

# The essence of Mathematics?

## The MU-puzzle

Idea and some of the text is from "Gödel, Escher, Bach: an Eternal Golden Braid" by Douglas R. Hofstadter.

Here you will be introduced to a *formal system* in the form of a little puzzle. "Can you produce **MU**?" is the puzzle. To begin with, you will be supplied with a *string* (which means a string of characters), and that string will be **MI**.

We call the formal system *the MIU-system*, and it is defined as follows:

### Strings

A string means a string of characters. The strings of the MIU-system consists of the letters M, I and U in any combination and number. Below are some examples of strings:

<b>MU</b>	<b>UIM</b>
<b>MIU</b>	<b>MIUIIU</b>
<b>MUUMUU</b>	<b>IIIIIII</b>

But although all of these are legitimate strings, they are not strings which are "in your possession". In fact, the only string in your possession so far is **MI**. Only by using the rules, about to be introduced, can you enlarge your private collection.

### Rules of inference

These are the rules of producing new strings from old ones.

**Rule 1:** If you possess a string whose last letter is **I**, you can add on a **U** at the end.

Examples:	<b>MII</b>	$\Rightarrow$	<b>MIIU</b>
	<b>UI</b>	$\Rightarrow$	<b>UIU</b>
	<b>MUI</b>	$\Rightarrow$	<b>MUIU</b>
	<b>I</b>	$\Rightarrow$	<b>IU</b>

**Rule 2:** Suppose you have **Mx**. Then you may add **Mxx** to your collection. Here the letter 'x' stands for any string ('x' itself is not a part of the MIU-system).

Examples:	<b>MIU</b>	$\Rightarrow$	<b>MIUIU</b>
	<b>MUM</b>	$\Rightarrow$	<b>MUMUM</b>
	<b>MU</b>	$\Rightarrow$	<b>MUU</b>
	<b>MII</b>	$\Rightarrow$	<b>MIII</b>

**Rule 3:** If **III** occurs in one of the strings in your collection, you may make a new string with **U** in the place of **III**.

Examples: **UMIIIMU**  $\Rightarrow$  **UMUMU**  
**MIII**  $\Rightarrow$  **MIU**  
**MIII**  $\Rightarrow$  **MUI**  
**MIII**  $\Rightarrow$  **MU**

**Rule 4:** If **UU** occurs inside one of your strings, you can drop it.

Examples: **UUU**  $\Rightarrow$  **U**  
**MUUI**  $\Rightarrow$  **MI**  
**MUUUIII**  $\Rightarrow$  **MUIII**  
**UU**  $\Rightarrow$  (empty string)

Don't think you can run any of these rules backwards!

Examples: **MIU**  $\Rightarrow$  **MI** not allowed!  
**MUU**  $\Rightarrow$  **MU** not allowed!  
**MU**  $\Rightarrow$  **MIII** not allowed!  
**II**  $\Rightarrow$  **IUII** not allowed!

### Using the rules

From the start you only possess the string **MI**. Now you may try to include new strings in your collection, using the four rules of inference. To create a new string from one of those you possess is called making a *derivation*.

Example: Here is a derivation of the string **MUIIU** (the numbers refer to the rule used in each step):

**MI**  $\xRightarrow{2}$  **MII**  $\xRightarrow{2}$  **MIII**  $\xRightarrow{1}$  **MIIIIU**  
 $\xRightarrow{3}$  **MUIU**  $\xRightarrow{2}$  **MUIUUIU**  $\xRightarrow{4}$  **MUIIU**

After doing this we have included **MUIIU** in our collection of strings, - and also **MII**, **MIII**, **MIIIIU**, etc.

Exercise: Try to create some new strings with **MI** as a starter, using the four rules of inference. Can you make **MU**?

## Theorems, Axioms, Proofs

You probably have made some attempts to produce **MIU**. In doing so, you have built up your own private collection of strings. Such strings, producible by the rules, are called *theorems* of the formal system. To produce a string is *to prove a theorem*. As a starting point you have got a theorem for free, namely **MI**. Such a "free" theorem is called an *axiom*. So in the MIU-system we have only one axiom:

**Axiom:**                    **MI**

To prove a new theorem you must start with a known theorem and use the rules of inference, creating a proof. The only known theorem from the very start is **MI**. The derivation of **MUIIU** means, that we have proved, that **MUIIU** is a theorem of the MIU-system.

Exercise:        Try to prove the following theorems (let the derivation start with the axiom **MI**):

**MUI, MUII, MIII, MU, MIIII.**

## M-mode

Now after having derived a lot of theorems you maybe have got the feeling, that doing this is very much like acting like a machine. It would certainly be possible - in fact it would be very easy - to program a computer to generate theorem after theorem of the MIU-system. The computer-program could do this in the following methodical way, - an *algorithm* for creating theorems of the MIU-system:

Step 1: Apply every applicable rule to the axiom **MI**. This yields two new theorems:  
**MIU, MII.**

Step 2: Apply every applicable rule to the theorems produces in step 1.

Step 3: Apply every applicable rule to the theorems produces in step 2.

...                    ...

In this way the computer is creating a "tree of theorems" where each branch corresponds to using a rule creating a new theorem from an old one. The tree is shown (upside down) later in the text.

Exercise:        Create the start of the tree of theorems of the MIU-system as far as you have time and energy. Can you create a branch ending with **MIU**?

## I-mode

When you have worked for some time with the MU-puzzle, acting like a machine, deriving a number of theorems, you will probably begin to notice some properties of the theorems you have made; that is where human intelligence enters the picture. For instance, it was probably not obvious to you that all theorems would begin with **M**, until you had tried a few. Then the pattern emerged, and not only could you see the pattern, but you could understand it by looking at the rules, which have the property that they make each new theorem inherit its first letter from an earlier theorem; ultimately, then, all theorems' first letter can be traced back to the first letter of the sole axiom **MI** - and that is a proof that theorems of the MIU-system must all begin with **M**.

What you did here was using your intelligence to jump out of the formal system. It is very important when studying formal systems to distinguish between working *within* the system, behaving like a machine (M-mode), and making statements or observations *about* the system, using your intelligence (I-mode).

One of the main purposes of stepping out of the formal system is to find a "theoremhood-test", that is a characterization of theorems in the system which make it possible to test if a given string is a theorem or not. Now, what about the "tree of theorems" which you started on, being in the M-mode? This method produces every single theorem sooner or later, because the rules are applied in every conceivable order. Couldn't we state the following "theoremhood-test"?

Create the "tree of theorems". Wait until the string in question is produced; when that happens, you know it is a theorem - and if it never happens, you know that it is not a theorem.

The problem with this "test" is, that you do not have any guarantee that you will get your answer in a finite length of time. What about the string **MU**? If you derive **MU** in the tree, then you know it is a theorem. But if you do not derive **MU** today you cannot make any conclusion about the theorem-hood of **MU**, - it could be, that you will derive it tomorrow! So the "tree of theorems" is not a useful test. To be useful the test must be done in a finite amount of time; such a test is called a *decision procedure* for the given formal system.

When you have a decision procedure, then you have a very concrete characterization of the nature of all theorems in the system. In the MIU-system to stay in M-mode deriving theorems is not a decision procedure; you must change to I-mode in the hope of getting a finite-time characterization of the theorems.

In the exercises to come I want to help you to find a decision procedure for the MIU-system.

Exercise: Can you find any rule concerning the numbers of **M**'s, the numbers of **I**'s and/or the numbers of **U**'s in the theorems of the MIU-system.

With basis in this investigation try to formulate a decision procedure for the MIU-system. And try to give arguments for it. If you do not succeed in this, you will get more help on the next page.

Exercise: Put focus on the number of **I**'s in the theorems of the MIU-system. Try to formulate a rule in the form:

"If the string  $x$  is a theorem of the MIU-system, then the number of **I**'s in  $x$  is a number ..."

The ... is called a *necessary condition* for a string to be a theorem. If you have a string for which the number of **I**'s don't ... , then you know, that the string is not a theorem.

Exercise: Is it possible to go the other way? That gives a rule in the form:

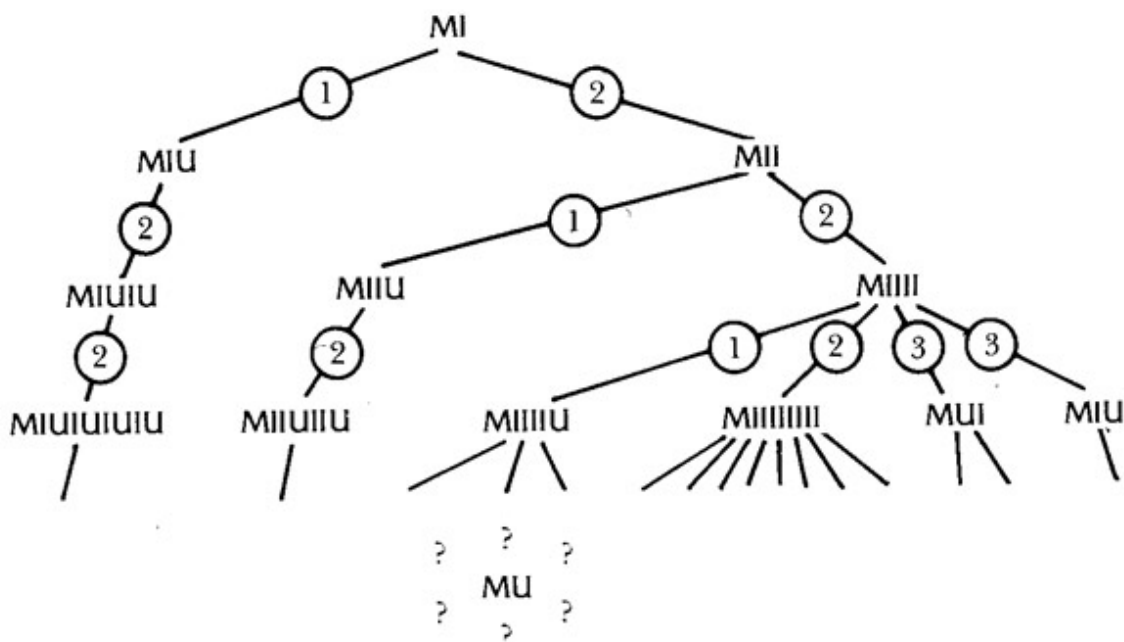
"If in the string  $x$  the number of **I**'s ... , then  $x$  is a theorem of the MIU-system"

Here ... is a *sufficient condition* for a string to be a theorem. This is more difficult. If it is too difficult then get help in the next two exercises.

Exercise: Try to find a method of proving theorems of the form **MII...I**, where the string has no **U**'s and the number of **I**'s is fulfilling ...

Exercise: Try to find a method of proving theorems in general, **M...**, where the string has a number of **I**'s fulfilling ...

Exercise: Now I hope you will be able to formulate the decision procedure of the MIU-system, - and decide if **MU** a theorem or not.



There is something very significant about what has happened here. It shows one difference between people and machines. It is possible to program a computer to do a routine task in such a way that the computer will never notice even the most obvious facts about what it is doing; but it is inherent in human consciousness to notice some facts about the things one is doing. If you punch "1" into an adding machine, and then add 1 to it, and then add 1 again, and again, and again, and continue doing so for hours and hours, the machine will never learn to anticipate you, and do it itself, although any person would pick up the repetitive behavior very quickly. Or, to take a silly example, a car will never pick up the idea, no matter how much or how well it is driven, that it is supposed to avoid other cars and obstacles on the road; and it will never learn even the most frequently traveled routes of its owner.

The difference, then, is that it is possible for a machine to act unobservant; it is impossible for a human to act unobservant. Notice I am not saying that all machines are necessarily incapable of making sophisticated observations; just that some machines are. Nor am I saying that all people are always making sophisticated observations; people, in fact, are often very unobservant. But machines can be made to be totally unobservant; and people cannot.

It is an inherent property of intelligence that it can jump out of the task which it is performing, and survey what it has done; it is always looking for, and often finding, patterns. The jumping out can be very simple, as for example getting bored watching a television program, therefor zapping to another channel; or realizing that you are wasting your time watching television in general, therefor turning it off. The jumping out can be very complex, as for example breaking a pattern of life, getting divorced, quitting your work or just stop short for some moments every day meditating on your existence and the meaning of your life.

How well have computers been taught to jump out of the system? Here is one example which surprised some observers. In a computer chess tournament not long ago in Canada, one program - the weakest of all the competing ones - had the unusual feature of quitting long before the game was over. It was not a very good chess player, but it at least had the redeeming quality of being able to spot a hopeless position, and to resign then and there, instead of waiting for the other program to go through the boring ritual of checkmating. Although it lost every game it played, it did it in style. A lot of local chess experts where impressed. Thus, if you define "the system" as "making moves in a chess game", it is clear that this program had a sophisticated, preprogrammed ability to exit from the system. On the other hand, if you think of "the system" as being "whatever the computer had been programmed to do", then there is no doubt that the computer had no ability whatsoever to exit from that system.

Exercise: Is there really any difference between M-mode and I-mode? Is I-mode not just a very advanced type of M-mode? Do you think it is OK to talk about a computer "jumping out of the system"? Is it a question of how to define "the system"? - Also, concerning we human beings: Are we "jumping out the system" when meditating, quitting work and travelling around the Earth, not following the traditions. Is this also a question of defining "the system"? Do you think we can maintain the distinction between computers (robots) and people in the future?