B.1 Probabilistic Boolean Circuits

A Boolean circuit describes an algorithm in the form of a flow chart that connects the single bit operations, see Appendix C.12 of Part II. It has two supplemental generalizations:

a probabilistic circuit formalizes probabilistic algorithms,

a family of circuits allows to express the complexity of an algorithm for increasing input sizes.

First we formalize the concept of a probabilistic algorithm for computing a map

$$f\colon A \longrightarrow \mathbb{F}_2^s$$

on a set A. To this end we consider maps (to be represented by circuits)

$$C\colon A \times \Omega \longrightarrow \mathbb{F}_2^s$$

where Ω is a probability space. We look at the probabilities that C "computes" $f(x)$ or f:

$$P(\{\omega \mid C(x,\omega) = f(x)\}) \quad (\text{"locally" at } x) \text{ and}$$

$$P(\{(x,\omega) \mid C(x,\omega) = f(x)\}) \quad (\text{"globally"})$$

that we want to be "significantly" $> \frac{1}{2^s}$, the probability of hitting a value in $\mathbb{F}_2^s$ by pure chance. In the local case we average over Ω for fixed x, in the global case we average also over $x \in A$. In general we assume that the probability spaces Ω and $A \times \Omega$ are finite and (in most cases) uniformly distributed.

In order to describe probabilistic algorithms we need circuits with *three* different types of input nodes:

- r **deterministic input nodes** that are seeded by an input tuple $x \in \mathbb{F}_2^r$, or x from a subset $A \subseteq \mathbb{F}_2^r$,
- some **constant input nodes**, each a priori set to 0 or 1,
- k **probabilistic input nodes** that are seeded by an element ("event") of the LAPLACEan probability space $\Omega = \mathbb{F}_2^k$ (corresponding to k "coin tosses"), or by an element of a subset $\Omega \subseteq \mathbb{F}_2^k$.—Sometimes also other probability distributions on Ω, different from the uniform distribution, might be taken into account.

The theory aims at statements on the probabilities of the output values $y \in \mathbb{F}_2^s$.

Examples

1. Searching a quadratic non-residue for an n bit prime module p. Here we choose a random $b \in [1 \dots p-1]$ and compute $(\frac{b}{p})$ (the LEGENDRE symbol that is 1 for quadratic residues, -1 for quadratic non-residues). The success probability is $\frac{1}{2}$, the cost $\mathrm{O}(n^2)$ (see Appendix A.8).

   More generally we ask whether an h-tuple

   $$(b_1, \dots, b_h) \in \Omega = [1 \dots p-1]^h$$

   of independently choosen elements contains a quadratic non-residue. There is a probabilistic circuit (for the given p) without deterministic input nodes (but with some constant input nodes to input p),

   $$C : \mathbb{F}_2^{hn} \longrightarrow \mathbb{F}_2^n,$$

   $$C(\omega) = \begin{cases} b_i, & \text{the first } b_i \text{ that is a quadratic non-residue,} \\ 0 & \text{if none of the } b_i \text{ is a quadratic non-residue,} \end{cases}$$

   of size $\mathrm{O}(hn^2)$ that outputs a quadratic non-residue with probability $1 - \frac{1}{2^h}$. Note the deviation of this example from the definition above: Here C doesn't compute an explicitly given function f but provides output with a certain property.

2. The strong pseudoprime test: Here the input is taken from the set $A \subseteq [3 \dots 2^n - 1]$ of odd integers. We want to compute the primality indicator function

   $$f : A \longrightarrow \mathbb{F}_2, \quad f(m) = \begin{cases} 1 & \text{if } m \text{ is composite,} \\ 0 & \text{if } m \text{ is prime.} \end{cases}$$

   The probabilistic input consists of a base $a \in \Omega = [2 \dots 2^n - 1]$. The strong pseudoprime test is represented by a circuit

   $$C : \mathbb{F}_2^n \times \mathbb{F}_2^n \longrightarrow \mathbb{F}_2$$

   of size $\mathrm{O}(n^3)$, and yields the result 1 if m fails (then m is proven to be composite), 0 if m passes (then m is possibly prime). Thus C outputs the correct result only with a certain probability.

Now we formalize the property of a (probabilistic) circuit C of computing the correct value of $f(x) \in \mathbb{F}_2^s$ with a probability that "significantly" differs from a random guess: Given $\varepsilon \geq 0$, a circuit

$$C : \mathbb{F}_2^r \times \Omega \longrightarrow \mathbb{F}_2^s$$

(with r deterministic input nodes) has an **ε-advantage** for the computation of $f(x)$ or f if

$$P(\{\omega \in \Omega \mid C(x,\omega) = f(x)\}) \geq \frac{1}{2^s} + \varepsilon \quad (\text{“local case”}) \text{ or}$$

$$P(\{(x,\omega) \in A \times \Omega \mid C(x,\omega) = f(x)\}) \geq \frac{1}{2^s} + \varepsilon \quad (\text{“global case”}).$$

Thus in the global case the probability with respect to ω of getting a correct result is additionally averaged over $x \in A$. The advantage 0, or the probability $\frac{1}{2^s}$, corresponds to randomly guessing the result.

C has an **error probability** δ for computing $f(x)$ or f if

$$P(\{\omega \in \Omega \mid C(x,\omega) = f(x)\}) \geq 1 - \delta \quad \text{or}$$

$$P(\{(x,\omega) \in A \times \Omega \mid C(x,\omega) = f(x)\}) \geq 1 - \delta.$$

Examples

1. For searching a quadratic non-residue $\bmod p$ we have

   $$P(\{\omega \in \Omega \mid C(\omega) \text{ is a quadratic non-residue}\}) = 1 - \frac{1}{2^h}.$$

   Thus the circuit has an $(\frac{1}{2} - \frac{1}{2^h})$-advantage and an error probability of $\frac{1}{2^h}$.

2. For the strong pseudoprime test we have for fixed m

   $$P(\{\omega \in \Omega \mid C(m,\omega) = f(m)\}) \begin{cases} \geq \frac{3}{4} & \text{if } m \text{ is composite,} \\ = 1 & \text{if } m \text{ is prime.} \end{cases}$$

   Averaging over m we get

   $$P(\{(m,\omega) \in A \times \Omega \mid C(m,\omega) = f(m)\}) \geq \frac{3}{4},$$

   hence an $\frac{1}{4}$-advantage and an error probability of $\frac{1}{4}$. (Since the number of composite integers is much larger than the number of primes, the value $\frac{1}{4}$ is not significantly changed by averaging over m.)