- classical vs. quantum computing (lect 16)

what is the difference between classical and quantum computing ?

- can a quantum computer do something that a classical computer cannot ?

  ⇝ NO ! foundation of q-comp. is still "usual" math. A classical computer can simulate what a quantum computer would do

- but to simulate an n-qubit system $(\mathbb{C}^2)^{\otimes n}$ we would need $\sim 2^n = \dim (\mathbb{C}^2)^{\otimes n}$ variables

this would violate the complexity-theoretic Church-Turing thesis :

any algorithmic process can be simulated in polynomial time / resource by a Turing machine (classical computer)

- classical universality (§16.2.1)

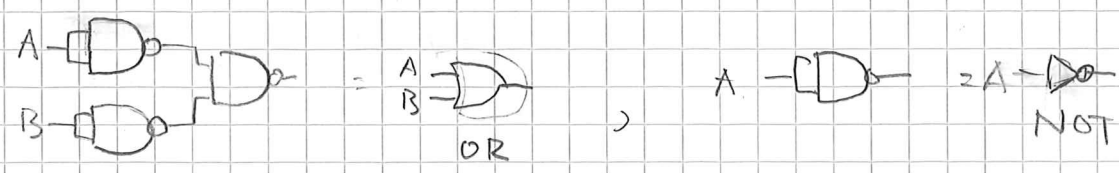
Def. a collection of (classical) gates $f_0, \dots, f_{n-1}$ called <u>universal</u> if any Boolean function $f : \{0,1\}^k \to \{0,1\}$ can be written as

$$f_{i_1}\left(f_{j_1}(\cdots), \dots, f_{j_m}(\cdots f_{k_1}(x_{p_1}, \dots, x_{p_q}))\cdots\right)$$

same x_i can appear many times

Ex. $f_{NAND}(x,y) = \begin{cases} 0 & x=1 \text{ and } y=1 \\ 1 & \text{otherwise} \end{cases}$ 

is universal



Ex. f_{AND}, f_{OR} not universal.

$f(*, 0, *') = 1 \Rightarrow f(*, 1, *') = 1$

(monotone functions are stable under composition.)

Variation for reversible gates:

reversible gate $F(x_0, \cdots, x_{k-1}) = (f_i(x_0, \cdots, x_{k-1}))_{0 \le i < k}$

corresponding to invertible map $\{0,1\}^k \to \{0,1\}^k$

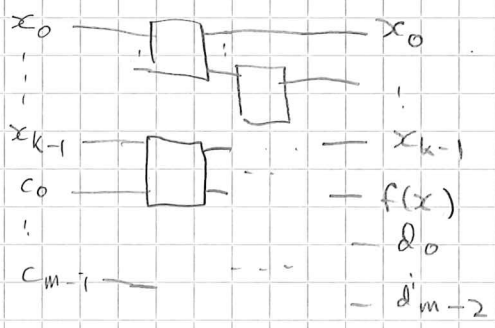
Def. a collection of rev. gates $F_0, \cdots, F_{n-1}$ is universal (in a strong sense) if

$\forall f : \{0,1\}^k \to \{0,1\}$ $\exists c_0, \cdots, c_{m-1} \in \{0,1\}$

$\exists G$ composition of f_i's s.t.

$G(x_0, \cdots, x_{k-1}, c_0, \cdots, c_{m-1}) = (x_0, \cdots, x_{k-1}, f(x_0, \cdots, x_{k-1}), *)$

in diagram:



□ : f_i or ✱

but ⊣ not allowed

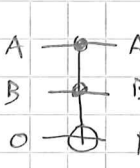
Ex. the Toffoli gate

controlled-controlled-NOT

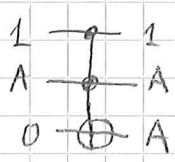
$|110\rangle \mapsto |111\rangle \mapsto |110\rangle$

otherwise preserve input

implementing NAND:



input splitting



- complexity (§ 10.1)

Thm (Shannon) $f_0, \dots, f_{k-1}$ any univ. collection of gates; for $n \gg 1$ almost every function $f: \{0,1\}^n \to \{0,1\}$ needs at least $O(\frac{2^n}{n})$ many f_i's to represent.

Sketch different choice of $(f_i)_{0 \le i \le k}$ would only give polynom. difference.

$\rightsquigarrow$ we can look at the case f_{NAND}.

Step 1. $\log_2(\#\{f: \{0,1\}^n \to \{0,1\}\}) = 2^n$

Step 2 with T copies of f_{NAND}, the possible config. of compositions $f_{NAND}(\dots f_{NAND}(x_i, x_j), \dots)$ is $\le (n+T)^{2T}$.

$\therefore$ each copy of f_{NAND} has $\binom{n+T-1}{2}$ possible comb. of inputs (at most)

$$\binom{n+T-1}{2}^T \le (n+T)^{2T}.$$

Step 3 to have $2^n \sim 2T \log_2(n+T)$

T needs to be $O(\frac{2^n}{n})$ ∎

- Universality for quantum gates

We want to consider an analogue of universality for reversi

but: $\{U: \mathbb{C}^N \to \mathbb{C}^N \text{ unitary}\}$ is an uncountable (continuum) infinite set, while (for fixed $U_0, \dots U_{k-1}$) compositions of $U_0, \dots, U_{k-1}$ up to $U_i \otimes I_N$, etc. is only countable infinite

$\rightsquigarrow$ should consider approximation

Def. a collection of unitary gates $U_0, \dots, U_{k-1}$

$(U_i : (\mathbb{C}^2)^{\otimes k_i} \to (\mathbb{C}^2)^{\otimes k_i})$

is called <u>universal</u> if any unitary gate

$U : (\mathbb{C}^2)^{\otimes n} \to (\mathbb{C}^2)^{\otimes n}$

is approximated by compositions of $U_i \otimes I_{2^m}$

and permutation of factors $(\mathbb{C}^2)^{\otimes n} \to (\mathbb{C}^2)^{\otimes n}$

$v_0 \otimes \cdots \otimes v_{n-1} \mapsto v_{\sigma(0)} \otimes \cdots \otimes v_{\sigma(n-1)}$

$\sigma \in S_n$