[1] Hello. I'm Eric Hehner, from the computer science department, of the University of Toronto. I want to talk to you about algebra, about the symbols and notations used in algebra, and about the way the subject is taught and explained, and about how algebra can be put into practice. I'm going to end with something called Unified Algebra, but to start with, I'll tell you about Boolean Algebra. [2] Boolean Algebra is the simplest kind of algebra. It has just [3] two values, and there are [4] very few operations you can perform on those two values. Each operation is defined by a [5] small table, saying what the result is for each combination of operand values. Boolean Algebra is taught in [6] university in an optional course. You don't have to learn Boolean Algebra even in the mathematics department. And what's it good for? [7] Its original application was for reasoning about true and false statements. It's used in mathematical proofs, where the two values represent theorems and antitheorems. In circuit design where the two values represent high voltage and low voltage. I use it all the time for specifications to describe satisfactory and unsatisfactory computer behavior. Programmers use Boolean expressions after the word if, and while. I've seen it used in law, where the two values are innocent and guilty. Boolean Algebra applies to anything that comes in two kinds. So who uses Boolean Algebra? Well, I do, but in the world at large, [8] in the general population, I would say very few people; almost nobody.

Now [9] let's look at number algebra. It has [10] infinitely many values and operators, and even the simplest number operation, which is counting, is defined by induction. Real numbers are defined by limits. This is way more complicated than Boolean Algebra. So when do we learn it? [11] In primary school. What is it used for? [12] measurement of any kind of quantity. And who uses it? [13] cashiers, architects, manufacturers, farmers, financial planners, scientists, -- everybody. Why is number algebra so much better known and used than Boolean Algebra? Maybe we need to look back in history for that answer. [14] After numbers were invented, but before algebra was invented, there were certainly problems that needed to be solved, but without algebra to solve them, they were probably solved by trial and error. For example, if someone needed to divide up their three goats and 20 chickens equally between two people, and it was agreed that a goat is worth 8 chickens, then perhaps they [15] drew a line, and then started moving the goats and chickens around [16] until they found a solution. To verify that it is a solution, you just need a little arithmetic. But to find a solution without moving goats and chickens around, you need algebra.

[17] Here are two pages from the earliest known book of algebra in the English language. It says it is [18] the rule of equation, commonly called algebers rule. It talks about [19] nombers denominate, which are numbers, and nombers abstracte, which are variables. It's written as a dialogue between a scholar and a master. [20] Here are the next two pages. There's something very famous right [21] here, which I'll read to you. And to avoid the tedious repetition of these words -- is equal to -- I will let, as I do often in work use, a pair of parallels, gemove lines, that means twin lines, of one length, thus, and then there are the lines, because no two things can be more equal. That's the invention of the equal symbol, although it's way too wide. Then there's some algebra, and then [22] some discussion. I'll read a bit. In the first, there appeareth two nombers, that is, 14x plus 15y. I have it written out [23] so it's easier to read, and even [24] easier if I use a modern font. It says: In the first there appeareth 2 nombers, that is $14x + 15y$ equal to one nomber, which is $71y$. But if you mark them well, you may see one denomination, on both sides of the equation, which never ought to stand. Wherefore abating, that means subtracting, the lesser, that is $15y$ out of both the nombers, there will remain $14x = 56y$ that is, by reduction, $1x = 4y$.

Scholar. I see, you abate $15y$ from them both. And then are they equal still, saying they were equal before. According to the third common sentence, in the pathway: If you abate even portions, that means equal portions, from things that be equal, the parts that remain

shall be equal also.
Master. You do well remember the first grounds of this art.
It's a long explanation saying it's ok to subtract the same amount from both sides of an equation. That's a special case of the transparency law. Later they want to add the same thing to both sides of an equation, so there's another long discussion, and then they say If you add equal portions, to things that be equal, what so amounteth of them shall be equal.

Each step in an abstract calculation is accompanied by a concrete justification. For example, we have the Commutative Law [25]. I'll make that [26] easier to read.
When the chekyns of two gentle men are counted, we may count first the chekyns of the gentleman having fewer chekyns, and after the chekyns of the gentleman having the greater portion. If the nomber of the greater portion be counted first, and then that of the lesser portion, the denomination so determined shall be the same. This version of the Commutative Law includes an unnecessary case analysis, and it has missed a case: when the two gentlemen have the same number of chickens, it does not say whether the order matters. Here's the Associative Law. Well, you can read it if you want to. The point is that for each simple step in a calculation, there's a lot of discussion. [27] The discussion is needed to reassure people that it's ok to apply the law. That's because [28] the algebra was not trusted. And that's because [29] algebra replaces meaning with symbol manipulation. We're supposed to be talking about goats and chickens, not x's and y's. [30] Those who were skilled at informal reasoning, that means natural language reasoning, thought that it helps your calculation to think about the goats and chickens. And, as usual in any [31] advance, the experts, the people who are most skilled in the old way, are the least likely to appreciate the advance. [32] Today, each step in a calculation is justified by an algebraic law, not by discussing the objects that the calculation is talking about. [33] And that's how we can go farther, faster, more succinctly, and with more certainty, in our quantitative reasoning.

[34] Here's a modern proof from a book on my shelf. It doesn't matter what this proof is about. I just want to [35] point out that there's some good sequences of the use of algebraic laws. The proof is actually two pages; [36] here's the second page, and you see lots more good algebra going on here. This algebra is talking about elements of rings and fields. But there's a second calculation going on [37] in the words in between the formulas. It's a boolean calculation. Since, and from which, and this would imply, and contradicting, and so on. These can all be formalized as boolean expressions, and the whole proof turned into one boolean algebra calculation, with subexpressions about ring and field elements. But mathematicians don't like to write their proofs formally because that would replace meaning with symbol manipulation. They think it helps to think about and talk about the objects that the proof is about. Especially the mathematicians who are best at writing informal proofs. Right now, boolean algebra is where number algebra was 5 centuries ago. If we could start to trust it and use it, we could go farther, faster, more succinctly, and with more certainty than today's informal, wordy arguments.

[38] We've all done number algebra since high school. The problem here is to simplify an expression. So we apply some laws, line by line, until we get the simplest equivalent expression we can find. [39] It doesn't have to be all equal signs on the left side of the page. Well, then we can't call it simplification, so I'll call it calculation. Here we calculate that x times x plus 2 is greater than or equal to minus one. [40] Boolean calculation is similar. We just apply laws, line by line. In this calculation, we can say [41] we are simplifying the top line to true. Simplifying to true is exactly what [42] proving is. [43] Here's an example that doesn't have all equal signs on the left side. The calculation shows that the top line is implied by true, and that's also proof. [44] Next we have a mixture of boolean and number algebra. You don't need to look at everything. The top line is a conjunction of equations, so we're solving simultaneous equations. The bottom line is a conjunction of simple equations, and that's a solution. On the left side, it isn't all equal signs, so this may not be the only solution. Actually, it's all equations down to the next-to-bottom line, and we see there that there are two solutions. Anyway, the point I'm trying to make, [45] is that simplifying, proving, and solving are all the same. They are all

calculation.

[46] The terminology that is in use for algebra is confusing. There are terms, and they have values, and separately there are formulas or propositions or sentences and they don't have values. Instead they are true or false. [47] There are operators like plus and times that operate on terms. And there are connectives like and and or that join formulas. And to make things more complicated, [48] terms can be boolean, so they have values true and false, which is different from being true or false. And then there are [49] predicates, but I'm not sure what the definition of predicate is traditionally. There are [50] at least three different equal signs, and although they all have identical algebraic properties, namely they are reflexive, symmetric, transitive, and transparent, there's supposed to be some sort of philosophical difference between them. [51] Here's an example. *a* and *b* are boolean variables, plus is a boolean operator called disjunction, *a* plus *b* is a boolean term which has value true or false, *a* plus *b* equals *a* and *a* plus *b* equals *b* are formulas, so they are true or false, and finally the v is a logical connective. It doesn't have to be that complicated. [52] If I ask students to prove that some expression *a* is equivalent to some other expression *b*, they give it a try. [53] But if I ask them to prove *a* exclusive or *b* they don't know what I'm asking. That's because exclusive or isn't a verb. [54] If I write *a* exclusive or *b* is equivalent to true, they're ok, because now they have a verb, even though equating to true is an identity operation, like adding zero or multiplying by one. Or I can just change the exclusive or symbol to [55] is not equivalent to they're ok, even though that's exactly the same operation. Now if [56] I equate that to true they're in trouble again because there are too many verbs. [57] You see the same thing in programming texts where they write while flag equals true do something. It would be equivalent, simpler, and more efficient to write: while flag do something, but that doesn't sound good. If our mathematics has to conform to English grammar, or any natural language, it won't serve its purpose, which is to replace the vagueness, ambiguities, wordiness, and inconsistencies of natural language with a language that's concise and consistent and designed for calculation.

[58] The notations of number algebra are standard. All over the world, people from the age of eight onward are familiar with expressions like 738+45=783. But there isn't any standard for boolean algebra notations, not even for the two boolean values. Sometimes it's true and false, or T and F, or one and zero, and I've even seen one and zero the other way around, which is confusing. [59] Quite often the arithmetic symbols are used, with plus for disjunction, and minus for negation, and just putting things next to each other for conjunction. So here are some boolean laws. [60] The first law looks ok because it's the same for number algebra, but these [61] three clash with number algebra. They are not laws of number algebra. [62] In this one, I needn't have written equals 1, but I did just to give it a verb. [63] A lot of mathematicians would say: who cares, just choose some symbols and get on with it. But mathematicians have to apply the laws, and to do that they have to recognize the patterns, and know when the law matches the expression you have, and that means the laws have to be familiar. And we want to be able to mix arithmetic and boolean variables and operators in the same expression, as we did in the mixed calculation a few pages ago. But we can't if we reuse the arithmetic symbols for boolean algebra in this way because it causes ambiguities. [64] Using the words true and false for the boolean values is just as clumsy as using words for numbers. And it identifies these boolean values with one of the many application areas of boolean algebra. [65] I would like to suggest the top and bottom symbols. They are neutral and equally applicable to all application areas. True and false, theorem and antitheorem, power and ground, and so on. [66] Now let's look at the way implication has been treated. There are all these symbols. And there's always confusion about the meaning of implication when the antecedent is false. [67] For example, “If my mother had been a man, I'd be the king of France.”. That's a famous quotation. [68] Even the chair of electrical engineering and computer science at Berkeley can get it wrong. He said: if the garage door is open and the car is running, then the car can be backed out of the garage. He says it means if either or both are false, then the car cannot be backed out. But that's not what it means. It means if they're both true, the car can be backed out, and doesn't say what can or can't happen

if either or both are false. [69] In my area, formal methods, the Z and B specification languages both got it wrong too. But if we stop identifying the boolean values with true and false, implication becomes [70] easy. It is the boolean ordering. Bottom is lower than or equal to top. That's implication.

[71] Now I want to talk a little about the design of mathematical symbols. One principle is symmetry. We should make a symmetric symbol for a symmetric operator, and an asymmetric symbol for an asymmetric operator. That's not just esthetics. [72] That saves us from having to learn so many laws. The principle says: just writing it all backwards is equivalent. Well, I didn't turn the variables backwards, but all the operators are turned backwards. [73] Mathematics has followed that principle for some operators, but not for others.

[74] Now here's a similar principle. Duality. You make self-dual operators vertically symmetric, and the others are not vertically symmetric. [75] Now when you turn an expression upside down, you are negating it. Well, we don't turn variables upside down so we have to negate them. This is a generalization of deMorgan's laws. [76] Again, mathematics has followed that principle for some operators, but not for others. The benefit we would get from following these two principles is to turn lots of laws into trivial visual transformations. And that would make algebra easier.

[77] Some boolean expressions are laws. That means, no matter what values the variables have, the expression has value top. And some boolean expressions are unsatisfiable. That means, no matter what values the variables have, the expression has value bottom. [78] In between, there are boolean expressions that have value top for some values of the variables, and bottom for other values of the variables. Finding an assignment of values for the variables that gives an expression the value top is called solving. The expression might be an equation, but it could also be any other boolean expression. Solving has been the driving force for much of mathematics. You choose an unsatisfiable expression, and you say, what a pity that it doesn't have any solutions. Let's give it one. For example, [79] when the only numbers were the natural numbers, the equation x plus one equals zero didn't have a solution. So we just give it a solution, which we call minus one, and we invent the integers. [80] The equation x times 2 equals one doesn't have an integer solution, so we give it one by inventing the rational numbers. [81] x squared equals two doesn't have a rational solution, so we invent the real numbers. [82] x squared equals minus one doesn't have a real solution, so we invent the complex numbers. [83] That's what happened historically, and it's also the progression we go through when we teach mathematics. [84] As we gain solutions, we lose laws. x plus 1 not equal to 0 was a law of the natural numbers, but it's not a law of the integers. And so on. [85] There are people who say that the naturals are a subset of the integers, and the integers are a subset of the rationals, and so on. And there are other people who say the integers are not a subset of the rationals, but isomorphic to a subset. Personally, I don't care. I do care what the laws and solutions are. And I care that we use the same notation for natural one and integer one and rational one and real one and complex one, because I don't want to have to learn all the solutions and laws over again for each domain. [86] One plus one equals 2, whether those are natural numbers or complex numbers. All laws of complex arithmetic that can be interpreted over the naturals *are* laws of natural arithmetic. And all boolean expressions over the natural numbers have the same solutions over the complex numbers. Because we are using a unified notation as we enlarge or shrink the domain, I don't have to relearn all the laws and solutions.

And that's the reason I want to unify boolean algebra with number algebra. We already know that using one and zero for the booleans doesn't work; the laws clash. For the unification that works, I need to extend the numbers with an infinite value. [87] Top is unified with infinity, and bottom with minus infinity. Boolean negation is number negation. Boolean conjunction is number minimum. Boolean disjunction is number maximum. Implication is ordering, as I already said earlier. Equivalence is equality, and exclusive or is inequality. [88] Now every number law that employs only these symbols corresponds to a boolean law. This is just a random sample of laws. [89] Look at this law. On the left, it says that anything implies true, or top. On the right, it says that anything is less than or equal to infinity. [90] And this one, on the left, says that false implies anything. And on the right it says that minus infinity is less that or equal to anything. [91] And these two. On the left, those are deMorgan's laws. And they have exactly corresponding number laws. But I don't want to have to learn all the laws and solutions twice. So I [92] unify the notations. I'm going with top and bottom, for both boolean algebra and for number algebra. I shouldn't say both, because I am unifying the algebras. There's only one algebra, with one set of symbols. The two sides here are showing the terminology from the two algebras that are being unified. [93] The negation sign is ok for both sides. [94] For the next two, I used the number symbols. Conjunction is the minimum of its operands, and disjunction is the maximum of its operands. You might like to pause this video and think about that for a minute. [95] Nand and nor don't have any standard symbols, so I made nand look like a combination of minimum and negation, and nor looks like a combination of maximum and negation. [96] I've already said that implication is just the boolean ordering. Bottom is less than or equal to top. I've gone with the arithmetic symbols for ordering [97] because that gives us three more comparisons, which are standard for numbers, but they've never been given symbols or names in boolean algebra. They've been treated like a dirty secret that we don't ever talk about. [98] The standard equal sign is perfectly good for booleans and numbers. [99] For unequals, I just stood up the slash, because now, all symmetric operators have symmetric symbols, and all asymmetric operators have asymmetric symbols. I'm sorry to say duality isn't respected. I would have had to stray too far from standard for the last six operators for that.

This is the unification that allows the greatest number of boolean laws to be interpreted as number laws. [100] For example, this is a boolean law. If a then b else c is the same as a and b or not a and c. In unified algebra [101] it looks like this, and if b and c are numbers, it's still a law, just like in a lot of programming languages. [102] Here are two laws written in traditional boolean notations. They are called antidistributive laws, and they give people trouble because conjunction changes into disjunction, and vice versa. The first says that *a* and b implies c is the same as *a* implies c OR b implies c. The second says that *a* or b implies c is the same as *a* implies c AND b implies c. And that doesn't sound right to some people. [103] But in unified algebra, the first says that the minimum of *a* and *b* is less than or equal to *c* when at least one of *a* or *b* is less than or equal to *c*, and the second law says that the maximum of *a* and *b* is less than or equal to *c* when both *a* and *b* are less than or equal to *c*. That sounds right. And these are laws for all numbers, not just the booleans. [104] Here's a little calculation that's not hard to understand. We start with x minus y. That expression varies directly with x and inversely with y. From the first line to the middle line, we're increasing x to x plus one, so that increases x minus y. From the middle line to the bottom line, we're decreasing y to y minus one. We're subtracting less, so again the expression increases. [105] Here's a very similar calculation. The top line says x is greater than or equal to y. That expression varies directly with x and inversely with y, just like x minus y. So again, if we increase x to x plus one, we increase the whole expression, and if we decrease y we increase the whole expression. If you're having trouble with this calculation, try reading the symbols on the left as implies. x is greater than or equal to y implies that x plus one is greater than or equal to y, and that implies that x plus one is greater than or equal to y minus one. With unification, boolean reasoning becomes the same as arithmetic reasoning.

[106] Here's how I think mathematics should be taught. We start with the simplest algebra there is. That's boolean algebra, and these are all its symbols. They are all defined with a small table of values. And of course we show lots of real world applications, and we practice simple calculations. [107] Then, when we're ready, we introduce a third value, zero, in between top and bottom. We keep all the same symbols as before, and add a few more. Now the applications are whatever has three values, like yes, maybe, and no, or large, medium, and small. By adding a new value, we gain solutions that we didn't have before, and therefore we lose some laws that we had before. But we also gain laws that use the new symbol. We could add one more value, or [108] maybe at this point we add in all the natural numbers, and some operations on them. Since we already have negation, that gives us all the integers. [109] Then we add in

division, and we have the rationals, always gaining solutions. We lose a few old laws, and we gain laws that use the new symbols. It's the standard development of mathematics, but starting with the simplest algebra.

[110] Somewhere in the development of mathematics, we introduce functions. The notation Alonzo Church used in 1930 was like this. A variable is introduced; here it's n. Then its domain; here it's the natural numbers. And then the body of the function. So this is the successor function on the naturals, mapping n to n+1. The hat notation comes from Russell and Whitehead's Principia Mathematica, and its purpose is to show the scope of the variable. But a hat is typographically inconvenient, especially when the body of the function is a long expression. So the typesetter convinced Church [111] to move it down in front, because the typesetter had a Greek lambda symbol. But the typesetter didn't have enough capital lambdas, so [112] small lambdas were used, and that was the birth of the lambda calculus. Unfortunately, that doesn't serve the purpose that the hat was for; it doesn't show the scope of the variable. So [113] I'm using this notation for functions. The angle brackets do show the scope, and they're sort-of sideways hats.

The only job of a function is to introduce a variable, so now we can unify [114] all those notations that introduce dummy variables. I'll call them all quantifiers. It just means applying an operator to a function. Applying plus to a function sums the values of the function, so that replaces the big sigma notation. Applying times replaces the big pi notation. Applying minimum gives you the minimum value of the function, and if the body happens to be boolean, that's the for-all quantifier. Applying maximum gives you the maximum value of the function, and if the body happens to be boolean, that's the existential quantifier. [115] And we can do parities and set formation and limits and integration the same way, but I won't get into that in this talk.

[116] If function f has domain D, then the function that maps variable x in domain D to f of x is just f. [117] So the sum, as x varies over D, of f of x can be written as just plus f. [118] Similarly the minimum, as x varies over D, of f of x can be written as minimum f. Well, if f has a boolean result, which we usually call a predicate, the left side of this equation is traditionally pronounced: for all x in D, f of x. [119] And the maximum, as x varies over D, of f of x can be written as maximum f. If f has a boolean result, the left side is traditionally pronounced: there exists x in D, such that f of x. But the right side is a much neater way to write it. [120] Here are deMorgan's laws, written traditionally. [121] In unified algebra we write them this way. Or even more briefly, [122] this way. It says that the negation of a universal quantifier is the existential quantifier of the negated operand. But f might have a numeric result, so it says the negation of the minimum f value is the maximum of the negated f values. [123] In traditional notations, the law of specialization says that if f is true for all values, then in particular, f is true of y. And generalization says that if f is true for y, then there exists a value for which f is true. [124] In unified algebra, it looks like this, and we can write it more briefly like [125] this. It says that the minimum f value is less than or equal to any f value, and any particular f value is less than or equal to the maximum f value. [126] The main point is that these laws hold for all numbers, including the booleans. I'm subtly trying to get you to think of the booleans as numbers, but not in the traditional sense of zero and one; they're numbers in the unified algebra sense of bottom and top so that they share laws and solutions.

[127] Here's a standard step in a basic number calculation. We're finding the minimum value of function f with y subtracted from each function value. The function has some nonempty domain, but I didn't bother to write it. The calculation step is factoring. We don't have to subtract y from each function value. [128] Instead, we can find the minimum function value and subtract y from that. And we can write it [129] briefly as in this bottom line. [130] Here's a similar calculation. The traditional reading of the top line is: for all x, f of x is greater than or equal to y. Now we [131] factor out greater than or equal to y. It says the minimum function value is greater than or equal to y. And that can be written [132] more briefly. On the top line, I pronounced the quantifier as for all. On the middle line, I pronounced it as minimum. But it's

the same operator. It doesn't matter whether it's applying to a boolean function body, as in the top line, or a number body, as in the middle and bottom lines. The law is the same as the subtraction example; just factoring. Or maybe f is a boolean valued function, and y is boolean, and that greater than or equal to sign is reverse implication. It doesn't matter; it's a law no matter what's boolean and what's numeric. [133] Here's another subtraction example, but this time factoring out the other side. In the top line, we're finding the minimum value of y minus each function value. When we [134] factor out this side, minimum changes to maximum. We take the maximum of the function values and subtract that from y. And we [135] can write it more briefly. The minimum difference between y and f of x equals the difference between y and the maximum f of x. [136] Here's a similar calculation. The top line might traditionally be read: for all x, y is greater than or equal to f of x. Or it might be saying: for all x, y is implied by f of x. [137] This line might be saying y is greater than or equal to the maximum f value. Or it might be saying y is implied by the existence of an x such that f of x is true. Since it's all the same law, I want to get rid of all the different ways of saying it, and settle on one terminology, one way of saying it.

[138] Here is another example. The first expression says for all x and y, P of x implies Q of y. The other expression says exists x such that P of x implies for all y Q of y. And the question is whether these two expressions are equivalent. Well, are they? It's not obvious to me. How would you even start to answer? People start by saying something like: suppose some x has property P. Or suppose all y have property Q. And then they consider various special cases. It's an inefficient, wordy, and uncertain way to reason. Let's [139] look at it in unified algebra. The first expression is this one. [140] We just factor P x less than or equal to out the left side of the inner function, and then [141] factor less than or equal to minimum Q out the right side, and we're done. Better than that, we've proven something more. Because we could be talking about numbers. The top line says that all the P values are below or equal to all the Q values. And the bottom line says that the maximum P value is below or equal to the minimum Q value. And that's the same.

[142] My next example is about lists. We can read this as saying the sum of a list divided by the length of the list is less than or equal to the maximum value in the list. In other words, the average value is less than or equal to the maximum value. Now [143] a quick calculation, which I won't read, and we get to the bottom line, which I will read. It says: if the total number of things is more than the number of places to put them, then there's a place with more than one pigeon in it. In a couple of steps we go from saying the average is less than the maximum to the pigeon-hole principle. On the top line I said less than or equal to, and on the bottom line I pronounced it if then. And what I called maximum on the top line, I said exists on the bottom line.

[144] Here's my last example. Suppose f is a function from the natural numbers to the real numbers. In traditional notation, the left side says that each f value is less than or equal to the next one. So f is a nondecreasing function. Let's say, an ascending function. The right side says that the first f value, f zero, is the minimum f value. It says that if f is ascending, then f starts with its minimum value. For a reason that will become clear in a moment, I want to weaken that to say f zero is [145] less than or equal to the minimum value. Now, [146] in unified algebra, it can still be read as saying that if f is an ascending function, then its first value is less than or equal to its minimum. The reason I changed to less than or equal is so that I can apply [147] the portation law. You might be familiar with this law [148] in boolean form. Or you might be familiar with it from type theory. It's the same law. [149] Applying it, I get this. And if f happens to have a boolean range, then I can read it as follows. If f is true of 0, and furthermore if f of n implies f of n plus one, then f is true of all naturals. It's induction. In one step, we show the equivalence between induction and the first item in a nondecreasing sequence is its minimum.

[150] This video comes from the paper “from Boolean Algebra to Unified Algebra”, which you can find on my website. There's more in the paper that I haven't talked about, but the

main point is that [151] there is a big advantage to unifying boolean algebra with number algebra, namely, they share laws and solutions. It's exactly the same motivation for unifying naturals with integers with rationals with reals with complex numbers, which we have already done. And we can [152] also unify values and types, and unify functions with function spaces. And in general unify logic and algebra. If you want to know how, [153] have a look at my paper on unified algebra. In it I present a development of algebra, with all the laws, all the way from the booleans to the reals. [154] Another take-away from this talk is that simplifying, solving, and proving are all the same. They are all just calculation. [155] In the future, I would like to see unified algebra become a tool that lots of people use routinely, not just something that mathematicians study. And for that to happen, [156] it has to be learned early, practiced, and applied.

I hope you found my talk interesting.