addendum *[posted on 26.02.14]*

1. Print and read can have only an ID as the argument. print(x), read(x2) are valid. But print(5), print("abc"),etc are invlid. Also read(1) or read(m[i,j]) are invalid.
2. Print and read do not operate on the list of identifiers. e.g. print (a,b) is invalid, while print(a); print(b) are valid.
3. Arithmetic expression can have numbers, string and matrix constants. Following are valid expressions written on the right hand side of the assignment statement.

```
s = "abc"+m;
m1 = [1,2;2,3;3,4] + m2;
s = "hello"+"world";
s = c + "hello";
```

4. Size operator cannot have an argument other than the ID. Examples @m, @s etc are valid. Following expression is invalid.

```
@([1,2;3,4;4,5] + m)
```

   The size operator can only be applied to a variable name and not on values of string (i.e. actual string data) or on matrix element values. Only @str or @a etc are valid but @"abcd" and @[1,2,3;4,5,6] etc are not.
5. Assignment statements will also have the expressions including size operators

```
a = @s;
[a,b] = @m;
```

   Note: Type checking is a semantic activity.
6. assignment statements of the following type must be handled properly. It is invalid syntactically.

```
[a,b] = 5;
```

   Modify the grammar to incorporate two different forms of assignment statement; one for an expression on the RHS assigned to the identifier (LHS), and the other for function call returning multiple values as mentioned in the language specifications document.
7. The logical operator not is a unary operator and precedes the boolean expression

```
.not.(a<b)
```

8. Expressions have the maximum precedence of the parenthesis pair.
9. The keyword 'function' must be included in the language.
10. The language does not include typecasting. An expression with one type cannot be converted to another type.
11. At least one statement is a must in the function definition.
12. The open and closed parentheses pair around each boolean expression is required as per the syntax. So

```
(a<b).and.((c<10).or.(b=/=c))
```

    should be correct. Modify the grammar accordingly.
13. Arithmetic operators are left associative.
14. Syntactically, all logical operators (and, or, not) must be applied to the boolean expressions and not on a number or arithmetic expressions.
