MATH 195: Gödel, Escher, and Bach (Spring 2001)

Notes and Study Questions for Thursday, April 19

Reading: Notes below replaces Chapter XIII (*BlooP and FlooP and GlooP*)
Air on G's String (pp.431-437)

Replacement for Chapter XIII: *BlooP and FlooP and GlooP*

Unfortunately, we don't have time to linger over the details of Chapter XIII. It's a shame, because besides providing us with an important concept used in the final proof of Gödel's theorem, the chapter also gives you a good idea how computer programs actually work. But,... next time. The few paragraphs that follow are intended to give you the main points that you'll need to understand the Gödel's proof.

The point of the last dialogue, *Aria and Diverse Variations* was to introduce you to the distinction between *searches that are guaranteed to terminate* and *searches that can potentially go on forever*. This shouldn't be such a surprising notion. You already know that you can list all theorems in the **pq** system in an ordered fashion (remember the bucket algorithm), thereby guaranteeing that every theorem is on the list. More than that, just by examining a **pq** string you can predict how long you'll have to look down your list before you find it (or know that you'll never find it). In this respect, testing a **pq** string for theoremhood is like testing wither a number has the Goldbach property. In each case, you can say that you need only test some finite list of strings or numbers in order to come to a decision.

Conversely, although you can list all the theorems in the **MIU** system in a systematic fashion (we did that), you can't know how far down the list you have to go before you reach a given string. Suppose you start with the axiom **MI** (level 1) and apply all the appropriate rules of production (level 2), then take what you produced and apply to *them* all the appropriate rules of production (level 3), and so forth. By the time you reach level 10, you haven't encountered **MU**. Does that mean you never will? No, we know you never will by reasoning outside the system, not by an exhaustive search. You can't tell how deep in the list a theorem might lie. So it is also with the Tortoise property and Wondrous Numbers. You can produce lots of them, but you can never be sure that a given number has or hasn't the property if it doesn't come up in your search. It may be that you just haven't searched long enough for two primes (Tortoise property) or an end of the journey (Wondrous Numbers).

45 **SQ1.** How would you go about figuring out whether a number is prime? Can you decide whether it is prime through a search guaranteed to terminate? How many numbers would you have to go through before deciding with confidence whether 1741 is or is not a prime?

This distinction between searches that have a known boundary (e.g., you need only examine the numbers from 1 to **n**) is very important to our goal of understanding Gödel's theorem, because ANY STATEMENT OF NUMBER THEORY THAT CAN BE DECIDED BY A

BOUNDED SEARCH CAN BE REPRESENTED IN TNT. Procedures that describe bounded searches are called *primitive recursive* (I bring up the term because Hofstadter makes great use of it in the next chapter). Hofstadter also uses the term "BlooP program" to describe the specific instructions of the search. "BlooP" stands for **Bounded loop** -- "bounded" because it has a definite limit on the number of levels of the search, and "loop" because it may use its instructions recursively, looping back to the beginning. Why Hofstadter chooses to capitalize "P", I have no idea.

36

SQ2. Outline an approach you might take for deciding whether a given string is or is not a theorem within the **pq** system. Turn your ideas and the actions you would take into an outline for a set of instructions that you could teach to another human. Important: can you know from the input string how many instruction steps will be necessary to decide the question? If so, your instructions can become a BlooP program.

45

SQ3. Is it possible to write a TNT statement that expresses the idea that **MU** is a theorem within the **MIU** system with **MI** as the sole axiom?

So a BlooP program can be written that determines whether a number has the Goldbach property, but the determination of whether a number is Wondrous takes you on a search that has no predetermined end. The procedure is not primitive recursive and cannot be represented by a BlooP program. But it CAN be represented by a FlooP program (Free **loop**), a program that is recursive but has no bounds on the number of times it loops. A Floop program might determine whether a number has the Wondrous Property. You could devise a strategy (very simple: start with the number, then follow the rules and see if you ever get to one), but you can't know how many iterations you have to use the strategy before reaching a decision.

45

SQ4. Outline an approach you might take for deciding whether a particular string of the **MIU**-system is derivable from **MI**. Turn your ideas and the actions you would take into an outline for a set of instructions that you could teach to another human. Important: can you know from the input string how many instruction steps will be necessary to decide the question? If not, your instructions may become a FlooP program, but not a BlooP program.

Floop programs can be divided into two classes: those that terminate (when, we can't predict) and those that never terminate. The former class is called *generally recursive* functions, and these two can be represented in TNT. Unfortunately, there's no way of looking at a Floop program to tell whether it is terminating or nonterminating. Thus, we can't know whether many interesting statements of number theory can be shown by using the rules of TNT to be true. Maybe, however, we can jump out of the system (as we did with **MIU**) and deal with these recalcitrant statements. Maybe, but Gödel says no. And in a week, we'll say no also.

Air on G's String

Bach's *Air for the G String* that inspired the title of this dialogue is one of the few pieces referred to this semester that almost all of you are likely to know. Even if you're not a Bach aficionado, you may know the piece indirectly through Procol Harum's *Whiter Shade of Pale*, which it inspired. To hear the original, go to Course Documents, Additional Material, Unit IV, and click on the piece. However, unlike some of the other pieces, this one has no connection to the corresponding dialogue except for the pun on the name. Pun... what is meant by "G string"? What is meant by "G's string"?

Hoffstadter uses this dialogue like the last one to illustrate a single major concept. This time it is "quining". If you grasp by the end what quining is about, you've gotten the main point of the dialogue. The dialogue eases you into hard-core quining. Be sure to pause at each example and feel its meaning.

27 **SQ5.** What exactly is the title that, so far as you know, is not connected with any book?

36 ~~43~~ **SQ6.** Why does Achilles say (p.435) that *...it might make more sense if it said 'sentence' instead of 'king'*?

27 **SQ7.** What is a specific example of a tortoise's love song?

45 **SQ8.** What exactly is "Sentence Q"? What's its relation to Sentence P? (don't be fooled that Tortoise gives TWO examples of Sentence Q. Use the one that's related to Sentence P).

27 **SQ9.** Think of a quined sentence of your own. Try one that's self-referential

45 **SQ10.** Is the sentence shouted on the telephone at Achilles true or false?

You might be able to guess from the last question where Hofstadter is going. Gödel was somehow able to import the Liar Paradox into TNT, which was supposed to be impervious to it. You see in this dialogue how quining can lead to that paradox. If quining can be translated into TNT language, then TNT is in deep trouble. We will cover only 12 more pages in this book. In the last three, Hofstadter will show how quining can be arithmetized and imported into TNT. The result is a statement analogous to that shouted at Achilles.