AMAST'91 Banquet Talk *

Yuri Gurevich

May 1991, Iowa City

Bull. of Euro. Assoc. for Theor. Computer Science 51, Feb. 1991, 90–106.

Prologue

Tuesday, May 7, 1991. I sign the last grade sheet and smile at the spring sun. Finally the semester is over. A message from Teo Rus arrives. "The second conference on Algebraic Methodology and Software Technology needs a banquet speaker", writes Teo. I am very flattered. And scared. I recall a recent banquet talk in Ann Arbor. The man went on and on. I left before he finished. On the other hand, the invitation is a challenge and an opportunity. You know, sometimes we feel like philosophers if only anybody would listen. I accept the invitation before the scare gets a hold of me.

I leave my office and meet Kevin Compton, another member of the small computer theory group in our huge Department of Electrical Engineering and Computer Science. "How are you?" asks Kevin. "Well, I was fine only a few minutes ago", and I tell him about the invitation to give a banquet talk. "I do not envy you", says Kevin. Soon a message from him tells me about 5 books on public speaking in the library. I thumb the books. They have witty things on almost any subject, but do not mention algebra or software, let alone algebraic methodology and software technology. The volumes of humor are depressing. This is not it. Teo could find a professional joker to entertain the conference. At that time in Iowa it could be a national politician.

After thinking it over, I decide to take a scientific approach and write a scholarly paper. You know, another paper never hurts your vita. The scientific approach explains the use of "we" in the sequel.

The AMAST Phenomenon

The organizing principles are given by the following observation attributed to Don Knuth: The two most important questions about AI are: What is A and what is I?

What is the question complexity of AMAST? There are 5 letters in the word, but A appears twice. A closer examination reveals that there are only 3 questions:

(1) What is algebraic methodology?

(2) What is software technology?

(3) What does AND mean in the AMAST context?

The third question is the toughest of the three.

*Bull. of Euro. Assoc. for Theor. Computer Science 51, Feb. 1991.

Algebraic methodology

According to Webster, methodology is "a system of methods, as in any science". Thus, algebraic methodology is a system of methods employed in algebra. Makes sense.

You may wonder how algebraic methodology is different from algebra. In algebra you search for definitions to formalize your theorems; in algebraic methodology you search for theorems to justify your definitions. It is clear that "algebraic methodology" sounds better on a grant proposal; it implies also some connection to applications.

Some folks ridicule the division of algebra or anything else into pure and applied. "Consider painting", they say, "if your paintings are bought by museums then you are a pure artist, and if your paintings are sold in a supermarket then you are an applied artist. But what if you intended to sell your paintings in a supermarket and a museum bought them? Are you a pure or applied artist?" We say: where do you find those clever folks? They all are in departments like Pure Mathematics or Physics. The distinction between pure and applied science is very important. How would DARPA know whom to support?

What is algebra?

It is clear that algebra is the essence of algebraic methodology. So let us examine what algebra is. Etymology often is a key to the meaning. We asked a few of our learned colleagues about the etymology of "algebra" and then consulted Webster. It turns out that folklore and Webster disagree on the etymology of "algebra".

Folklore: "algebra" as well as "algorithm" come from the name Al-Khowarazmi of a 9th century mathematician.

Webster: "algebra" comes from Arabic "al-jabr" which means the reunion of broken parts.

The folklore explanation would be more useful to us because it connects AM with ST in a very natural way. Nevertheless, being committed to a scholarly approach, we adapt Webster's explanation as more scientific and will try to find a good use for it as well.

Is algebra a part of mathematics?

Yes and no.

Why yes? This is obvious and well documented; see [Jane Doe], [Robert Roe], [John Smith].

Why no? We give 2 proofs: By contradiction and by authority. These proofs are specially designed to work on banquets, after a good meal with plenty of wine and before the dessert.

The proof by contradiction. If algebra is a part of mathematics then mathematics is broken into parts. The reunion of broken parts is algebra. Thus algebra = mathematics, which is not true.

The proof by authority. The famous Communist prophet Vladimir Ilich Lenin spoke about the algebra of social revolution. This places algebra into a different college, let alone a different department.

Is "yes and no" a legitimate answer? Sure. Since "AM and ST" is a legitimate title, "yes and no" is a legitimate answer. The question of what "yes and no" means will be discussed later on when we come to the second A of AMAST.

Algebra and logic

Logic methodology has been used in AMAST talks as much as algebraic methodology. This is not surprising. Algebra and logic are like Michigan and Ohio. Do you know that there was a war between Michigan and Ohio? It was about Toledo. You may think that each side wanted the other one to have Toledo, but this is not true. Each side wanted Toledo for itself. The federal government intervened and gave Toledo to Ohio. This explains the famous Michigan slogan OH-HOW-I-HATE-OHIO-STATE. Further, the federal government gave a portion of Wisconsin to Michigan. This is how Michigan became topologically disconnected. The reaction of Wisconsin is not documented.

The Toledo of algebra and logic is called "universal algebra" in algebra and "model theory" in logic. Maybe, Iran/Iraq is a better analogy because each side has its own name for the disputed part: Persian Gulf vs. Arabian Sea.

In any case, algebra and logic have a large intersection as witnessed by numerous AMAST talks. However we have

Theorem 1 *Algebra* $\neq$ *Logic.*

Proof The proof is by contradiction and related to the Russian journal "Algebra and Logic". It would be silly to have a journal "Algebra and Algebra", and the Russian Academy would not approve such a thing. □

Logics

There are many logics in the literature. Female logic, male logic, email logic, dialectical logic, mathematical logic, etc.

Male logic is all too known to be discussed here.

Email logic is all too painful to be discussed here.

Female logic is all too dangerous to be discussed here. The field of AMAST is dangerous as it is. As a matter of fact, we are going to discuss the dangers of the field. But there are prudent limits to risks taken.

Dialectical logic is sort of an art of being logical and illogical at the same time. In the SU (which means Soviet Union and is quite different from US; concatenation is not commutative), logic was divided into dialectical and formal. The first was always supported, the second was forbidden for years. Why? This is a wrong question, it is a question from a wrong logical system. A Soviet dissident logician Essenin-Volpin divided formal logical systems into two classes: democratic and totalitarian. In a democratic system, the rules tell you what is forbidden. By default, the rest is allowed. In a totalitarian system, the rules tell you what is allowed. By default, the rest is forbidden. (For those of you who understand only the language of categories, democratic and totalitarian systems are the final and initial objects of the appropriate category.) You wouldn't ask why Mr. A had not been allowed to go abroad. This would be a wrong question. You might ask why Mr. B had been allowed to go? That should have a good reason. For example, Mr. B might work for the secret police. Now you can see why the question "Why was formal logic forbidden?" is a wrong one. (Actually, they had a "reason" to forbid formal logic: the connection to philosophical positivism. Is positivism so exceptionally bad? Not necessarily. But it is certainly different from dialectical materialism, the only true philosophy.)

In the rest of this talk, logic means mathematical logic.

What is software technology?

This question is easy. We all know what hardware technology is. Software technology is the direct opposite of hardware, except it is a little harder.

Boom and gloom. Software technology is booming, but it goes through a severe crisis as well: reliability, compatibility, verifiability, etc. You name it. Some hackers do not realize that. They happily hack and change our world. They should be explained to that there is a severe crisis out there. The poor devils badly need guidance and organizing principles. This is where AMAST comes in.

What does "x AND y" mean?

The third question about AMAST was about the AND of AMAST. We stumbled also upon the meaning of "yes and no". Let us generalize and consider a more general question: what does "x and y" mean where x, y are arbitrary things (not statements)? Our discussions with learned colleagues turned up a couple of possible answers.

(1) The set {x,y}. This answer may be blatantly wrong. AM and ST = AMAST which isn't a set of two elements. The organizing committee, all by itself, has more than two elements.

(2) The fact that the intersection of x and y is nonempty. That sounds a little more convincing, but cannot be quite right because x and y are not necessarily sets.

Notice that in both answers, AND is commutative, which is not true in general. It is well known for example that the Communist founders are Karl Marx and Friedrich Engels, not Friedrich Engels and Karl Marx.[1]

(3) "x vs. y". This third answer is not necessarily true as well. For example, the relations between AM and ST are not adversarial; AM loves ST, and ST couldn't care less about AM.

One hazard of the trade: wrong abstraction level

It is clear by now that we overabstracted our third question. A wrong abstraction level is one of the greatest hazards of our trade.

If the abstraction level is too low, you have too many details. There are no theorems to prove or apply.

If you abstract too much, you may find yourself in a sterile atmosphere with no theorems (well, with only shallow theorems) to prove. Alternatively, this may be a delightful trap. You may find yourself in a very fertile atmosphere with numerous attractive theorems but this could be a problem too, because you may lose sight of the original question. For example, we find it very tempting to proceed with the investigation of the meaning of "x and y" in its full generality.

It may be in the eye of the beholder whether you abstract too much or not. For example, define programs equivalent if they compute the same thing, and find yourself in a delightful world of logic. Play with lambda calculus and types. (Didn't you really want to be a logician?) Ignore those silly programs that do not behave properly. The Unix kernel, for example. What does it compute? Nothing. It doesn't even converge. Modulo some side effects, it is equivalent to a trivial infinite loop.

[1] At this point during the talk, Vaughan Pratt said, "See Paris and die".

What is the right level of abstraction? This is the art of our science. That is what AMAST is all about.

So what does AND mean in the AMAST context?

Is it "motivated by"? There is a good precedent for this interpretation. That is what AND often means in the famous phrase "logic and computer science". We believe that "motivated by" isn't the main meaning in our case. As we mentioned above, there is an implication of [desired] applicability in the phrase "algebraic methodology".

Is it "applied to"? Hardly.

The most appropriate meaning seems to be: To be applied to [indirectly][eventually]. In other words, the meaning is "for".

Another delightful trap

You dive into mathematics and ... never come back. This trap is similar to but different from the one we discussed earlier.

For example, you write a book on Principles of Programming Languages. You have to give some formal semantics, of course. Denotational semantics seems fun. It requires domain theory though, and domain theory requires fixed-point theory. You explain all this carefully. The project goes along quite nicely. Suddenly, you panic! You have to say something about programming languages as well. A real language, like C, would be too much detail and trouble, this is obviously too low an abstraction level. You already gave the semantics of lambda calculus which is, all by itself, a programming language par excellence. How about the while language? Good. This should satisfy all those imperative freaks.

More on the AND of AMAST

There are of course other cases of "x and y" where AND means FOR. But there is something special about the AMAST use of AND. Consider, for example, the case when x = math and y = physics. Imagine you would like to apply some beautiful mathematics to some physics that does not quite fit your mathematics. What can you do? You can write science fiction but you cannot change the physical world. The situation is quite different if x = AM and y = ST. In principle, you can change ST. Why do they use those silly imperative languages that do not fit my mathematics? They would be much better off using functional languages or logic programming.

Future research

AMAST is in a business of changing the world of software technology. AMAST activities are hazardous, delightful and blessed with opportunities. They are approved and supported by the highest offices of the land like the Office of Naval Research.

Theorem 2 *AMAST is A MUST.*

Proof sketch At this moment, we can only give a very preliminary sketch of our proof. The next AMAST will be in Europe. In one of the dialects of Europese, AND is UND. This accounts for the crucial change of A to U. □

Acknowledgement I am greatly thankful to Neil Jones and Zoe Gurevich for patiently listening to successive versions of this talk, for useful comments and for encouragement.