I want to generalize the functions I made to work for more operators, and I want to be able to assign a penalty for each operator. First I will have to decide on which operators to include and how I will represent each one. Ideally I will make something that can be easily adjusted if I want to include more operators later. I also want it to work for multiple gate types.

I will start by only programming in some of these, but I still want to make it possible to add others later on. The focus will be on two input gates.

Possible Operators (see 7.03.15\_Supplementary\_Notes):

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Representation** |
| NOT | True when the input is False | (~a) |
|  |  |  |
| AND | True when both a and b are true | (a&b) |
| NAND | Opposite of AND; True when at least one is False | (a@b) |
|  |  |  |
| OR | True when at least one is true | (a+b) |
| XOR | True when only one is True not both | (a^b) |
| NOR | Opposite of OR; True when neither are True | (a.b) |
| XNOR | True if both are true or both are False | (a=b) |
|  |  |  |
| IMPLIES | True if a is False or both a and b are True | (a>b) |
| NIMPLIES | Opposite of IMPLIES | (a$b) |

Highlighted ones are the ones that I am unsure of.

I’m not sure how I will make the circuit finder remove things that give the same truth as things with only one gate. Also, I must think about how I will approach things if the cost for an operator is 0.

I completed the logic operator functions and the circuit cost evaluator functions. Along with the function to write circuits to a file and retrieve them from a file. These all seem to be functioning properly from the small tests I performed.

I got the getTruthValue and containsEqualSubCircuit to work for 3 inputs. I need to think of a way to generalize this to varying inputs.

Went into detail about how RNA-seq data is gathered with Jing. She showed me where I can get the raw data. I will practice going from that raw data to the numbers presented in the papers.

Continued working on the minCircuitFinders. I plan to make two. One will be a general minCircuitFinder that will try and find all truth values for the number of inputs. The other will just try to find the min circuits for a specific truth value. I will get as much of the first one done today as possible before I have to leave.

I have to be careful with nonsymmetrical operators a>b is different from b>a

Consider allowing operations with 1,0, and self for certain operations.

Considered:

(a.a) 🡪not(a) \*ignored for (a.0)

(a.0) 🡪 not(a)

(a^1) 🡪not(a)

(a=0) 🡪 not(a)

(a>0) 🡪 not(a)

(1$a) 🡪 not(a)

(a@a) 🡪not(a)

Ignored:

(~0) ==1

(~1)==0

(a&0) == 0

(a&1)==a

(a&a) == a

(a@0) == 0

(a@1) == a

(a+0) == a

(a+1) == 1

(a+a) == a

(a^0) == a

(a^a) == 0

(a.1) == 0

(a=1) == a

(a=a) == 1

(a>1) == 1

(a>a) == 1

(a$0) == a

(a$1) == 0

(a$a) == 0

(0>a) == 1

(1>a) == a

(0$a) == 1

And any operation involving only 1 and 0.

Confirm with Bryan that the notAllowed list still applies. Figure out how to make it dependent on numInputs. Also must be careful of when cost is 0. Don’t want to be stuck infinitely on the same level because nor costs 0. Find out what is allowed to be 0.

There seems to be a problem with the function. It is eliminating too many circuits. Nothing with more than 3 gates is made. I need to check the filtering mechanisms.

I fixed the problem and it seems to be working, except it is slower than the original version that only worked for nor gates. See 7.03.15\_Supplementary\_Notes2.txt. This would most likely be greatly sped up by creating a class item for circuits. This way we don’t need to find the truth value from scratch each time. Instead we just use the truth value of the two components. It would probably be a similar story for the circuitCost function, but this probably still wouldn’t speed things up enough for the function to reach 7 or 8 gates if we wanted it to go that far. I’ll try to look into doing things with classes tomorrow.