Logically Complete

A *set of logic gates* (or logic operators), L, is **logically complete** if it can be used to build a logical circuit (or expression) that is **logicallly equivalent** to *any* arbitrary logical circuit (or expression).

For any arbitrary *logical circuit*, C, there is a logical circuit, C', such that $C \equiv C'$ and C' is constructed using **only** gates from L.

Proving a Set of Gates is Logically Complete

- Approach (Reduction)
  Give a *canonical* set of logically complete gates, say LCG, a *different* set of gates, K, could be proven to be logically complete by showing that $\forall g \in LCG \ \ \exists g' \ni g \equiv g' \ \wedge \ g'$ only uses gates in K. This reduces the set K to a solved problem.
  - Proving LCG is Logically Complete This is the hard part. Picking the first set to prove logically complete and then proving it.
- $\{\neg, \wedge, \vee\}$ Is Logically Complete
  TBP: $\forall K \in \{\text{LogicalExpression}\}\ \exists K' \ni K \equiv K' \ \wedge \ K'$ uses only operators from the set $\{\neg, \wedge, \vee\}$.
  Given logical expression K: K has some number, n, logical variables and a *truth table* with 2^n rows. Logic variables will be named x_i starting from $0 \leq i < n$.
  Consider some row where K is 1.
  Build a *conjunction* of all of the variables or their negation. For each column, i, if x_i is 1, include x_i in the conjunction; if x_i is 0, include its negation:
  $\bigwedge_{i=0}^{n-1} (x_i == 1) ? x_i : \overline{x_i}$
  With three logical variables and a K with four rows that are 1, four conjunctions are created is in this next table.

				A	B	C	D	K'
x_0	x_1	x_2	K	$\overline{x_0}\overline{x_1}x_2$	$\overline{x_0}x_1x_2$	$x_0\overline{x_1}\overline{x_2}$	$x_0x_1x_2$	$A \vee B \vee C \vee D$
0	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	1
0	1	0	0	0	0	0	0	0
0	1	1	1	0	1	0	0	1
1	0	0	0	0	0	0	0	0
1	0	1	1	0	0	1	0	1
1	1	0	0	0	0	0	0	0
1	1	1	1	0	0	0	1	1

  Note that in the table $K \equiv K'$ because the values are the same for every combination if input variables.
  The K' column is generated by making a *disjunct* of all of the *conjuncts* built for the rows where K is 1.
  Because each conjunct is 1 in only the row that was used to construct it, there is one conjunct with a 1 in each row where K is 1 and none with 1 in any row where K is 0. Joining them with a logical or creates a column with exactly as many 1 rows as there are conjuncts (none overlap) and in *exactly the rows where K is 1.* Thus the disjunct of the constructed conjuncts is the sought K'.
  $\therefore$ We can build a conjunct for any row with a 1 in K using just $\wedge$ and $\neg$ operators. These conjuncts are combined in a disjunct with only the $\vee$ operator. The resulting K' is logically equivalent to K by construction and uses only operators in the set $\{\neg, \wedge, \vee\}$.
  $\therefore \forall K \in \{\text{LogicalExpression}\}\ \exists K' \ni K \equiv K' \ \wedge \ K'$ uses only operators from the set $\{\neg, \wedge, \vee\}$.
- Anything Else
  - Make an And, an Or, and a Not