

Correspondence-based analogies for choosing problem representations

Aaron Stockdill*, Daniel Raggi*, Mateja Jamnik*,
Grecia Garcia Garcia†, Holly E. A. Sutherland†, Peter C.-H. Cheng† and Advait Sarkar‡

*University of Cambridge, Cambridge, UK

{aaron.stockdill, daniel.raggi, mateja.jamnik}@cl.cam.ac.uk

†University of Sussex, Brighton, UK

{g.garcia-garcia, h.sutherland, p.c.h.cheng}@sussex.ac.uk

‡Microsoft Research, Cambridge, UK

advait@microsoft.com

Abstract—Mathematics and computing students learn new concepts and fortify their expertise by solving problems. The representation of a problem, be it through algebra, diagrams, or code, is key to understanding and solving it. Multiple-representation interactive environments are a promising approach, but the task of choosing an appropriate representation is largely placed on the user. We propose a new method to recommend representations based on correspondences: conceptual links between domains. Correspondences can be used to analyse, identify, and construct analogies even when the analogical target is unknown. This paper explains how correspondences build on probability theory and Gentner’s structure-mapping framework; proposes rules for semi-automated correspondence discovery; and describes how correspondences can explain and construct analogies.

Index Terms—analogy, problem-solving, representation

I. INTRODUCTION

People adapt their explanation for their audience [1]. This adaptation can be motivated by the concept being explained, the audience’s knowledge, or the motivation behind sharing the concept [2], [3]. Each factor must be balanced by the explainer before choosing a representation for the explainee.

In this work, we suggest a paradigm for encoding the analogical links between different representational systems as a means of suggesting alternative representations, and thus easing the burden on the explainer. We do this through a formalisation of *correspondence* which: is probabilistic, allowing us to derive links between representational systems from examples; has mathematical properties that enable composition and other forms of analogical discovery; and allows us to compute a measure of *informational suitability* of representational systems, which can be used for representation selection.

Consider the problem of finding a formula for *the sum of integers between 1 and n*, using some of the mathematical operators $+$, $-$, \times and \div . We may start by writing the expression in a ‘formal’ notation, that is,

$$\sum_{i=1}^n i \quad (1)$$

which is compact and unambiguous. But it presupposes an understanding of summation notation, and it gives no clues to

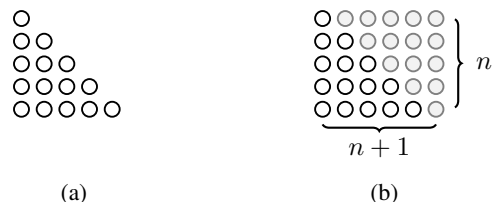


Fig. 1: The sum of integers in a dots representation. (a) The numbers 1 through to 5 as rows of dots. (b) The general relationship for n dots, extrapolated from $n = 5$; counting and halving the grid, we have $n \times (n + 1) \div 2$ black-edged dots.

the shape of our desired formula. The solution might involve induction, or identifying and manipulating series; both are advanced techniques to reach $n \times (n + 1) \div 2$. The algebraic representational system is *informationally* appropriate, but not *cognitively* appropriate for, say, a novice.

We can re-represent Formula (1) as in Figure 1a, instantiated to $n = 5$, where numbers are rows of dots. Modulo generalisation, these representations show the same key information, but their cognitive features are different: the dot representation hints at the solution, but in the algebraic representation this is missing. Observing that we have a triangle, applying simple counting and reflection rules yields the solution in Figure 1b.

Symbolic algebra and grids of dots are not equivalent representational systems: the former can encode concepts that the latter cannot. Nor are the two forms of the problem equivalent: ‘modulo generalisation’ exposes different levels of abstraction. But the statements are *analogous*, capturing the same *important concepts* [4]: integers, associativity, commutativity, etc.

Preserving the relevant structure across representations is complicated when the components take different forms. The symbol 2 becomes $\circ\circ$, while $+$ becomes the stacking operation. The notion of correspondence that we present in this paper precisely captures such links between the *properties* of different representational systems, which can be used to explain existing analogies, and to help construct novel analogies. Related work on analogy and representation is summarised in Appendix A.

Our previous work on the *rep2rep* framework describes

representational systems in terms of properties, completely describing a system, and a problem within that system [4]. We described *correspondences* as links between representational systems, which act as the basis of a representation recommendation. Given problem q and representational system r , we define the *informational suitability* of r for representing q as

$$\sum_{\langle p_1, p_2, s \rangle \in C} s \cdot \text{importance}_q(p_1) \cdot \text{sat}_{p_1}(q) \cdot \text{sat}_{p_2}(r) \quad (2)$$

where C is the set of correspondences, s is the strength of a correspondence, $\text{importance}_q(p_1)$ is the importance of property formula p_1 for problem q [4], and $\text{sat}_{p_1}(q)$ is an indicator function equal to 1 when property formula p_1 is satisfied by the properties of q , otherwise 0 (similarly for p_2 and r). We expand on each of these components in the following section. Intuitively, Equation (2) counts reasons why representational system r is good, weighting those reasons by both how important they are to the problem q and how ‘good’ those reasons are (the correspondence strength s). From informational suitability we can recommend effective representational systems to solve a problem. In this paper, we formalise correspondence in terms of conditional probabilities and propositional logic. We show how this definition allows us to automatically derive correspondences from data, and subsequently, how given a set of correspondences we can automatically infer novel ones.

II. FORMALISING CORRESPONDENCES

A. Transformations

A correspondence associates an encoding of informational content in one representational system to an encoding of content in another representational system, with a measure of how similar the content is. When both source and target representations are specified, the correspondences describe how they are analogous; when only the source representation is specified, the correspondences specify *requirements* on the target representation. In our sum example, the dots representation fulfils the requirements of the problem: they can encode integers and summation. A representational system like the lambda calculus would also fulfil the requirements, but Venn diagrams would not. If instead our problem involved trigonometric functions, our dot and Venn diagram representational systems would fail to satisfy the requirements, receive a low informational suitability score, and hence not be recommended.

The ‘requirements’ of a problem are *properties*, either present or absent. We take our definition of *property* from our previous work [4], where properties are triples of kinds, values, and attribute sets. The *kind* is one of five possible values: tokens (the visible pieces of a representation), types (the roles that tokens can take in a representation), patterns (higher level structures that are composed of tokens), laws (facts in the representation such as associativity), and tactics (ways of manipulating the representation). *Values* are the instances of the kinds that exist within a representational system: for example, a property might have kind ‘type’ and value ‘number’ in our algebraic representation in Formula (1). Finally, *attributes*

describe how the value is used within the representational system at a more granular level. Common attributes for specific representation instances are ‘number of occurrences’ or ‘is of type’ for tokens. We omit the attribute set when empty.

Formally, a correspondence is a triple $\langle p_1, p_2, s \rangle$ where p_1 and p_2 are propositional formulae of properties and s is a real value between 0 and 1 [4]. Property formulae use the connectives AND, OR, and NOT, with the expected interpretations, allowing many-to-many correspondences. We expand on property formulae in Appendix B. Value s denotes the correspondence *strength*: when $s = 0$, the content is unrelated; when $s = 1$, the content is the same.

If we wanted to encode the correspondence ‘a \sum is like stacking *either* horizontally or vertically’, we first identify the properties involved. In this case, we have a ‘token’ kind of property with value \sum from the algebraic representation, so $p_1 = \text{token } \sum$ (ignoring any attributes for this example), while we have two ‘pattern’ properties from the dots representation, $p_2 = \text{pattern stack-horizontal}$ and $p_3 = \text{pattern stack-vertical}$ (where the attributes are similarly ignored). The correspondence would thus be $\langle p_1, p_2 \text{ OR } p_3, 1.0 \rangle$. Note ‘*either*’ in our specification: these target properties are not independent, so we do not write two correspondences. Instead, we use the OR connective to make the sufficiency of one property or the other explicit. This is a strong correspondence, and hence has strength $s = 1.0$. We will explore strength further in Section II-C.

Correspondences are a potential source of *explanations*. Because we can inspect which correspondences are matched (i.e., in Equation (2), for which subset of correspondences C both $\text{sat}_{p_1}(q)$ and $\text{sat}_{p_2}(r)$ are true) we can describe the analogy they form. Similarly to Gentner’s [5] structure-mapping, correspondences encode *how* two structures are analogous; with strength, we can extend structure-mapping and quantify how good the analogy is. A description of how and why an analogy was made can be explicated to the student. Further cognitive grounding discussion is in Appendix C.

B. Property probabilities

If we consider words to be the atomic units of written English, we can compute an occurrence probability of a word derived from its frequency in a particular corpus. Similarly for representational systems, properties are the atomic units; each has an occurrence probability derived from its frequency in a corpus. Thus, each property p in a representational system is associated with a prior probability $\text{Pr}(p)$.

In addition to baseline probabilities, we can consider the context in which properties occur. For two analogous problems in different representational systems, knowledge of the properties from one representational system will update our knowledge of the properties present in the other. Returning to our example of summing integers, observing the \sum operator in Formula (1) primes us to expect stacking—whether horizontal or vertical—in Figure 1a. The conditional probability $\text{Pr}(p_2 | p_1)$ captures our knowledge about p_2 after observing that p_1 is present.

When defining the conditional probability, we consider the co-occurrence of two properties to depend on the source prob-

$$\frac{R(x_1, \dots, x_n) \quad R(y_1, \dots, y_n) \quad \forall_{i=1}^k [\langle x'_i, y'_i, s_i \rangle \wedge (x_i \rightarrow x'_i) \wedge (y'_i \rightarrow y_i)]}{\langle x_{k+1}, y_{k+1}, s' \rangle \quad \dots \quad \langle x_n, y_n, s' \rangle} \text{ [REL] \quad where \quad } s' = \frac{1}{n-1} \sum_{i=1}^k s_i \quad (8)$$

lem and its *transformations* into alternative representational systems. As in our dots example, the transformations of a problem capture the same important concepts, but are not necessarily isomorphic. Stacking is a transformation of adding; dots are a transformation of natural numbers.

An extended discussion can be found in Appendix D.1.

C. Strength

Correspondence strength must reflect the analogical similarity of its constituent property formulae. Naively, this is the conditional probability of the second property formula given the first, but this is not directly comparable to the prior: Is there a big difference to the prior probability? How much *could* the probabilities be different?

To address these concerns, we define the strength of the correspondence $\langle p_1, p_2, s \rangle$ to be

$$s = \frac{\Pr(p_2 | p_1) - \Pr(p_2)}{1 - \Pr(p_2)}. \quad (3)$$

When $\Pr(p_2 | p_1) < \Pr(p_2)$, we redefine the correspondence to be between p_1 and NOT p_2 , using $\Pr(\text{NOT } p) = 1 - \Pr(p)$:

$$\frac{\Pr(\text{NOT } p_2 | p_1) - \Pr(\text{NOT } p_2)}{1 - \Pr(\text{NOT } p_2)} = \frac{\Pr(p_2) - \Pr(p_2 | p_1)}{\Pr(p_2)}. \quad (4)$$

Strength is undefined when $\Pr(p_1)$ or $\Pr(p_2)$ are 0 or 1; these indicate either the property never occurs, or the property carries no information (in the Information Theory sense).

Informally, Equation (3) states that s is the increase in probability of observing p_2 relative to the maximum potential increase of observing p_2 . The numerator is the difference between the informed and uninformed probability of p_2 , while the denominator is the difference between perfect prediction of p_2 and the uninformed probability of p_2 . Similarly, Equation (4) informally states the *decrease* in probability of p_2 relative to the potential decrease of p_2 ; see Appendix D.2 for more details.

Mutual information [6] and Kullback-Leibler (KL) divergence [7] are similar to correspondence strength. Mutual information is a symmetric measure of how much information is shared between two distributions; correspondences are not symmetric, so mutual information is inappropriate for strength. KL divergence is asymmetric, but not bound to the interval $[0, 1]$. Because properties are either present or absent and, so, behave as Bernoulli random variables, we can normalise the KL divergence to the interval $[0, 1]$ with information content: $\text{KL}(\Pr(p_1 | p_2) || \Pr(p_1)) / I(p_1)$. But KL divergence forms a leaky abstraction. As we will see in Section IV, strength as defined in Equation (3) encapsulates relationships between property formulae only in terms of prior probabilities and strength; KL divergence requires calculations on posterior probabilities that are not necessarily available.

III. DISCOVERING CORRESPONDENCES

We turn to exploring how correspondences are discovered. Additional points for this discussion are in Appendices D.3–D.5. Consider the correspondence between 2 and $\circ\circ$: these fill the same role in their respective representations. Hence,

$$\langle \text{token } 2 : \{\text{hasType} := \text{number}\}, \\ \text{token } \circ\circ : \{\text{hasType} := \text{dot-arrangement}\}, 1.0 \rangle$$

For a human this is ‘obvious’, but it must be deduced from somewhere. We propose four rules to automatically discover correspondences: identity, reversal, composition, and relation.

The rule of *identity* states that two properties with the same kind and same value are perfectly corresponding:

$$\frac{p_1 \equiv p_2}{\langle p_1, p_2, 1.0 \rangle} \text{ [IDY]} \quad (5)$$

where \equiv is the string match on kinds and values. This rule implies that correspondences are reflexive with strength 1.0, and allows naturally overlapping representations (e.g., using Hindu-Arabic numerals) to map cleanly into one another.

Correspondence-based rules build new correspondences from existing ones. The first is the rule of *reversal*:

$$\frac{\langle p_1, p_2, s \rangle}{\langle p_2, p_1, s' \rangle} \text{ [REV] where } s' = s \cdot \frac{\Pr(p_1)(1 - \Pr(p_2))}{\Pr(p_2)(1 - \Pr(p_1))}. \quad (6)$$

This allows us to ‘walk backwards’ along a correspondence. The second is the rule of *composition*:

$$\frac{\langle p_1, p_2, s \rangle \quad p_2 \rightarrow p_3 \quad \langle p_3, p_4, s' \rangle}{\langle p_1, p_4, s \cdot s' \rangle} \text{ [CMP]} \quad (7)$$

allowing us to chain together correspondences. Typically we assume $p_2 \equiv p_3$, but using $p_2 \rightarrow p_3$ generalises this rule to property formulae. The strength for each rule is derived from the definition of correspondence strength in Equation (3), treating properties as Bernoulli random variables, and the assumption that properties are *conditionally* independent of each other. That is, p_1 and p_4 are not independent, but we assume they are independent *given* p_2 .

Finally, the *relation* rule [REL] exploits the internal structure of each representation to suggest new ‘parallel’ correspondences. Properties can be linked by the relation ‘ p_1 has attribute with entry p_2 ’. More generally, let $R(p_1, \dots, p_n)$ be an n -ary relation on properties. Relation R defines a context with some properties already in correspondence: the remaining properties may also be in correspondence. This is the rule of *relations*, Equation (8). Assume, without loss of generality, that the properties already in correspondence are the first k pairs. At least one pair of properties is *already* in correspondence, and at least one pair is a *potential* correspondence: thus $1 \leq k < n$. If k is high—many property pairs already correspond—we can be more confident that the remaining $n - k$ property pairs correspond. Hence their strength s' will be high. But if k

TABLE I: Properties of the algebraic and dots representations.

Algebraic	type number, token 1 : {hasType := number}
Dots	type dot-arrangement, token \circ : {hasType := dot-arrangement}

is low—few property pairs already correspond—we cannot be so confident that the remaining property pairs correspond. Hence their strength s' will be low. In the case where $n = 2$ and $k = 1$, we have $s' = s$, and so a useful simplification occurs: if two properties have the corresponding attributes, then the properties themselves may correspond; conversely, if two properties correspond, the values of their attributes may also correspond. We call these [VAL] and [ATR], respectively.

Let us explore these rules using our example of algebra and dots using the properties present in Table I. Because these two representations are disjoint—no properties share a kind and value—the rule of identity [IDY] cannot be used. A human *analyst* must identify a simple initial correspondence:

$$\langle \text{token } 1 : \{\text{hasType} := \text{number}\}, \\ \text{token } \circ : \{\text{hasType} := \text{dot-arrangement}\}, 1.0 \rangle$$

This states that 1 corresponds perfectly to \circ . We can now apply [REV] to associate \circ with 1:

$$\frac{\langle \text{token } 1 : \{\text{hasType} := \text{number}\}, \\ \text{token } \circ : \{\text{hasType} := \text{dot-arrangement}\}, 1.0 \rangle}{\langle \text{token } \circ : \{\text{hasType} := \text{dot-arrangement}\}, \\ \text{token } 1 : \{\text{hasType} := \text{number}\}, 0.9 \rangle} \text{ [REV]}$$

Notice the strength reduced slightly: this is because a 1 has a higher prior probability than a single \circ : all dot arrangements are numbers, but not all numbers are dot arrangements.

Using the correspondence between 1 and \circ , we can discover correspondences between their attributes. Applying the rule of attributes, we have a correspondence between numbers and dot arrangements (eliding the attribute assertions):

$$\frac{\langle \text{token } 1 : \{\text{hasType} := \text{number}\}, \\ \text{token } \circ : \{\text{hasType} := \text{dot-arrangement}\}, 1.0 \rangle}{\langle \text{type number, type dot-arrangement, } 1.0 \rangle} \text{ [ATR].}$$

Applying rules in this way discovers new possible correspondences. A richer set of properties would allow for more rule applications and more discoveries, but even over small domains this support is valuable. Analysts find filtering correspondence suggestions easier than inventing correspondences themselves [8]: with tooling support based on these rules, large domains can viably be linked with correspondences, opening up future applications for this framework.

IV. CORRESPONDENCES AS EXPLANATIONS

Understanding both the definition and source of correspondences, we can interpret their function in making a recommendation. Correspondences provide two modes of explanation: descriptive explanation, and constructive explanation. A descriptive explanation is useful when two structures are suspected to be analogous; for example, our statements about summing numbers in Formula (1) and arranging dots in

Figure 1a. *How* they are analogous might be unclear, but the correspondences that link them form an explanation. Consider our correspondence $\langle \text{token } 1, \text{token } \circ, 1.0 \rangle$ linking 1 and \circ (ignoring attributes). Or consider the more sophisticated correspondence earlier linking the \sum operator with stacking. By informing the user that these are the strongest correspondences which are matched in both the source and target statements (perhaps with generated text) we can explain the analogy.

Conversely we can consider constructive explanations. If a target structure is not known, we can use correspondences to describe what properties the analogous statement should have. Consider again Formula (1), our algebraic problem statement, but this time without knowing the equivalent statement in dots. Using the same correspondences from our descriptive explanation, we can suggest to a student that this problem might be solved by representing 1 as \circ , and to use either vertical or horizontal stacking to convey the \sum . Such hints could support progress through the problem, and potentially reveal deep insights into numbers and summation.

Appendix E describes two evaluations of correspondences. Appendix F exemplifies correspondences in a new domain.

V. SUMMARY AND CONCLUSIONS

Re-representing a problem is a burden on the solver, particularly learners, yet is often key to reaching a solution. We have described correspondences, which capture links between representational systems, to automatically analyse a problem and suggest alternative representations. We explored how correspondences describe or form analogies: numbers can be represented by dots, and addition by stacking. Through correspondences we enable reactive, contextual, and explainable interactions between people and software.

The current automatically discovered correspondences are overly general; we will consider how domain-specific heuristics can be integrated in a principled way. We plan to explore alternatives to the objective function given in Equation (2). We will explore how programmers choose programming languages; this framework would allow programmers to evaluate languages specifically for their current problem. We plan to run user studies to empirically evaluate our formulation.

We propose discovering correspondences in an interactive process, with an analyst acting as arbiter: what is valid in one instance is invalid in the next, even between the same representations. This freedom makes correspondences useful: analogies can be imperfect, but they still guide our reasoning in surprising and insightful ways.

An implementation of the work described in this paper is available at <https://github.com/rep2rep/robin>. This paper, without appendices, is available at <https://dx.doi.org/10.1109/VL/HCC50065.2020.9127258>

ACKNOWLEDGEMENTS

Aaron Stockdill is supported by the Hamilton Cambridge International PhD Scholarship. This work was supported by the EPSRC Grants EP/R030650/1, EP/R030642/1, EP/T019034/1, and EP/T019603/1. The authors wish to thank the reviewers and Gem Stapleton for their comments.

REFERENCES

- [1] J. D. Moore, "What makes human explanations effective?" in *Proceedings of the 15th Annual Conference of the Cognitive Science Society*. Hillsdale, NJ, USA: Lawrence Erlbaum Associates, 1993, pp. 131–136.
- [2] I. Vessey, "Cognitive fit: A theory-based analysis of the graphs versus tables literature," *Decision Sciences*, vol. 22, no. 2, pp. 219–240, 1991.
- [3] D. Moody, "The "physics" of notations: Toward a scientific basis for constructing visual notations in software engineering," *IEEE Transactions on Software Engineering*, vol. 35, no. 6, pp. 756–779, 2009.
- [4] D. Raggi, A. Stockdill, M. Jamnik, G. Garcia Garcia, H. E. A. Sutherland, and P. C.-H. Cheng, "Inspection and selection of representations," in *Intelligent Computer Mathematics*, C. Kaliszyk, E. Brady, A. Kohlhase, and C. Sacerdoti Coen, Eds. Springer, 2019, pp. 227–242.
- [5] D. Gentner, "Structure-mapping: A theoretical framework for analogy," *Cognitive Science*, vol. 7, no. 2, pp. 155–170, 1983.
- [6] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: John Wiley & Sons, Ltd, 2005.
- [7] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.
- [8] R. D. Freund, J. John W. Brelsford, and R. C. Atkinson, "Recognition vs. recall: Storage or retrieval differences?" *Quarterly Journal of Experimental Psychology*, vol. 21, no. 3, pp. 214–224, 1969.
- [9] S. Ainsworth, *The Educational Value of Multiple-representations when Learning Complex Scientific Concepts*. Springer, 2008, ch. 9, pp. 191–208.
- [10] A. Shimojima, "Logical constraints in diagrammatic reasoning," in *Logical Reasoning with Diagrams*, G. Allwein and J. Barwise, Eds. New York, NY, USA: Oxford University Press, 1996, ch. 2, pp. 27–48.
- [11] G. Stapleton, M. Jamnik, and A. Shimojima, "What makes an effective representation of information: A formal account of observational advantages," *Journal of Logic, Language and Information*, vol. 26, no. 2, pp. 143–177, June 2017.
- [12] A. Shimojima and D. Barker-Plummer, "Operations on single feature indicator systems," in *Diagrammatic Representation and Inference*, P. Chapman, G. Stapleton, A. Moktefi, S. Perez-Kriz, and F. Bellucci, Eds. Springer, 2018, pp. 296–312.
- [13] K. Stenning and O. Lemon, "Aligning logical and psychological perspectives on diagrammatic reasoning," *Artificial Intelligence Review*, vol. 15, no. 1, pp. 29–62, March 2001.
- [14] B. Falkenhainer, K. D. Forbus, and D. Gentner, "The structure-mapping engine: Algorithm and examples," *Artificial Intelligence*, vol. 41, no. 1, pp. 1–63, 1989.
- [15] D. Raggi, A. Bundy, G. Grov, and A. Pease, "Automating change of representation for proofs in discrete mathematics (extended version)," *Mathematics in Computer Science*, vol. 10, no. 4, pp. 429–457, 2016.
- [16] J. L. Herlocker, J. A. Konstan, and J. Riedl, "Explaining collaborative filtering recommendations," in *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, ser. CSCW '00. New York, NY, USA: ACM, 2000, p. 241–250.
- [17] G. Minnameier, "The logicity of abduction, deduction, and induction," in *Ideas in action: Proceedings of the applying Peirce conference*. Nordic Pragmatism Network Helsinki, 2010, pp. 239–251.
- [18] L. Damas and R. Milner, "Principal type-schemes for functional programs," in *Proceedings of the 9th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, ser. POPL '82. New York, NY, USA: ACM, 1982, pp. 207–212.

A. Related work

Diverse representations benefit education. Ainsworth discussed the value of multiple representations when they are presented together, and thus can be compared and contrasted by the learner [9]. One benefit of having access to several different encodings is gaining deeper understanding in how representations relate to each other. Ainsworth also acknowledged the importance—and difficulty—of being able to select a representation.

To choose appropriate representations, we must be able to compare representations. The diversity of representations makes them difficult to describe: a representation relates to the domain it is describing, but the mapping can be complex. The specific strengths of diagrammatic representations were described by Shimojima as *free rides* [10], and then generalised by Stapleton et al. as *observational advantages* [11]: representations change the nature of accessing information, and can provide seemingly direct access to the underlying domain objects. Recent work by Shimojima formalises the mapping between a representation and the domain it represents [12], specifically focusing on single indicator feature systems (representations with bijective mappings to their domains). This semantic approach contrasts with work from Stenning et al., who observed the characteristically diagrammatic aspects of representations [13]; the syntax defines the interface we use for reasoning. Moody’s ‘physics of notations’ [3] also examined the visual aspects of representations, aiming to identify desirable aspects. These physical features directly influence our ability to understand and work with representations. We consider representations broadly, including sentential and diagrammatic ones, formalised ones and even representations that do not have a known formalisation.

Gentner considered representations from the perspective of analogy, and concluded that representations are well-considered through their internal structure [5]. Her work on structure-mapping constructs links between two analogous concepts by unifying their internal associations [14]. This requires well-defined internal structure—a requirement the correspondences framework shares. But structure-mapping assumes that two concepts are known to be related, we just need to work out *how* they are related. When we are searching for an analogy, we do not know this. Thus, the purpose of our work is to *discover* related concepts through analogy.

B. Property formulae

When introducing correspondences, we mentioned *property formulae*. These formulae consist of properties combined with the connectives AND, OR, and NOT. We now spend some time understanding the motivation of these formulae, and the meaning we ascribe to them.

Correspondences are, in the simplest case, a one-to-one relation. One property in the first representation relates to one property in the second representation. For example, a dot corresponds to a number. While tidy and simple, this is also

rare. A representation might need to compose several tokens and patterns to capture what is a single token in another. Or a representation might have *redundancy*, where there are several properties that can all do the same thing. For example, adding numbers corresponds to stacking dots, but this can be either a vertical stack or a horizontal stack.

To allow for these more complex relations, we introduce combinations of properties: the connectives AND and OR, meaning that both properties must be present, or at least one of the properties must be present, respectively. We note a semantic quirk: OR is similar, but different, to having two correspondences. Consider our stacking example again: we write

$$\langle \text{token } +, \text{ pattern stack-horizontal OR pattern stack-vertical, } 1.0 \rangle$$

whereas we could use two correspondences

$$\langle \text{token } +, \text{ pattern stack-horizontal, } 1.0 \rangle$$

and $\langle \text{token } +, \text{ pattern stack-vertical, } 1.0 \rangle$.

The two individual correspondences state that + is like stacking horizontally, and that + is like stacking vertically. If a representation has only one kind of stacking then that is acceptable, but if it has both then that is better—twice as good, in fact. In contrast, the combined correspondence using OR states that having one kind of stacking is acceptable, as is having both, but it is not *better* to have both than just one or the other.

This difference in semantics does not apply to AND, as there is no way to re-encode AND as a combination of correspondences.

The final connective NOT describes inverse relationships; that is, observing a property in the first representation makes it *less* likely we will observe a property in the second relation. That is, they conflict. We see this used most often in two ways: either to express a direct negative correspondence (as in the case of Equation (4)) or to preclude conflicting properties: if I have x but not y , then that corresponds to z .

Notice that our combining logical formulae with probabilistic measures in correspondences provides benefits over each approach individually. Logically defined analogies require formalisations and guarantees about their domains. This restricts them to well-defined systems and prohibits ‘good enough’ analogies which can enlighten without being totally formalised or correct [15]. Purely statistical analogy mechanisms allow for these fuzzy analogies, but sacrifice explainability. Without any internal structure, the analogy suggestions are opaque to the user: there is no justification or support on *why* this is an analogy [16]. Our formalisation of correspondences allows for considering informal representations and fuzzy analogies between them as well as explaining the analogy.

C. Cognitive foundations

Human reasoning takes many forms, but broadly we can describe reasoning as *deductive*, *inductive*, or *abductive* [17]. The simplest to automate—captured by the identity, reversal, and

transitivity rules—is deductive reasoning. Through operations on existing correspondences we can deduce new correspondences. Thus, correspondences form a sort of calculus.

While the rules of identity, reversal, and transitivity are motivated by deductive reasoning, relation correspondence rule is motivated by *inductive* reasoning. Specifically, we assume a meaningful structure must be preserved, and this structure is exposed through relationships. By observing these relationships, we can infer new correspondences.

Foundational work by Gentner explored how analogical power is proportional to the relations preserved, in contrast to the superficial features preserved [5]. Indeed, Gentner defines an analogy to be ‘a comparison in which relational predicates, but few or no object attributes,¹ can be mapped from base to target’ [5]. Our rules address this with the ‘hasType’ attribute: through the types of patterns or relation-tokens, we can extract these relation mappings. We use a Hindley-Milner-style type system [18], to accommodate higher-order relations.

Beyond typing relations, our *rep2rep* framework allows for other attributes which can impose novel types of structure on the domain. These attributes extend the set of relations available to create correspondences, fully realising the potential of the structure-mapping concept. All of the contributions in this paper naturally scale over these attributes.

Further extending Gentner’s structure-mapping framework, we remove the need for an explicit target structure to form an analogy with. We work between representational *systems*, such that *any* source structure in a known representation can be used to *construct* an analogy in another known representation.

D. Deriving strengths

The definition of correspondence strength gives us an object to manipulate mathematically. We take full advantage of this, and in this section there are five calculations we wish to inspect in detail: the conditional probability of properties, strength between conflicting property formulae, the strength of reversed correspondences, the strength of composed correspondences, and the strength of correspondences induced by [REL].

1) *Conditional property probabilities*: Conditional probability is defined as

$$\Pr(X | Y) = \frac{\Pr(X \cap Y)}{\Pr(Y)}.$$

We define $\Pr(p_2 | p_1)$ in the same way, adapted to our domain and corpus. In representation A we can encode problems a_i . Each can be transformed into analogous problems $T_B(a_i)$ in representation B . Note that $T_B(a_i)$ is a set; multiple transformations might be appropriate. Using the count operator $|\cdot|$, and $\text{sat}_p(q)$ as a predicate on whether the property formula p is satisfied by the properties in question q , we have

$$\Pr(p_2 | p_1) = \frac{\sum_i |\{b_j \in T_B(a_i) \mid \text{sat}_{p_1}(a_i) \wedge \text{sat}_{p_2}(b_j)\}|}{\sum_i |\{b_j \in T_B(a_i) \mid \text{sat}_{p_1}(a_i)\}|}.$$

¹*Attribute* here is used in the manner of Gentner [5]; this is not an attribute as we define it. In this paper, we use the word ‘feature’ with similar meaning.

The transformation set $T_B(a_i)$ is more general than a set of formal translations. As in our dots representation example, the problems in the transformation set capture the same important concepts, and include problems not provably equivalent. We consider the transformation appropriate if a human expert would be able to reach the same solution as under the original problem statement.

2) *Conflicting properties*: Strength is defined to be between 0 and 1, but the formula in Equation (3) becomes negative when $\Pr(p_2 | p_1) < \Pr(p_2)$. To account for this, we redefine the correspondence to be between p_1 and NOT p_2 , that is observing p_1 informs us about *not* observing p_2 . The strength is now calculated as

$$s = \frac{\Pr(\text{NOT } p_2 | p_1) - \Pr(p_2)}{1 - \Pr(\text{NOT } p_2)}$$

which, after applying the probability identity $\Pr(\text{NOT } a | b) = 1 - \Pr(a | b)$ yields

$$s = \frac{\Pr(p_2) - \Pr(p_2 | p_1)}{\Pr(p_2) - 0}.$$

The derived formula has an interpretation that is consistent with the regular strength interpretation. We can read this as: the strength of the correspondence is the change in likelihood of p_2 after observing p_1 relative to the amount it could potentially have decreased (our leaving -0 in the denominator is to remind the reader of the lowest possible probability).

3) *Reversed correspondences*: The rule [REV] transforms the correspondence $\langle p_1, p_2, s \rangle$ into $\langle p_2, p_1, s' \rangle$ where s does not necessarily equal s' . By Equation 3, we have

$$s' = \frac{\Pr(p_1 | p_2) - \Pr(p_1)}{1 - \Pr(p_1)}.$$

Further, by Bayes’ Theorem, we have

$$\Pr(p_1 | p_2) = \Pr(p_2 | p_1) \cdot \frac{\Pr(p_1)}{\Pr(p_2)}.$$

Thus we can rewrite s' in terms of only s , $\Pr(p_1)$, and $\Pr(p_2)$.

$$\begin{aligned} s' &= \frac{\Pr(p_1 | p_2) - \Pr(p_1)}{1 - \Pr(p_1)} \\ &= \frac{\Pr(p_2 | p_1) \cdot \Pr(p_1) / \Pr(p_2) - \Pr(p_1)}{1 - \Pr(p_1)} \\ &= \frac{\Pr(p_2 | p_1) - \Pr(p_2)}{\Pr(p_2)} \cdot \frac{\Pr(p_1)}{1 - \Pr(p_1)} \\ &= \frac{\Pr(p_2 | p_1) - \Pr(p_2)}{1 - \Pr(p_2)} \cdot \frac{1 - \Pr(p_2)}{\Pr(p_2)} \cdot \frac{\Pr(p_1)}{1 - \Pr(p_1)} \\ &= s \cdot \frac{1 - \Pr(p_2)}{\Pr(p_2)} \cdot \frac{\Pr(p_1)}{1 - \Pr(p_1)} \end{aligned}$$

But why is this useful? We ensure we have a clean abstraction: we do not need to know $\Pr(p_1 | p_2)$, or even know that correspondence is defined in terms of conditional probability. Knowing only the correspondence strength and the prior probabilities of p_1 and p_2 is sufficient to calculate the reversed correspondence strength.

A further algebraic manipulation reveals that

$$s \cdot \Pr(p_1) \cdot (1 - \Pr(p_2)) = s' \cdot \Pr(p_2) \cdot (1 - \Pr(p_1)).$$

Multiplying by the probability of the ‘starting’ property and the complimentary probability of the ‘ending’ property equalises the strengths. In fact, this is equal to

$$\Pr(p_1 \cap p_2) - \Pr(p_1) \cdot \Pr(p_2),$$

the difference between the true conjunctive probability of the properties, and the ‘naive’ conjunction of the properties if they were independent.

4) *Composed correspondences*: The greatest strength of our definition of strength is when we are composing correspondences with [CMP]. Given $\langle p_1, p_2, s \rangle$ and $\langle p_2, p_3, s' \rangle$, we can derive $\langle p_1, p_3, s \cdot s' \rangle$; that is, strengths multiply during composition. The proof is long and uninteresting, but does rely on the conditional independence of p_1 and p_3 given p_2 ; that is,

$$\Pr(p_1 \cap p_3 | p_2) = \Pr(p_1 | p_2) \cdot \Pr(p_3 | p_2).$$

As with reversal, this allows for clean abstraction. The definition of correspondence strength is concealed: if you are composing correspondences you simply multiply the strength. It also gives us an interesting insight into correspondences: by assuming conditional independence, the correspondence will get weaker as we extend the chain. This is sensible: the more we extrapolate, the weaker our confidence becomes. The ‘true’ strength of the correspondence between p_1 and p_3 can be derived from a data set in the same way the first correspondence strengths are, but as a first approximation the multiplication is suitable.

5) *Relation correspondences*: The final strength formula is that associated with the [REL] rule in Equation (8). The strength s' of all the derived correspondences based on an n -ary relation with k already-corresponding arguments is

$$s' = \frac{1}{n-1} \sum_{i=1}^k s_i$$

As we explain in Section III, the motivation for this definition is how confident we are: if most of the n arguments already correspond, the remaining $n - k$ arguments are more likely to correspond, so the strength is higher. Conversely, when few of the n arguments already correspond, the remaining $n - k$ arguments are less likely to correspond, so the strength is lower.

This calculation is a heuristic, and is not based on any fundamental properties of correspondences. Domain-specific knowledge about the nature of the relation R would allow us to more precisely set s' .

E. Feasibility assessment

Having thoroughly explored correspondences in isolation, we return to the motivating goal: how can we suggest an appropriate representation for a specific problem? By describing representations as in [4], we can apply Equation (2) over a set C of correspondences that are both manually and automatically

TABLE II: Ratings for representations, as assigned by human experts and the recommendation algorithm. Alg-X is the algorithm with X disabled: I for importance, S for strength, IS for both. Magnitudes between columns are incomparable.

Representation	Experts	Algorithm	Alg-I	Alg-S	Alg-IS
Bayesian	6.0	17.4	31.0	17.4	31.0
Areas	4.8	11.4	21.7	12.3	23.0
Contingency	4.9	8.4	18.5	9.1	20.0
NatLang	3.5	6.9	16.0	10.4	20.0

derived between the representations, and thus assess how sensible our automatic analogies are.

We presented 11 experts² with the following problem statement:

Problem 1% of the population has disease D . A test is reliable 98% if you have the disease and 97% if you do not have the disease. Assuming the test comes out positive, what is the probability of having the disease?

This problem is stated in a Natural Language representational system, but is amenable to many alternative representations. Along with the problem description, we provided short descriptions of five representational systems:

Natural Language	English language consisting of words, punctuation, etc., grouped into sentences and paragraphs;
Bayesian Algebra	Formal algebraic manipulation, notably including the function $\Pr(\cdot \cdot)$;
Contingency Tables	Two-dimensional tables filled with numeric values, with rules governing the sums of rows and columns;
Rectangular Areas	A unit square which can be partitioned using horizontal and vertical lines; and
Euler Diagrams	Overlapping circles representing sets and their relations, notably unable to represent <i>size</i> .

We discard the Euler representation from our results, as through feedback we discovered that participants implicitly ‘upgraded’ the representation by adding the assumption that the size of the regions indicated cardinality.

Participants used a 7-point Likert scale to rate the degree to which each representation is sufficient for solving the problem. The mean rating of each representation is shown in Table II. Simultaneously, using our framework, we evaluated the informational suitability of each representation. The informational suitability score from Equation (2) for each representation is also shown in Table II. The one-tailed Pearson’s correlation between the two ratings is $r = 0.89$ ($p = 0.053$). We conclude that the algorithm does agree with expert opinion on the informational suitability of representations.

²In screening participants, we defined an ‘expert’ as someone who uses maths skills in day-to-day work such as Lecturers, Professors, Research staff and PhD students in Engineering, Informatics, and Computer Science departments.


```

type title  type character  type actor
token Blade Runner : {isThe = title}
token Harrison Ford : {isThe = actor}
token Rick Deckard {isThe = character,
                    playedBy = Harrison Ford}

```

Fig. 2: Example properties for a Films category.

To understand how the different facets of the algorithm influence the result, we performed an ablation study over the two factors of Equation 2: property importance and correspondence strength. The ratings for the algorithm without these two factors are also included in Table II; Alg-I is the algorithm *without* importance measure, that is, the importance factor is switched off ($r = 0.91$, $p = 0.047$), Alg-S is the algorithm *without* strength, ($r = 0.74$, $p = 0.130$), and Alg-IS is the algorithm with neither ($r = 0.83$, $p = 0.086$). We can conclude that strength has more influence on the result than importance, but the score is largely driven by the count of correspondences: even with both disabled, the algorithm performs moderately well.

F. Generalising correspondences to other domains

Correspondences and the rules to discover them were developed in the context of mathematical problem solving, representation recommendation, and kind-value-attribute property triples, but the ideas are more general. Consider again the rule of relations, [REL], a generalisation of the attribute and value rules. There is no reference to kinds, values, or attributes; [REL] requires only sets of objects and relations on them. Similarly, [REV] and [CMP] work over any sets of objects—no properties required. When we abstract the equivalence relation from the rule of identity [IDY] to work on abstract objects rather than properties, the correspondence framework becomes fully generalised. Correspondences can thus be used to find analogies between any sets of objects equipped with relations.

To demonstrate this generalisation, we consider the problem of discovering analogous *entertainment across media* rather

```

type title  type character  type writer
token The Caves of Steel : {isThe = title}
token Isaac Asimov : {isThe = writer}
token Elijah Baley : {isThe = character}

```

Fig. 3: Example properties for a Books category.

than *mathematical problem statements*. Our sets of objects are no longer representations, but product categories: books and films, for example. We use the *rep2rep* property structure to manage the object encoding, so our objects remain kind-value-attributes, but in principle this is not required. Our kinds are limited to types and tokens, but the values are a diverse collection of names, both peoples’ and titles. We keep our relations as attributes, and thus can continue to use [ATR] and [VAL], but use new labels such as ‘isThe’, or ‘playedBy’. The equivalence relation \equiv remains string equality on kinds and values. We give some sample properties in Figures 2 and 3.

In our simple example, we can apply the rule of identity to types *title* and *character*, because both categories share these two properties. From this we can apply the rule of value using the ‘isThe’ attribute and *character* correspondence, thus deducing that *Rick Deckard* corresponds to *Elijah Baley*. The derivation is:

$$\begin{aligned}
 \text{Let } p_1 &= \text{token Rick Deckard } \{ \text{isThe} = \text{character}, \\
 &\quad \text{playedBy} = \text{Harrison Ford} \} \\
 \text{and } p_2 &= \text{token Elijah Baley } : \{ \text{isThe} = \text{character} \} \text{ in} \\
 &\quad \langle \text{type character}, \text{type character}, 1.0 \rangle \\
 &\frac{p_1 \{ \text{isThe} = \text{character} \} \quad p_2 \{ \text{isThe} = \text{character} \}}{\langle p_1, p_2, 1.0 \rangle} \text{ [VAL]}
 \end{aligned}$$

Over such constrained property sets this is trivial: an analogy was found between domains of books and films. However, it exemplifies the generality and application of our framework in a new domain.