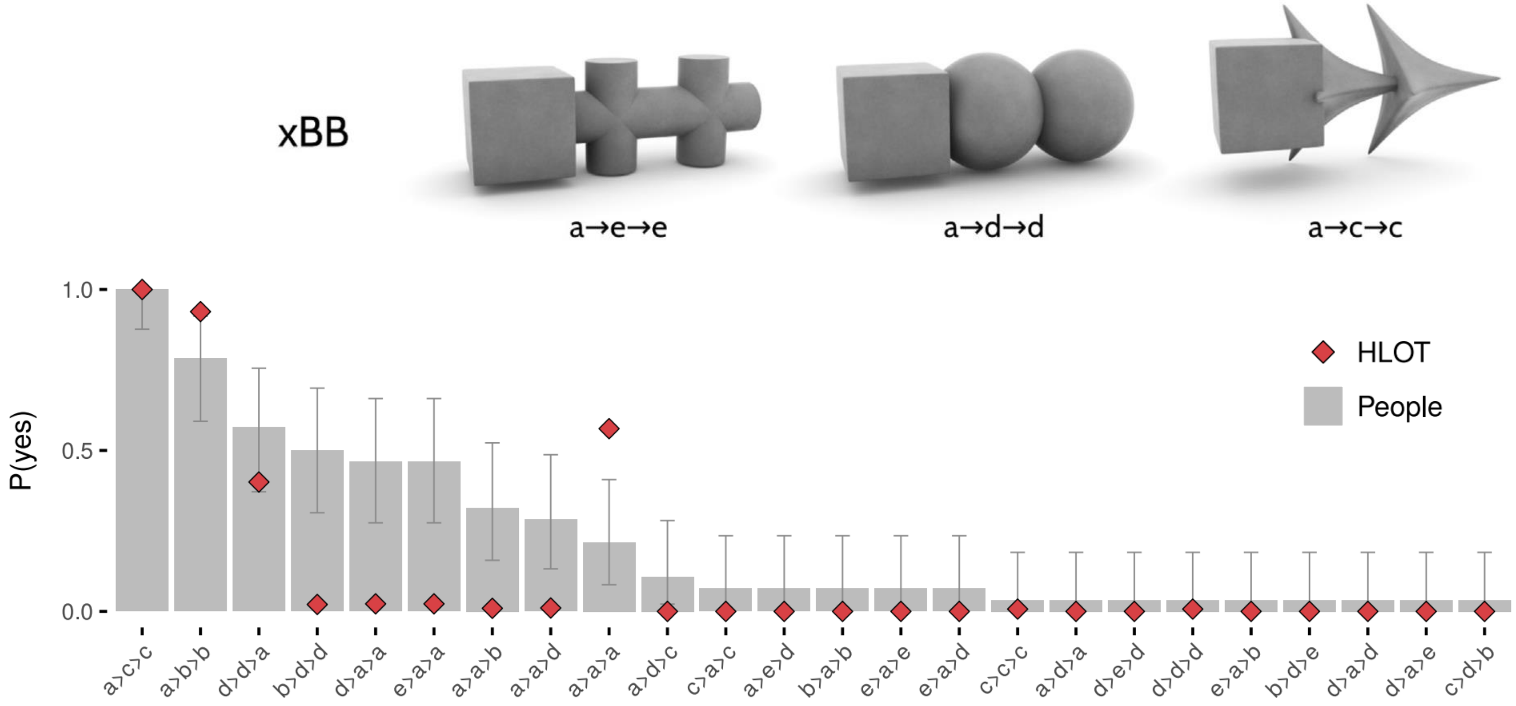
Show 24 new items

Ask whether the concept applies

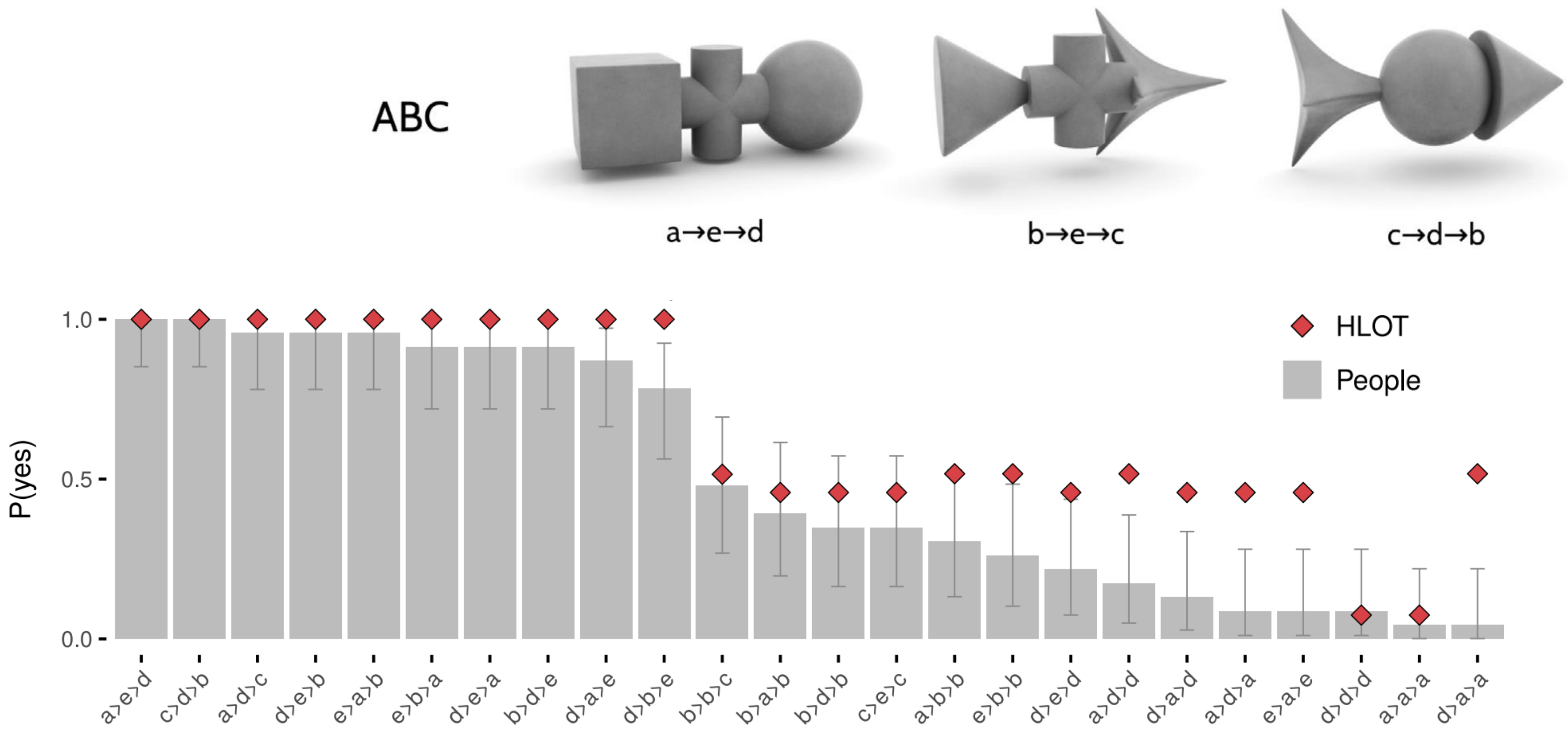
Experimental results



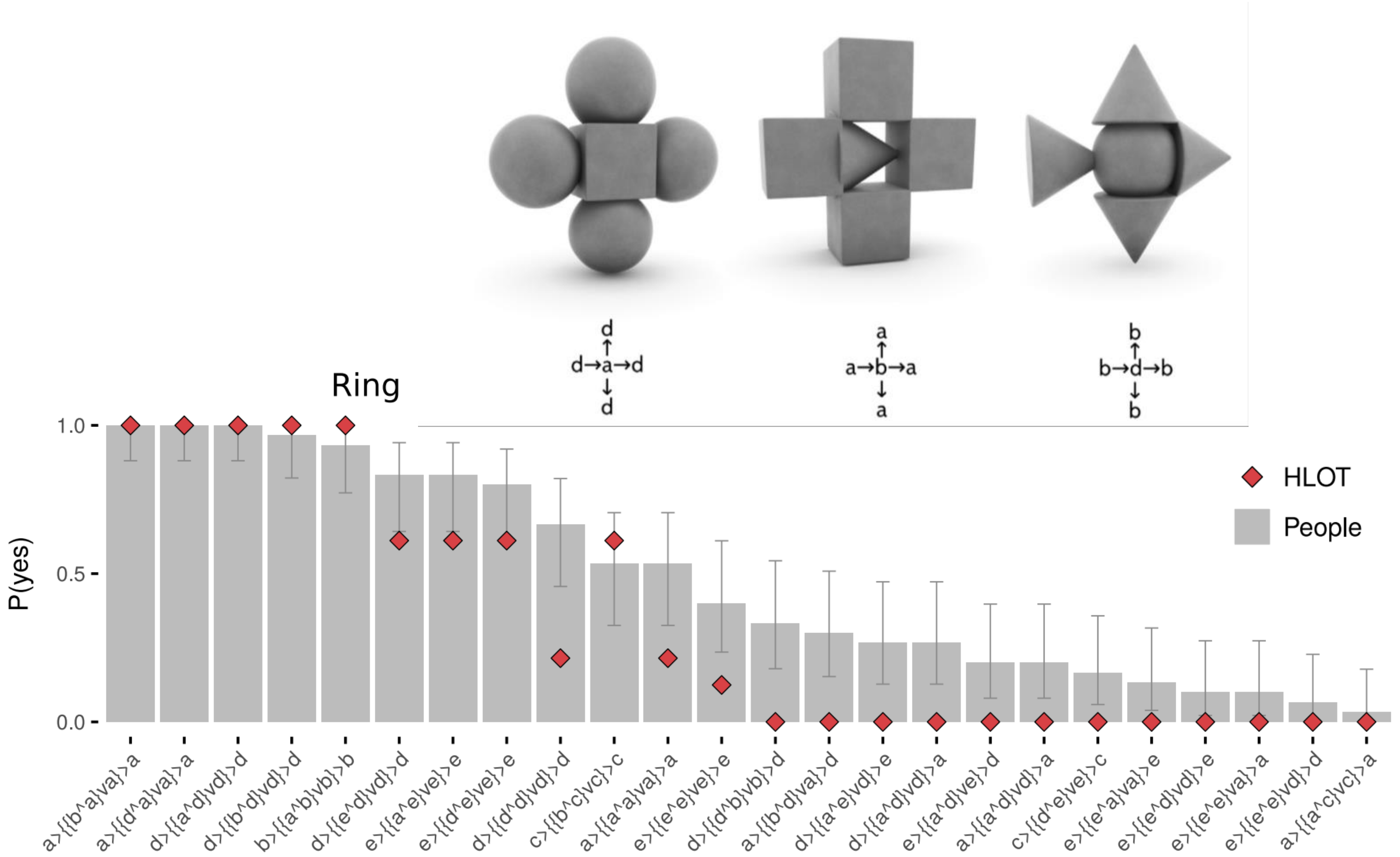
Experimental results



Experimental results



Experimental results



Sequences - Planton et al (2021)

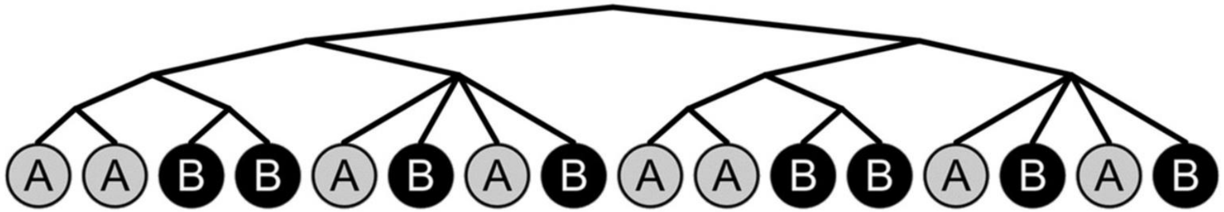
Sequence Learning in an LoT

Planton et al (2021), *A theory of memory for binary sequences: Evidence for a mental compression algorithm in humans*

New: Doesn't use natural language.

pLoT model of how humans deal with *binary sequences*

- E.g., {0,1}*
- But e.g. it can be two pitches (auditory stimuli)



Sequence Learning in an LoT

Staying (“+0”)

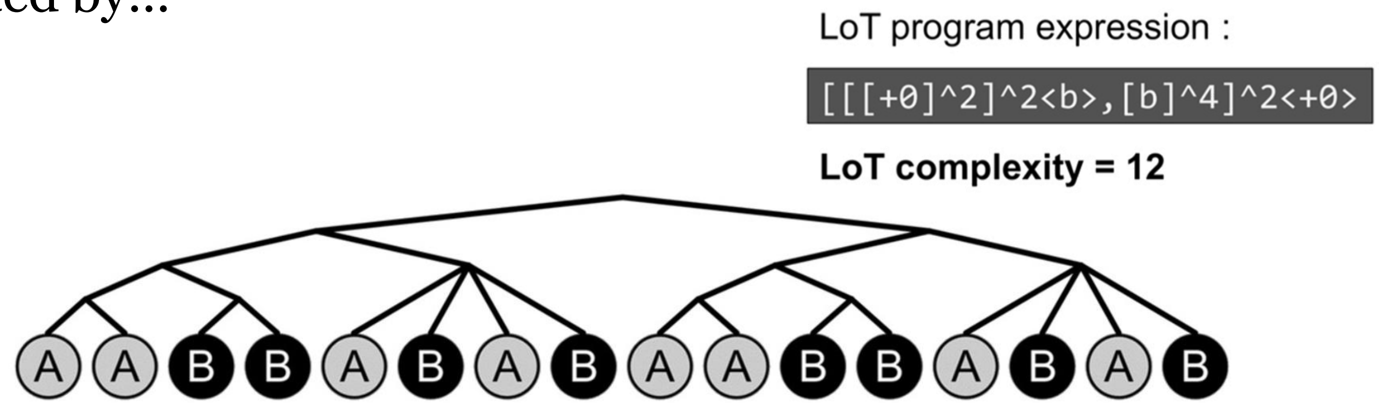
Moving to the other item (here denoted ‘b’)

Repetition (“^n”, where n is any number),

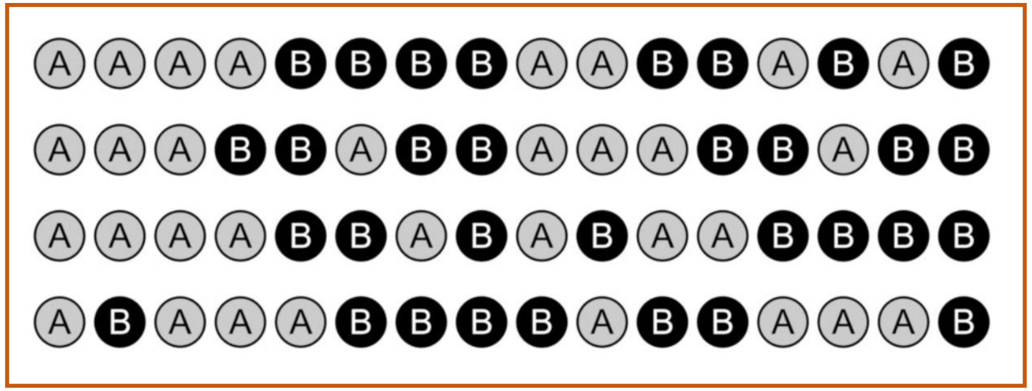
- Possibly with a variation in the starting point
- Denoted by <x> where x is an elementary instruction, either +0 or b

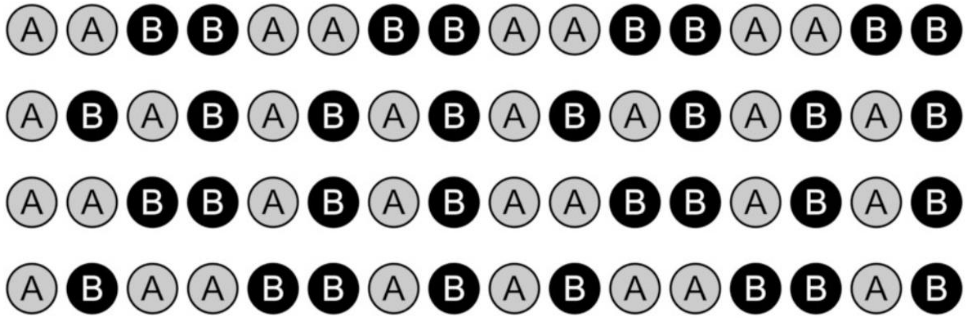
Embedding of expressions is represented by…

- brackets (“[. . ..]”)
- concatenation by commas (“,”)



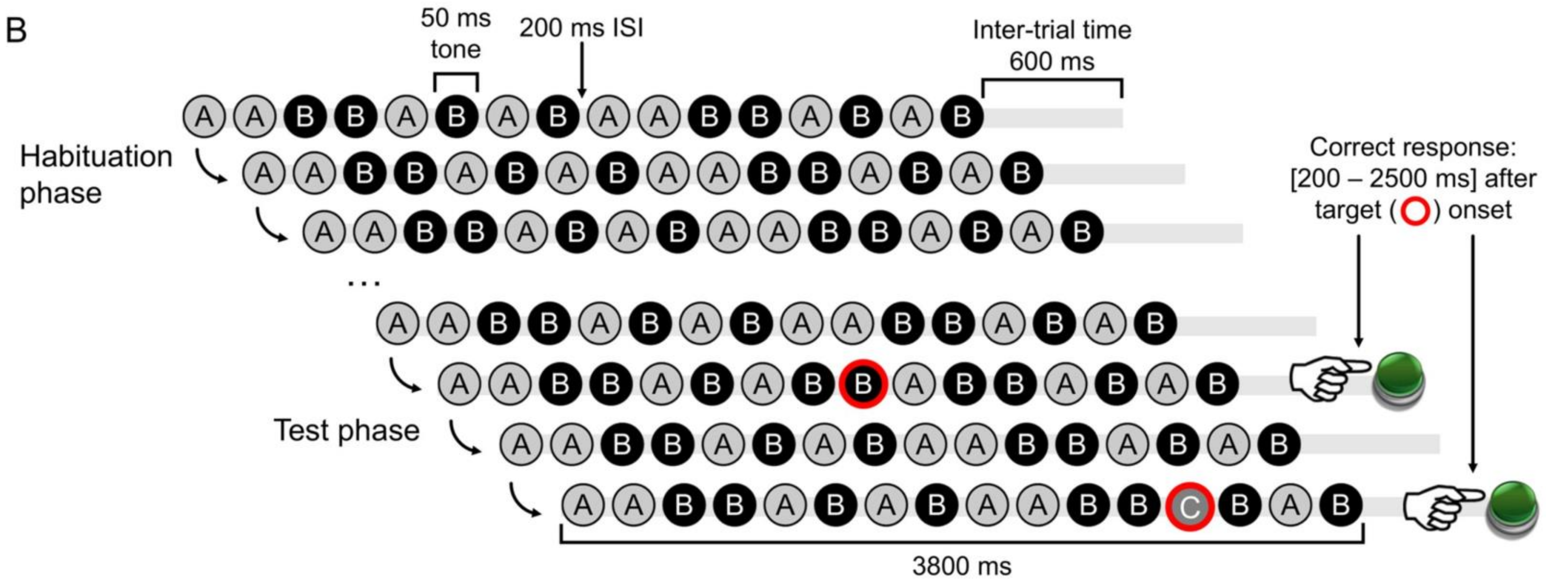
Which set has most complex sequences?





Sequence Learning in an LoT

‘Sequence violation’ experimental paradigm:



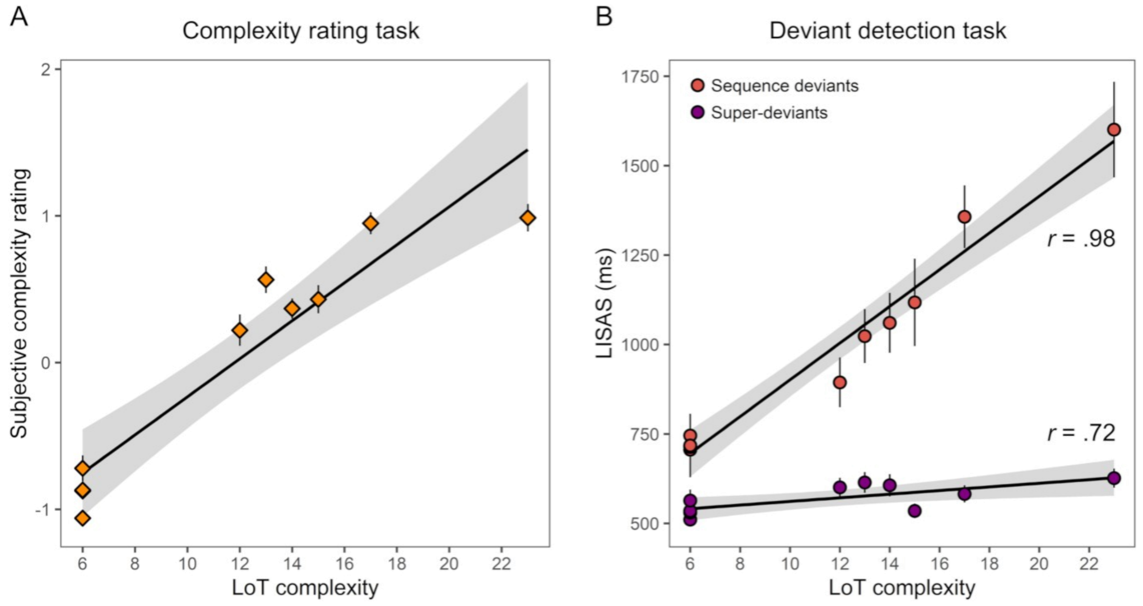
Empirical hypothesis: for **equal sequence length**, **error rate and response time** in violation detection increase with sequence complexity.

Sequence Learning in an LoT

First experiment results (16-items sequences)

Deviants identification

- Sequence deviants
    - Switched note btw A and B
- Superdeviant
    - A new note is introduced



Participants' subjective complexity evaluations (left)

LISAS Linear Integrated Speed-Accuracy Score combines response times & error scores

Taking stock

The variety of applications of the pLoT

Linguistic	Non-linguistic
Kinship	Visual concepts
Numerals	Binary sequences

The variety of applications of the pLoT

Kinship

Visual concepts

Reflected categorization

Numerals

Binary sequences

Lower-level cognition (Reaction times, errors)

The variety of applications of the pLoT

Familiar	New
Kinship	Visual concepts
Numerals	Binary sequences

The variety of applications of the pLoT

Typological

Kinship

Visual concepts

Numerals

Binary sequences

Acquisition

Experimental

The variety of applications of the pLoT

Abstract

Visual

Kinship

Numerals

Visual concepts

Binary sequences

Auditory

Intro & structure

Four examples today:

- Kinship
- Numerals
- Visual concepts
- Binary sequences

Part I	Introduction: On the very idea of an LoT
Part II	Technical background
Part III	Bayesian program induction (LOTlib3)
Part IV	**Case studies**
Part V	Summary & Future prospects