



Solving Hofstadter's analogies using Structural Information Theory

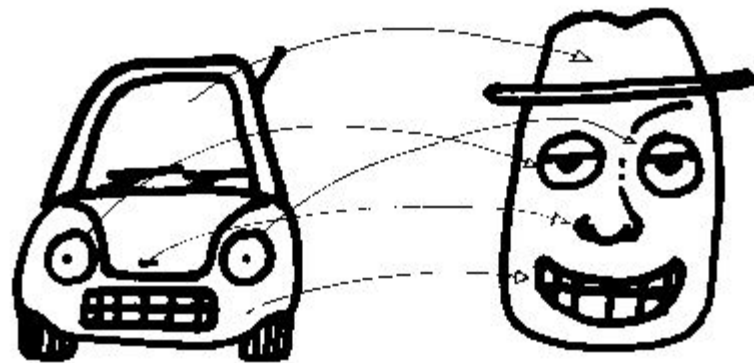
19 November 2020, BNAIC conference @ Leiden (virtual)

Geerten Rijdsdijk (geerten.rijdsdijk@student.uva.nl)
and Giovanni Sileno (g.sileno@uva.nl)

University of Amsterdam

Analogical inference

- Analogies, in all forms, are fundamental to human cognition
- Traditional study subject in psychology and AI
- Contemporary studies in ML



visual analogy

$\text{Paris} \approx \text{France} + \text{Berlin} - \text{Germany}$

analogical inference as operations on word vectors

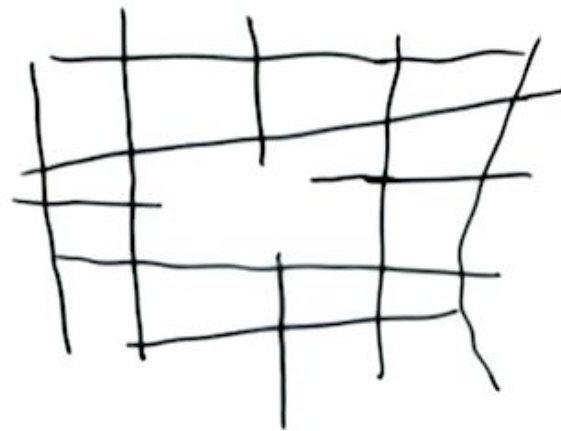
Analogical inference on symbolic analogies

- Hofstadter suggested to study phenomena about analogies on a limited domain whose primitives are meaningless symbols

ABC : ABD :: IJK : ?

string-based proportional inference

- Proposed solutions:
 - Copycat and Metacat



"Analogy is the Interstate Freeway System of cognition"

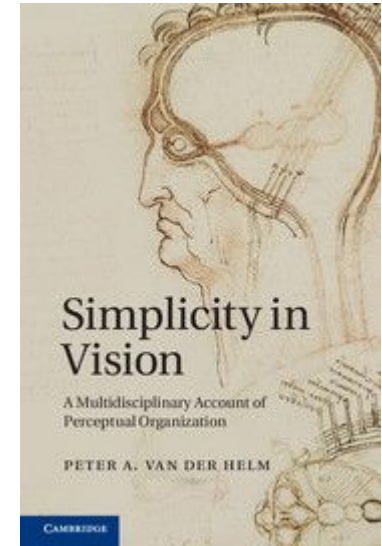
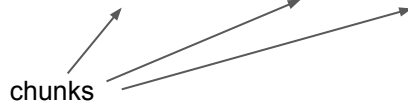
Structural Information Theory

- A perceptual theory about Gestalt effects (emerging organization)
- Non-probabilistic
- Symbol-based

- SIT coding language based on 3 operators:

- *Iteration* $3^*(A)$ $\Rightarrow$ **AAA**
- *Symmetry* $S[(A)(B),(C)]$ $\Rightarrow$ **ABCBA**
- *Alternation* $\langle(A)\rangle/\langle(B)(C)\rangle$ $\Rightarrow$ **ABAC**

chunks



This research: can *Structural Information Theory* be used to analyze Hofstadter's analogies?

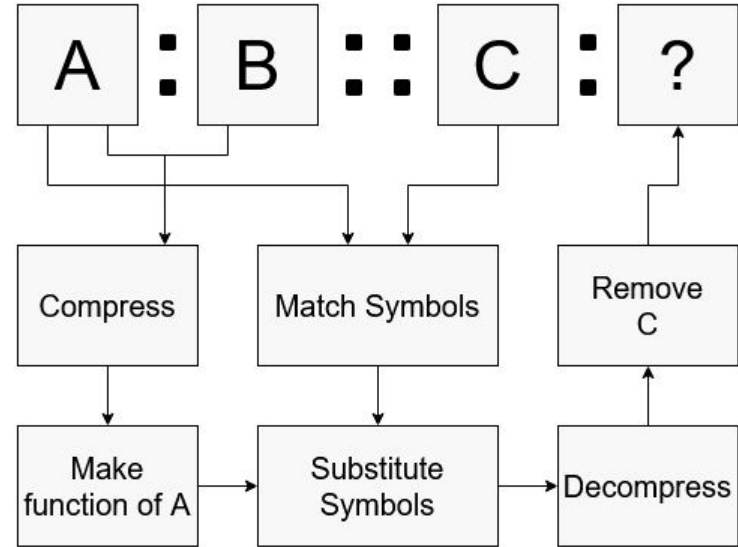
- Thanks to an anonymous reviewer we discovered previous work [Dastani et al., 2003] investigating the same question from a theoretical point of view (algebraic properties of the encoding language)
- Still, our work is distinct as we focused rather:
 - on **concretizing** the **inferencial mechanism**, building upon a more recent algorithm for SIT (PISA [VanderHelm2015]),
 - **evaluating** our proposal with **human responses**
 - **comparing** our proposal with **Metacat** (plausibly state of the art solver)

Method based on several components

(Here explained by means of an example)

- Analogy test given in input:

1. ABC:ABD::IJK:?

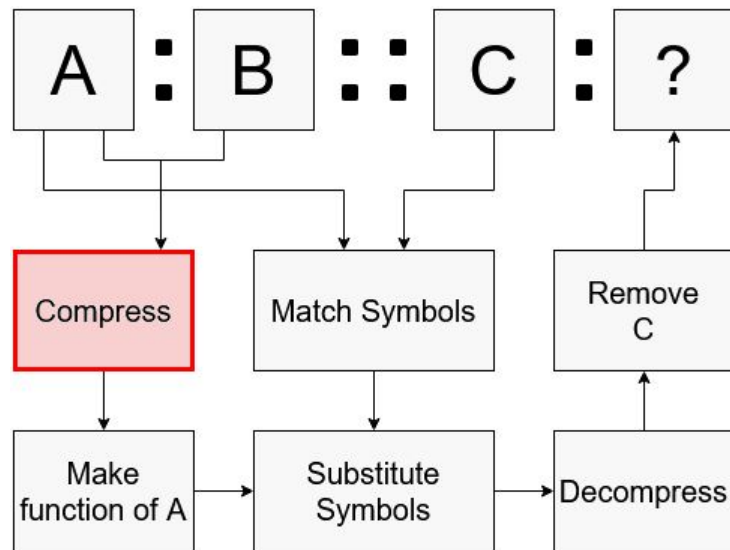


Compression of A concatenated with B

- We use the PISA algorithm [VanderHelm2015]
- We reimplemented it in Python, extending it with new features, e.g. a more general computation of complexity

1. **ABC:ABD::IJK:?**

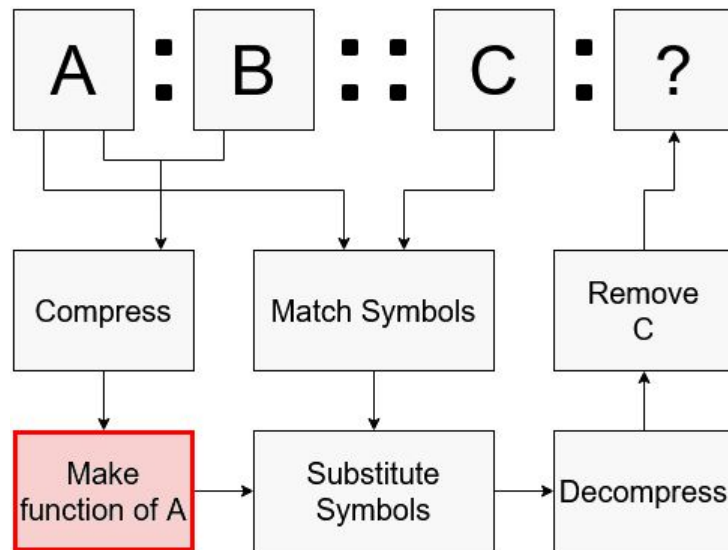
2. **<(AB)>/<(C)(D)>**



Making the code a function of A

- We introduce a metric/distance between symbols.
- *Problem*: which symbol should the distance be calculated from?

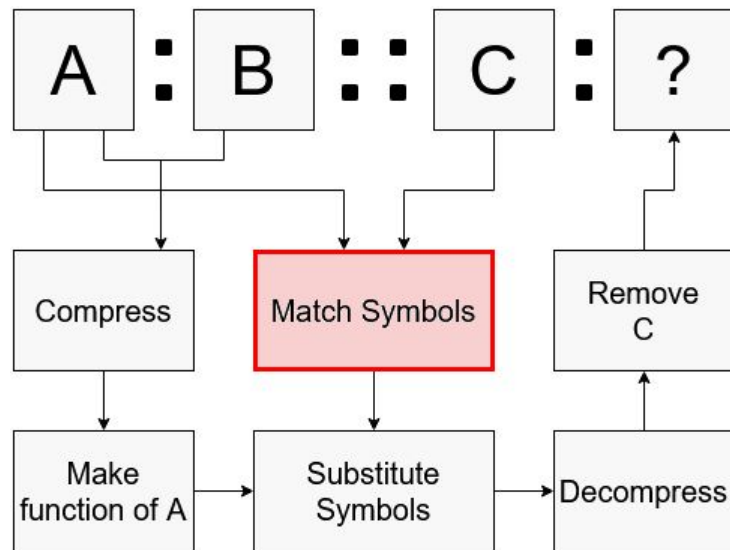
1. **ABC:ABD::IJK:?**
2. **<(AB)>/<(C)(D)>**
3. **<(AB)>/<(C)(\$+1)>**



Symbol matching between C and A

- We formulate a mapping between symbols in A and C
- *Problem:* Difficult when A and C are of different lengths

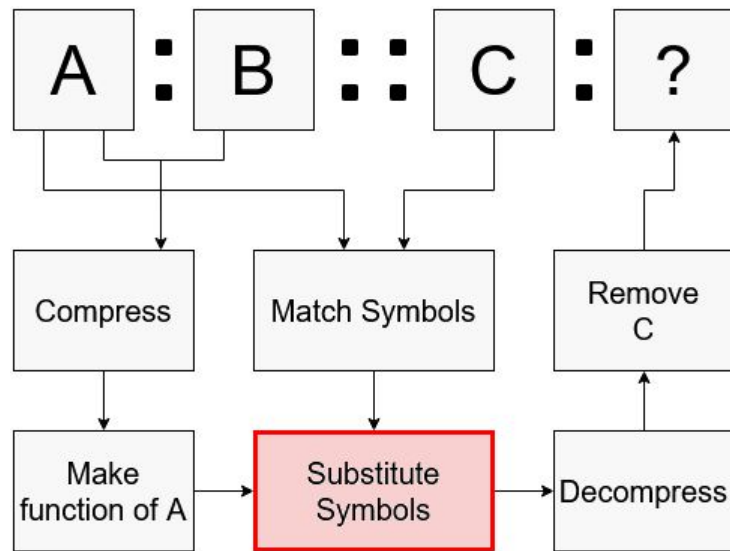
1. **ABC:ABD::IJK:?**
2. **<(AB)>/<(C)(D)>**
3. **<(AB)>/<(C)(\$+1)>**
4. **A:I, B:J, C:K**



Symbol substitution

- We create a new SIT code with the mapping

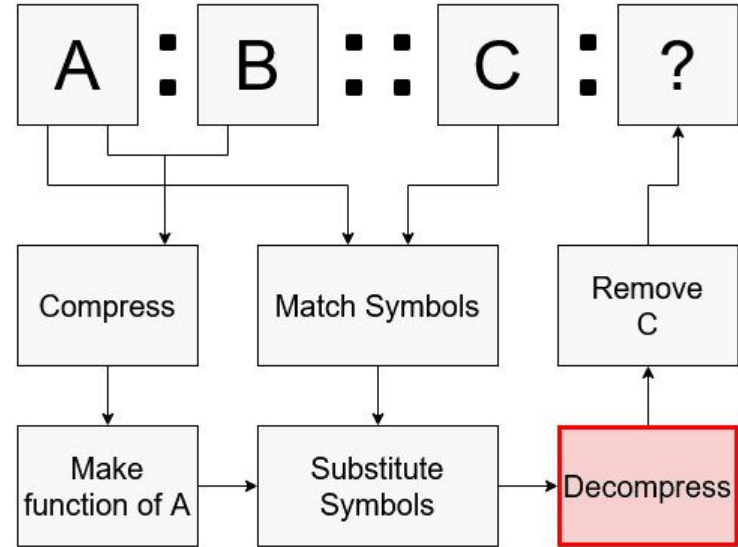
1. **ABC:ABD::IJK:?**
2. **<(AB)>/<(C)(D)>**
3. **<(AB)>/<(C)(\$+1)>**
4. **A:I, B:J, C:K**
5. **<(IJ)>/<(K)(\$+1)>**
6. **<(IJ)>/<(K)(L)>**



Decompressing

- We decompress the SIT encoding

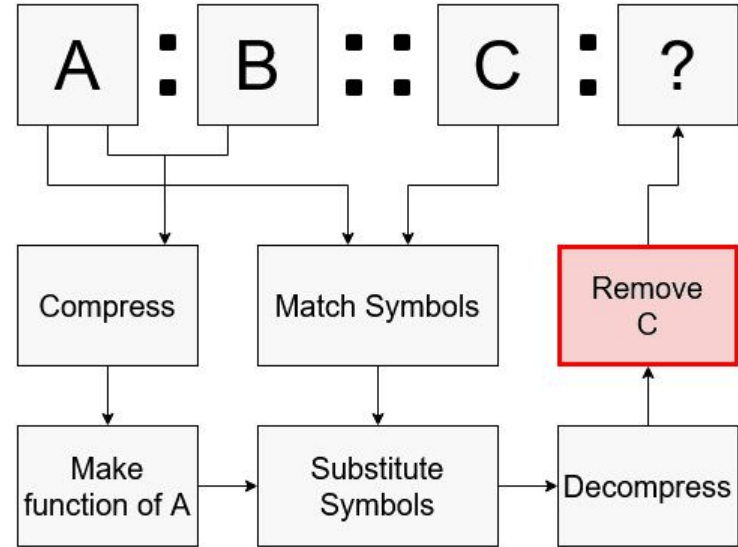
1. **ABC:ABD::IJK:?**
2. **<(AB)>/<(C)(D)>**
3. **<(AB)>/<(C)(\$+1)>**
4. **A:I, B:J, C:K**
5. **<(IJ)>/<(K)(\$+1)>**
6. **<(IJ)>/<(K)(L)>**
7. **IJKIJL**



Removing C

- If **C** is not in the decompressed code, the answer is dismissed

1. **ABC:ABD::IJK:?**
2. **<(AB)>/<(C)(D)>**
3. **<(AB)>/<(C)(\$+1)>**
4. **A:I, B:J, C:K**
5. **<(IJ)>/<(K)(\$+1)>**
6. **<(IJ)>/<(K)(L)>**
7. **IJKIJL**
8. **IJL**



Evaluation

- Comparison with human performance
 - Existing dataset [Murena2017]
 - 11* questions of type: **ABC:ABD::X:?**
 - 65 participants
 - Novel dataset
 - 20 more complex analogy questions
 - 35 participants
- Comparison with state of art solver
 - **Metacat** [Marshall2006], updated 2016
<http://science.slc.edu/~jmarshall/metacat/>

Results (Murena's dataset)

Given X	Solutions	Selected by	P_s	P_M	Given X	Solutions	Selected by	P_s	P_M
IJK	IJL	93%	1	1	BCD	BCE	81%	2	1
	IJD	2.9%	-	-		BDE	5.9%	1	-
BCA	BCB	49%	3	2	IJJKKK	IJJLLL	40%	1	2
	BDA	43%	1	1		IJJKKL	25%	2	1
AABABC	AABABD	74%	1	1	XYZ	XYA	85%	1	-
	AACABD	12%	-	-		IJD	4.4%	-	-
IJKLM	IJKLN	62%	1	1	RSSTTT	RSSUUU	41%	1	1
	IJLLM	15%	-	-		RSSTTU	31%	2	-
KJI	KJJ	37%	1	1	MRRJJJ	MRRJJK	28%	2	1
	LJI	32%	-	2		MRRKKK	19%	1	2
ACE	ACF	63%	1	1					
	ACG	8.9%	-	-					

generating given human answers: our solver 16/22 (73%), Metacat 14/22 (64%)
most common human answer in the top 2 generated answers: 10/11 (91%) both
best answer matching most common human answer: 8/11 (73%) both

Results (our dataset)

Given problem	Solutions	Selected by	P_s	P_M
ABA:ACA::ADA:?	AEA	97.1%	1	1
	AFA	2.9%	-	-
ABAC:ADAE::BACA:?	DAEA	60%	2	-
	BCCC	28.6%	21	-
AE:BD::CC:?	DB	68.5%	3	1
	CC	17.1%	-	2
ABBB:AAAB::IIJJ:?	IIJJ	57.1%	1	-
	JJII	14.3%	-	-
ABC:CBA::MLKJI:?	IJKLM	88.6%	1	1
	-	-	-	-
ABCB:ABCB::Q:?	Q	100.0%	1	-
	-	-	-	-
ABC:BAC::IJKL:?	JKLM	54.3%	-	-
	KIJL	14.3%	2	-
ABACA:BC::BACAD:?	AA	57.1%	1	-
	BCD	31.4%	-	-
AB:ABC::IJKL:?	IJKLM	85.7%	1	1
	IJKLMNOP	11.4%	-	-
ABC:ABBACCC::FED:?	FEEFDDD	91.4%	2	1
	-	-	-	-

ABC:BBC::IKM:?	JKM	57.1%	7	-
	KKM	37.1%	2	-
ABAC:ACAB::DEFG:?	DGFE	68.6%	2	-
	FGDE	14.3%	1	-
ABC:ABD::CBA:?	DBA	51.4%	1	2
	CBB	45.7%	2	1
ABAC:ADAE::FBFC:?	FDFE	94.3%	1	-
	FDFA	2.9%	-	-
ABCD:CDAB::IJKLMNOP:?	LMNIJK	80.0%	-	-
	-	-	-	-
ABC:AAABBBCCC::ABCD:?	AAABBBCCCDDD	74.3%	1	1
	AAAABBBBCCCCDDDD	17.1%	-	-
ABC:ABBCCC::ABCD:?	ABBCCCDDDD	85.7%	-	-
	ABBCCCDDD	8.6%	1	-
ABBCCC:DDDEEF::AAABBC:?	DEEEFF	77.1%	1	-
	DCCDDF	8.6%	-	-
A:AA::AAA:?	AAAAAA	62.8%	1	-
	AAAA	25.7%	2	1
ABBA:BAAB::IJKL:?	JILK	71.4%	-	-
	JIJM	11.4%	5	-

generating most common human answer: our solver 16/20 (80%), Metacat 7/20 (35%)
best answer in the top 2 human answers: 13/20 (65%) vs 8/20 (40%)
best answer matching most common human answer: 10/20 (50%) vs 6/20 (30%)

Conclusion and Future developments

- We presented a novel method to solve analogy question by means of SIT, adequately extended with metrical information and a different measure of complexity. We present a new implementation of PISA in Python.
- We performed an evaluation of our proposal w.r.t. observed human responses, in comparison to Metacat. Our solver exhibits
 - **similar performance** on *Murena's dataset*
 - **better performance*** on *own dataset*
 - is incomparably **faster!**
- Future developments:
 - Further study heuristics/strategies for symbols matching
 - Looking for a unifying theory based on Kolmogorov's complexity



Solving Hofstadter's analogies using Structural Information Theory

19 November 2020, BNAIC conference @ Leiden (virtual)

Geerten Rijdsdijk (geerten.rijdsdijk@student.uva.nl)
and Giovanni Sileno (g.sileno@uva.nl)

University of Amsterdam