

PROFESSOR: In this final segment on predicate logic, there are two issues that I'm going to talk about. The first is some problems with translating A E quantifiers and E A quantifiers into English-- or rather from English into logic. We've seen examples in class that English is ambiguous. And I want to show you two that are I think interesting and provocative. Just as a warning that the translation is not always routine.

And then the second topic is an optional one, just to kind of make some comments about the amazing results in metamathematics, the mathematics of mathematics, or more precisely, the mathematics of mathematical logic. And two fundamental theorems about properties of predicate calculus, which go beyond this class and are optional, I would suggest it's worth listening to the A E in English example. And if you want to skip the short discussion of the meta theorems, that's fine, because it's never going to come up again in this class.

So let's look at this phrase in English, where the poet says, "all that glitters is not gold." Well, a literal translation of that would be that, if we let G be glitters, and I can't use G again, so we'll say Au is gold, then this translated literally would say for every x, G of x, if x is gold implies that not gold of x. So is that a sensible translation?

Well, it's clearly false, because gold glitters like gold. And you can't say that gold is not gold. So this is not what's meant. It's not a good translation. It doesn't make sense.

Well, what is meant, well, when the poet says, "all that glitters is not gold," he's really leaving out a key word to be understood from context. All that glitters is not necessarily gold. He was using poetic license. You're supposed to fill in and understand its meaning.

And the proper translation would be that it is not true that everything that glitters is gold. It is not the case that for all x, if x glitters, then x is gold. So it's just an example where a literal translation without thinking about what the sentence means and what the poet who articulated this sentence intended will get you something that's nonsense. It's one of the problems with machine translation from natural language into precise formal language.

Let's look at another example of the same kind. The poet says this time, "there is a season to every purpose under heaven." This is a variant of a biblical phrase. So what does it mean?

Well, the literal translation would be, there exists an s that's a season, such that for every p

that's a purpose, s is for p . Well, that from the way the quantifiers work means that there's some season, say summer, that's supposed to be good for all purposes. Well, that's not right, because summer is not good for snow shoveling. And if your purpose is to shovel snow, then summer will not do for you as a season.

So even though it's phrased, there is a season to every purpose under heaven, it's not the case that the intended translation is there is a season for every purpose. In fact, the poet really means to flip the quantifiers, which is what's shown here. We're going to switch them around so that we are really saying, for every purpose, there is a season, such that s is for p .

For snow shoveling, winter is good. For planting, spring is good. For leaf watching, fall is good. And that is, in fact, the intended translation here, although I remind you that there's a famous historical man, Sir Thomas More, who was described as a man for all seasons. That would be a case where there was one man who was good for all seasons. He was a polymath-- a writer, a cleric, and the chancellor of England for many years until he had a falling out with Henry VIII, which served him ill.

That's the end of those two examples, whose point is just to warn you that translation from English into math is not something that can be done in a mindless mechanical way.

Sometimes, the quantifiers really are meant to go the other way from the way that they literally appear.

Now, we're going to shift to another topic, which is just two profound theorems from mathematical logic about the properties of predicate calculus that are worth knowing about and that describes sort of the power and limits of logic. So these are called meta-theorems, because they're theorems about theorems. They're theorems about systems for proving theorems. And that phrase, meta, means going up above a level.

So the first theorem is a good news theorem. It says that, if you want to be able to prove every valid assertion of predicate calculus, there really is only a few axioms and rules that will do the job. As a matter of fact, the rules that you need are just rules that we've seen already, namely modus ponens and universal generalization and a few valid axioms which we've seen already.

So let's go back a little bit. So there's a remark here that says that, in practice, if you're really going to try to do automatic theorem proving, you need much more than this minimal system. But it's intellectually interesting and satisfying that a fairly economical set of axioms and inference rules are in theory sufficient to prove every logically valid sentence. And this is

known as Godel's completeness theorem.

Godel was the great mathematician, German mathematician, who spent the latter part of his life at the Institute for Advanced Study in Princeton as an emigre. And he has two major theorems, at least, that are results of his. One is the completeness theorem-- this one. Then there's an incompleteness theorem, which maybe we'll talk about in a few lectures. But for now, the good news is you can prove everything that's valid using a few simple rules.

Now, the bad news is that there's no way to tell whether your attempt to find a proof for something that you think is valid is going to succeed. There's no way to test whether or not a quantified formula is valid. This is in contrast to the case of propositional formulas, where you can do it with a truth table. A truth table may blow up, so it becomes pragmatically infeasible. But at least, theoretically, there's an exhaustive search that will enable you to figure out whether a propositional formula is valid.

That's not the case with predicate calculus. Predicate calculus is undecidable, meaning that it's logically impossible to write a computer program that will take an arbitrary predicate calculus formula in and print out true or false depending on whether or not it's valid. Can't be done.

Now, as I said, we're not going to go further into these theorems. These are the kind of basic results that would be proved in an introductory course in logic. Usually, they take about a half a term to do, maybe a little less.

And it goes beyond our course. You can look over in the math department for introductory courses in logic. And you will learn about these two profound meta-theorems about logic and math.