MODULE *SimpleMath*

This is a TLA+ module that asserts some simple mathematical formulas to be true. Each formula is preceded by the TLA+ keyword ASSUME , which means that it is to be taken as an assumption. *TLC* checks that assumptions are valid, so it can be used to check the truth of formulas.

You can modify this file and run *TLC* to check your own formulas. However, observe the following constraints:

- Use only the built-in TLA operators, which are listed in Tables 1 and 2 of the book. (To use others, you either have to add their definitions or the TLA+ statements that import their definitions from other modules.)
- The only variables you should use are bound variables–for example, the ones introduced by existential quantification.
- In your formulas, you can use natural numbers (0, 1, 2, ... ), strings like "abc", and the values a, $b$, $c$, $d$, $e$, $f$, and $g$, which *TLC* will interpret as arbitrary values that are unequal to each other and to any other value.
- Use only bounded quantifiers; *TLC* cannot handle the unbounded quantifiers. (It also cannot handle the unbounded CHOOSE operator).

This file contains a number of ASSUME statements. They could be replaced by a single ASSUME that assumes the conjunction of all the formulas, but using separate *ASSUMEs* makes it easier to locate an error.

Note: Table 8 tells you how to type all the symbols that do not have obvious *ASCII* equivalents.

CONSTANTS $a, b, c, d, e, f, g$

This statement declares the values a, ... , $g$ so they can be used in formulas.

This example shows how you can check propositional logic tautologies.

ASSUME

$$\forall F, G \in \{\text{TRUE}, \text{FALSE}\} : (F \Rightarrow G) \equiv \neg F \lor G$$

Here is an example showing how you can check that a formula is NOT a tautology of propositional logic.

ASSUME

$$\neg\forall F, G \in \{\text{TRUE}, \text{FALSE}\} : (F \lor G) \Rightarrow (F \land G)$$

The following examples illustrate the operators of set theory.

ASSUME

$$\{1, 2, 2, 3, 3, 3\} = \{3, 1, 1, 2\}$$

ASSUME

$$\{1, 2\} \cup \{2, 3, 4\} = \{5, 4, 3, 2, 1\} \cap \{1, 2, 3, 4\}$$

ASSUME

$$\{1, 3\} \subseteq \{3, 2, 1\}$$

ASSUME
$\{a, b, c\} \setminus \{c\} = \{a, b\}$

ASSUME
$\{a, b\} \in \{\{a, b\}, c, \{d, e\}\}$

The following defines *SomeSets* to be the set of all subsets of the set $\{a, b, c, d, e\}$. The ASSUME that follows shows how you can use this set to have *TLC* check that a property of sets hold for all the sets in *SomeSets*. (This doesn't imply that the property is valid for all sets, but it's likely to discover if the property is not valid.)

$SomeSets \triangleq \text{SUBSET } \{a, b, c, d, e\}$

ASSUME
$\forall S, T \in SomeSets : (S \subseteq T) \equiv S = (S \cap T)$