Conceptualizing Language for Mathematical Problem Types Across Domains

Nicholas H. Wasserman
Teachers College, Columbia University

Paul Christian Dawkins
Texas State University

In this preliminary theoretical report, we problematize the language used in mathematical problems – in particular, with respect to the objectives and the directives of problems. We use this to conceptualize a set of eight problem types, described in three clusters, that run across mathematical domains. This paper explores the motivation for such work, as well as a rationale for why existing problem classifications in mathematics do not accomplish our intended aims; we identify potential applications of the framework, as well as challenges, for discussion.

Keywords: Problem types, Mathematical domains, Language, Mathematics register

Much of the study of mathematics is reducible to the problems students are asked to solve. While university syllabi identify the content of a course in terms of concepts and topics, the list of tasks assigned operationalize what students must learn and do. Problem solving, and the posing of mathematical tasks, are both central themes of mathematics education (Schoenfeld, 1992); as is the language used within them – which may be referred to as part of the mathematics register (Halliday, 1975; Pimm, 1987) or even its own genre (Bakken & Andersson-Bakken, 2021). However, if we consider the role of language in posing tasks to students, two key issues arise. By the time students take university mathematics they have years of experience with the genre of mathematics problems. Yet, whereas mathematical terms are often both well-defined and consistently used, the language used for posing mathematical problems is neither. For instance, the directive "find" in one problem context might mean perform an operation (e.g., "Find $f(2)$ when $f(x) = x^2$"), in another provide a singular example (e.g., "Find an x such that $f(x) = 4$ if $f(x) = x^2$), and elsewhere solve an equation (e.g., "Find all x such that $x^2 = 4$"). This suggests that decoding the nature and objective of a task could serve as part of a hidden curriculum students must navigate. To lay the groundwork for an investigation of the relationship between the language used in posing mathematics tasks and the actual objective being asked of students, this preliminary theoretical paper considers: *How can we categorize mathematical problems, across a variety of domains, that might help specify the nature of the task assigned?* We wanted categories that applied across domains because students experience the language of mathematics tasks across different courses. Although the current paper – a preliminary theoretical report – does not compare whether the same language is used for different tasks, or whether one task is expressed in various language, it lays the foundation for such investigation.

Motivation and Background

As a somewhat humorous illustration, consider the common meme reproduced in Figure 1 – of a student's solution to a mathematical problem. Here the student interprets the imperative "find" in an everyday sense, rather than according to any mathematical sense(s).

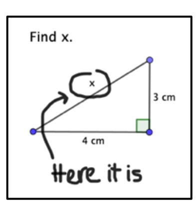

Figure 1. A student solution to the problem's directive to "Find…"

A mathematical problem uses language to give a directive (sometimes in question form) and tends to have a particular kind of mathematical goal associated with it. The problem's directive – e.g., the imperative verb – is supposed to signify this objective to the reader. Yet, we often do not give clear meaning and definition to these problem-posing terms. So, on one side, they are left ambiguous with *no* mathematics-specific meaning, partly because they are part of a meta-language about the doing of mathematics. On the other, we avoid defining them because, as we illustrated above with "find," they are polysemous and may be used to express a variety of goals, depending on the specific task (e.g., Schleppegrell, 2007). That is, an imperative verb can have *many* mathematics-specific meanings. In Figure 1, the task used the imperative "Find…" but the goal is to solve for the real-numbered value(s) of x given the properties of the figure – it is a problem about a solution set.

Rationale, Goals and Assumptions

Why do we need another way to categorize mathematics problems? Naturally, mathematics educators have developed categorizations, which reasonably tend to relate to 1) learning goals, 2) students' prior knowledge, 3) intended solution methods, or 4) expressions of epistemologies or mathematical practices. These features often make the categories specific to a mathematical topic (features 1-3), contextual within an instructional setting (features 1-3), or situated within broader social practice (feature 4). For instance, Schoenfeld's (1992) famous distinction between exercise and problem relates to the solver's prior knowledge. Stein et al.'s (2009) framework for cognitive demand similarly takes student experience into account. "Open" versus "closed" problems (c.f., Bahar & Maker, 2015) tend to account for solution methods, whereas Stanic and Kilpatrick (1989) elaborated five ways that problems can be a means to other ends: problems get used as justification, as motivation, as recreation, as vehicle (for learning), and as practice. This primarily categorizes problems in terms of their learning goals. Of course, one might also specify learning goals in other ways. Many frameworks categorize problems based on content area – whether broad (e.g., algebra, geometry) or more specific (e.g., systems of equations). Others categorize goals in terms of cognitive activities, such as recall or analyze, intended to capture processes. Some do both (e.g., the TEDS-M study, Tatto et al., 2008).

Such frameworks help us understand certain aspects of problems – often informative in textbook analyses (e.g., Gracin, 2018; Herbel-Eisenmann, 2007; Li, 2000). However, although we can imagine similar uses, we instead want categories of what students must produce in response to the problem, regardless of solution method, that can be applied across contexts and without reference to broader practices – which existing frameworks do not accomplish. These choices express our interest in understanding the *nature* of the problem posed in relation to the *language* used in the problem. We do not mean to deny the importance of all of these factors in how students interpret new tasks, as though the language used to express a problem can be isolated from all of these other factors. Rather, we seek to understand what information is (and is not) conveyed by the specific imperative or interrogative language used in a problem. It may be that most of the information about what the problem requires must be inferred from other sources of information, such as prior experience with similar problems. But this would imply that the language used to express a task remains largely inert and could be made more productive.

In setting down this path, we had four underlying assumptions that guided how we developed our framework. First, we wanted a framework that was broadly applicable across domains, which meant we did not want domain-specific problem types (e.g., "Differentiate"). The aim was for problem types to be comprehensible in any area of mathematics. Doing so affords looking at underlying connections across domains, and also allows us to compare across domains.

Second, our goal was to establish a framework that characterizes the core "objective" that students are asked to accomplish in a problem. As such, we were not necessarily concerned with how students might do so – the procedures used to get there. Accordingly, we focused on the general nature of a problem's solution rather than the path to that solution.

Third, we wanted to aim for a coding framework that was less subjective. That is, guessing at the particular ways students might accomplish a problem is challenging, as is guessing at an author's rationale for a problem. We wanted to classify more objective aspects.

Fourth, we sought a framework that would reduce the linguistic complexity in the posing of mathematical problems. We are dubious of being able to exhaust all problem types given the diversity of interesting tasks out there. Since we ultimately cared about the linguistic genre of mathematics problems, we restricted ourselves to common types even if we missed special cases.

Framework Definitions and Descriptions and Examples of Each

Table 1 provides our initial framework of 8 problem types, along with working definitions. We approached this task with a perspective of mathematics as being oriented around sets of objects, and operations on those objects[1]. For collections of objects, we also might talk about descriptions of or relationships between them. This orientation means we need to have clarity about the nature of the elements within those sets to consider problems with respect to our categories. We can understand the 8 types in terms of three clusters depending upon the requested response: something related to a solution set, the outcome of a procedure to be applied, or a statement or justification. We also attempt to convey each category as not being domain specific (i.e., they would cover many problems in most undergraduate mathematics courses).

Table 1. Problem type framework and definitions

Cluster	Problem Type	Definition
Solution set	SOLVE	Respond with *the complete solution set of mathematical objects* that satisfies a given set of properties; the complete solution set may come in a particular representation, such as a variable expression or a graph
	EXEMPLIFY	Respond with *an example solution of a mathematical object* – or demonstrate the existence of a solution – that satisfies a given set of properties; or show no such solution exists
	COUNT	Respond with how many; this may be the *cardinality* of the solution set
	APPROXIMATE	Respond with *an approximate solution* that satisfies a given set of properties; an estimate, sketch, or hypothesis of the solution
Operation output	EVALUATE	Respond with the outcome of a procedure or operation; simplify, compute, or determine the outcome of a specific process
Statement or argument	CONJECTURE	Respond with a mathematical statement about – possibly the answer to a question about – given objects, properties, or relationships between them
	JUSTIFY	Respond with a justification or explanation for a mathematical statement about given objects, properties, or relationships between them
	DERIVE	Respond with a mathematical statement about given objects, properties, or relationships between them [CONJECTURE], and provide a justification or explanation for the truth of that statement [JUSTIFY]

SOLVE, EXEMPLIFY, COUNT, APPROXIMATE

With these four, problem responses can be framed as being chosen from a larger set of objects, with the objective being to identify what objects satisfy a given set of properties. These four problem types specify four different objectives with respect to a solution set. SOLVE means

[1] Granted, we can talk about an operation as also being reducible to a set, but we kept both ideas here.

to specify the entire solution set; EXEMPLIFY to specify one element of the solution set; COUNT to specify how many are in a set; and APPROXIMATE to approximate a solution.

In mathematics, one might think of numbers as the default set of objects under consideration. But in combinatorics it is a set of outcomes; in graph theory it is a graph (a collection of vertices and edges); in geometry it is collections of points, lines, and planes on a particular kind of surface or space; and so on. Solutions to problems, then, might consist of very different types of objects. We classify a problem as a SOLVE problem if it requests all solutions be found. Equations and inequalities common to the domain of algebra (e.g., $x + 2 = 2x - 7$), which specify a relation between two expressions, are common ways we find problems about solution sets, but they are not the only way. In graph theory, for instance, we might see, “Find all graphs (up to isomorphism) with vertex-degree sequence {2,2,2,2,2,2}.” The critical point is that, amongst a larger collection of objects, a given set of properties is being specified and the objective of the problem is related to identifying objects that maintain those properties.

Notably, when we talk about a set of objects, or a solution set, the way in which those objects are represented is not especially important. The problem would still be about a solution set regardless of whether we chose to represent a solution as a decimal or fraction; or an expression (possibly with variables) or a visual representation of that solution as a point on a number line or a Cartesian graph. The representational form of the solution is not a key differentiator.

EVALUATE

In contrast to the first four, this problem type is centrally about an operation. Here, the task instructs students to apply some procedure rather than specifying the properties the final response must satisfy (as do problems of the first four types).

Operations is a broad category indeed. Symbolic expressions in algebra (rather than equations and inequalities), followed by an equals sign, are a common way EVALUATE problems are posed (e.g., $5 + 7 =$, or $\int_0^4 x^2 =$), though certainly not the only way. Here, the “=” is indicative of the directive to perform the operation and record its result. Nonetheless, it puts basic arithmetic in the same category as some calculus problems. There are also many other types of operations in mathematics for which we tend to use words (not symbolic expressions). For example, problems that ask to “Simplify” an algebraic expression or “Transform” a geometric figure similarly indicate an operation for the solver to perform on an object. Frequently, there are reasonably clear procedures to accomplish these operations, but the critical point is that the objective of the problem is to produce the intended output of that operation.

CONJECTURE, JUSTIFY, DERIVE

Especially in collegiate mathematics, students are asked to produce mathematical statements or justifications (in this context, this includes explanations, arguments, and proofs). In most any mathematics course, students can be asked to explain or conjecture – though the proportion of such tasks is often low. The last three categories of problems are distinct because the response should be a statement or justification. Responses to CONJECTURE tasks should be statements specifying properties or relationships, to JUSTIFY tasks should be justifications, and to DERIVE tasks should be a combination of both. Opposite of the first four, in these tasks a (smaller) collection of objects is often given, and the request is to identify distinguishing properties.

Justifications are prominent in proof-based courses such as analysis or algebra, but problems inviting statements and justification can abound in every mathematical domain. In graph theory, for example, one might be asked to "Justify that a tree with n edges must have $n + 1$ vertices." Providing explicit rationale for any claim is essential across all of mathematics.

Discussions, Opportunities, and Questions

As we hope was evident from the examples, these 8 problem types provide distinguishable objectives that a mathematical problem might set for a student, while remaining general enough to be applicable across the various mathematical domains. In what follows, we comment on the opportunities this theoretical perspective might offer the field, as well as raise some preliminary tensions and questions.

This framework helps (objectively) describe and classify the objective of different types of mathematical problems. It gives a defined set of language that we can compare to the very messy collection of imperative verbs used in posing problems. Such theory would help us inquire into questions about the intersection of language and mathematical problems, such as: *What everyday language is used to convey these particular problem objectives? Is the variety and variation of problem posing language a source of confusion for students in general, or on particular problem objectives?* (In an initial use of the framework coding textbook problems, there were dozens of verbs used for EVALUATE problems, and the imperative verb "find" could have signified *any* of the problem types!) Since the framework is not domain-specific, another opportunity provided by the theory is to inquire into the types of problems commonly asked in various domains. That is, *are there differences between mathematical domains in terms of the types of objectives posed in problems?* (e.g., is it true that algebra is really "solving"-heavy, or geometry more skewed toward "justification"?) The critical point is that we find potential value in pursuing this theoretical development because it affords opportunities to answer questions around the nature of problem prompts and language that do not exist quite yet.

Beyond broad questions for the audience in this preliminary theoretical report (e.g., *What challenges or opportunities do you see with the framework?*), here we highlight two tensions, with the purpose of leading into some more specific audience questions. First, consider a simple equation such as $2x + 5 = 8x - 1$. Very apparently, this is a search for a solution set – a SOLVE problem; which $x \in \mathbb{R}$ have this property? However, one might also conceptualize this as a search for (x, y) *pairs* (in $\mathbb{R}^2$) that are shared amongst the linear solutions sets of $y = 2x + 5$ and $y = 8x - 1$, the x-values of these pairs being solutions. Furthermore, there are also standard algebraic procedures to solve – an EVALUATE. If we consider an equation as our "base" object (and not a real number), then we might simply be performing an operation(s) on that object – and the outcome of performing that operation(s) is indeed the solution set. This points to the need to clarify "base object" in problems with this framework; or, alternatively, to preference one coding over another in specific situations. *Does having to specify base object go against the aim of less subjectivity? Or, alternately, is having to specify those objects in talking about the goals of problems a further benefit to communication?* Second, the inclusion of COUNT seems reasonable, given that there are many interesting questions about "how many" across mathematical domains – e.g., how many solutions to a polynomial of degree n? how many distinct triangles with lengths 4, 5, and 6? etc. However, enumeration would also be very closely aligned with the field of combinatorics. *Does this make the category type too domain-specific? Are all counting problems COUNT, or are some EVALUATE – and, if so, what is the distinguishing line? Moreover, if COUNT is a category, should there be a category that captures, for instance, the essence of the field of probability, which is about assigning likelihood to an event set (e.g., solution set) within a broader landscape of outcomes?*

References

Bahar, A., & Maker, C. J. (2015). Cognitive backgrounds of problem solving: A comparison of open-ended vs. closed mathematics problems. *Eurasia Journal of Mathematics, Science and Technology Education, 11*(6), 1531-1546.

Bakken, J., & Andersson-Bakken, E. (2021). The textbook task as a genre. *Journal of curriculum studies*, *53*(6), 729-748.

Gracin, D. G. (2018). Requirements in mathematics textbooks: A five-dimensional analysis of textbook exercises and examples. *International Journal of Mathematical Education in Science and Technology, 49*(7), 1003-1024.

Halliday, M.A.K. (1975). Some Aspects of Sociolinguistics. In, *Interactions between linguistics and mathematics education* (pp. 64-73). UNESCO.

Herbel-Eisenmann, B. A. (2007). From intended curriculum to written curriculum: The “voice” of a mathematics textbook. *Journal for Research in Mathematics Education, 38*(4), 344-369.

Li, Y. (2000). A comparison of problems that follow selected content presentations in American and Chinese mathematics textbooks. *Journal for Research in Mathematics Education, 31*(2), 234-241.

Pimm, D. (1987). *Speaking Mathematically: Communications in the mathematics classroom*. London, UK: Routlege and Kegan Paul.

Schleppegrell, M. (2007). The linguistic challenges of mathematics teaching and learning: A research review. *Reading and Writing Quarterly, 23*, 139-159.

Schoenfeld, A. (1992). Learning to think mathematically: Problem solving, metacognition, and sense making in mathematics. In D. A. Grouws (Ed.), *Handbook of research on mathematics teaching and learning* (pp. 334-370). Reston, VA: NCTM.

Stanic, G., & Kilpatric, J. (1989). Historical perspectives on problem solving in the mathematics curriculum. In R. Charles & E. Silver (Eds.), *The teaching and assessing of mathematical problem solving* (pp. 1-22). Reston, VA: NCTM.

Stein, M. K., Smith, M. S., Henningsen, M. A., & Silver, E. A. (2009). Implementing standards-based mathematics instruction: A casebook for professional development (2nd ed.). Reston, VA: NCTM.

Tatto, M. T., Schwille, J., Senk, S., Ingvarson, L., Peck, R., & Rowley, G. (2008). *Teacher Education and Development Study in Mathematics (TEDS-M): Policy, practice, and readiness to teach primary and secondary mathematics. Conceptual framework.* East Lansing, MI: Teacher Education and Development International Study Center, College of Education, Michigan State University.