

Chapter 7

Human-Like Computational Reasoning: Diagrams and Other Representations



Mateja Jamnik

Abstract In this chapter, I give a personal account of my experience in Alan Bundy’s DReaM group in the Department of Artificial Intelligence at the University of Edinburgh between the years of 1995 and 1998. Of course, the impact of this experience has been profound and long-lasting to this day. The culture and the nature of research work, the collaborations, the interests and the connections have endured, evolved and multiplied throughout this time. My own work in the DReaM group started by investigating human “informal” reasoning and formalising it in a diagrammatic theorem prover. After leaving Edinburgh, this work naturally evolved into combining diagrams with other representations in a uniform framework, as well as applying visual representations in other domains, such as reasoning with ontologies. But one of the fundamental questions remained unanswered, namely, how do we choose the right representation of a problem and for a particular user in the first place?

7.1 The DReaM Research Environment

Few factors influence a researcher’s ethos regarding their work more than where and with whom they did their PhD project. I arrived to Alan Bundy’s DReaM research group in the autumn of 1995, fresh from finishing a post-graduate Diploma in Computer Science at Cambridge. This was not exactly planned: I actually applied to do a PhD in the Cognitive Science Department at the University of Edinburgh. I was interested in humans, not machines. But given that I was a mathematician by my undergraduate degree and that I just finished a post-graduate degree in Computer Science, my application made it to Alan Bundy in the Department of Artificial Intelligence. I am so glad for this serendipity because the privilege has been immeasurable.

M. Jamnik (✉)

Department of Computer Science and Technology, University of Cambridge, Cambridge,
United Kingdom

e-mail: mateja.jamnik@cl.cam.ac.uk

The DReaM group at that time was a large and thriving community across multiple sites, covering the Universities of Edinburgh, Heriot-Watt and Napier, of very diverse people working on equally diverse research topics. The academic staff were Alan Bundy, Alan Smaill, Andrew Ireland and Helen Lowe.¹ They were working with numerous post-doctoral researchers including Ian Green, Toby Walsh, Richard Boulton, Julian Richardson and Geraint Wiggins. During my time in the group, I was part of a cohort of PhD students including Louise Dennis, Jon Whittle, Raul Monroy, Simon Colton, Francisco Cantu, Ian Frank, Jeremy Gow, Stephen Creswell and Jim Molony. We also collaborated internationally with scientists like Fausto Giunchiglia, Alessandro Armando and their groups in Italy, and Jörg Siekmann, Erica Melis, Dieter Hutter and their groups in Germany. Of course, the DReaM's ethos of creativity of thought and rigour of methodology have since spread around the world as we have pursued our careers across the globe and inevitably passed these values and skills to the next generation of researchers.

I could perhaps describe the diverse topics in mathematical reasoning that the DReaM members pursued as either formalising symbolic reasoning or formalising human-like reasoning.² Symbolic reasoning directions included proof planning, rippling, the systems Clam and λ Clam, induction, co-induction and hardware verification (Chaps. 1–4). Human-like reasoning directions included analogy, diagrammatic reasoning, ontologies and concept formation (Chaps. 5–8). Inevitably, this list is only partial, and all the chapters of this volume hopefully fill some of the gaps. I was particularly interested in the kinds of human-like reasoning that we could perhaps call “intuitive”, or the kind that is inherently human and that is quite different to machine-oriented reasoning. Examples include the use of analogy, symmetry and diagrams.

Our daily lives as researchers were enriched by the visits of numerous scientists who shared their expertise and thoughts with us. Three visitors strongly shaped the direction that I took in my PhD research. Erica Melis from Saarbrücken was working on analogy reasoning at the time [23]. I was intrigued at how one can use examples of solutions in one problem to inspire and help us find a solution to a related problem. Erica mechanised this process in the context of proof. Whilst I did not use her work directly in my PhD, it turns out that my first project after my PhD was to mechanise learning of proof methods by analogy [17]. The second most memorable visitor was Predrag Janičić from Belgrade. He was interested in geometrical reasoning [18], which very much coincided with my interest in human visual reasoning. Predrag also became a close friend, and I could speak his language, so we had our own way of communicating. Finally, perhaps the most influential visitor in the DReaM group for me was Alan Robinson. He came to Edinburgh early

¹I apologise if my poor memory is not serving me well and I mixed people up or inevitably forgot to mention some.

²This divide is perhaps a little artificial since all of our work was motivated by the goals of artificial intelligence, namely we were trying to computationally model human mathematical reasoning. Alan Bundy's Chap. 1 of this volume gives a more precise overall description of our work.

on in my PhD and was interested, like me, in “intuitive” or “informal” reasoning. Our discussions surrounded the distinction between a visual or spatial representation and the more usual machine symbolic representation. Alan Robinson showed me a number of “proofs without words” that he encountered, and some of them became my toy or working examples to represent, solve and mechanise the process. He believed that our ability to “see” the truthfulness of a statement is one of the really fundamental components of the human mathematical cognitive repertoire. Later, when my PhD research was published in a book [15], he kindly wrote in the foreword that in this work I “*found an explanation of at least part of the mystery of how humans are able to ‘see’ the truth of certain mathematical propositions merely by contemplating appropriate diagrams and constructions*”.

What struck me most about the DReaM group was his openness to discuss any topic that anybody was interested in. Clearly, Alan Bundy nurtured a kind and supportive research environment in which everyone could do their best work. His intellectual generosity was boundless. The Blue Book Notes (see Chap. 1) provided the opportunity for sharing and discussing our research with the group. They laid the ground for exploring novel and half-baked ideas that most often developed into mature and original scientific contributions. Alan led the DReaM group in an organised way that taught me how to be a supervisor and a mentor to my own students and post-docs. This is perhaps best demonstrated by numerous “How-to...” guides that Alan wrote, for example, “How to be my student”, “How to write an informatics paper”, “Writing a good grant proposal”, “The Researchers Bible” and “How to say no”.³

If I were to summarise the enduring influence that the DReaM group had on me, then I would put in the first place the intellectual generosity that I try to bestow on my own research group today. I learnt the importance of rigorous methodology and the place for heuristics to guide the automation of reasoning. Perhaps uniquely at that time, our work provided a human-oriented perspective on artificial intelligence that remained the main motivation for my research. Finally, my time in Alan’s DReaM group instilled in me the importance of an interdisciplinary and collaborative approach to research, which I think is key to innovation in AI today.

7.2 Diagrammatic Reasoning

Despite the fact that diagrams have been used in mathematics since the time of Aristotle and Euclid, the invention of formal axiomatic logic at the end of the nineteenth century in the sense of Frege, Russell and Hilbert denied diagrams a *formal* role in theorem proving. Diagrams were only used informally for illustrating a formal proof and for suggesting proof steps but were formally superfluous.

³Many of these can be found on Alan Bundy’s web page: <https://sweb.inf.ed.ac.uk/bundy/>.

Fortunately, the end of the twentieth century started to see a redressing of this issue [2, 5]. Examples include formalised logical systems of diagrams [10, 13, 35]. This directly abolished the widely held Hilbertian theoretical objections to diagrams being used in proofs. Our work on Diamond was amongst these: it pioneered the construction of purely diagrammatic proofs where diagrams and their manipulations are the proof [15, 16].⁴ The motivation for this work was rooted in formalising some of the “informal” reasoning that humans do in mathematics when using diagrams.

Take for example, the diagram in Fig. 7.1. It takes only secondary school-level knowledge of mathematics to understand that the diagram is about the sum of odd natural numbers. We can “see” that the theorem is true not only for the example in the diagram of $n = 6$, but for any value of n . In other words, the simple procedure of splitting a square into the so-called ells works in general. Diamond tackles this problem, in addition to a number of other, the so-called proofs without words, many of which can be found in Nelsen’s books [25, 26] and Gardner’s mathematical recreations [8, 9].

Diamond’s theorems are in the domain of algebraic mathematics about natural numbers that can be expressed as diagrams in a discrete space and are inductive over a parameter. But there is a problem, namely, such diagrams are concrete in nature, so abstractions such as ellipsis need to be used to express the general diagram (and proof) for all values of the parameter. These abstractions are difficult to keep track of whilst manipulating. So we proposed a solution: to use schematic proofs.

Schematic proofs are based on the mathematical notion of the ω -rule that says that for the natural numbers $0, 1, 2, \dots$:

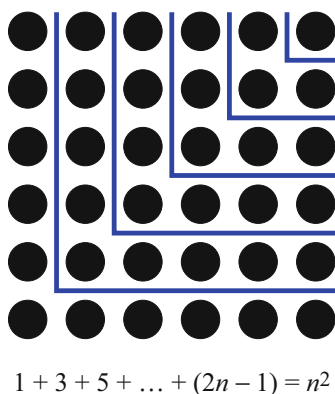


Fig. 7.1 The theorem is about the sum of the first n odd natural numbers. It represents the example of a case for $n = 6$. The proof starts from the RHS of the theorem n^2 and takes a square. Then, the square is split into a sequence of nesting and increasing in size, the so-called ells. Each ell represents a subsequent natural number: there are two edges, each of size n , but the joining vertex has been counted twice; hence an ell is $2n - 1$

⁴This work was done for my PhD with Alan Bundy and Ian Green as my supervisors.

$$\frac{\phi(0), \quad \phi(1), \quad \phi(2), \quad \dots}{\forall x.\phi(x)}.$$

That is, if we can prove $\phi(n)$ for $n = 0, 1, 2, \dots$, then we can infer that $\phi(x)$ for all natural numbers x . Clearly, the ω -rule is not very practical for automation, since it requires the proof of an infinite number of premises to prove its conclusion. A more practical alternative is the constructive ω -rule that has an additional condition: if all premises $\phi(n)$ can be proved in a *uniform* way, that is, there exists an effective procedure, $proof_\phi$, which takes a natural number n as input and returns a proof of $\phi(n)$ as output, then we can conclude the universal statement:⁵

$$proof_\phi(n) \vdash \phi(n).$$

One such effective procedure is, for example, a recursive program. Now $proof_\phi$ can be a recursive procedure that formalises our notion of schematic proof where the number of steps in the proof depends on the parameter n . We used this notion in formalising diagrammatic proofs in Diamond.⁶

Diamond's theorems are expressed as diagrams for some concrete values, that is, ground instantiations of a theorem. The initial diagram is manipulated using some geometric operations. The sequence of geometric operations on a diagram represents the inference steps of a diagrammatic proof. In the above example, the inference step is splitting an ell from a square to produce an ell and a smaller square. The set of all available operations defines the proof search space. Next, Diamond automatically extracts a general pattern from these proof instances and captures it in a recursive program that constitutes a general diagrammatic proof for the universally quantified theorem. The constructive ω -rule justifies the step from schematic proofs to theoremhood. In Diamond, the diagrammatic schematic proof is formalised as

$$\begin{aligned} proof_\phi(n+1) &= \mathcal{A}(n+1), proof_\phi(n) \\ proof_\phi(0) &= \mathcal{B}, \end{aligned}$$

where $\mathcal{A}(n+1)$ consists of a sequence of diagrammatic operations, and the number of applications of each operation is (linearly) dependent on n . \mathcal{B} is a possibly empty basis, that is, no additional operation is required to complete the proof.

The generated program capturing the schematic proof still needs to be verified to be correct. This is something that human mathematicians often omit, and hence history of mathematics is full of erroneous proofs (see Cauchy's proof of Euler's theorem as reported by Lakatos in [21] and in Chap. 1 of this volume). The

⁵From the logical point of view, the constructive ω -rule (and also the ω -rule) is a stronger alternative to mathematical induction, where the generation of proofs for all instances is satisfied by the requirement for the effective procedure, such as a recursive function.

⁶The constructive ω -rule and schematic proofs have previously been implemented for arithmetic theorems and their symbolic proofs by another DReaM member, Siani Baker [3].

verification requires meta-level reasoning about the proof, rather than the object-level theorem, and is done by induction:

$$\begin{aligned} & \text{proof}_\phi(0) \vdash \phi(0) \\ & \text{proof}_\phi(n) \vdash \phi(n) \implies \text{proof}_\phi(n+1) \vdash \phi(n+1). \end{aligned}$$

The work on automation of diagrammatic proofs in Diamond provides important information on proof procedure construction. It exposes the importance of representing diagrammatic expressions so that general reasoning techniques can be applied to them. Furthermore, it provides an insight into how diagrams and purely diagrammatic inferences can be used in formal proofs.

7.3 Heterogeneous Reasoning

Picking up any mathematical book reveals that many theorems are proved using symbolic inference steps as well as diagrams. We call these heterogeneous proofs: examples of two such proofs can be seen in Fig. 7.2. In the first example, a theorem about triangular numbers is proved by transforming it with symbolic inferences into an expression that then has a compelling diagrammatic proof.⁷ In the second example, the theorem asserts a statement about a bitmap image that clearly requires the use of image processing steps to then combine them with symbolic inferences.

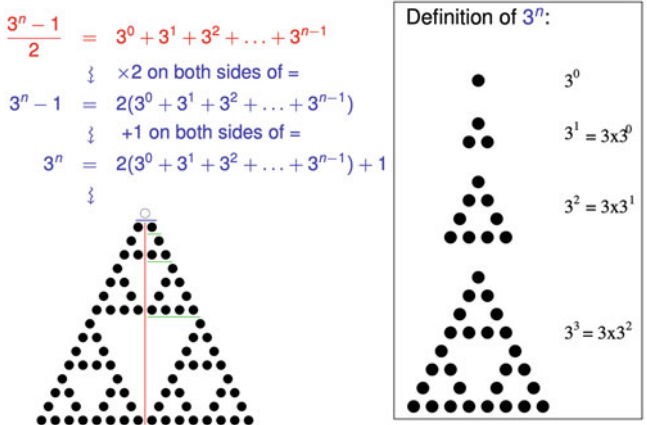
There exist tools for combining diverse systems (e.g., OpenBox [4], Omega [36], HETS [24]), but they do not allow mixing of representations. Indeed, most mechanised theorem provers use only symbolic representations, like different types of logic. Whilst Diamond (and other diagrammatic theorem provers like Speedith [42]) constructs proofs using only diagrammatic inference steps, not all theorems can be expressed with diagrams. Moreover, human mathematicians typically use not only multiple, but also informal representations such as natural language or images within the same problem for different parts of the solution.

We designed and built a heterogeneous reasoning framework MixR [41] where different existing symbolic as well as diagrammatic reasoners can be used at the same time so that symbolic and diagrammatic proof steps can be interleaved within the same proof.⁸ Furthermore, when logical formalisation of a particular representation (e.g., images, natural language or audio) is not tractable, we can embed such data in existing provers and still enable informal heterogeneous reasoning with these opaque objects within an otherwise formal proof.

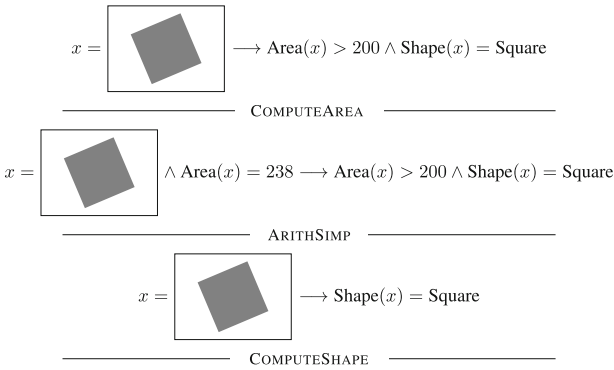
The MixR framework provides a generic infrastructure for extending existing general-purpose theorem provers with heterogeneous reasoning in the form of

⁷Notice that there is no compelling completely diagrammatic proof of the original expression of the theorem, thus the need to mix symbolic and diagrammatic inference steps.

⁸This work was done in collaboration with my PhD student Matej Urbas.



(a)



□

(b)

Fig. 7.2 Examples of two heterogeneous proofs: (a) a few symbolic steps are followed by transformation of the formula into a diagram followed by diagrammatic proof steps to prove a theorem about triangular numbers; (b) here the heterogeneous proof consists of three proof steps: the ComputeArea inference step is heterogeneous and takes a bitmap image and extracts some information (the area of the square) that is expressed in the symbolic language; the ArithSimp inference step is symbolic; the ComputeShape is also a heterogeneous inference step—it extracts that the bitmap shape is a square and thus resolves the implication

heterogeneous logic. The crucial part of our heterogeneous logic is the mechanism, called placeholders, which embeds foreign data into formulae of existing theorem provers so that it can be dealt with using external tools. This data is directly embedded into formulae of a prover that treats them as primitive objects that can be reasoned with its standard inference engine. When required, the reasoner can invoke external tools on this data to obtain new knowledge. Our approach using

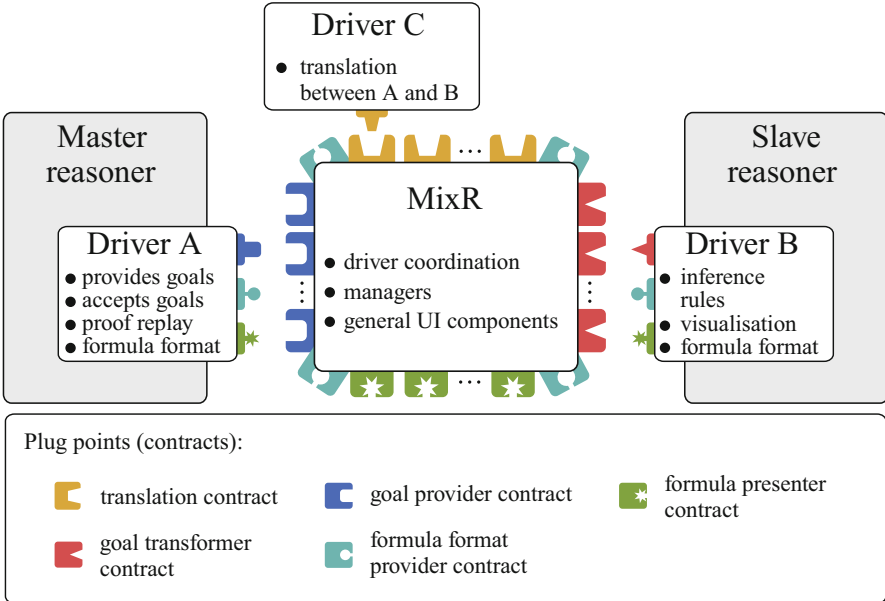


Fig. 7.3 MixR’s architecture with hypothetical drivers. The central box represents MixR’s core. It contains the implementation of heterogeneous logic components, general UI components and driver plug points. Drivers surround MixR’s core and plug into it through the plug points

placeholders removes the need for translations between representations, which is particularly useful when no such translation is available or even possible (e.g., diagrammatic representations from CAD tools, images and signal processing).

MixR is an implementation of this heterogeneous logic and placeholders and enables the integration of arbitrary existing theorem provers of any modality with each other into new heterogeneous systems. A tool developer can plug their chosen reasoners into MixR by writing MixR drivers for them. MixR, in turn, integrates them with each other into a new heterogeneous reasoning system. For example, we plugged Speedith [42] for spider diagrams and Isabelle [28] for sentential higher-order logic into MixR to create the Diabelli [40] heterogeneous reasoning system. We also integrated image processing with symbolic reasoning into PicProc [41] that can prove a theorem in Fig. 7.2b. MixR provides a user interface as well as an application programming interface (API) for drivers. Using the API, the drivers can share, translate and visualise formulae of various modalities. They may also apply foreign inference steps and query other drivers to invoke foreign reasoning tools. The architecture of MixR is illustrated in Fig. 7.3.

Many reasoning tools, representations and visualisation aids in artificial intelligence exist mostly in isolation, specialised in their specific domains. Bringing them together in a simple, flexible and formal way, as in MixR, allows them to contribute to the problem-solving/theorem-proving tasks. This better models what people do

in problem solving, it allows developers to easily design systems that are flexible according to the needs of the end users, and it enables us to take advantage of the existing powerful technology in a novel and sustainable way.

7.4 Accessible Diagrammatic Reasoning About Ontologies

One of our main motivating factors for computationally modelling reasoning with diagrams has been the fact that people use them and find them intuitive and accessible. The barrier to entry for explaining problems and their solutions is lower using diagrams than symbolic logical formalisms. One domain that routinely requires some level of formal reasoning but involves a range of different stakeholders is ontologies. Ontologies are a common knowledge representation paradigm, but they frequently have accessibility issues due to unfamiliarity of domain experts with symbolic notations (e.g., DL, OWL). Some visualisation facilities have been implemented [14, 22], but their focus is expressing and editing ontologies rather than reasoning with and about ontologies.

Ontologies represent knowledge in a domain with definitions of concepts, their properties and relations between concepts. Reasoning with ontologies is done with a justification algorithm [19] that selects a minimal set of axioms responsible for entailment. There is empirical evidence [12] that confirms that stakeholders find it difficult to get from the justification to the explanation of the reasons for the particular selected axioms entailing the problem. Thus, a number of symbolic theorem provers have been implemented, which construct a symbolic explanation for justification–entailment pair. Unfortunately, these proofs have the same inaccessibility issues as before: domain experts are not familiar with their symbolic notations.

In order to address the inaccessibility of symbolic notations, we devised a visual theorem prover, *iCon*, that uses a visual language to represent and reason with ontologies.⁹ The input to *iCon* is a justification–entailment pair expressed as diagrammatic axioms (justifications) and a diagrammatic theorem (ontology entailment). The output is an interactively constructed proof using applications of diagrammatic inference rules that explains how the entailment follows from the axioms.

The visual language of *iCon*, concept diagrams [37], covers almost all of the standard ontology language OWL 2. Empirical studies demonstrate the accessibility of concept diagrams compared to competing diagrammatic and symbolic notations [33]. Concept diagrams consist of curves (circles, as in Euler and Venn diagrams) that represent ontology classes (they are sets), dots and spiders that

⁹This work was done during the Leverhulme Trust funded project “ARD: Accessible Reasoning with Diagrams” in collaboration with Gem Stapleton, Zohreh Shams, Yuri Sato, Sean Mcgrath and Andrew Blake.

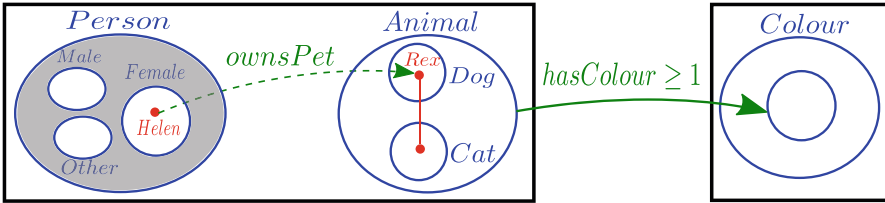


Fig. 7.4 Example of a concept diagram

represent individuals in classes, and arrows that represent object properties. There are also boundary rectangles to denote all individuals in the world, and shading to place an upper bound on the cardinality of the sets. Complete formalisation of concept diagrams is given in [38].

Figure 7.4 shows a concept diagram that has 2 bounding rectangles. Spatial relationships between parts of the diagram convey information, for example, that *Person* and *Animal* represent disjoint sets, since the two corresponding curves are disjoint. We can also see that *Helen* is a *Female* person, due to the location of the (red) dot labelled *Helen*. A dot connected by a line to another dot is called a *spider*, and it signals that it is not clear which set an individual belongs to. For example, in Fig. 7.4, *Rex* could be either a *Cat* or a *Dog*. The region outside of *Other*, *Male* and *Female* is shaded, which means that there is no person who is neither a *Female*, a *Male* nor *Other*. The dashed arrow *ownsPet* connects the dot *Helen* to *Rex*. This means that *Helen* owns *Rex* as her pet, but she can own pets of other types too. Unlike dashed arrows, solid arrows mean that the source is related to *only* the target. So, the colours that an animal can have cannot be outside the set *Colour*. Together with the arrow annotation ≥ 1 , this means that all animals have *at least* one colour.

iCon consists of an inference engine and the graphical user interface. The inference engine contains a collection of inference rules, applies inference rules to diagrams and manages proofs. The inference rules can be either symbolic (conjunction elimination or identity) or diagrammatic. The diagrammatic inference rules come from the ontology community's standard set of inference rules for OWL 2 RL [27], introduced by the W3C in [43]. In order to construct a proof for a justification–entailment pair, we equipped iCon's inference engine with diagrammatic versions of the symbolic inference rules for OWL 2 RL. Diagrammatic inference rules rewrite the diagrams representing the premises of a proof state in order to make them identical to the goal of the proof state. In contrast to a symbolic proof, which is typically inaccessible to domain experts, this results in a diagrammatic proof, which is empirically evidenced to be more accessible [1, 33]. Figure 7.5 illustrates an example of such a diagrammatic inference rule. Reasoning in ontologies most commonly involves entailments, that is, checking if the set of axioms is consistent, coherent or for query answering. Thus, proofs will often be about finding out why a set of axioms is inconsistent or incoherent so that the ontology can be repaired. An example of both a symbolic and iCon's diagrammatic proofs of a theorem about inconsistency can be seen in [34].

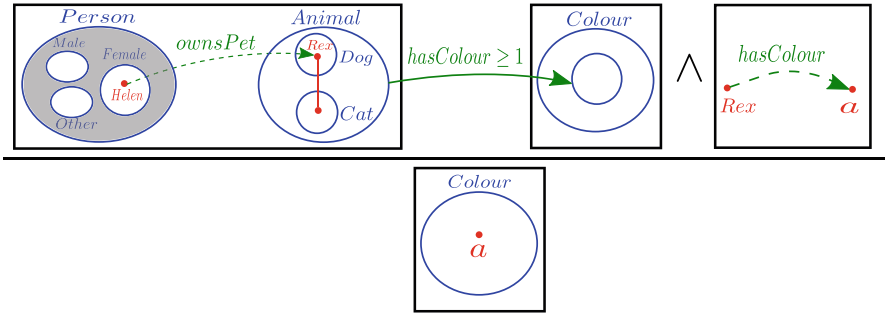


Fig. 7.5 Example of iCon’s diagrammatic inference rule

Ontologies are frequently used in the real world by diverse stakeholders, so it is paramount to make working with them accessible. Current symbolic reasoners for ontologies provide only a minimal set of axioms for entailments without explanations for these entailments or indeed lack of entailment. In contrast, iCon’s diagrammatic proof provides not only an explanation for the entailment that exposes the interaction between the minimal set of axioms, but also an accessible evidence and clues for how to repair the ontology when it is found inconsistent or incoherent. Thus, iCon can be effectively used for reasoning about and debugging of ontologies.

7.5 How to Choose a Representation

So far, we showed how diagrams can be used for formal reasoning, how architectures can be built to enable reasoning with diverse types of representations and indeed tools, and how we can formally reason with diagrams about ontologies. But the question remains: given a problem that we want to solve, how do we choose the representation that is best suited for solving it and that is most appropriate for the user who is trying to solve it? Cognitive science has firmly established that choosing an effective representation can yield dramatic improvements in human problem-solving performance [7, 20] and substantially enhance learning [6]. This is what we are currently investigating in an interdisciplinary project on human-like computing, which has Alan Bundy as one of its advisers.¹⁰ We are combining artificial intelligence, mathematics and cognitive science to investigate human cognitive abilities to find representations that suitably match problems, and the process by which humans adapt or switch between representations. We are devising a foundational theory and building computational models of the critical role that

¹⁰This work started during the EPSRC funded projects “How to (Re)represent it?” and “Automating Representation Choice for AI Tools” in collaboration with Peter Cheng, Daniel Raggi (also an ex-DReaMer), Grecia Garcia Garcia, Aaron Stockdill, Holly Sutherland and Gem Stapleton.

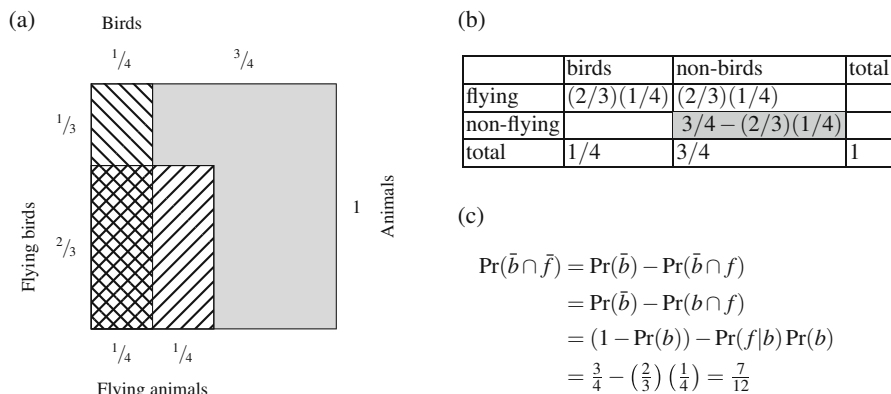


Fig. 7.6 Bird probability example. (a) Geometric representation—the solution is the area of the solid shaded region $\frac{3}{4} - \left(\frac{2}{3}\right)\left(\frac{1}{4}\right) = \frac{7}{12}$. (b) Contingency table representation—the solution is in the shaded cell. (c) Bayesian representation

representations play in problem solving, and automating them in a new generation of adaptive AI systems [30–32, 39].

To illustrate our approach, consider this problem in probability:

One quarter of all animals are birds. Two thirds of all birds can fly. Half of all flying animals are birds. Birds have feathers. If X is an animal, what is the probability that it’s not a bird and it cannot fly?

Here are three different ways one can go about solving this (see Fig. 7.6):

1. You could divide areas of a rectangle to represent parts of the animal population that can fly and parts that are birds.
2. You could use contingency tables to enumerate in its cells all possible divisions of animals with relation to being birds or being able to fly.
3. You could use formal Bayesian notation about conditional probability.

Which of these is the most effective representation for the problem? It depends; the first is probably best for school children; the last for more advanced mathematicians. How can this choice of appropriate representation be mechanised? We are interested to find out:

- What are the formal mathematical and cognitive foundations for choosing an effective representation of a problem?
- Can we develop new cognitive theories that allow us to understand the relative benefits of different representations of problems and their solutions, including taking into account individual differences?
- How can we automate an appropriate choice of problem representation for both humans, taking into account individual differences, and machines to improve human–machine communication?

- Can we build an AI tutoring system, aimed at mathematical problems, that incorporates personalised representation choices and improves users' abilities to solve problems?

We distinguish between *cognitive* and *formal* properties of a representation, in an approach that radically, but systematically, reconfigures previously descriptive accounts of the nature of representations [11]. We use this to devise methods for measuring competency in alternative representation use and also to engineer a system to automatically select representations. *Cognitive properties* characterise cognitive processes demanded of a particular representation (e.g., problem state space characteristics; applicable state space search methods; attention demands of recognition; inference operator complexity [6]). *Formal properties* characterise the nature of the content of the representation domain (e.g., operation types like associative or commutative, symmetries, coordinate systems, quantity or measurement scales).

We devised a novel encoding for taxonomising formal and cognitive properties of problems and representational systems [30, 32]. We catalogue formal properties using templates of attributes that (currently) the developer of the system assigns values to. The attributes encode the informational content of the question and a representational system. Table 7.1 gives snippets from a formal property catalogue for the above *Birds* problem stated in the natural language representation. The colours code the importance of the property relative to the information content (top to bottom in decreasing importance). Table 7.2 gives snippets of the catalogue of formal properties for the Bayesian representational system (used in the solution in Fig. 7.6c). Any representational system and problem expressed in it can be encoded using this description language.

We built algorithms that automatically analyse these encodings for a given problem (like the one in Table 7.1) with respect to candidate representational systems (like the one in Table 7.2) in order to rank the representations, and ultimately suggest the most appropriate one. This analysis is largely based on *correspondences* between the properties of representational systems and their relative importance for a given problem. For example, the correspondences between the natural language formulation of the example and the Bayesian one are translational/morphism-like pairs, such as $\text{ratio} \rightsquigarrow \text{real}$, $\text{given} \rightsquigarrow |$, $\text{probability} \rightsquigarrow \text{Pr}$ and $\text{intersection} \rightsquigarrow \cap$.

Similarly to formal properties, we devised a catalogue of 9 critical cognitive properties. They span spatial and temporal scales (icons to whole displays and seconds to tens of minutes), numerous cognitive processes and the mapping of information between symbols/expressions and concepts. The attributes of cognitive properties characterise the cognitive cost, that is, the difficulty of using that representational system for problem solving. We designed weighting functions to compute overall values of the cognitive cost for each property: they are based on a problem at hand, a typical user and utilise the taxonomy of formal properties.

To adjust cognitive costs from a typical user to individual's abilities, we devised a small but diverse set of user profiling tests. The measures extracted from these

Table 7.1 Formal properties of the *Birds* problem in its natural language representation (note colour)

Kind	Value
Error allowed	0
Answer type	Ratio
Primitives	Probability, and, not
Types	Ratio, class
Patterns	_:ratio of _:class are _:class, probability of _:class and _:class
Facts	Bayes' theorem, law of total probability, unit measure, additive inverse, . . .
Tactics	Deduce, calculate
Primitives	One, quarter, all, animals, birds, two, thirds, can, fly, half, flying, X, animal, probability, cannot
Related primitives	Times, divided_by, plus, minus, equals, union, intersection, probability, zero, . . .
# of primitives	67
# of distinct primitives	31
# of statements	5
Primitives	Feathers
Related primitives	Beast, animate, creature, wing, aviate, flock, fowl, dame, carnal, being, fauna, . . .

Table 7.2 A section of formal properties for Bayesian representational system

Kind	Value
Types	Real, event
Primitives	Ω , \emptyset , 0, 1, =, +, -, *, \div , \cup , \cap , \setminus , $\bar{\cdot}$, Pr,
g-complexity	Type-2
Facts	Bayes' theorem, law of total probability, non-negative probability, unit measure, sigma additivity, commutativity . . .
Tactics	Rewrite, arithmetic calculation
i-complexity	3
Rigorous	TRUE

profiles enable us to scale the level of contributions of each cognitive property to the overall cost of a representational system for an individual. We operationalised the encoding of cognitive properties by automating heuristics that encode user preferences and level of expertise to influence the ranking of potential candidate representational systems.

In this chapter, we are laying the foundations for understanding formal and cognitive properties that affect the choice of representations in problems solving. Our prototype implementations of the algorithms that carry out this analysis show that it is possible to model such processes computationally. We are now applying these foundations in applications such as personalised AI tutoring systems.

7.6 Future Directions

The overarching theme of the work reported here, and common to many past and present DReaM group members, is about computationally modelling human reasoning. The enduring legacy of the DReaM group and our common interests mean that in a number of these projects we continue with existing and establish new collaborations with the past and present DReaMers. For example, Alan Bundy is serving on the advisory board of my project about representation choice and AI tools, and Alison Pease is helping us with her HRL system [29] in our mathematical education project.

The aim of my work is to make AI systems more human-like in the way they interact with users, in the representations that they choose for this interaction, in the methods that they employ to solve problems and in the explanations that they provide alongside their solutions. There are many future directions, especially with respect to fully automating some of these processes and scaling them up to general real-world AI systems. In particular, we are currently developing automated methods for a diagrammatic reasoning system to *discover* new, intuitive solutions to mathematical problems. We are also investigating how we can make theorem provers construct proofs with methods at a level of abstraction and with a level of automation that human mathematicians find appealing. Furthermore, we are marrying statistical with symbolic and knowledge-based approaches to machine learning in order to enhance machine-oriented with human-oriented inference. The results are AI systems that produce solutions from fewer examples and with better explanations of the solutions. There are many applications of this work, but we are focusing on education and developing a new generation of adaptive AI tutoring systems, and on medicine and building integrative data models for clinical decision support systems in personalised cancer medicine.

There is currently much excitement about artificial intelligence and its impact on society. Most of the work that is generating this excitement is due to impressive results of statistical machine learning. However, these machine-oriented methods produce solutions that often lack explanations and use representations that are inaccessible to humans. My research is motivated by human reasoning, so I employ symbolic learning and knowledge-based reasoning as well as diverse representations to enhance this learning and inference. Interdisciplinarity and collaboration have always been at the centre of the DReaM group research ethos, and they have therefore undoubtedly shaped me and my work. Both are key to advancing the field and building a new generation of AI systems that are transparent and have a good cognitive model of the user to be adaptable and to produce explanations understandable to humans.

Acknowledgments I am thankful to all of my collaborators in the work reported here, including Alan Bundy, Ian Green, Matej Urbas, Gem Stapleton, Zohreh Shams, Yuri Sato, Sean Mcgrath, Andrew Blake, Peter Cheng, Daniel Raggi, Aaron Stockdill, Grecia Garcia Garcia and Holly Sutherland.

References

1. Alharbi, E., Howse, J., Stapleton, G., Hamie, A., Touloumis, A.: Visual logics help people: An evaluation of diagrammatic, textual and symbolic notations. In: IEEE Symposium on Visual Languages and Human-Centric Computing, pp. 255–259. IEEE (2017)
2. Anderson, M., Meyer, B., Oliver, P. (eds.): Diagrammatic Representation and Reasoning. Springer (2001)
3. Baker, S., Smaill, A.: A proof environment for arithmetic with the omega rule. In: J. Calmet, J. Campbell (eds.) Integrating Symbolic Mathematical Computation and Artificial Intelligence, no. 958 in Lecture Notes in Computer Science, pp. 115–130. Springer (1995)
4. Barker-Plummer, D., Etchemendy, J., Liu, A., Murray, M., Swoboda, N.: Openproof: A flexible framework for heterogeneous reasoning. In: G. Stapleton, J. Howse, J. Lee (eds.) Diagrams, *Lecture Notes in Artificial Intelligence*, vol. 5223, pp. 347–349. Springer (2008)
5. Chandrasekaran, B., Glasgow, J., Narayanan, N. (eds.): Diagrammatic Reasoning: Cognitive and Computational Perspectives. AAAI Press/MIT Press, Cambridge, MA (1995)
6. Cheng, P.: Electrifying diagrams for learning: principles for effective representational systems. *Cognitive Science* **26**(6), 685–736 (2002)
7. Cheng, P., Lowe, R., Scaife, M.: Cognitive science approaches to diagrammatic representations. *Artificial Intelligence Review* **15**(1-2), 79–94 (2001)
8. Gardner, M.: *Mathematical Circus*. Vintage, New York (1981)
9. Gardner, M.: *Knotted Doughnuts and Other Mathematical Entertainments*. W.H. Freeman and Company, New York (1986)
10. Hammer, E.: *Logic and visual information*. CSLI Press, Stanford, CA (1995)
11. Hegarty, M.: The cognitive science of visual-spatial displays: Implications for design. *Topics in Cognitive Science* **3**, 446–474 (2011)
12. Horridge, M., Parsia, B., Sattler, U.: Lemmas for justifications in OWL. In: 22nd International Workshop on Description Logics, vol. 477. CEUR-WS.org (2009)
13. Howse, J., Stapleton, G., Taylor, J.: Spider Diagrams. *LMS JCM* **8**, 145–194 (2005)
14. Itzik, N., Reinhartz-Berger, I.: SOVA - A tool for semantic and ontological variability analysis. In: Joint Proceedings of the CAiSE 2014 Forum and CAiSE 2014 Doctoral Consortium, vol. 1164, pp. 177–184. CEUR-WS.org (2014)
15. Jamnik, M.: *Mathematical Reasoning with Diagrams: From Intuition to Automation*. CSLI Press, Stanford, CA (2001)
16. Jamnik, M., Bundy, A., Green, I.: On automating diagrammatic proofs of arithmetic arguments. *Journal of Logic, Language and Information* **8**(3), 297–321 (1999)
17. Jamnik, M., Kerber, M., Pollet, M.: Automatic learning in proof planning. In: F. van Harmelen (ed.) Proceedings of 15th ECAI, pp. 282–286. European Conference on Artificial Intelligence, IOS Press (2002)
18. Janjic, P.: GCLC - A tool for constructive Euclidean geometry and more than that. In: A. Iglesias, N. Takayama (eds.) *Mathematical Software - ICMS, Lecture Notes in Computer Science*, vol. 4151, pp. 58–73. Springer (2006)
19. Kalyanpur, A.: Debugging and repair of owl ontologies. Ph.D. thesis, The University of Maryland (2006)
20. Kotovsky, K., Hayes, J.R., Simon, H.A.: Why are some problems hard? *Cognitive Psychology* **17**, 248–294 (1985)
21. Lakatos, I.: *Proofs and Refutations: The Logic of Mathematical Discovery*. Cambridge University Press, Cambridge, UK (1976)
22. Lohmann, S., Negru, S., Haag, F., Ertl, T.: Visualizing ontologies with VOWL. *Semantic Web* **7**(4), 399–419 (2016)
23. Melis, E.: A model of analogy-driven proof-plan construction. In: C. Mellish (ed.) Proceedings of the 14th IJCAI, pp. 182–189. International Joint Conference on Artificial Intelligence, Morgan Kaufmann, San Francisco, CA (1995)
24. Mossakowski, T., Maeder, C., Lüttich, K.: The Heterogeneous Tool Set, HETS. In: TACAS, *LNCS*, vol. 4424, pp. 519–522. Springer (2007)

25. Nelsen, R.: Proofs without Words: Exercises in Visual Thinking. Mathematical Association of America, Washington, DC (1993)
26. Nelsen, R.: Proofs without Words II: Exercises in Visual Thinking. Mathematical Association of America, Washington, DC (2001)
27. The OWL2 web ontology language. URL <https://www.w3.org/TR/owl2-direct-semantics/>. Retrieved Dec 2019
28. Paulson, L.: Isabelle: A generic theorem prover. No. 828 in Lecture Notes in Computer Science. Springer (1994)
29. Pease, A.: A computational model of Lakatos-style reasoning. Ph.D. thesis, Edinburgh University, UK (2007)
30. Raggi, D., Stapleton, G., Stockdill, A., Jamnik, M., Garcia Garcia, G., C.-H. Cheng, P.: How to (Re)represent it? In: 32th IEEE International Conference on Tools with Artificial Intelligence, pp. 1224–1232. IEEE (2020)
31. Raggi, D., Stockdill, A., Jamnik, M., Garcia Garcia, G., Sutherland, H., C.-H. Cheng, P.: Dissecting representations. In: A. Pietarinen, P. Chapman, L. Bosveld-de Smet, V. Giardino, J. Corter, S. Linker (eds.) Diagrams: Diagrammatic Representation and Inference, *LNCS*, vol. 12169, pp. 144–152. Springer (2020)
32. Raggi, D., Stockdill, A., Jamnik, M., Garcia Garcia, G., Sutherland, H., Cheng, P.: Inspection and selection of representations. In: C. Kaliszky, E. Brady, A. Kohlhase, C. Sacerdoti-Coen (eds.) Intelligent Computer Mathematics (CICM), *Lecture Notes in Computer Science*, vol. 11617, pp. 227–242. Springer (2019)
33. Sato, Y., Stapleton, G., Jamnik, M., Shams, Z.: Human inference beyond syllogisms: an approach using external graphical representations. *Cognitive Processing* **20**(1), 103–115 (2019)
34. Shams, Z., Jamnik, M., Stapleton, G., Sato, Y.: iCon: A diagrammatic theorem prover for ontologies. In: F. Wolter, M. Thielscher, F. Toni (eds.) Principles of Knowledge Representation and Reasoning: Proceedings of the 16th International Conference, KR 2018, pp. 204–205. AAAI Press (2018)
35. Shin, S.: The Logical Status of Diagrams. Cambridge University Press, Cambridge, UK (1995)
36. Siekmann, J., Benz Müller, C., Brezhnev, V., Cheikhrouhou, L., Fiedler, A., Franke, A., Horacek, H., Kohlhase, M., Meier, A., Melis, E., Moschner, E., Normann, I., Pollet, M., Sorge, V., Ullrich, C., Wirth, C.P., Zimmer, J.: Proof development with Ω . In: A. Voronkov (ed.) 18th Conference on Automated Deduction, no. 2392 in Lecture Notes in Artificial Intelligence, pp. 144–149. Springer (2002)
37. Stapleton, G., Compton, M., Howse, J.: Visualizing OWL 2 using diagrams. In: IEEE Symposium on Visual Languages and Human-Centric Computing, pp. 245–253. IEEE (2017)
38. Stapleton, G., Howse, J., Chapman, P., Delaney, A., Burton, J., Oliver, I.: Formalizing concept diagrams. In: Visual Languages and Computing, pp. 182–187. Knowledge Systems Institute (2013)
39. Stockdill, A., Raggi, D., Jamnik, M., Garcia Garcia, G., Sutherland, H., Cheng, P., Sarkar, A.: Correspondence-based analogies for choosing problem representations. In: C. Anslow, F. Hermans, S. Tanimoto (eds.) IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC 2020, pp. 1–5. IEEE (2020)
40. Urbas, M., Jamnik, M.: Diabelli: A heterogeneous proof system. In: B. Gramlich, D. Miller, U. Sattler (eds.) IJCAR, *Lecture Notes in Artificial Intelligence*, vol. 7364, pp. 559–566. Springer (2012)
41. Urbas, M., Jamnik, M.: A framework for heterogeneous reasoning in formal and informal domains. In: T. Dwyer, H. Purchase, A. Delaney (eds.) Diagrams, *Lecture Notes in Computer Science*, vol. 8578, pp. 277–292. Springer (2014)
42. Urbas, M., Jamnik, M., Stapleton, G.: Speedith: A reasoner for spider diagrams. *Journal of Logic, Language and Information* **24**(4), 487–540 (2015)
43. Reasoning in OWL 2 RL and RDF graphs using rules. https://www.w3.org/TR/owl2-profiles/#Reasoning_in_OWL_2_RL_and_RDF_Graphs_using_Rules. Retrieved Dec 2019