

Assignment 3

1. The Let expression in L3 is a special form, because it is computed in a specific way which is encoded by the way we defined L3.
However, we can make every let as “define” and “lambda” special forms . That means, Let is a syntactic abbreviation.
2. Evaluating a let expression equals to applying a closure.
That’s because when we evaluate a lambda expression we create a closure and in class we learned a method in which we transformed a let expression to a equal lambda expression.
3. Here are 4 examples for semantic errors:
 - a. assigning an argument illegally . example: (lambda(x) (+xx) true)
We assigned a Boolean type to x, who must be a number in order for the primitive + operator to be applied successfully.
 - b. dividing by zero. Example : (lambda (x) (1/x) 0)
 - c. The varRef doesn’t exist. Example: (+a 2)
If a wasn’t defined earlier, an error will be returned because a is not a reference to anything.
 - d. an error of var types. Example: (* "a" 2)
because the primitive operator * expects receiving two numbers.
4. The role of valueToLitExp is to solve the typing problem that occurs in the substitution model when executing a function. Before executing the function the arguments are evaluated, But when substituting the arguments, it creates a problem, the body of the function is an expression type, while the values are a value type. Therefore, we convert the values to their literal expression.
5. The valueToLitExp function is used when we are about to substitute variables with values. However, In normal evaluation, we make the substitution before the evaluation of the arguments. Therefore this function is not needed.
6. The difference is, Special allow us to use our own evaluation rules in contrast to primitive operators, which All of them have to follow the same evaluating rules.
7. In the substitution method, each time we apply a procedure, we re-write the body of the code; computing each value of the arguments, renaming all the parameters, returning the values back to expression types and placing the arguments in every VarRef. However, in the environment model we evaluate the body once, and place in it when needed.
As an example we would write a function which being called many times in the algorithm, in the substitution method it will be heavier because of the double-code,

and on the environment method the body of the function will be held only once.

8.

