Nick Peters
COMP 469
02/02/09
Assignment #1

**1. Define and explain the 4 rules of the MIU System.**

- Rule #1 - If any string ends with the character “I,” you can append the character “U” to the end of the string.
- Rule #2 - If any string begins with “M,” you can duplicate the string after “M” and append it to the end.
- Rule #3 - If a string contains "III," you can replace it with the character “U”.
- Rule #4 - If any string contains "UU," you can delete the string "UU."

**2. Why is the MIU System called "typographical"?**

According to Hofstadter the MIU system is typographical because it has the following operations:
- Reading and recognizing any of a finite set of symbols.
- Writing down any symbol belonging to that set.
- Copying any of those symbols from one place to another.
- Erasing any of those symbols.
- Checking to see whether one symbol is the same as another.
- Keeping and using a list of previously generated theorems. (8)

**3. Why is the MIU System called a "deduction" system.**

“A deductive system consists of axioms and rules of inferences that can be used to derive the theorems of the system.” (2). By definition of a deductive system, the MIU system is a deduction system since we use a single axiom and rules of inference to derive theorems that belong to the system.

**4. Does the MIU System use "induction" in any way? Explain.**

Mathematical induction proves that if the first statement, or basis case, is true, then the next statement in the infinite sequence is true (4). The problem here is that we could look at the next statement for an infinite amount of time and never find the string “MU.” Induction would not prove, nor disprove the existence of the string we are looking for.

**5. Explain any realistic "interpretation" of the symbols, or strings of symbols, in the MIU System.**

The MIU system gives us a top-level view of the derivation of theorems. Using a derivation tree we are able to see how the theorems are produced and in what order.

**6. Give an example of an "axiom" in the MIU System.**

An Axiom is a postulate in which the truth is taken for granted (1). In this case, our given axiom is the string "MI." All other theorems are derived from this single axiom using the ruleset given to us.

**7. What does it mean for a string of symbols to be "well formed" in the MIU System?**

Being well formed means the string of symbols are generated by a formal grammar (3). We can look at the MIU system as a grammar and the theorems generated from our axiom as a derivation tree. All strings of symbols, or theorems, that are generated by the grammar, or derivation rules, are all considered to be well formed. The strings of symbols that will never be a production of the derivation tree are considered to be not well formed.

8. **Give examples of strings that are well formed and those that are not well formed.**

Well Formed: MI, MIU, MIUIU
Not Well Formed: MU, MIII, anything that doesn't start with an M

**9. Explain how the MIU System uses "modus ponens".**

Modus ponens takes the form: If P Then Q. P. Therefore Q (5). In this case, P is whether or not a given string meets the requirement of a given rule. In this case the rule takes the form of a regular expression and Q is the modified string. As an example we'll take a look at rule one.

P - The string ends in "I"
Q - Old String + "U"

Using a regular expression we determine if the string ends in I. If it does, then we return a new string with a "U" appended to the end.

**10. Explain how your system implements "conflict resolution".**

There are a few conflicts that the system resolves. These conflicts include: which order to perform the rules, when to stop running the system, and what strings to evaluate.

The first conflict is resolved by coding an arbitrary order in which the system perform the rules. In this case I coded it such that the system performs the rules in order (Rule 1, Rule 2, etc.).

For the next conflict, there are two cases in which we stop running the system. The first case is when we find the string "MU." The second case is when we reach a certain string length. We know that the system will never find the string "MU" so we can safely make the assumption that the program will not terminate as a result of this case. The second case will terminate the derivation of future theorems when the string length gets too large.  For performance purposes, I limited the string length to 20 characters. As a result, the system derives all theorems whose string representation is less than 20 characters.  As an aside, there are some exceptions to this rule.  For example say we run rule 2 on a string that is 19 characters long.  The resulting string would be 37 characters long.  At that point the system would stop deriving theorems from that particular string representation.

The last conflict, determining which strings to evaluate, is resolved in a couple ways.  First, regular expressions are used to determine whether or not a string is applicable to a particular rule.  If it is not, we cannot derive anymore theorems using that particular rule and theorem.  The next way we figure out which strings to evaluate is through the use of a look-up table.  The look-up table is a hash table that uses strings that we have already derived as indexes.  If a given string has already been derived, there is no need to derive more theorems from it, since it has already been done in another branch of the derivation tree.

**11. Does the MIU System use forward or backward chaining?  Explain.**

According to Tanimoto, forward chaining is the strategy in which we start at a given node, and search for our goal node (7).  In the MIU system, we are given the axiom "MI" to start at.  Using our rules, we derive other theorems and search for our goal node.  As a result of the nature of the search, the system uses forward chaining.

**12. The MU Puzzle: Given only the axiom MI, is MU deducible in the MUI System.**

No.

**13. Provide a convincing argument that MU is, or is not, deducible by the MUI System given only the axiom MI to start with.**

To get the string MU we need to first get the string MIII.  Using Rule 3, we can derive MU from MIII.  In order to prove that we cannot derive “MU,” we can prove that we cannot derive the string 'MIII' by looking at the number of Is.  In the beginning we have only one “I” which is not divisible by the number 3.  If we apply rule 2 to “MI,” we will derive the theorem "MII."  This string has two Is which is also not divisible by 3.  If we continue to apply rule 2, the number of Is in a string is equal to 2^a where a is the number of times we apply rule 2.

Say after two applications of rule 2 we apply rule 3.  We will start with the string MIIII and get either MIU or MUI.  Regardless, we still have 1 I.  If we continue to apply rule 2, we will always have a number of Is that is not divisible by 3.  As a result we will never derive “MIII” and thus not derive “MU.”

References:


1. http://en.wikipedia.org/wiki/Axiom
2. http://en.wikipedia.org/wiki/Deductive_system
3. http://en.wikipedia.org/wiki/Formula_(mathematical_logic)
4. http://en.wikipedia.org/wiki/Mathematical_induction
5. http://en.wikipedia.org/wiki/Modus_ponens
6. http://en.wikipedia.org/wiki/MU_puzzle
7. Hunter, Geoffrey. Metalogic: An Introduction to the Metatheory of Standard First-Order Logic, University of California Pres, 1971.
8. Hofstadter, Douglas R. Godel, Escher, Bach: An Eternal Golden Braid. Basic Books, Inc. 1972.