Chapter 2

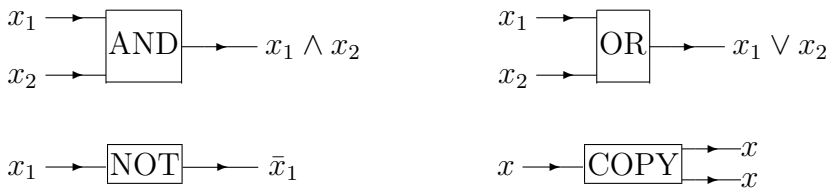
Circuit models of computation

2.1 Introduction

A computation is ultimately performed by a physical device. Thus it is a natural question to ask what are the fundamental limitations that the laws of physics impose on a computation. An early work on such issues was that of Landauer who argued that a "bit erasure" – a logically irreversible process – is always accompanied by heat dissipation and is thus a thermodynamically irreversible process. Consequently any computation using irreversible gates (AND, OR, etc.) will dissipate heat. But is there a fundamental principle that requires a minimum amount of heat dissipation? A negative answer to this question was put forward by Bennett, Benioff and others. More precisely (as we will see later) any logically irreversible computation can be made logically reversible, with appropriate elementary gates, provided that we are willing to increase the work space.

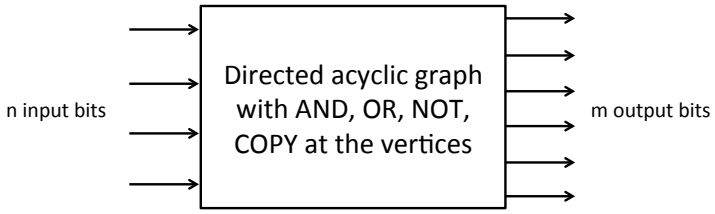
2.2 Classical circuit model of computation

We begin with classical computations done with classical circuits. Consider the basic net of logical gates acting on bits $x_i \in \mathbb{F}_2 = \{0, 1\}$.

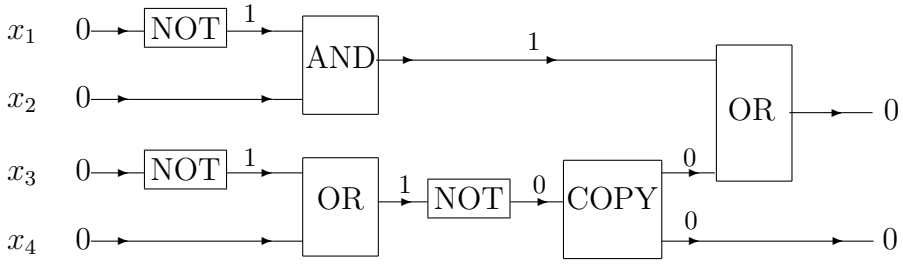


(COPY is also called FANOUT sometimes.)

Definition 1 *A boolean circuit is a directed acyclic graph with n input bits and m output bits. The input can always be initialized to* $(0 \dots 0)$ *because any* $(x_1 \dots x_m)$ *is obtained by a series of appropriate NOT gates.*



For example, the following figures illustrates the circuit of $f(x_1, x_2, x_3, x_4) = ((\bar{x}_1 \wedge x_2) \vee (\overline{\bar{x}_3 \wedge \bar{x}_4}), \overline{\bar{x}_3 \wedge x_4})$:



A celebrated theorem of Emil Post (circa 1950) says that for any function $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$ one can construct a Boolean circuit that computes it.

Theorem 2 *For any function* $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$ *there exists a Boolean circuit that maps inputs* $(x_1 \dots x_n)$ *to outputs* $(y_1 \dots y_m) = f(x_1 \dots x_m)$*. The Boolean circuit is constructed out of* NOT, AND, OR, COPY *and is a directed acyclic graph.*

One says that the set of gates {NOT, AND, OR, COPY} is universal. Note that AND, OR are not reversible. We come back later to the issue of reversibility.

Proof. A function $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$ can be represented by component functions $f_i : \mathbb{F}_2^n \to \mathbb{F}_2$,$i = 1 \dots m$. So if we can COPY the input m times, we just need to show that there exists a Boolean circuit for each $f_i : \mathbb{F}_2^n \to \mathbb{F}_2$. The problem is then reduced to finding Boolean circuits for functions $f : \mathbb{F}_2^n \to \mathbb{F}_2$. For each $\vec{a} = (a_1 \dots a_m) \in F$ we define

$$C_{\vec{a}}(x_1 \dots x_n) = \phi_1(x_1) \wedge \phi_2(x_2) \wedge \dots \phi_n(x_n)$$

where

$$\begin{cases} \phi_i(x_i) = x_i & \text{if } a_i = 0, \\ \phi_i(x_i) = \bar{x}_i & \text{if } a_i = 1. \end{cases}$$

This is built out of AND, NOT gates only, and since $\wedge$ is associative it can be done recursively (directed acyclic graph). We note that $C_{\vec{a}}(x_1 \dots x_n) = 1$ if $(x_1 \dots x_n) = (a_1 \dots a_n)$. Now given a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ let $\{\vec{a}^{(1)}, \dots, \vec{a}^{(k)}\}$ be the set of inputs in $\mathbb{F}_2^n$ for which f takes values 1. For all other input f takes value 0. A little thought shows that

$$f(x_1 \dots x_m) = C_{\vec{a}^{(1)}}(x_1 \dots x_n) \vee C_{\vec{a}^{(2)}}(x_1 \dots x_n) \vee C_{\vec{a}^{(k)}}(x_1 \dots x_n)$$

It remains to see that $\vee$ is associative and can thus be done in a recursive way (*i.e.*, with a directed acyclic graph). So f is computed from OR and COPY. ■

2.3 Reversibility versus irreversibility

The NOT gate is logically reversible. This means that from the output one can recover the input. However the AND, OR gates are logically irreversible. We will now show that any Boolean circuit can be simulated by a *logically* reversible circuit. Moreover a universal set of reversible gates exists.

From $f : \mathbb{F}_2^m \to \mathbb{F}_2$ we construct $\tilde{f} : \mathbb{F}_2^m \oplus \mathbb{F}_2 \to \mathbb{F}_2^m \oplus \mathbb{F}_2$ as follows :

$$\tilde{f}(x_1 \dots x_m, y) = (x_1 \dots x_m, f(x_1 \dots x_m) \oplus y)$$

Now, $\tilde{f}$ is invertible since from $(x_1 \dots x_m, f(x_1 \dots x_m) \oplus y)$ we can recover $x_1 \dots x_m$ and $f(x_1 \dots x_m)$ [since we have a circuit for f] and then

$$y = (f(x_1 \dots x_m) \oplus y) \oplus f(x_1 \dots x_m).$$

Thus any f can be computed in a reversible way from the circuit for $\tilde{f}$. To compute f reversibly we start with the input $(x_1 \dots x_m, 0)$ compute

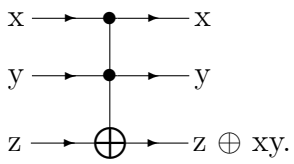
$$\tilde{f}(x_1 \dots x_m, 0) = (x_1 \dots x_m, f(x_1 \dots x_m)),$$

copy the last bit $f(x_1 \dots x_m)$ and run back the computation to get

$$\tilde{f}^{-1}(x_1 \dots x_m, f(x_1 \dots x_m)) = (x_1 \dots x_m, 0).$$

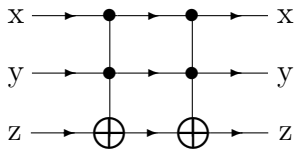
In this way we have $f(x_1 \dots x_m)$ and the circuit is left in its initial state $(x_1, \dots, x_m, 0)$. So we can achieve logical reversibility. We want to show that we can achieve physical reversibility (no heat dissipation).

What remains to be shown is that $\tilde{f}$ can be represented by a circuit containing only irreversible elementary gates. We already know that $\tilde{f}$ can be represented by a circuit containing AND, OR, NOT, COPY. We want to replace AND, OR by a reversible gate. This can be achieved by using the 3 bit gate known as *Toffoli gate* which is a CCNOT (controlled-controlled NOT):



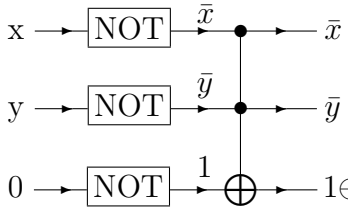
or $\quad \mathrm{T}(x, y, z) = (x, y, z \oplus xy).$

This gate flips the target bit z if both control bits x and y are equal to 1. Otherwise z is left unchanged. The Toffoli gate is reversible because:



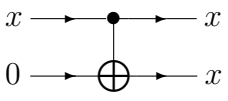
or $\quad \mathrm{T}^2(\mathrm{x},\ \mathrm{y},\ \mathrm{z}) = (\mathrm{x},\ \mathrm{y},\ \mathrm{z})$

If we set $z = 0$, $T(x, y, 0) = (x, y, xy)$ outputs $x \wedge y$ in the third bit. Thus the AND gates can be replaced by a Toffoli gate provided we increase our workspace to have a target input bit $z = 0$ and the additional outputs x & y. For the OR gate we can use

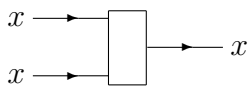


$$\begin{cases} 1 \vee 1 = 1 \\ 1 \vee 0 = 1 \\ 0 \vee 1 = 1 \\ 0 \vee 0 = 0 \end{cases}$$

Finally the COPY gate can be replaced by



which is a CNOT gate (reversible) with the target bit set to 0 in the input. Note that the pure COPY gate, if reversed:



erases a bit so there is heat dissipation. This is the reason why we replace it by a CNOT.

Summarizing we have shown that a Boolean circuit made of the universal set $\{\text{AND}, \text{OR}, \text{COPY}, \text{NOT}\}$ can be simulated by a reversible circuit made of the universal set $\{\text{CNOT}; \text{Toffoli}; \text{NOT}\}$.

The set $\{\text{AND}, \text{OR}, \text{COPY}, \text{NOT}\}$ involves single and two bit gates. On the other hand $\{\text{CNOT}, \text{Toffoli}, \text{NOT}\}$ involves single, two and three bit gates. Is it possible to build reversible circuits using only single and two bit gates? It is possible to show that the answer to this question is no. In fact it suffices to produce a counterexample: the Toffoli gate cannot be simulated reversibly with single & two bit gates. We will see that (perhaps surprisingly) in the quantum case single and two bit gates suffice for reversible computation.