Formal Systems

Nathan Albin

October 26, 2005

Gödel vs. *Principia Mathematica*. In the early 1900s, Alfred North Whitehead and Bertrand Russell published three volumes of a work they called *Principia Mathematica*. The purpose, as Russell said, was "to show that all pure mathematics follows from purely logical premises and uses only concepts definable in logical terms." The idea was to start from some simple agreed-upon truths (or axioms) and derive deeper mathematical truths by using only simple rules of logic for making new true statements out of old ones. The authors succeeded in reproducing many important mathematical truths in this way. There were many who believed that *Principia Mathematica* (or something like it) would be strong enough to produce *every* mathematical truth! In 1931, Kurt Gödel proved that this is impossible. He showed that there is no way to make something like *Principia Mathematica* tell us every true thing about math. In other words, Gödel showed that there are true statements that are beyond the reach of *Principia Mathematica* or any improvement of it. The purpose of the next two lectures is to explore exactly what it was that Gödel proved and what he was talking about.

A toy problem: The MU-puzzle. The **MU**-puzzle comes from the book *Gödel, Escher, Bach: an Eternal Golden Braid* by Douglas R. Hofstadter. The puzzle "pieces" are the letters **M**, **U** and **I** which are placed together into *strings*. Some examples of strings in this puzzle are

UMUUIM
MUUU
IUMU
UUIUU

It is important to realize that the order of the letters is important, so that **UMM** is not the same as **MUM** or **MMU** for example.

In order to solve the puzzle, you try to form the string **MU** starting with the string **MI** and using the following rules for forming new strings from old ones. By using these rules, you will be able to produce more and more strings in addition to your initial string **MI**. Your goal is to show that **MU** is one of them.

> Rule I: If you have a string which ends in **I**, then you can form a new string by adding **U** to the end.

So right away from **MI**, you can make **MIU**. That gives you two strings to work with. **MU** isn't one of them, though. We need more rules.

> Rule II: If you have a string of the form **M**x (where x stands for any fixed string of letters), then you can also use the string **M**xx.

x is NOT one of the puzzle "pieces" and it should never show up in one of your strings. Instead, it is a *variable* which represents any string of the letters **M**, **U** and **I**. While you can apply the rule several times with different xs, x cannot change its meaning during any single application of the rule. Using Rule I and Rule II, you now have a lot of strings to work with (infinitely many in fact).

Exercise: Explain how to use the first two rules to obtain the following strings.

MIUIU
MIIIIIIIIU
MIIUIIU

Exercise: With only these two rules, solving the puzzle is impossible. Can you think of a reason why?

> Rule III: If you have any string which contains **III**, then you can form a new string by replacing the **III** with **U**.

Exercise: Form three new strings from **MIIIIIIIIU** using Rule III.

> Rule IV: If you have a string which contains **UU**, you can form a new string by simply removing the **UU**.

Exercise: Using Rule IV, form three new strings from the string **MUUIUUU**.

Exercises:

1. Decide whether each of the following strings can be formed by starting with **MI** and following the four rules. If so, describe how. If not, explain why.

   (a) **MUM**

   (b) **MUI**

   (c) **IIM**

   (d) **MUIIU**

   (e) **MIUIUIU**

   (f) **MUII**

2. Form groups of 3 or 4. Each person in the group should form a string by starting with **MI** and using about 10 steps. (A step involves one application of one of the four rules.) Now give the string to each of the other group members and see if they can recover your steps. Did everyone make the string using the same steps?

3. Can the **MU** puzzle be solved? If so, show how. If not, explain why. (Don't be worried if you don't get this one right away.)

The MIU-system. Mathematically, the **MU**-puzzle can be seen as a *formal system*, which we could call (as Hofstadter does) the **MIU**-system. A formal system consists of three parts.

(i) Every formal system has a finite set of *symbols*. For the **MIU**-system, this is the set $\{\mathbf{M}, \mathbf{I}, \mathbf{U}\}$.

(ii) Every formal system has a set (not necessarily finite) of *axioms*. Each axiom is a string composed of one or more symbols. For the **MIU**-system, the set of axioms is $\{\mathbf{MI}\}$.

(iii) Every formal system has a finite set of *production rules* which describe how to form new strings from old ones. For the **MIU**-system, the set of production rules is {RULE I, RULE II, RULE III, RULE IV}.

Strings which can be produced from the axioms in a finite number of steps by following the production rules are called *theorems*. The set of axioms is automatically included in the set of theorems. In this language, the **MU**-puzzle can be rephrased as follows.

Is **MU** a theorem in the **MIU**-system?

Decision procedures. Given a formal system like the **MIU**-system, one natural question to ask is whether there exists some procedure which can tell us (in finite time) if a given string x is a theorem or not. Such a procedure is called a *decision procedure*. Imagine a decision procedure as a computer program. We should be able to run this program, enter a string of symbols, and set the computer to work. After a while, the computer should either respond by `YES` if the string is a theorem, or by `NO` if it is not. There is no requirement on how fast the computer gives us an answer, but it must answer eventually.

The LT-system. The **LT**-system is a formal system whose set of symbols is $\{\mathbf{L}, \mathbf{T}, \mathbf{o}\}$ and whose single axiom is $\{\mathbf{oLToo}\}$. (There's just one axiom, like in the **MIU**-system.) The rules are

RULE I: If $x\mathbf{LT}y$ is a theorem, then so is $x\mathbf{LTo}y$.

RULE II: If $x\mathbf{LT}y$ is a theorem, then so is $x\mathbf{oLTo}y$.

Exercise: Describe a decision procedure for the **LT**-system.