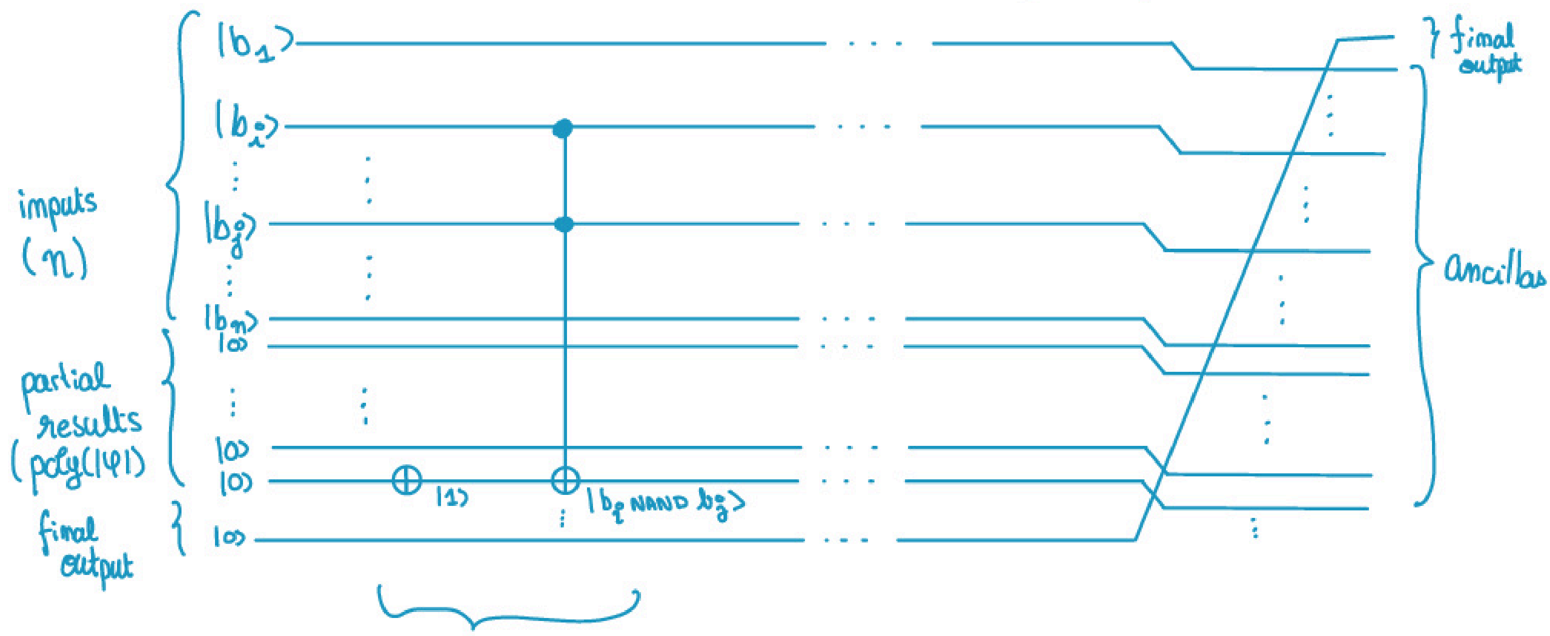
Assignment # 6

Hugo SALOU

Consider a formula φ with n variables. We can construct an oracle for φ by using swaps and NANDs (as any propositional formula can be transformed into an equivalent formula using with only variables and NANDs in poly-time). Using ancillas, we can evaluate φ on some valuation $b_1, \dots, b_n$ with

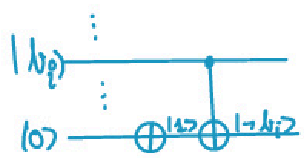


Evaluate a subformula b_i NAND b_j

In the above circuit, we implicitely assumed that $i < j$. This can be done as the NAND operation is symmetric (same thing if ϕ's partial result is "higher" than ψ's).

To evaluate a subformula of the form ϕ NAND ψ, we use a very similar tactic but we place the Toffoli's controls on the two qubits responsible for ϕ's and ψ's partial result.

The only case that is somewhat ambiguous is $i = j$, but we can simply use a CNOT gate whose control is on $b_i = b_j$:

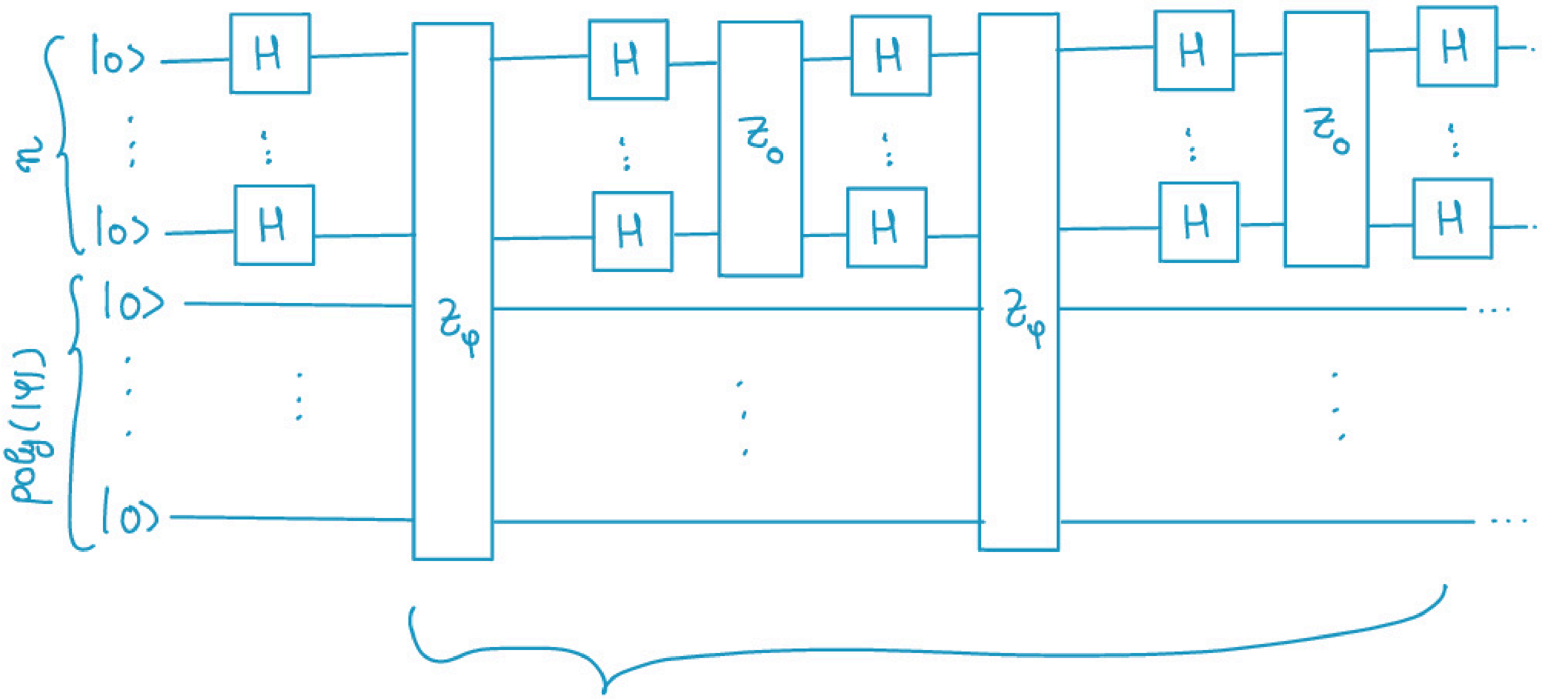


Using tutorial #5's results, we can obtain a query oracle for evaluating φ:

$$Z_\varphi |b_1 \cdots b_n\rangle |0^{poly(|\varphi|)}\rangle = (-1)^{\varphi} |b_1 \cdots b_n\rangle |0^{poly(|\varphi|)}\rangle.$$

(Q 1, part III of Tutorial #5).

We can thus apply Grover's algorithm to the n first inputs of Z_φ:



Repeat k times

As multiple valuations $(b_1^l, \ldots, b_n^l)_{l \in [\![1,m]\!]}$ can satisfy φ, we can run Grover's algorithm for decreasing values of k: $k = 2^n / 2^{iter} = 2^{n-iter}$ where iter is the number of iterations of (not in) Grover's algorithm.

With high probability, we have $k = \Theta(\sqrt{2^n/l})$, thus we have

a circuit of size

$$\mathcal{O}\big(k \cdot (\mathrm{size}(U_\varphi) + \underbrace{\mathrm{size}(Z_0)}_{\substack{\leq \mathrm{poly}(n) \\ \leq \mathrm{poly}(|\varphi|)}}\,)\big) = \mathcal{O}\big(2^{n/2}\,\mathrm{poly}\,(|\varphi|)\big)$$

END of Assignment #6