Lecture 5

10 FEB 2014

Quickly define BPP as the class of all promise problems that can be decided efficiently by a probabilistic Turing machine (i.e., one that can use coin tosses / random bits to decide the next action)

promise problem is $L = A_{yes} \cup A_{no} \subseteq \{0,1\}^*$ such that $A_{yes} \cap A_{no} = \emptyset$

BPP consists of all promise problems L for which there exists (bounded error probabilistic polynomial time)

a prob. TM M such that

1) if $x \in A_{yes}$, then M accepts w/ prob. $2/3$

2) if $x \in A_{no}$, then M rejects w/ prob. $2/3$

Chernoff bound shows that constant is arbitrary. can even be very close to 1/2 as long as only longer than an inverse polyno

Why? Consider that $X_1, \ldots, X_n$ are independent Bernoulli RVs, each w/ $p > 1/2$. Then the probability of $n/2$ or more of events has an exact value

MAJORITY VOTE

$$P_{succ} = \sum_{i=\lfloor n/2 \rfloor + 1}^{n} \binom{n}{i} p^i (1-p)^{n-i}$$

From Chernoff bound, we get

$$P_{succ} \geq 1 - \exp\left\{-\frac{n}{2p}\left(p - 1/2\right)^2\right\}$$

So in order to meet any desired error ε, we have

$$\varepsilon \leq \exp\left\{-\frac{n}{2p}\left(p-1/2\right)^2\right\}$$

$\Rightarrow$ can take n to be no longer than

$$\frac{1}{(p-1/2)^2} \log\left(\frac{1}{\varepsilon}\right)$$

In fact, from this we can see that even if $p = 1/2 + \frac{1}{\text{poly}(n)}$, we only require a polynomial # of repetitions to get the error to be $\exp\{-\text{poly}(n)\}$

Another resource that is important in computation is energy in addition to time & space. The most famous result in this area is Landauer's erasure principle.

For a reversible computation, in principle no energy is required to conduct the computation. However, irreversible computations erase information & there is a fundamental cost associated to doing so. This is what ~~quantum~~ Landauer's erasure principle quantifies.

We will prove this using techniques from quantum information theory...

approach outlined in 1306.4352 (QIP 2014)

Minimal Assumptions:

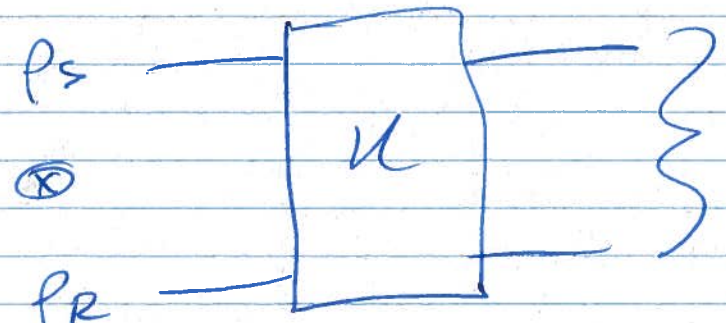
1) process involves a system S & a reservoir R

2) reservoir R is initially in a thermal state $\rho_R = \frac{e^{-\beta H}}{Tr\{e^{-\beta H}\}}$ for Hamiltonian H & inverse temperature β

3) S & R are initially uncorrelated $\rho_{SR} = \rho_S \otimes \rho_R$

4) process proceeds by unitary evolution.

Picture



β = initial inverse temp.

$\Delta S = S(\rho_S) - S(\rho_S^*)$

entropy decrease of system

$$\Delta Q = \mathrm{Tr}\{H\rho_R'\} - \mathrm{Tr}\{H\rho_R\}$$

heat dissipated to reservoir

Landauer's bound:

$$\beta \Delta Q \geq \Delta S$$

if information is erased (i.e., $\Delta S \geq 0$)
i.e., entropy decrease is positive

then heat is dissipated to the environment.
(there is an energy cost to this.)

other point = no heat dissipation needed in principle for reversible computation

simpler statement: erasure of a bit requires $kT \log 2$ joules of work

Definitions:

von Neumann entropy $S(\rho) = -\mathrm{Tr}\{\rho \log \rho\}$

rel. entropy $D(\rho|\sigma) = \mathrm{Tr}\{\rho \log \rho\} - \mathrm{Tr}\{\rho \log \sigma\} \geq 0$

mutual info. $I(A;B)_\rho = S(\rho_A) + S(\rho_B) - S(\rho_{AB}) \geq 0$

We prove that

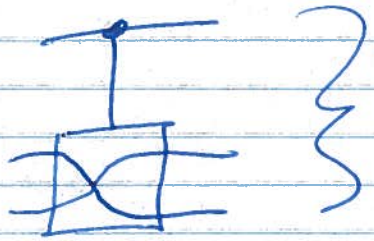
$$\beta \Delta Q = \Delta S + I(S';R') + D(\rho_R' \| \rho_R)$$
$$\geq \Delta S$$

$$\Delta S + I(S';R') = S(\rho_S) - S(\rho_S') + S(\rho_S') + S(\rho_R') - S(\rho_{SR}')$$
$$= S(\rho_S) - S(\rho_S') + S(\rho_S') + S(\rho_R') - S(\rho_{SR}) \quad \left\| \{S(\rho_S) + S(\rho_R)\} \right.$$
$$= S(\rho_R') - S(\rho_R)$$
$$= -\mathrm{Tr}\{\rho_R' \log \rho_R'\} + \mathrm{Tr}\{\rho_R \log \frac{e^{-\beta H}}{\mathrm{Tr}\{e^{-\beta H}\}}\}$$
$$= " \quad + \mathrm{Tr}\{\rho_R(-\beta H - \log \mathrm{Tr}\{e^{-\beta H}\})\}$$
$$= " \quad - \beta \mathrm{Tr}\{H\rho_R\} - \log \mathrm{Tr}\{e^{-\beta H}\}$$
$$= " \quad " \quad " \quad + \beta \mathrm{Tr}\{H\rho_R'\} - \beta \mathrm{Tr}\{H\rho_R'\}$$

$$= \beta \mathrm{Tr}\{H(\rho_R' - \rho_R)\} - \mathrm{Tr}\{\rho_R' \log \rho_R'\}$$
$$+ \mathrm{Tr}\left\{\rho_R' \log \frac{e^{-\beta H}}{\mathrm{tr}\{e^{-\beta H}\}}\right\}$$
$$= \beta \Delta Q - D(\rho_R' \| \rho_R)$$

□

We will show that reversible computation is possible by using just a single reversible logic gate: the Fredkin gate

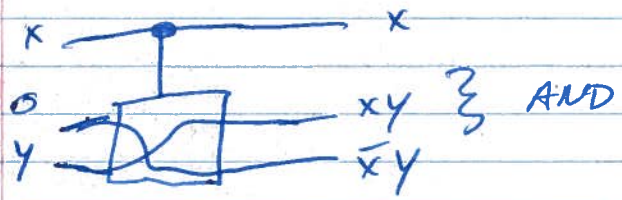


"controlled-SWAP"

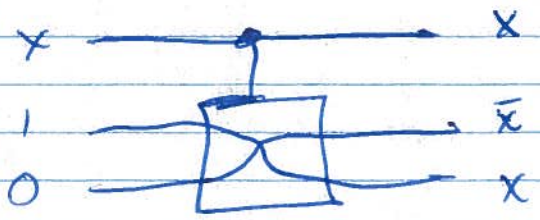
if 1st bit is zero, do nothing
otherwise swap the last two

clearly reversible (in fact "self-inverse")

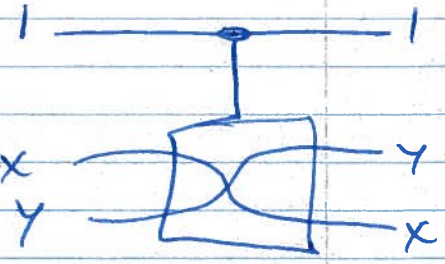
can simulate AND



can simulate NOT



can simulate CROSSOVER / SWAP



FANOUT also

only needs extra ancilla bits

since we can compute any function $f(x)$

$f: \{0,1\}^n \to \{0,1\}^m$ w/ a Boolean circuit, we can do so w/ reasonable overhead w/ Fredkin gates. Can represent action of computation as

$$(x, a) \to (f(x), g(x))$$

a: ancilla; $g(x)$: garbage

computation produces unwanted garbage bits. It is desirable to get rid of these, especially when we move on to quantum computation.

add the NOT gate + CNOT gate

NOT: $0 \to 1$, $1 \to 0$

CNOT: $00 \to 00$, $01 \to 01$, $10 \to 11$, $11 \to 10$

to our gate set for convenience,

w/ NOT gates, we can write action of computation as

$$(x, 0) \to (f(x), g(x))$$

can also add CNOTs at the beginning of the circuit to copy the input as

$(x, 0, 0) \rightarrow (x, x, 0) \rightarrow$ run circuit on last two registers

$(x, f(x), g(x))$

can now discuss an important idea called uncomputation

initialize four-register computer to

$(x, 0, 0, y)$

1) run above circuit on 1st 3 registers

$\rightarrow (x, f(x), g(x), y)$

2) copy 2nd register to fourth w/ CNOTs

$\rightarrow (x, f(x), g(x), y \oplus f(x))$

Since all steps to compute $f(x)$ are reversible, we can run reverse of circuit in step 1 to get

$\rightarrow (x, 0, 0, y \oplus f(x))$

can then omit the final zeros to say that the action of the circuit is

$(x, y) \rightarrow (x, y \oplus f(x))$

resource overhead involved is ~~none~~ constant for replacing ANDs & NOTs & others w/ Fredkin.

since nothing is erased, in principle, this computation can be done w/ zero work cost.

linear overhead for CNOTs & NOTs another run of circuit for uncomputation.

So any irreversible circuit has a reversible simulation. To have a fully reversible implementation, we need reversible Turing machines, but we won't go there...

classes like P, BPP, & BQP do not change ...than using a reversible