**Logic II: Simple Sentences** June 14, 2012

**CCSS for Grade 6 (page 39)**

*Students understand the use of variables in mathematical expressions. They write expressions and equations that correspond to given situations, evaluate expressions, and use expressions and formulas to solve problems. Students understand that expressions in different forms can be equivalent, and they use the properties of operations to rewrite expressions in equivalent forms. Students know that the solutions of an equation are the values of the variables that make the equation true. Students use properties of operations and the idea of maintaining the equality of both sides of an equation to solve simple one-step equations. Students construct and analyze tables, such as tables of quantities that are in equivalent ratios, and they use equations (such as $3x = y$) to describe relationships between quantities.*

**Equivalent expressions**

Arithmetic expressions denote numbers. Algebraic expressions, on the other hand, don't denote anything but they have the form of arithmetic expressions, and consequently if numerical values are assigned to the variables, then the result denotes a number. Two arithmetic expressions are said to be *equal* if they denote the same number. Two algebraic expressions are said to be *equivalent* if, whenever values are assigned to the variables in the two expressions in the same way, the resulting expressions denote the same number. To emphasize the idea here, we call this *equivalence in meaning*. The subtle importance of this phrase is clarified in the next paragraph.

There is another way to understand equivalence of expressions which is dependent upon rules such as the associative, commutative and distributive laws. Each of these laws gives us a rule for rewriting expressions—for example, using the commutative law, we may rewrite $a + b$ as $b + a$. If one expression can be transformed into another by applying the laws of arithmetic, then they are said to be "symbolically equivalent." Since the laws do not change what an expression denotes (or will denote when values are assigned to its variables), if two expressions are symbolically equivalent, then they are equivalent in meaning. The converse is also true. If two expressions are equivalent in meaning, then they are symbolically equivalent. This is a remarkable fact; it shows that the laws of arithmetic give a complete description of addition and multiplication of the real numbers.

**Equality**

So far, the symbolic formations we have been discussing can only point to numbers, or serve as directions for making numbers from numerical ingredients. We are at the "Me Tarzan" stage of language, since we have not yet introduced symbols that enable us to make complete, intelligent statements, e.g., "I am Tarzan and you are Jane." The equality symbol = takes us into a new realm. If we write the equality symbol = between two arithmetic expressions, we get a mathematical sentence—a complete thought, (though possibly an an incorrect one). For example $1 = 1$, $2 = 1 + 1$, $1 + 1 + 1 = 3$ are sentences. $1 = 2$ is also a sentence, but it is false. (Grammatical but false: "Tarzan is Jane.")

In developing the ideas of logic, we have been careful to distinguish between the informal ideas of constants, variables and functions and the formal ideas of constant symbols, variable

symbols and function symbols. The informal concepts are used in a variety of ways, and the exact meanings are dependent upon context. On the other hand, *constant*, *variable* and *function symbols* have explicitly stated meanings and are used according to explicit rules. The common-sense idea of an equation is also flexible. For example, we often speak of scientific laws as equations. Ohm's Law is the equation

$$I = \frac{V}{R},$$

where $I$ is the electrical current through a load, $V$ is the voltage across the load and $R$ is the resistance of the load. It describes the behavior of direct current circuits, and it a fact of nature. But Ohm's equation is always used in a context where this specific interpretation is understood. When we speak of a *formal equation*, on the other hand, we are referring to an array of symbols consisting of two formal expressions with an equals sign between them. We experienced the kind of thing I am referring to here yesterday, when we were thinking about the equivalent expressions:

$$(m^2 - n^2)^2 + (2mn)^2 \qquad \text{and} \qquad (m^2 + n^2)^2.$$

We introduced these expressions in the context of thinking about Trina's Triangles. But the expressions themselves and their meaning as functions can be separated from this context. It is true that no matter what numbers we substitute for $m$ and $n$, the two expressions above will have the same value, and we can—and did—prove that by using the rules of arithmetic.

A formal equation is true if the expressions on the two sides denote the same number. It is false if the two expressions denote different numbers. If a formal equation contains algebraic expressions then it is neither true nor false, since algebraic expressions don't denote anything. Such an equation can be changed into true or false one by replacing the variable symbols by constant symbols.

Here are some examples:

a) $3 + 5 = 8$. This equation is true.

b) $x + 5 = 8$. This equation is neither true nor false.

c) $a^2 + b^2 = c^2$. Without a knowledge of the meaning of $a$, $b$ and $c$, we cannot say anything at all about this string of symbols. If $a$, $b$ and $c$ are constant symbols, then the equation is either true or false, but we cannot tell which until we know more about the symbols. If $a$, $b$ and $c$ are variable symbols, then this equation is neither true nor false, but if we assign to the variables the lengths of the sides of a right triangle, with $c$ being assigned the length of the longest side, then the resulting assertion is true.

A *law* or an *identitiy* is an equation between algebraic expressions that is true no matter how the variables are interpreted (in some domain). For example, the Commutative Law of Addition is the equation $x + y = y + x$, which we know it true of all real numbers.[1]

[1] We have said that a formal equation with variables is neither true nor false. If this is so, then it would appear that to be consistent we would have to say that identities are neither true nor false when viewed as formal equations. But remember that we are describing the

An algebraic equation that is not a law is neither true nor false. The *solution set* of an equation is the set of all assignments of values to the variables that makes the equation true. In some cases, the solution set is obvious:

- For $x = 6$, the solution set has one element.
- For $(x-1)(x-2)(x-3) = 0$, the solution set is $\{1, 2, 3\}$.

Two equations are said to be *equivalent* if they have the same solution set. The process of "solving" an equation amounts to translating the equation into equivalent equations (or possibly into sentences more complex than equations) until a sentence whose solutions are obvious is obtained.

**Problems**

1. What is a linear equation? What is the solution set of a linear equation in one variable? In two variables? In three variables?
2. An equation is said to be "polynomial" if the expressions in it can be made by applying only the operations of addition and multiplication to number constants and variables. If a polynomial equation involves only one variable, then what generalizations can you make about the solution set? How about two variables?
3. Give an example of an equation in one variable that has the integers as a solution set. Give an example of an equation in one variable that has the positive integers as a solution set.
4. If I add the same expression to both sides of a given equation, can the solution set change? (Assume that the added expression contains no variables that are not already in the given equation.)

**Project**

*Mathematica* treats expressions and equations in a manner that is consistent with how logicians think. If a numerical expression is given as input, *Mathematica* evaluates it:

```
In[1]:= (1 + 2 + 3) 4
Out[1]:= 24
```

If an algebraic expression is given as input, *Mathematica* evaluates numerical subexpressions, combines "like terms," and puts the parts in canonical order. It repeats these operations until they no longer lead to a simpler expression, and returns the result:

```
In[2]:= x + 2 (x + x) + 3 (x + 2(x + x) + y)
Out[2]:= 5 x + 3 (5 x + y)
```

---

structure of formal logic, not giving advice for how to talk to your math class. In formal logic, we solve this problem by including symbols that make the fact that we are asserting something for all substitutions explicit. Formal logic actually contains a special symbol, $\forall$, to mean "for all". The Commutative Law is

$$\forall x, y : x + y = y + x.$$

It does not expand expressions

```
In[3]:= (1 + x) (2 + x)
Out[3]:= (1 + x) (2 + x)
```

unless asked to do so:

```
In[4]:= Expand[(1 + x) (2 + x)]
Out[4]:= 2 + 3 x + x^2
```

In *Mathematica*, the sign `==` is used to form input strings that play the role of equations, in the sense that they may be true or false, or—if they contain variables—may be solved. The following input essentially asks *Mathematica* if two numerical expressions have the same value:

```
In[5]:= 1 + 2 + 4 + 8 == 16 - 1
Out[5]:= True
```

If we present *Mathematica* with an algebraic equation, the program will often throw the question back at us.

```
In[6]:= 2 + 3 x + x^2 == 0
Out[6]:= 2 + 3 x + x^2 == 0
```

We may interpret this to mean that the program cannot determine whether the equation is true or false without more information. This may happen even if the equation is an identity:

```
In[7]:= 2 + 3 x + x^2 == (1 + x) (2 + x)
Out[7]:= 2 + 3 x + x^2 == (1 + x) (2 + x)
```

However, *Mathematica* sometimes counts expressions as "equal":

```
In[8]:= 4 x == 2 (x + x)
Out[8]:= True
```

We can check identities using `Solve`. This function returns a list of conditions on the variable that make the equation true. {{}} indicates that the empty condition {} suffices, so the equation is true for all substitutions.

```
In[9]:= Solve[2 + 3 x + x^2 == (1 + x) (2 + x), x]
Out[9]:= {{}}
```

Compare with the following:

```
In[10]:= Solve[2 + 3 x + x^2 == 0, x]
Out[10]:= {{x → -2}, {x → -1}}
```

**Problem:** What determines if *Mathematica* returns `True` when asked if two expressions are equal?