# The General, the Abstract, and the Generic in Advanced Mathematics

**Guershon Harel**
Purdue University
West Lafayette
Indiana 47907
USA

**David Tall**
University of Warwick
Coventry
CV4 7AL
UK

The terms generalization and abstraction are used with various shades of meaning by mathematicians and mathematics educators, each representing both a process and the product of that process. In this article we attempt to rationalize the use of these terms in a cognitive context, in a manner intended to shed some light on the different qualities of generalization in advanced mathematics. Using this analysis we will be able to suggest pedagogical principles designed to assist students' comprehension of advanced mathematical concepts.

## Generalization: Expansive, Reconstructive, and Disjunctive

The term "generalization" is used both within and outside mathematics to mean the process of applying a given argument in a broader context. However, the cognitive processes demanded by mathematical generalization will depend on the individual's current knowledge. For instance, students A, B and C may all know how to solve linear equations in one variable. However, suppose that Student A has a relational understanding (in the sense of Skemp 1976) of the solution process: (s)he understands that when adding an expression to each side, or multiplying by a (non-zero) constant, the solution set does not change. Meanwhile students B and C have only an instrumental understanding, simply carrying out manipulations on the equation ("do the same thing to both sides," "get the *x* terms on one side and the numbers on the other," "change side and change signs," "collect together like terms"). They all generalize the method to solving linear equations in two variables, by "eliminating *x* to obtain a solvable linear equation in *y*, then substituting the value found for *y* back into the equations to obtain an equation for *x*". Student A is expanding and enriching his (or her) schema for solving equations, whereas students B and C are simply adding new, unconnected facts to their list of things to do ("eliminate *x* and solve for *y*," "substitute *y* back to find *x*").

Later a further generalization is given, from solving 2¥2 equations to 3¥3, and "more generally," to *m*¥*n* equations using row operations on the matrix of coefficients. It happens that student A understands the essence of the process and can proceed by enriching and expanding his or her schema, seeing the early methods of equations in one and two

This article appeared in *For the Learning of Mathematics*, **11** 1, 38–42 (1989).

variables as simply being special cases of the new generalized procedure. Student B and C now diverge in what they do. Student B begins to see the underlying meaning of the solution process and struggles with the new ideas to attain a cognitive reconstruction by which (s)he can see the 2¥2 case as a special case of the *m*¥*n* procedure. In contrast, student C adds yet another system of solving equations to his/her list of procedures: solving *n* equations in *m* unknowns. We therefore distinguish *three* different kinds of generalization which depend on the individual's mental construction:

1. *Expansive generalization* occurs when the subject expands the applicability range of an existing schema without reconstructing it.
2. *Reconstructive generalization* occurs when the subject reconstructs an existing schema in order to widen its applicability range.
3. *Disjunctive generalization* occurs when, on moving from a familiar context to a new one, the subject constructs a new, disjoint, schema to deal with the new context and adds it to the array of schemas available.

Note that expansive generalization is a true generalization in the sense that earlier schemas are included directly as special cases in the final schema. Reconstructive generalization differs in that the old schema is changed and enriched before being encompassed in the more general schema, but then it gives a true generalization of the enriched schema.

Disjunctive generalization seems to an observer (such as a university professor marking a students attempts at solving linear equations) to be a successful case of generalization. It certainly allows the individual to cope with a broader range of mathematical examples. But it fails to be a *cognitive* generalization in the sense that the earlier examples (linear equations in one and two variables) are not seen by the individual as special cases of the general procedure. The reverse move from the *m*¥*n* case to the case of linear equations in a single variable is seen not as a specialization, but as a shift to a disjoint schema. Disjunctive generalization is also a recipe for failure for the weaker student in that it increases the number of procedures that the individual requires to solve the more general class of problems. It gives the weaker student an additional burden to carry under which he or she is prone to collapse.

Long term, therefore, expansive and reconstructive generalizations are far more appropriate for cognitive development and, of these, expansive generalization is cognitively easier than reconstructive generalization. However, although expansive generalization may be easier in the short term, in the long term there are times when the reorganization of knowledge becomes essential, in which reconstructive generalization is far more appropriate.

At such a time, even when the student's schema is a relational one, the relations may not extend in an appropriate direction to allow an expansive generalization, and a re-construction becomes essential. For instance, the successive generalizations of vector sum and scalar multiples from $\boldsymbol{R}^2$ to $\boldsymbol{R}^3$ to $\boldsymbol{R}^n$ is essentially a case of applying the same techniques to each coordinate in successively broader systems. The algebraic aspects of this process are likely to be an expansive generalization for most students. But the geometric aspects – modifying the geometrical ideas in two and three dimensional space to a mental concept of space of $n$ dimensions – is likely to require a re-constructive generalization which few achieve.

The passage from $\boldsymbol{R}^n$ to the abstract notion of a vector space $V$ over a field $F$, on the other hand, requires a re-constructive generalization in most students. The learner is presented with a name for the concept ("the vector space $V$") and some of its properties (the axioms) and – usually guided by an expert – must follow a subtle and difficult process of construction of the meaning of $V$ and its properties by deduction from the axioms. This is further complicated in the learner's mind by the fact that the properties *to be deduced* in $V$ *are known* to hold in $\boldsymbol{R}^n$, causing the problem for the student that, although these properties are "obvious" in the (only) examples (s)he understands, judgement must be suspended on their truth in $V$ until they are shown to follow by deduction from the axioms. It is a process which requires massive cognitive reconstruction.

In principle we believe that the most desirable approach to generalization is to provide experiences which lead to a meaningful understanding of the current situation, to allow the move to the more general case to occur by expansive generalization, but that there are times when the situation demands a re-construction and, in such cases, it is necessary to provide the learner with the conditions in which this reconstruction is more likely to take place.

## Abstraction

An abstraction process occurs when the subject focuses attention on specific properties of a given object and then considers these properties in isolation from the original. This might be done, for example, to understand the essence of a certain phenomenon, perhaps later to be able to apply the same theory in other cases to which it applies.

Such application of an abstract theory would be a case of *reconstructive generalization* – because the abstracted properties are reconstructions of the original properties, now applied to a broader domain. However, note that once the reconstructive generalization has occurred, it may then be possible to extend the range of examples to which the arguments apply through the simpler process of expansive generalization.

For instance, when the group properties are extracted from various contexts to give the axioms for a group, this must be followed by the re-construction of other properties (such as uniqueness of identity and of inverses) from the axioms. This leads to the construction of an abstract group concept which is a re-constructive generalization of various familiar examples of groups. When this abstract construction has been made, further applications of group theory to other contexts (usually performed by specialization from the abstract concept) are now expansive generalizations of the original ideas.

## The case of definition

The process of formal definition in advanced mathematics actually consists of two distinct complementary processes. One is the *abstraction* of specific properties of one or more mathematical objects to form the basis of the definition of the new abstract mathematical object. The other is the process of *construction* of the abstract concept through logical deduction from the definition.

The first of these processes we will call *formal abstraction*, in that it abstracts the form of the new concept through the selection of generative properties of one or more specific situations; for example, abstracting the vector-space axioms from the space of directed-line segments alone or from what it is noticed to be common to this space and the space of polynomials. This formal abstraction historically took many generations, but is now a preferred method of progress in building mathematical theories. The student rarely sees this part of the process. Instead (s)he is presented with the definition in terms of carefully selected properties as a *fait accomplit*. When presented with the definition, the student is faced with the *naming* of the concept and the statement of a small number of properties or axioms. But the definition is more than a naming. It is the selection of generative properties suitable for deductive construction of the abstract concept.

The abstract concept which satisfies *only* those properties that may be deduced from the definition *and no others* requires a massive reconstruction. Its construction is guided by the properties which hold in the original mathematical concepts from which it was abstracted, but judgement of the truth of these properties must be suspended until they are deduced from the definition. For the novice this is liable to cause great confusion at the time.

The newly constructed abstract object will then generalize the properties embodied in the definition, because any properties that may be deduced from them will be part of it. Because of the difficulties involved in the construction process, this is a *reconstructive generalization*.

Occasionally the process leads to a newly constructed abstract object whose properties apply only to the original domain, and not to a more general domain. For instance, the formal abstraction of the notion of a complete ordered field from the real numbers, or the abstraction of the group concept from groups of transformations. Up to isomorphism there is only one complete ordered field, and Cayley's theorem shows that every abstract group is isomorphic to a group of transformations. In these cases the process leads to an abstract concept which does not extend the class of possible embodiments.

We include these instances within the same theoretical framework for, though they fail to generalize the notion to a broader class of examples, they very much change the nature of the concept in question. The formal abstraction process coupled with the construction of the formal concept, when achieved, leads to a mental object that is easier for the expert to manipulate mentally because the precise properties of the concept have been abstracted and can lead to precise general proofs based on these properties.

Formal abstraction leading to mathematical definitions usually serves two purposes which are particularly attractive to the expert mathematician:

(a) Any arguments valid for the abstracted properties apply to all other instances where the abstracted properties hold, so (provided that there are other instances) the arguments are more general.

(b) Once the abstraction is made, by concentrating on the abstracted properties and ignoring all others, the abstraction should involve less cognitive strain.

These two factors make a formal abstraction a powerful tool for the expert yet – because of the cognitive reconstruction involved – they may cause great difficulty for the learner.

## Generic abstraction

Having focussed on the fact that a formal abstraction is valuable for the expert, yet difficult for the learner to attain, we pose the question as to how we may help students pass through the difficult transition and to attain the reconstructive generalization required for the formal abstraction. We suggest that this can be done more effectively by focussing on a mid-way development in which a specific example is seen by the teacher as a representative of the abstract idea, which we term a generic example.

Initially the student will be presented with one or more *prototypes* for the abstract concept. To the teacher these ideas represent instantiations of the abstract concept, but the student has not yet performed the abstraction, and so these prototypes may function in a seriously erroneous way in which the student abstracts the wrong properties. This seems to happen with the introduction of the function concept in mathematics. So difficult is this abstract concept that it seems not possible to present it in a sufficiently generic manner. Instead we see that pupils presented with an informal introduction to the function concept develop a menagerie of examples from which they abstract inappropriate properties. Tall & Bakar (to appear) suggest that the prototypes that students in the UK develop for the function concept are ideas such as $y=x^2$, any “typical polynomial”, $y=1/x$, a sine curve, a relationship between two variables in which $y$ varies with $x$, a “continuously varying graph” and so on. The result is that when asked if a graph or a formula represent a function, in the absence of a formal definition, the students seem to scan their prototypes to see if there is any resonance. These may produce a false resonance (such as the sense that a circle $x^2+y^2=1$ is a function because it is given by a formula relating $x$ and $y$ or because it has a familiar graph) or erroneously fail to produce a resonance (such as the fact that $y$=constant is not a function because here $y$ does not vary, or is not dependent on $x$).

However, if the process is successful and the student sees one or more specific examples as typical of a wider range of examples embodying an abstract concept, then this is a (relatively painless) form of abstraction which we call a *generic abstraction*.

This process clearly involves generalization (because it embeds the examples in a broader class of example embodied by the generic abstraction). But it is also a mild form of abstraction because it lifts the student’s cognitive consciousness to a higher level in which the more general concept is sensed and abstracted, at least implicitly, from the generic examples.

A wide range of computer experiences to learn mathematical concepts seem to be instances of generic abstraction. Tall (1986, 1989) defined a class of software which enables the user to abstract a higher order concept from examples of the concept a generic organizer. For instance, a generic organizer for the notion of the gradient of a graph consists of a magnification program to see that a tiny part of many familiar graphs will look straight under magnification. The learner who comprehends this property may now glance along a curve and see the changing gradient. A second generic organizer moves a secant along the graph (through two moving points a fixed small difference apart) and plots gradient values enables the student to see the gradient plot as a representation of the changing gradient of the graph. The generic abstraction that the student attains is not the formal idea of the derivative, but a gestalt appreciation of the dynamically growing gradient function. This generic process, seen only with specific graphs, is abstracted by the student in a manner which allows him or her to apply the process to a new graph. The software offers the student the possibility to guess the formula for the gradient and test if this is appropriate. Thus the generic abstraction links intuitively with the later formal abstraction of the derivative function as a symbolic process. Students also develop an ability to apply their generic abstraction of the gradient process to have a significantly better ability to sketch gradients of a given graph.

Thomas and Tall (1988) generate a concept of variable through programming the computer in BASIC and using software which accepts normal algebraic notation. Through comparing specific evaluations of expressions such as $2(x+y)$ and $2x+2y$ for different numerical values the pupils are encouraged to carry out the generic abstraction that such expressions are equal. In this process the letters $x$ and $y$ are not general symbols, but letters standing for numbers. This computer approach forms a basis for further development in the solution of equations and inequalities in which the letters stand for unknown numbers (or even one of a set of possible numbers), leading to further abstraction of the variable concept. Students develop significantly better understanding of the concept of a letter as a variable. For instance, they are more likely to see that the equations

$$3p–1=5 \text{ and } 3(p+1)–1=5$$

are “essentially the same equation”, in which, having found the solution of the first to be $p$=2, realize that the solution to the second is $p$+1=2, so $p$=1. Meanwhile, control pupils who have taken a more traditional route are more likely to see the solution of an equation as a *process*, which means that they multiply out the brackets in the second case and follow through to “move the terms involving $p$ to one side and the numbers to the other” to obtain the solution.

Breidenbach *et al* (to appear) use the computer language ISETL to introduce the notion of function through programming. The syntax of ISETL encourages the student to think of the function as a *process*, in which the function and the variable(s) it takes is named and the process defined by which the output of the function is calculated for given input. For instance, a function in the form

$$\mathrm{V(t)} = \begin{cases} 26.7t^2 & \text{if } 0<t\leq 50 \\ (4/3)\pi t^3 & \text{if } 50<t \end{cases}$$

is defined in ISETL as

```
    V := func(t);
            if t < 0 return “out of domain”; end;
            if t <= 50 then return 26.7*(t**2);
            else return (4/3) * 3.146*(t**3);
            end;
    end;
```

This program embodies the notion of checking that the input *t* is in the domain of *V*, then instructing the computer how to carry out the process to give the appropriate output. Such a method of embodying the function process leads to a generic abstraction of the function concept. Students show radical improvements in their ability to deal with properties of functions at the *specific* level, (computing composites, inverses and intricate related problems involving specific functions) but may be less successful at the general level (for instance, in proving or disproving the statement that “if *f* and *g* are 1-1 then the product *fg* is 1-1”).

## Principles for Generic Abstraction

The idea of generic example is coupled with three pedagogical principles which guide the instruction in helping students in the process of abstraction. The three principles, which were developed in a three year teaching experiment of linear algebra, with advanced high-school students (Harel, 1985), were found to be applicable in other contexts, such as those discussed in this paper. We start with these principles and then discuss their contributions to the construction of the abstraction process.

**The entification principle.** This principle states that, for a student to be able to abstract a mathematical structure from a given model of that structure, the elements of that model must be conceptual entities in the student’s eyes; that is to say, the student has procedures that can take these objects as inputs (Greeno, 1983, Harel & Kaput, *in press*). This principle was derived from the finding that advanced high-school students were able to abstract the vector-space structure from the geometric spaces of directed line segments, but not from the spaces of polynomials (see Harel, 1989). It has been established by several researchers (e.g., Dubinsky, *to appear*) that polynomials, or functions in general, are not conceived by students as objects. Directed line-segments, on the other hand, are manipulable objects which serve as inputs for operations familiar to students.

**The necessity principle.** This principle states that the subject matter has to be presented in such a way that learners can see its necessity. For if students do not see the rationale for an idea (e.g., a definition of an operation, or a symbolization for a concept), the idea would seem to them as being evoked arbitrarily; it does not become a concept *of the students* (à la Steffe, 1988). In Harel's (1985) linear algebra program, this principle was implemented by providing instructional activities through which students, individually and working in groups, participated in the construction of the concepts and their relations. Through this work students discussed new ideas, their relations to familiar ones, their contribution to the reconstruction of previous concepts, and their use in solving problems. Special attention was given to the symbolization form and definitions of operations. For example, why certain variables of a concept are included in its symbols and others are not, or why the matrix operation are defined as they are (see Harel, 1989).

**The parallel principle**. When instruction is concerned with a "concrete" model, that is, a model which satisfies the entification principle, the instructional activities within this model should be designed to parallel the processes that will later apply within the abstract structure. This will mean that the instruction potentially involves only an expansive generalization, in which the concrete model is manipulated in a generic way. But it is designed to lay the seeds for a much easier reconstructive generalization at a later stage when the abstraction of the formal concept occurs in a corresponding abstract manner.

The parallel principle is vital to the abstraction process. It provides structures in the mind which allow the abstractions to root in familiar processes and it also enables abstraction to operate more in the nature of an expansive generalization. For example, when dealing with the three-dimensional geometric model of a vector space, a basis could be defined as three non-collinear directed line segments. But such a definition is restrictive and model-dependent because it does not transfer to abstract vector spaces. In Harel (1985) the concept of basis was explored by the students starting from the concept of minimal spanning set. Initially such a spanning set may be less appropriate because the scalars involved might not be unique. But this in turn leads to the *necessity* of considering minimal spanning sets, which need also to be ordered in a specific way to give a unique scalar representation.

None of these three principles were explicitly used in the research mentioned in the previous section. It is an interesting exercise to attempt to identify where they occur implicitly. In retrospect it is possible for us to identify the principles in action in the two developments in which one of us was involved.

Certainly the entification principle is important in the development of the concept of derivative, in that it is assumed that the students already have a mental image of the notion of linear functions, their graphs and their gradients. These are the entities, together with the process of magnification, upon which the development is based. The computer programs used in the development pass from this knowledge to generalize to the notion of locally straight curves. The necessity to study such curves is given by the existence of highly wrinkled curves which are *not* locally straight. It is seen as essential to realize that not all functions are locally straight. Thus theorems in analysis must explicitly state in what way differentiability is required and show how this property is necessary for the deduction of other properties. Furthermore it is necessary to focus on how the numerical derivative is calculated as $\frac{f(x+h)-f(x)}{h}$ and what happens for small $h$. and show how this numerical process parallels the symbolic process of taking the symbolic limit of $\frac{f(x+h)-f(x)}{h}$ as $h$ tends to zero. This lays the foundations for a formal abstraction which may follow.

Likewise the introduction of algebra through computer programming occurs through building on the arithmetic experience of the pupils, playing a physical game in which numbers written on card are stored in boxes marked with the names of the variables to give a concrete meaning to the notion of variable. The necessity to become acquainted with algebra is partly shown through the power of the notation to encapsulate general processes such as $T=P*R/100$: $S=P+T$ to calculate the tax $T$ and total price $S$ given the initial price $P$ and the (percentage) rate $R$. Thus it is only necessary to specify this algorithm and the rate $R$ to give a computer process that takes the price $P$ as input and gives back the sale price $S$ as output. The necessity to study the peculiarities of algebraic notation also arises from the fact that the computer is programmed to function using such conventions and can be relied upon always to use the notation in the same way, thus $2+3*5$ gives 17, instead of 25. The purpose therefore is to understand how the computer calculates any given expression. The parallel principle arises because the students learn to parallel the computation of algebraic expressions in their physical game following the same conventions used by the computer.

## From Generic Abstraction to Formal Abstraction

A generic abstraction of a mathematical concept gives the student a sense of the concept that is *operative*. The student is likely to feel secure in carrying out operations generically within the context (e.g. solve linear equations or find eigenvalues in $\boldsymbol{R}^n$), but may fail to be able to prove formal properties, for instance that

> “if two linear maps $f{:}U\to V$, $g{:}V\to W$ are such that $gf{:}U\to W$ is 1-1, then $f$ is 1-1”.

Until the student has constructed the concepts of an abstract vector space and of linear functions, (s)he may find it difficult to handle a *definition* of 1-1, say in the form:

> $f{:}U\to V$ is said to be 1-1 if and only if:
> given $f(x_1)=f(x_2)$ for $x_1, x_2 \in U$, then $x_1=x_2$.

How can one handle a statement like $f{:}U\to V$ is 1-1, until one has constructed a meaning for its constituent parts?

The proof is simple enough:

> To show that $f$ is 1-1,
> $\underline{\text{suppose that } f(x_1)=f(x_2) \text{ for } x_1,x_2 \in U}$,
> then $g(f(x_1))=g(f(x_2))$,
> so $gf(x_1)=gf(x_2)$,
> $\underline{\text{then}}$ (because $gf$ is 1-1) $\underline{\text{we conclude that } x_1=x_2}$.

The underlined part of this proof is the statement that $f$ is 1-1. It is simple enough to learn by rote, and many students may do this. But this may produce not relational, but instrumental understanding of the proof, simply adjoining it to the separate items of knowledge that the student has acquired through a process of disjunctive generalization.

To be able to attain such a proof within the context of a formal abstraction requires a meaningful understanding of the definitions. As we have seen earlier, this requires a process of re-construction. However, the construction of such concepts in generic examples are more likely to be achieved by the entification principle and the necessity principle, and the provision of an appropriate procedure for proof construction in the abstract context is more likely to follow from the parallel principle. It must still occur within a context where the individual knows that certain properties are true in all familiar examples but must be deduced without prejudice within the abstract concept. But the parallel principle allows a generalization of the procedure to be passed from the examples to the abstract concept by a process more akin to an expansive generalization, clouded only by the conflict that the properties, known to be true in the examples, must be re-constructed in the abstract context. Thus the passage from generic abstraction to formal abstraction remains one requiring reconstruction, but a reconstruction with potentially less cognitive strain.

## References

Breidenbach D., Dubinsky E., Hawks J, and Nichols D. (*in preparation*): "Development of the Process Concept of Function".

Dubinsky (to appear): "Reflective abstraction in Advanced Mathematical Thinking", in *Advanced Mathematical Thinking* (ed. Tall D. O.) *(in preparation)*

Greeno, G. J. 1983: "Conceptual entities" in D. Genter & A. L. Stevens (Eds.). *Mental Models*, 227-252.

Harel G. 1985: *Teaching linear algebra in high school*, unpublished doctoral dissertation, Ben-Gurion University of the Negev, Beersheva, Israel.

Harel, G. 1987: "Variation in linear algebra content presentations", *For the Learning of Mathematics, 7*, 29-31.

Harel G. & Kaput J. (in press): "The role of conceptual entities in the construction of advanced mathematical concepts and their symbols", in D. Tall (Ed.) *Advanced Mathematical Thinking.*

Skemp R.R. 1976: "Relational understanding and instrumental understanding", *Mathematics Teaching* 77, 20-26.

Steffe L. 1988: Children's construction of number sequences and multiplying schemes. In M. Behr and Hiebert (Eds.) *Number Concepts and Operations in the Middle Grades*. Reston, VA, National Council of Teachers of Mathematics.

Tall D.O., 1986:*Building and Testing a Cognitive Approach to the Calculus Using Interactive Computer Graphics*, Ph.D. Thesis, University of Warwick.

Tall D.O. 1989: 'Concept Images, Generic Organizers, Computers & Curriculum Change', *For the Learning of Mathematics*, 9,3 37-42.

Thomas M.O.J. and Tall D.O., 1988: "Longer-Term Conceptual Benefits From Using a Computer in Algebra Teaching", *Proceedings of the 12th Conference of PME*, Hungary, 601-608.