An Information-Processing Account of Representation Change: International Mathematical Olympiad Problems are Hard not only for Humans

Takuya Matsuzaki (matuzaki@nuee.nagoya-u.ac.jp)
Nagoya University, Furo-cho, Chikusa-ku, Nagoya, 464-8603, JAPAN

Munehiro Kobayashi (munehiro-k@math.tsukuba.ac.jp)
University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8571, JAPAN

Noriko H. Arai (arai@nii.ac.jp)
National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8634, JAPAN

Abstract

In this paper, we present a new information-processing model of math problem solving in which representation change theory can be implemented. Specifically, we divided the problem representation process into two. One is to straightforwardly translate problem texts into formulas in a conservative extension of Zermelo-Fraenkel's set theory, and the other is to interpret the translated formulas in local mathematical theories. A ZF formula has several interpretations, and representation change is thus implementable as a choice of an appropriate interpretation. Adopting the theory of real closed fields as an example of local theory and its quantifier elimination algorithms as an approximate process of searching for solutions, we develop a prototype system. We use more than 400 problems from three sources as benchmarks: exercise books, university entrance examination, and the International Mathematical Olympiad problems. Our experimental results suggest that our model can serve as a basis of a quantitative study on representation change in the sense that the performance of our prototype system reflects difficulties of the problems quite precisely.

Keywords: problem solving; information-processing model; insight; representation change

Introduction

Some math problems are much more difficult than others to solve even though they do not require higher levels of mathematical knowledge or techniques. Nine dot problem and mutilated draughtboard problem are examples of such problems. Where does the difficulty come from?

In classical information-processing models, the difficulty of a given problem is explained by its computational complexity: the cost of search (Kaplan & Simon, 1990; MacGregor, Ormerod, & Chronicle, 2001). In contrast, Gestalts explain the phenomena by the term *insights* (Isaak & Just, 1995; Ohlsson, 1992). A problem is called an *insight problem* when solving it requires a key feature of the problem to be recognized or restructured (representation change).

One of the major criticisms of classical information-processing account is that it has no mechanism to implement representation change since problem solving is understood as a search within a well-defined problem space (Öllinger, Jones, & Knoblich, 2014). If one tries to enlarge the framework (theory) of the problem to implement representation change inside it, then search space explosion is almost always inevitable. For example, it is a well-known fact that almost all the mathematical activities can be formalized in *Zermelo-Fraenkel's set theory* (ZF), thus representation change can be formalized as proof search in ZF. However, we cannot expect the search to be terminated in a realistic time since the search space of ZF is too vast. On the other hand, the representation change account also has some downsides. It does not provide any process model, and the analysis remains qualitative but not quantitative (MacGregor et al., 2001).

In this paper, we present a new information-processing model that enables us to include the representation change account. On the basis of the flow chart of insight problem solving (Öllinger et al., 2014), we first specify the perceptual process as translation of a given problem into a formula in ZF. We extend the language of ZF so that the translation is kept as straightforward as possible. In other words, we assume that the perceptual process requires no insight but rather corresponds to natural language and image processing. This is worth mentioning since the inputs of the existing information-processing account are usually not obtainable without insight regardless of the theories in which the problems are represented (Newell & Simon, 1972; Chou, 1988; Kerber & Pollet, 2006). The obtained ZF formula is considered to be the *primary* problem representation. There are usually many possible interpretations of the primary problem representation in different mathematical local theories. For example, the mutilated draughtboard problem can be embedded to not only propositional logic but also Peano Arithmetic and Presburger Arithmetic. The possible interpretations of the primary representation are called *secondary* representation.

We take the theory of real closed field (RCF) as an example of local theories and implement an interpretation process from the primary to secondary representation. We adopt a quantifier elimination (QE) algorithm as an approximate process of searching for solutions (Iwane, Yanami, & Anai, 2014) and develop a prototype system to solve geometry and introductory calculus problems.

We manually formalize more than 400 math problems from three different sources in our extended ZF language as a benchmark. The problems are translated so that they can be obtainable automatically from the problem text using state-of-the-art natural language processing theories and techniques (Kamp & Reyle, 1993; Steedman, 2001; Zettlemoyer & Collins, 2005). One source of the problems is exercise books, another is university entrance examinations, and

Suppose that x and y are real numbers. Find the range of a satisfying $x^2+ax+1>0$ for all x.

$$\forall x \in \mathbb{R}(x^2+ax+1>0) \quad \Leftrightarrow \quad a^2-4<0$$
$$\Leftrightarrow \quad -2<a<2$$

Suppose that x and y are real numbers. Find the range of a such that there exists x satisfying $x^2+ax+1<0$.

$$\exists x \in \mathbb{R}(x^2+ax+1<0) \quad \Leftrightarrow \quad a^2-4>0$$
$$\Leftrightarrow \quad a<-2 \vee a>2$$

Figure 1: Problem solving and quantifier-elimination

the other is the International Mathematical Olympiad (IMO). Though all the problems require mathematical knowledge and techniques no higher than high-school level, they have different levels of difficulty. We naturally assume that more insight problems can be found in the IMO than in the other two because IMO problems are known to be solvable by only a few mathematically talented students. The highlight of our paper is the experimental results on the benchmark. This is the first paper to report the automated problem solving results on not only a few problems or a set of artificial problems but a large number of real high-school-level problems.

Preliminaries

Let us first redefine what we mean by "mathematical problem solving." A math problem is usually expressed as a combination of sentences, formulas, and figures. In principle, it can be expressed as a logical formula in a theory. A theory consists of a set of symbols called a *language* and a set of *axioms*. A language consists of *constants*, *variables*, *relations*, *functions*, and *logical symbols*. Constants and variables are *terms*, and also $f(t_1,\ldots,t_n)$ is a *term* if f is an n-ary function symbol and all t_is are terms. $R(t_1,\ldots,t_n)$ is an *atomic formula* if R is an n-ary relation symbol and all t_is are terms. For example, $2x+1=y$ and $x>y+z$ are atomic formulas in arithmetic. In the first-order mathematical logic, *formulas* are defined recursively from atomic formulas and logical symbols. In classical logic, we have seven connectives: $\wedge$ (and), $\vee$ (or), $\neg$ (not), $\rightarrow$ (implies), $\leftrightarrow$ (if and only if), $\forall$ (for all), and $\exists$ (there exists). The last two connectives, $\forall$ and $\exists$, are called *quantifiers*. When a variable is quantified, it is called a *bound variable*. For example, the variable x is bound in the formula $\exists x(f(a)=x)$ though a remains *free* (not bound). A formula containing no free variable is called a *sentence*. A formula containing no quantifier is called *quantifier-free*. The set of seven connectives are known to be complete in a sense that any mathematical assertion can be expressed as a first-order formula provided that an appropriate language and a set of axioms are given.

A mathematical problem is *solved* when we find a formal procedure to show the problem is equivalent to a quantifier free formula of the simplest form. Fig. 1 gives examples.

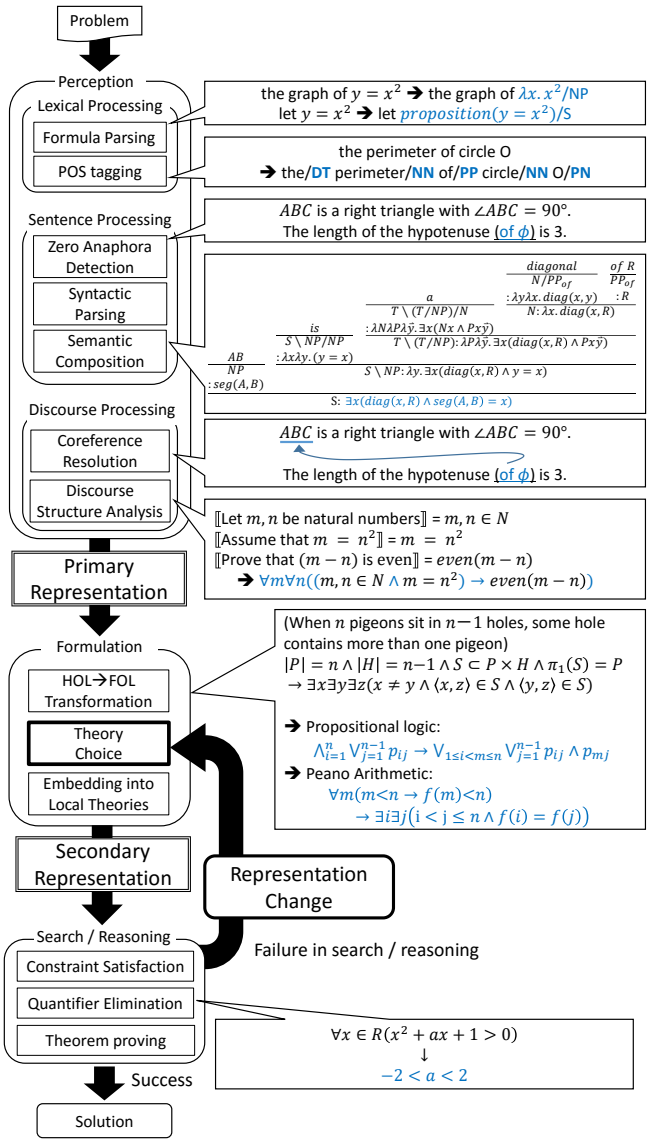

Figure 2: End-to-end problem solving model

Specifically, we say that a problem is *proved* when we show that a given problem is equivalent to True.

A theory is called *decidable* when there is an algorithm to determine whether any sentence is true or not. Gödel's incompleteness theorem shows that any theory containing Peano Arithmetic is undecidable.

Propositional logic, RCF, and Presburger Arithmetic are rare exceptions that are known to be decidable. However, computational complexity of the decision procedures is quite high. The theoretical lower bound of the decision procedure for propositional logic is superpolynomial to the size of input formulas assuming that P≠NP, and those for RCF and Presburger Arithmetic are doubly exponential (Tarski, 1951; Fischer & Rabin, 1974). These lower bounds reflect the phenomena of search space explosion.

An End-to-end Math Problem Solving Model

Fig. 2 presents an overview of our problem solving model. It consists of three modules. The perception module translates a problem into a primary representation expressed in ZF by language processing. The formulation module transforms the primary representation to another formula in ZF that is interpretable in a local theory such as RCF. Finally, the search/reasoning module works on the secondary representation. Once a failure is detected in the reasoning, the process backtracks to the formulation module and seeks another problem representation that makes the reasoning easier. The rest of this section provides more details on the three modules.

Perception Module

The perception module is organized along a hierarchy in natural language: words, sentences, and discourses (i.e., sequences of sentences). The lexical processing unit identifies the parts-of-speech and other syntactic properties of the words and math formulas in a problem. Since math formulas have their own grammar, they are analyzed by a specialized parser. Fig. 2 provides an example in which the same formula $y = x^2$ has different syntactic roles, noun phrase (NP) and embedded sentence (S), in accordance with its context.

The sentence processing unit translates each sentence in the problem into a formal representation. We assume a grammar-driven translation model here, which composes the semantic representation of a sentence along its syntactic structure (Carpenter, 1997; Heim & Kratzer, 1998). Specifically, we developed a Japanese grammar in the formalism of Combinatory Categorial Grammar (CCG) (Steedman, 2001). Fig. 2 depicts the process of semantic composition with CCG for the sentence "*AB* is a diagonal of *R*."

We need to detect omissions (zero pronouns) in the text before the semantic composition. Our current implementation detects them using a list of words and their syntactic arguments (i.e., case frames). Fig. 2 provides an example where an omission ("of ϕ," where ϕ, a zero pronoun, stands for *something*) is detected as the argument of 'hypotenuse.'

The discourse processing unit combines the sentence-level semantic representations into a single formula. We adopt the discourse representation theory (Kamp & Reyle, 1993) as the basic mechanism of the inter-sentential composition. Fig. 2 depicts an example where the semantic representations of three sentences are combined into one with the two connectives $\wedge$ and $\rightarrow$, and two universal quantifications ($\forall m \forall n$). The discourse processing unit also determines the antecedents of the anaphoric expressions including zero pronouns.

Formulation Module

The formulation module receives a primary problem representation and transforms it into a secondary representation that is amenable to reasoning. The process consists of two steps. One is the transformation of the higher-order formulas produced by the perception module to first-order formulas in ZF. The other is the transformation of the ZF formulas to those interpretable in a local theory, such as RCF and propositional logic. The former is usually a routine procedure for a person with the necessary math knowledge. The latter requires a target theory to be chosen beforehand. In the experiments, we chose RCF as the target local theory and confirmed that many pre-university math problems can be mechanically reformulated in RCF. This suggests that, once an appropriate local theory is chosen, the reformulation can be modeled as a heuristic search that seeks a formula in the local theory that is equivalent to the primary representation.

What is missing in our prototype implementation is a mechanism to choose an appropriate local theory. Our hypothesis is that it is the key ability in the representation change, which truly requires 'insight.' Our information-processing model thus serves as a test bed for *computational* models of insight problem solving by plugging-in a theory choice model to it. The rest of this section summarizes the implementation of the two reformulation steps and elucidates the contribution of the problem solving model.

Higher-order to first-order transformation The primary representation often includes higher-order elements (λ-abstractions), which denotes functions (e.g., $\lambda x.x^2$) and conditions (e.g., $\lambda y.(|y| < 1)$). They are necessary to translate the natural language expressions such as "The function that maps $x \in \mathbf{R}$ to x^2" and "The absolute value of y is less than 1. The same condition also applies to x." We eliminate such higher-order elements to obtain a first-order formula. In the current implementation, this is done by iteratively applying a handful of transformation rules such as β-reduction and variable elimination by substitution ($\exists x(x = \alpha \wedge \phi(x)) \Leftrightarrow \phi(\alpha)$).

Reformulation in RCF In the prototype implementation, a primary representation is rewritten into the language of RCF. The first-order language of RCF consists of polynomial equations and inequalities, logical connectives, and quantifiers. We developed a set of axioms that define various math concepts in the (higher-order) language of RCF, such as:

$$\forall x \forall f(\text{minimize}(x, f) \leftrightarrow \forall x'(f(x) \leq f(x'))).$$

The primary representation is iteratively rewritten with these axioms until an equivalent formula is found in the first-order language of RCF. There is no theoretical guarantee that such a formula will be eventually found even when it exists. We empirically examined how often it succeeds in the experiments.

Where in the process does *insight* come? The vocabulary of a problem usually tells us in which theory it should be solved. However, this is not always the case. For instance, the wording in the mutilated draughtboard problem does not suggest it should be formulated in arithmetic but not in propositional logic. Human solvers thus usually start by searching for the solution in propositional logic, putting dominoes on the board in trial-and-error manner. It is inevitable to change

Table 1: Subject areas of the benchmark problems

	Ex	Univ	IMO
Algebra	0	10	21
Linear Algebra	14	62	0
Geometry	81	65	94
Pre-calculus	0	75	0
Calculus	6	33	0
total	101	245	115

Table 2: Overall benchmark results

Problem Source	Solved (%)	Solved Time (sec) min/med/avg/max	Failed FM (%)	Failed TO (%)	Failed WR (%)
Ex	75.2	1 / 4 / 20 / 1069	7.9	13.9	3.0
Univ	65.3	1 / 7 / 38 / 1061	7.3	22.9	4.5
IMO	26.1	2 / 10 / 56 / 513	10.4	60.0	3.5

the representation of the problem to solve it in a realistic time. When and where does representation change happen in cognitive process? The main contribution of our processing model lies in pinning it down to a specific step in the problem formulation process, namely the theory choice.

Our information-processing model helps discriminate between different kinds of 'insight' problems. Nine-dot problem and mutilated draughtboard problem have been considered typical insight problems of the same kind. However, the reasons why people have difficulties are different in nature. Failure in solving nine-dot problem is at least partially due to the ambiguity of the term "line (segment)". Disambiguation of terms is a part of the perception process, but not of the formulation or representation change in our model. In contrast, they fail to solve mutilated draughtboard problem because they cannot choose an appropriate theory to solve it only from the superficial properties of the problem.

Solution Search/Reasoning

In the current implementation, we adopt a QE algorithm for RCF (Iwane et al., 2014) as an example for solution search. Note that we do not argue the QE algorithm per se is the model of human answer-deduction process. We utilize it to approximately measure the difficulty of mathematical reasoning on a given problem representation. The computational cost of the QE algorithm is quite sensitive to the problem representation; its time complexity is doubly exponential to the number of the variables in the representation. We regard a long running time of the algorithm as a sign of the *impasse* in the reasoning, which has been considered as a trigger of representation change (e.g., (Öllinger et al., 2014)). In the experiments, we examined to what extent this failure detection mechanism correctly reflects the difficulty of the problems.

Experimental Procedure

Aim of the Experiment

We developed a prototype implementation of the model described in the previous section. The theory choice process and representation change mechanism is not yet implemented. The aim of the experiment is to test if we can use the model as a basis for developing a computational model of theory choice and representation change. We thus need to verify: A) the model can solve many *non-insight* problems, which do not require representation change and B) the response of the model correlates with the difficulty of the problems. B) means the model is usable to *quantify* the difficulty of a problem in a certain representation, which is a requirement for a quantitative study on representation change.

Material

We collected more than 400 problems taken from three sources: exercise books (**Ex**), Japanese university entrance exams (**Univ**), and International Mathematical Olympiads (**IMO**). The **Ex** problems were sampled from a popular exercise book series. The problems in the books are marked with one to five stars in accordance with their difficulty: one to three stars signify textbook exercise level and four and five stars signify university entrance exam level. We sampled approximately the same number of problems from those marked with one, two, and three stars. The **Univ** problems were taken from the past entrance exams of seven top Japanese national universities. The **IMO** problems were taken from the past IMOs held from 1959 through 2014.

We examined the problems and exhaustively selected those that can be formulated (by humans) in the theory of RCF. The distinction between RCF and non-RCF problems was made solely on the basis of the essential mathematical content of the problems. The selected problems thus contain problems in several subject areas as shown in Table 1.

The problems were manually formalized in a higher-order language. Operators, who all majored in computer science and/or mathematics, were trained to translate the problems as faithfully as possible to the original natural language statements following the design of the perception module.

Experimental Results

The prototype system was run on the benchmark problems with a time limit of 3600s per problem. Table 2 shows the number of successfully solved problems; minimum, median, average, and maximum (wallclock) time spent on solved problems; number of failures in the reformulation of the primary ZF representation in RCF (FM); number of failures due to timeout (TO); and wrong answers (WR). Wrong answers were due to bugs in the current implementation.

Overall, the performances on the **Ex**, **Univ**, and **IMO** problems seem to well reflect the inherent differences in their difficulty levels. We conducted χ^2-test on the difference in the rates of success on them. The difference between **IMO** and other sources were statistically significant ($p < 0.01$) though that between **Ex** and **Univ** was not ($p = 0.09$).

We further examined how well the system performance correlates with the fine-grained difficulty level assessed by human experts. Table 3 lists the performance figures for

Table 3: Results for **Ex** problems by number of stars

#★	Succeeded		Failed	
	Success %	Time (sec) min/med/avg/max	FM (%)	TO (%)
1	82.4 (28/34)	1 / 4 / 5 / 39	11.8	5.9
2	73.5 (25/34)	2 / 4 / 6 / 39	5.9	11.8
3	69.7 (23/33)	2 / 4 / 51 / 1069	6.1	24.2
4	63.2 (24/38)	2 / 6 / 36 / 589	10.5	23.7
5	54.3 (19/35)	3 / 10 / 198 / 3245	2.9	42.9

Table 4: Syntactic profiles of the formalized problems

	Ex	**Univ**	**IMO**
# of $\forall$	2.2	2.0	5.8
# of $\exists$	5.3	9.3	3.1
# of λ	1.3	2.1	0.1
# of relations	12.5	19.8	13.8
# of functions	19.9	36.3	21.9
# of bound variables	8.8	13.4	9.1
# of free variables	3.0	3.1	1.8

the **Ex** problems (one to three stars) and additional problems sampled from those marked with four and five stars in the same exercise books. The overall correlation between the difficulty level and the system performance is clear although the difference in the success rates was statistically significant only between the problems with one star and five stars ($p < 0.05$, χ^2-test).

Analysis of the Experimental Results

Can we estimate the difficulty of a problem just by seeing it? If we can, the difficulty of the problems shall be attributed more to its inherent search cost (e.g., the time complexity determined by the number of variables) rather than the necessity of representation change. Table 4 presents several syntactic features of the benchmark problems. The figures are averaged over the problems taken from each source. It reveals that the syntactic features of the **IMO** problems are not very different from the exercise problems in **Ex** except for the distribution of variable binders ($\forall$, $\exists$, λ).

In addition to the basic features listed in Table 4, we may be able to estimate the difficulty of a problem by the vocabulary (i.e., distribution of function/relation symbols). To see this, we trained a binary classifier that predicts whether or not a problem can be solved by the prototype system in one hour. We used the features in Table 4 and the number of each symbol in a problem as the input and trained the classifier on the results of the benchmark test. Table 5 lists the precision and the recall of the classification obtained by 5-fold cross-validation. The definitions of the precision and recall are: precision = $TP/(TP+FP)$ and recall = $TP/(TP+FN)$, where TP (resp. FP) is the number of problems correctly (resp. wrongly) predicted 'solvable' and FN is the number of problems wrongly predicted 'unsolvable.' The overall prediction accuracy in Table 5 is way above the majority baseline of 57% but the accuracy is not very high especially on **Univ** and **IMO** problems.

Table 5: Accuracy of the solvability prediction

Source	Precision	Recall
Ex	88% (67/ 76)	93% (67/ 72)
Univ	73% (116/160)	78% (116/149)
IMO	57% (17/ 30)	47% (17/ 36)
All	75% (200/266)	78% (200/257)

The analysis presented above revealed that certain types of the difficulty are not captured by the superficial properties of the problems including the problem size and the vocabulary. This is a partial indication of the necessity of representation change or other kinds of insight for solving the problems. A future work is to examine such problems and clarify why they are difficult and what kinds of theory choice appear in human solutions of such problems.

Discussions

A first-order theory consists of a language and axioms. A formal theory is expressed in propositional logic, the first-order predicate logic, or higher-order predicate logic (typed lambda calculus). In our model, we set the primary representation expressed in the first-order ZF. Thus, there are three kinds of theory changes: axiom change, language (and axiom) change, and change from propositional to predicate logic.

Axiom Change

There are infinitely-many possible representations for propositional logic. Among them we can find analytic tableaux (cut-free LK), resolution, and Frege system (LK). The former two are the major systems used as the basis for automated theorem proving. Cut rule (axiom) allows one to introduce "lemmas" to prove theorems.

The pigeonhole principle is known to require exponential size proofs both in analytic tableau (Cook & Reckhow, 1979) and resolution (Haken, 1985), but it has polynomial-size proofs in Frege system (Buss, 1987). This is because we can introduce concepts of "addition", "subtraction" and "counting", and manipulate them to do some "restricted arithmetic" in Frege system. However, search-cost for appropriate cut-formulas is extravagant, and there is almost no hope that someone comes up with appropriate cut-formulas.

Another way to shorten proofs is to introduce a "symmetry rule" (Arai, 1996) as a new axiom. Propositional variable $p_{i,j}$ stands for "the i^{th}-pigeon sitting in the j^{th}-hole", and "$\bigvee_{j=1}^{n} p_{1,j}$" for "the first pigeon sitting in some hole" when expressing the pigeonhole principle in propositional logic (Fig. 2). If we have to check all the possible pigeons' positions, proofs blow up exponentially. Proofs will be shortened if we, without loss of generality, assume that the first pigeon sits in the first hole. In other words, "insight" realizing that a given problem has the property of symmetricity helps us to escape from an exhaustive search. There are some heuristics known to detect symmetricity, and it is implemented on computer (Arai & Masukawa, 2000).

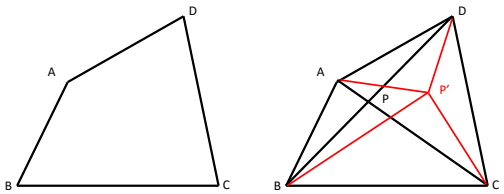

Figure 3: Solution to the quadrangle problem

Language Change

Elementary (Euclidean) geometry is known to be embeddable into the Cartesian coordinate system, and finally to RCF. However, languages and sets of axioms are different. As a result, the difficulties of problems do not remain the same.

Consider the following problem: "Let *ABCD* be a quadrangle. Find the point P that minimizes the sum of *AP*, *BP*, *CP*, and *DP*." Fig. 3 illustrates that the intersection of the diagonals minimizes the sum because of triangle inequality.

Insight may be required to line up the intersection of the diagonals as a candidate for *P*. However, the idea is easier to conceive when it is represented in Euclidean Geometry than in RCF since the intersection of the diagonals has salience in Euclidean Geometry.

Propositional or Predicate?

Mutilated draughtboard problem ($2n \times 2n$ version) is a good example of problems which is solvable when one changes the setting radically. The problem requires exponential-size proofs in resolution and analytic tableaux. It is not known whether or not it has short proofs in tableaux with symmetry rule. However, it has a short proof in arithmetic.

Conclusion

An end-to-end model of math problem solving has been presented. In the model, representation change is explained as the result of a choice of a local theory and the reformulation of a primary problem representation in it. Experimental results on more than 400 problems show that our prototype implementation reflects the difficulties of the problems quite precisely. Specifically, IMO problems require the system "theory change" more often than others when interpreting the timeout as "impasse". It indicates the model correctly captures the difficulty of the problems and hence it can serve as a basis of a quantitative study on representation change. Future work includes further analysis of the difficulty of math problems in light of our information-processing account and development of computational models of theory choice.

References

Arai, N. H. (1996). Tractability of cut-free gentzen type propositional calculus with permutation inference. *Theoretical Computer Science*, *170*(1), 129–144.

Arai, N. H., & Masukawa, R. (2000). How to find symmetries hidden in combinatorial problems. In *Proceedings of the eighth symposium on the integration of symbolic computation and mechanized reasoning.*

Buss, S. R. (1987). Polynomial size proofs of the propositional pigeonhole principle. *The Journal of Symbolic Logic*, *52*(04), 916–927.

Carpenter, B. (1997). *Type-logical semantics*. MIT Press.

Chou, S.-C. (1988). *Mechanical geometry theorem proving* (Vol. 41). Springer Science & Business Media.

Cook, S. A., & Reckhow, R. A. (1979). The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, *44*(01), 36–50.

Fischer, M. J., & Rabin, M. O. (1974). Super-exponential complexity of presburger arithmetic. In *Proc. of the siam-ams symposia in applied mathematics* (Vol. 7, pp. 27–41).

Haken, A. (1985). The intractability of resolution. *Theoretical Computer Science*, *39*, 297–308.

Heim, I., & Kratzer, A. (1998). *Semantics in generative grammar*. Wiley.

Isaak, M. I., & Just, M. A. (1995). Constraints on thinking in insight and invention. In R. J. Sternberg & J. E. Davidson (Eds.), *The nature of insight* (pp. 281–325). MIT Press.

Iwane, H., Yanami, H., & Anai, H. (2014). Synrac: A toolbox for solving real algebraic constraints. In *Mathematical software–icms 2014* (pp. 518–522). Springer.

Kamp, H., & Reyle, U. (1993). *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory*. Kluwer Academic.

Kaplan, C. A., & Simon, H. A. (1990). In search of insight. *Cognitive psychology*, *22*(3), 374–419.

Kerber, M., & Pollet, M. (2006). A tough nut for mathematical knowledge management. In *Mathematical knowledge management* (pp. 81–95). Springer.

MacGregor, J. N., Ormerod, T. C., & Chronicle, E. P. (2001). Information processing and insight: A process model of performance on the nine-dot and related problems. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *27*(1), 176.

Newell, A., & Simon, H. A. (1972). *Human problem solving* (Vol. 104) (No. 9). Englewood Cliffs, NJ: Prentice-Hall.

Ohlsson, S. (1992). Information-processing explanations of insight and related phenomena. In *Advances in the psychology of thinking* (pp. 1–44). Harvester Wheatsheaf.

Öllinger, M., Jones, G., & Knoblich, G. (2014). The dynamics of search, impasse, and representational change provide a coherent explanation of difficulty in the nine-dot problem. *Psychological research*, *78*(2), 266–275.

Steedman, M. (2001). *The syntactic process*. MIT Press.

Tarski, A. (1951). *A decision method for elementary algebra and geometry*. University of California Press.

Zettlemoyer, L. S., & Collins, M. (2005). Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proc. of the 21st conference in uncertainty in artificial intelligence* (pp. 658–666).