Outline

Introduction

From Level 1 to Level 1.5
   Gathering data and searching for a model
   Testing hierarchical predictions

Looking for familiar algorithms (Level 2)

Designing level 3 experiments

Discussion and Conclusions
Outline

Introduction

From Level 1 to Level 1.5
  Gathering data and searching for a model
  Testing hierarchical predictions

Looking for familiar algorithms (Level 2)

Designing level 3 experiments

Discussion and Conclusions
Goals of the talk

Discuss examples of:

1. Logics motivated by CogSci;
2. Experiments motivated by logics;
3. Selection of experimental methods.
Goals of the talk

Discuss examples of:

1. Logics motivated by CogSci;
Goals of the talk

Discuss examples of:

1. Logics motivated by CogSci;
2. Experiments motivated by logics;
Goals of the talk

Discuss examples of:

1. Logics motivated by CogSci;
2. Experiments motivated by logics;
3. Selection of experimental methods.
How can logic contribute?

1. In building cognitive theories;
How can logic contribute?

1. In building cognitive theories;
2. In computational modeling;
How can logic contribute?

1. In building cognitive theories;
2. In computational modeling;
3. In designing experiments.
Classically 3 levels of Marr

1. Computational level:
   ▶ specify cognitive task:
     ▶ f: initial state \(\rightarrow\) desired state
     ▶ problems that a cognitive ability has to overcome

Marr, Vision: a computational investigation into the human representation and processing visual information, 1983
Classically 3 levels of Marr

1. Computational level:
   - specify cognitive task:
     - $f$: initial state $\rightarrow$ desired state
     - problems that a cognitive ability has to overcome

2. Algorithmic level:
   - the algorithms that may be used to achieve a solution
   - compute $f$
Classically 3 levels of Marr

1. Computational level:
   ▶ specify cognitive task:
      ▶ f: initial state → desired state
      ▶ problems that a cognitive ability has to overcome

2. Algorithmic level:
   ▶ the algorithms that may be used to achieve a solution
   ▶ compute f

3. Implementation level:
   ▶ how this is actually done in neural activity

Marr, Vision: a computational investigation into the human representation and processing visual information, 1983
Observation

 Logical analysis informs about intrinsic properties of a problem.
Observation

*Logical analysis informs about intrinsic properties of a problem.*
Observation
Logical analysis informs about intrinsic properties of a problem.

←→ Level 1.5
Outline

Introduction

From Level 1 to Level 1.5
  Gathering data and searching for a model
  Testing hierarchical predictions

Looking for familiar algorithms (Level 2)

Designing level 3 experiments

Discussion and Conclusions
Outline

Introduction

From Level 1 to Level 1.5
  Gathering data and searching for a model
  Testing hierarchical predictions

Looking for familiar algorithms (Level 2)

Designing level 3 experiments

Discussion and Conclusions
The game consists of:

- a decoding board;
- code pegs of $n$ colours;
- key (feedback) pegs (black and white).

Players:

- The codemaker: chooses a secret pattern.
- The codebreaker: guesses the pattern.
Mastermind

Moves:

- Each guess: placing a row of code pegs.
- The codemaker provides feedback.
  
  - Black key for each code peg of correct color and position.
  - White key for each peg of correct color but wrong position.

- After that another guess is made.

Winning conditions for $k$ rounds:

- The codebreaker: obtains the solution within $k$ rounds.
- The codemaker: otherwise.
Mastermind: an inductive inquiry game

- Trials of experimentation and evaluation.
- Interactive game.
- How to transform it into a reasoning task?
MM Setting in Rekentuin
MM Setting in Rekentuin
MM in Rekentuin

- massive data bank (over 150 schools in The Netherlands);
- the next step: a logical reasoning system;
- perhaps similar to the one for syllogisms.

Gierasimczuk et al., Static Mastermind in Rekentuin. A computational, logical, and cognitive perspective, under construction
Syllogistic Reasoning: Meta-data analysis

Chater and Oaksford, The probability heuristic model of syllogistic reasoning, Cognitive Psychology, 1999

<table>
<thead>
<tr>
<th>premisses &amp; figure</th>
<th>conclusion</th>
<th>premisses &amp; figure</th>
<th>conclusion</th>
<th>premisses &amp; figure</th>
<th>conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA1 90 5 0 0</td>
<td>A</td>
<td>OA1 1 6 1 57</td>
<td>E</td>
<td>IA1 0 72 0 6</td>
<td>I</td>
</tr>
<tr>
<td>AA2 58 8 1 1</td>
<td>A</td>
<td>AO2 0 6 3 67</td>
<td>E</td>
<td>IA2 13 4 3 12</td>
<td>I</td>
</tr>
<tr>
<td>AA3 57 29 0 0</td>
<td>A</td>
<td>AO3 0 10 0 66</td>
<td>E</td>
<td>IA3 2 85 1 4</td>
<td>I</td>
</tr>
<tr>
<td>AA4 75 16 1 1</td>
<td>A</td>
<td>AO4 0 5 3 72</td>
<td>E</td>
<td>IA4 0 91 1 1</td>
<td>I</td>
</tr>
<tr>
<td>AI1 0 92 3 3</td>
<td>O</td>
<td>OA1 0 3 3 68</td>
<td>E</td>
<td>IA1 0 72 0 6</td>
<td>I</td>
</tr>
<tr>
<td>AI2 0 57 3 11</td>
<td>O</td>
<td>OA2 0 11 5 56</td>
<td>E</td>
<td>IA2 13 4 3 12</td>
<td>I</td>
</tr>
<tr>
<td>AI3 1 89 1 3</td>
<td>O</td>
<td>OA3 0 15 3 69</td>
<td>E</td>
<td>IA3 2 85 1 4</td>
<td>I</td>
</tr>
<tr>
<td>AI4 0 71 0 1</td>
<td>O</td>
<td>OA4 1 3 6 27</td>
<td>E</td>
<td>IA4 0 91 1 1</td>
<td>I</td>
</tr>
</tbody>
</table>

| A = all           | E = no     |
| 1 = some           | O = some … not |

---

* All figures have been rounded to the nearest integer; valid conclusions are shaded. Whenever two conclusions in the same row are valid, only the first one is valid in predicate logic.
Monotonicity calculus

- Logic rendering many valid arguments.
- Including syllogistic.
- Pivoting on monotonicity, e.g.,
Monotonicity calculus

- Logic rendering many valid arguments.
- Including syllogistic.
- Pivoting on monotonicity, e.g.,

<table>
<thead>
<tr>
<th>Rule 1</th>
<th>Rule 2</th>
<th>Example 1</th>
<th>Example 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha \implies \beta$</td>
<td>$\beta \implies \alpha$</td>
<td>$\text{all}(A, B)$</td>
<td>$\text{all}(C, B)$</td>
</tr>
<tr>
<td>$\ldots \alpha^+ \ldots$</td>
<td>$\ldots \alpha^- \ldots$</td>
<td>$\text{some}(A^+, C)$</td>
<td>$\text{no}(B^-, A)$</td>
</tr>
<tr>
<td>$\ldots \beta^+ \ldots$</td>
<td>$\ldots \beta^- \ldots$</td>
<td>$\text{some}(B^+, C)$</td>
<td>$\text{no}(C^-, A)$</td>
</tr>
</tbody>
</table>
Monotonicity calculus

- Logic rendering many valid arguments.
- Including syllogistic.
- Pivoting on monotonicity, e.g.,

<table>
<thead>
<tr>
<th>Rule 1</th>
<th>Rule 2</th>
<th>Example 1</th>
<th>Example 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha \implies \beta$</td>
<td>$\beta \implies \alpha$</td>
<td>$all(A, B)$</td>
<td>$all(C, B)$</td>
</tr>
<tr>
<td>$\ldots \alpha^+ \ldots$</td>
<td>$\ldots \alpha^- \ldots$</td>
<td>$\text{some}(A^+, C)$</td>
<td>$\text{no}(B^-, A)$</td>
</tr>
<tr>
<td>$\ldots \beta^+ \ldots$</td>
<td>$\ldots \beta^- \ldots$</td>
<td>$\text{some}(B^+, C)$</td>
<td>$\text{no}(C^-, A)$</td>
</tr>
</tbody>
</table>

Conversion

<table>
<thead>
<tr>
<th>$Q(A, B)$</th>
<th>No/All-not</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q(B, A)$, $Q = \text{some}$</td>
<td>$all(A, \text{not } B)$</td>
</tr>
</tbody>
</table>
Example

1. \textit{no}(B, C) premiss
2. \textit{some}(B, A) premiss
Example

1. $no(B, C)$ premiss
2. $some(B, A)$ premiss
3. $some(A, B^+)$ Conv from 2
Example

1. $\textit{no}(B, C)$ premiss
2. $\textit{some}(B, A)$ premiss
3. $\textit{some}(A, B^+)$ Conv from 2
4. $\textit{all}(B, \text{not } C)$ No/All-not from 1
Example

1. _no_\((B, C)\) premiss
2. _some_\((B, A)\) premiss
3. _some_\((A, B^+)\) Conv from 2
4. _all_\((B, not\ C)\) No/All-not from 1
5. _some_\((A, not\ C)\) Mon from 3 and 4
1. The shorter the proof the easier the syllogism.
1. The shorter the proof the easier the syllogism.
   - Level 1.5,
Processing model: example

1. The shorter the proof the easier the syllogism.
   ▶ Level 1.5,
   ▶ Rule application may be empirically weighted,
1. The shorter the proof the easier the syllogism.
   - Level 1.5,
   - Rule application may be empirically weighted,

2. It gives a good fit with data.

Table 4
Predicted difficulty of valid syllogisms according to the model described in the text, compared with Chater and Oaksford’s scores (in parentheses)

<table>
<thead>
<tr>
<th>Type</th>
<th>Predicted</th>
<th>Chater</th>
<th>Oaksford</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA1A</td>
<td>80</td>
<td>(90)</td>
<td></td>
</tr>
<tr>
<td>EA1E</td>
<td>80</td>
<td>(87)</td>
<td></td>
</tr>
<tr>
<td>EA2E</td>
<td>80</td>
<td>(89)</td>
<td></td>
</tr>
<tr>
<td>AE2E</td>
<td>80</td>
<td>(88)</td>
<td></td>
</tr>
<tr>
<td>AE4E</td>
<td>80</td>
<td>(87)</td>
<td></td>
</tr>
<tr>
<td>IA3I</td>
<td>80</td>
<td>(85)</td>
<td></td>
</tr>
<tr>
<td>IA4I</td>
<td>80</td>
<td>(91)</td>
<td></td>
</tr>
<tr>
<td>AI1I</td>
<td>80</td>
<td>(92)</td>
<td></td>
</tr>
<tr>
<td>AI3I</td>
<td>80</td>
<td>(89)</td>
<td></td>
</tr>
</tbody>
</table>

Geurts, Reasoning with quantifiers, Cognition, 2003
Outline

Introduction

From Level 1 to Level 1.5
   Gathering data and searching for a model
   Testing hierarchical predictions

Looking for familiar algorithms (Level 2)

Designing level 3 experiments

Discussion and Conclusions
## Complexity of quantifiers

<table>
<thead>
<tr>
<th>Definability</th>
<th>Examples</th>
<th>Recognized by</th>
</tr>
</thead>
<tbody>
<tr>
<td>FO</td>
<td>“all”, “at least 3”</td>
<td>acyclic FA</td>
</tr>
<tr>
<td>$\text{FO}(D_n)$</td>
<td>“an even number”</td>
<td>FA</td>
</tr>
<tr>
<td>PrA</td>
<td>“most”, “less than half”</td>
<td>PDA</td>
</tr>
</tbody>
</table>

Quantifiers, definability, and complexity of automata

Mostowski, Computational semantics for monadic quantifiers, 1998.
All flowers are blue.

\[ q_0 \xrightarrow{\text{not-b}} q_1 \]

At least 3 flowers are blue.

Most of the flowers are blue.
All flowers are blue.

At least 3 flowers are blue.
All flowers are blue.

At least 3 flowers are blue.

Most of the flowers are blue.
Does it say anything about processing?

Question

*Do minimal automata predict differences in verification?*
Complexity and reaction time

Complexity and working memory

- Compare performance of:
  
  - Healthy subjects.
  
  - Patients with schizophrenia.
  
  - Known working memory deficits.
Complexity and working memory

- Compare performance of:
  - Healthy subjects.
Complexity and working memory

- Compare performance of:
  - Healthy subjects.
  - Patients with schizophrenia.
Complexity and working memory

- Compare performance of:
  - Healthy subjects.
  - Patients with schizophrenia.
    - Known working memory deficits.
RT data

![Bar chart showing data for Patients and Control groups across Aristotelian, Numerical, Parity, and Proportional categories.](chart.png)
Accuracy data

Zajenkowski et al., A computational approach to quantifiers as an explanation for some language impairments in schizophrenia, under review.
Tractability/Intractability:
 extending difficulty/complexity analogy

1. Most villagers and most townsmen hate each other.
2. All/Most of the dots are connected to each other.

**Conjecture**

*Subjects avoid intractable interpretations.*

Gierasimczuk and Szymanik, Branching quantification vs. two-way quantification, Journal of Semantics, 2009


Bott et al., Interpreting Tractable versus Intractable Reciprocal Sentences, Proceedings of the International Conference on Computational Semantics, 2011.
Outline

Introduction

From Level 1 to Level 1.5
   Gathering data and searching for a model
   Testing hierarchical predictions

Looking for familiar algorithms (Level 2)

Designing level 3 experiments

Discussion and Conclusions
Higher-order social reasoning

Marble Drop Game
Question

- Subjects are good in second-order reasonings (Mean Acc = 0.91; Mean RT = 7.8).
- And they even get better with training.
Question

Subjects are good in second-order reasonings (Mean Acc = 0.91; Mean RT = 7.8).
And they even get better with training.

Question

*How are they doing it? Do they apply backward induction?*
Question

Subjects are good in second-order reasonings (Mean Acc = 0.91; Mean RT = 7.8).
And they even get better with training.

Question
How are they doing it? Do they apply backward induction?

Question
How can we try to answer the question?
Method

- Registering subjects’ behavior.
Method

- Registering subjects’ behavior.
- Tracking eye fixations.
Method

- Registering subjects’ behavior.
- Tracking eye fixations.
- Using BI suggests fixation pattern:
  - Bin 4, 3, next Bin 2, finally Bin 1.
Method

- Registering subjects’ behavior.
- Tracking eye fixations.
- Using BI suggests fixation pattern:
  - Bin 4, 3, next Bin 2, finally Bin 1.
- Area of Interests pattern for BI:
  - 4321
  - 3421
Results

Data consistent with AOIs: 1234 against BI hypothesis!

Meijering et al., Context facilitates theory of mind: What eye movements tell about higher-order strategic reasoning, 2011.
Outline

Introduction

From Level 1 to Level 1.5
  Gathering data and searching for a model
  Testing hierarchical predictions

Looking for familiar algorithms (Level 2)

Designing level 3 experiments

Discussion and Conclusions
Differences in brain activity.

McMillan et al. fMRI studies

Szymanik, A Note on some neuroimaging study of natural language quantifiers comprehension, Neuropsychologia, 2007

Szymanik & Zajenkowski, Quantifiers and working memory, LNCS, 2010
McMillan et al. fMRI studies

Differences in brain activity.

- All quantifiers are associated with numerosity: recruit right inferior parietal cortex.
- Only higher-order activate working-memory capacity: recruit right dorsolateral prefrontal cortex.
Differences in brain activity.

- All quantifiers are associated with numerosity: recruit right inferior parietal cortex.
- Only higher-order activate working-memory capacity: recruit right dorsolateral prefrontal cortex.

McMillan et al., Neural basis for generalized quantifiers comprehension, 2005
Szymanik, A Note on some neuroimaging study of natural language quantifiers comprehension, Neuropsychologia, 2007
Szymanik & Zajenkowski, Quantifiers and working memory, LNCS, 2010
Outline

Introduction

From Level 1 to Level 1.5
  Gathering data and searching for a model
  Testing hierarchical predictions

Looking for familiar algorithms (Level 2)

Designing level 3 experiments

Discussion and Conclusions
To sum up

Level 1.5  Mastermind, Syllogisms, Verification
Level 2  Marble Drop Game
Level 3  Quantifiers and Definability
Discussion

- Adequacy of Marr’s Levels.
- Idealized logical agents.
- How to measure difficulty?
- Logic & CogSci: can the benefits be mutual?
धन्यवाद
Static Mastermind (Chvatal 1983)

▶ finding the minimum number of guesses the codebreaker can make all at once at the beginning of the game;
▶ without waiting for the answers;
▶ and upon receiving the answers;
▶ completely determine the code in the next guess.

Observation (Greenwell 1999)
Static Mastermind ($n = 6, \ell = 4$) can be solved with six initial guesses. In particular: $(1, 2, 2, 1), (2, 3, 5, 4), (3, 3, 1, 1), (4, 5, 2, 4), (5, 6, 5, 6), (6, 6, 4, 3)$.

Conjecture
It is not possible to reduce to five (exhaustive check: approx $3.7 \times 10^{15}$ computations).
Mastermind (satisfiability) decision problem:

**Input** A set of guesses $G$ and their corresponding scores.

**Question** Is there at least one valid solution?

**Theorem**

*Mastermind Problem in NP-complete wrt $\ell$ (positions).*

**Objective computational measure!**
Monotonicity profiles determine difficulty

1. Some of the sopranos sang with more than three of the tenors.
2. None of the sopranos sang with fewer than three of the tenors.
3. Some of the sopranos sang with fewer than three of the tenors.
Monotonicity profiles determine difficulty

1. Some of the sopranos sang with more than three of the tenors.
2. None of the sopranos sang with fewer than three of the tenors.
3. Some of the sopranos sang with fewer than three of the tenors.

\[
Q_1 \text{A played against } Q_2 \text{B} \\
\downarrow \text{All B were C.} \\
\uparrow \text{Q}_1 \text{A played against } Q_2 \text{C}
\]
Monotonicity profiles determine difficulty

1. Some of the sopranos sang with more than three of the tenors.
2. None of the sopranos sang with fewer than three of the tenors.
3. Some of the sopranos sang with fewer than three of the tenors.

\[ Q_1 A \text{ played against } Q_2 B \]
\[ \text{All } B \text{ were } C. \]
\[ Q_1 A \text{ played against } Q_2 C \]

\[ \uparrow Q_1 \uparrow Q_2 \downarrow \downarrow Q_1 \downarrow Q_2 < \]
\[ \uparrow Q_1 \downarrow Q_2 \]
\[ \downarrow Q_1 \uparrow Q_2 \]

Geurts and Van der Slik, Monotonicity and Processing Load, Journal of Semantics, 2005
Discussion

Conclusion

*Automata model is psychologically plausible.*
Discussion

Conclusion
Automata model is psychologically plausible.

Conclusion
Computational complexity $\approx$ cognitive difficulty.
Discussion

Conclusion
Automata model is psychologically plausible.

Conclusion
Computational complexity $\approx$ cognitive difficulty.

- As far as we know this is the first empirical proof.
Discussion

Conclusion
Automata model is psychologically plausible.

Conclusion
Computational complexity $\approx$ cognitive difficulty.

- As far as we know this is the first empirical proof.
- Between Marr’s level 1 and 2.
P-Cognition Thesis

Hypothesis

*Human cognitive (linguistic) capacities are constrained by polynomial time computability.*

Hintikka’s branching reading

- Most girls and most boys hate each other.

\[
\exists A \exists A’[\text{most}(G, A) \land \text{most}(B, A’) \land \forall x \in A \forall y \in A’ H(x, y)].
\]
Most girls and most boys hate each other.
Branching readings are intractable

**Theorem**

*Proportional branching sentences are NP-complete.*

What about a tractable alternative?
Two-way quantification

\[(Q_1 Q_2) \land (Q_2 Q_1)\]

Subjects are happy to accept such interpretation.

Gierasimczuk and Szymanik, Branching quantification vs. two-way quantification, Journal of Semantics, 2009
Two-way quantification

\[(Q_1Q_2) \land (Q_2Q_1)\]

Subjects are happy to accept such interpretation.

Gierasimczuk and Szymanik, Branching quantification vs. two-way quantification, Journal of Semantics, 2009