Section 13.3 Random variables

In 13.3 MODIFY the text (following Syntax 13-1) as follows:

— The solver can randomize singular variables of any integral type.

— Arrays can be declared **rand** or **randc**, in which case all of their member elements are treated as **rand** or **randc**.

— Individual array elements can be constrained, in which case the index expression must be a literal constant may include iterative constraint loop variables, constants, and state variables.

— Dynamic and associative arrays can be declared **rand** or **randc**. All of the elements in the array are randomized, overwriting any previous data.

— The size of a dynamic array declared as **rand** or **randc** can also be constrained. In that case, the array shall be resized according to the size constraint, and then all the array elements shall be randomized. The array size constraint is declared using the size method. For example,

```
rand bit [7:0] len;
rand integer data[];
constraint db { data.size == len; }
```

The variable len is declared to be 8 bits wide. The randomizer computes a random value for the len variable in the 8-bit range of 0 to 255 and then randomizes the first len elements of the data array.

When a dynamic array is resized by randomize, the resized array is initialized (see 5.6.1) with the original array. That is, the resize grows or shrinks the array. This is significant for a dynamic array of class handles. Randomize does not allocate any class objects. Up to the new size, existing class objects are retained and their content randomized. If the new size is greater than the original size, each of the additional elements has a **null** value (requiring no randomization).

In resizing a dynamic array (by randomize or new), the rand_mode of every element is set to active.

If a dynamic array's size is not constrained, then the array shall not be resized and all the array elements shall be randomized. randomize() randomizes all the elements in the array

An object handle can be declared rand, in which case all of that object's variables and constraints are solved concurrently with the variables and constraints of the object that contains the handle.
 Randomization shall not modify the actual object handle and shall treat it as a state variable Objects handles shall eannot be declared randc.