Boolean bases

The existence of the disjunctive normal form shows that the operations of AND, OR, NOT suffice to express any Boolean function whatsoever. Can we do without one of these operations?

Yes, in fact, just AND and NOT suffice, because

$$(x + y) = (x' \cdot y')'.$$

Using this, we could replace every OR gate

```
x -------|
         |OR>----- z
y -------|
```

in a circuit by the circuit

```
x ---|NOT>----|
              |AND>-----|NOT>------ z
y ---|NOT>----|
```

Similarly, using the fact that

$$(x \cdot y) = (x' + y')',$$

we could replace every AND gate in a circuit by a circuit involving only OR and NOT. Thus, the functions NOT and AND are a complete Boolean basis, as are the functions NOT and OR. Can we do without NOT? No, because the Boolean functions definable using just AND, OR, and constants have a monotone property; in particular, changing any input 0 to a 1 cannot change the output from 1 to 0. since NOT does not have this monotone property, it is not definable using just AND, OR and constants.

NAND and NOR

The two input Boolean function NAND is a combination of NOT and AND. Its truth table is

x	y	(x NAND y)
0	0	1
0	1	1
1	0	1
1	1	0

You can check that a Boolean expression for this function is $(x \cdot y)'$, and a circuit implementing it is the following.

```
x ----|
      |AND>---|NOT>--- z
y ----|
```

Various transistor technologies make it very convenient to implement NAND as a primitive operation. Its gate symbol is similar to the AND gate, with a little circle (signifying negation) at the start of the output wire. Thus implementing other Boolean functions purely out of NAND gates became a design goal. This is possible because, all by itself, the NAND gate is a complete Boolean basis. In particular, `(x NAND x)` is the NOT function, so `((x NAND y) NAND (x NAND y))` is the AND function, and we have previously seen that NOT and AND are a complete Boolean basis.

We should suspect a dual result for OR. The corresponding function is NOR, for NOT-OR, which can be represented by the Boolean expression $(x + y)'$. It also is a complete Boolean basis, and has a gate symbol similar to that for OR, with a little circle on the output wire. NOR is also called Sheffer's stroke.

We've show the following sets to be be complete bases for the Boolean functions:

```
{AND, OR, NOT}
{AND, NOT}
{OR, NOT}
{NAND}
{NOR}
```

Food for thought: are XOR and NOT a complete Boolean basis?