A logic diagram consists of the following symbols (AND, OR, NOT, InputPort, OutputPort) called gates:

Each gate has 0..1 outputs and * (or 0..2) inputs. Ports are used to indicate inputs and outputs of logic diagrams (a.k.a. logic circuits). A simple logic diagram is shown below:

This circuit has:

- 4 input ports (a,b,c,d). Each input port, as shown above, has precisely 1 output.
- 2 output ports (r,t). Each output port, as shown above, has precisely 1 input.
- Every wire connects a gate output to a gate input. So the output of a is connected by a wire to the first input of or1; a wire connects the output of or3 to the input of t.
- 10 gates are shown (3 or gates, 1 and gate, 4 inputPorts, 2 outputPorts)
- 10 wires are shown
Your assignment is to:

- Use an UML tool to create a class diagram of logic diagrams.
- Carefully define any constraints in **english** that you need in.
- Derive a schema from your class diagram.
- Populate your tables with tuples that define the above logic diagram.

Submit your answers in a PDF file.

**Hint:** this is a typical nasty design problem because there are many different types of objects that are "slightly" different. In such cases, sometimes it is better to generalize the concept of "object", reduce the number of classes in your diagram, and add more constraints.
Here was my first attempt at creating a “solution” to this problem:

And afterthought, I thought: this is awful. Too many “slightly” different concepts or too many variations of the same concept pollutes the simplicity of what is going on.

So here is my attempt to simplify: There are different kinds of gates. Even in my above design I allowed room for this. But why can’t logic diagram be another kind of gate – a gate that has any number of inputs and any number of outputs? And why can’t a port be another kind of gate – one that has either an input or an output. At which point everything simplifies:
This is progress. Of course, there has to be some constraints, so here goes:

- an OR gate has precisely 2 inputs and 1 output.
- an AND gate has precisely 2 inputs and 1 output.
- a NOT gate has precisely 1 input and 1 output.
- an input port has precisely one output, and 0 inputs.
- an output port has precisely one input, and 0 outputs.

Of course, there the obvious “simple” constraints too:

- all gates have unique names

So I have 4 simple tables. In the following I don’t define a gate for a logic diagram – I could have done so, but I’d need a name for it. Encoding a nameless logic diagram as a gate is optional.