For my free will project I implemented a basic form of type casting and pointers for the integer type. The type casting was used to change a value from a integer to a integer pointer or back the other way. When creating a pointer, you cannot assign a value to it as that will be seen as an integer. If you want to make a pointer, then you assign it the reference of the thing that it will point to. This can be a variable, function, or procedure. If it is a function or procedure then it has to return an integer.

The syntax is described as such:

^integer == integer pointer type

^ == dereferencing operator

@ == reference operator

Thus a passage of code such as:

var a : integer;

var b : ^integer;

b := @a;

a := ^b;

means that you are setting b to the reference of a and then setting a to be the dereferenced value of b. In other words you are setting a equal to a.

The semantics for the pointers are fully implemented. The code generation is not finished though as there were two parts that are incomplete. There code for calling a function from a function pointer is missing. This was due to the fact that the code generation for the functions was not complete. In order to accomplish this, all that would have to be done is update the “asm\_label.” This is a variable inside all tree nodes that can hold a label for the node. This would be used for conditionals, loops, and functions. When a function pointer is created, the asm\_label is set to the label assigned to the original function. Then at compile time, when you use the dereferenced function pointer it just sets the asm\_label as where it jumps to. This can all be determined at compile time when generating the code.

The other part that is not working is a slight bug in the normal pointers. It can be seen below as the value of the address that I store in the register is actually cut off and not the full value (only the bottom 32 bits are stored). The 0x7fff should be appended to the front of it before gathering the data. This ruined all attempts of attaining valid values (most of the time a segmentation fault occurred).

