

# Stochastic Switching Circuit Synthesis

Daniel Wilhelm

Dept. of Computation and Neural Systems  
California Institute of Technology  
Pasadena, CA 91125  
wilhelm@caltech.edu

Jehoshua Bruck

Dept. of Electrical Engineering  
California Institute of Technology  
Pasadena, CA 91125  
bruck@caltech.edu

## Abstract

In his 1938 Master's Thesis, Shannon demonstrated that any Boolean function can be realized by a switching relay circuit, leading to the development of deterministic digital logic. Here, we replace each classical switch with a probabilistic switch (pswitch). We present algorithms for synthesizing circuits closed with a desired probability, including an algorithm that generates optimal size circuits for any binary fraction. We also introduce a new duality property for series-parallel stochastic switching circuits. Finally, we construct a universal probability generator which maps deterministic inputs to arbitrary probabilistic outputs. Potential applications exist in the analysis and design of stochastic networks in biology and engineering.

## I. INTRODUCTION.

Claude Shannon, in his 1938 Master's Thesis, discovered a systematic synthesis procedure to realize a given Boolean function using deterministic switches [Sha38]. This classical contribution led to the development of modern digital logic design and is at the foundation of our ability to design and manufacture digital circuits with millions of transistors.

Most importantly, Shannon showed how logic (Boolean algebra) can be mapped to physics (relay-based switching circuits). Shannon focused on deterministic variables and functions; by closing a subset of switches, a switching circuit and its associated Boolean function yield a deterministic output.

The natural question is: can we create a similar theory for stochastic variables and functions? Namely, given a desired probability distribution and a set of probabilistic switches (that we call *pswitches*) as building blocks, can we systematically design a switching circuit that realizes a desired probability distribution? Our main contribution is a positive answer to this question for the case where the probability distributions involved are Bernoulli.

Shannon's work focused on the so-called two-terminal series-parallel circuits. A *two-terminal circuit* is an undirected graph with exactly two nodes labeled as terminals, where each node can be visited by a path between the terminals. A two-terminal circuit  $C$  is *series-parallel* (sp) iff  $C$  is: (1) a single switch, or (2) a series or parallel combination of two sp circuits.

In this paper, we will also investigate a surprisingly powerful subset of series-parallel circuits that we call *simple series-parallel* (ssp). A two-terminal circuit  $C$  is ssp iff  $C$  is: (1) a single switch, or (2) a single switch in series or parallel with a ssp circuit.

Shannon's work focused on deterministic switching circuits, circuits where each switch is associated with a Boolean variable defining whether the switch is closed. We instead focus on *stochastic switching circuits*, circuits where each pswitch is associated with a Bernoulli random variable defining the (independent) probability that the pswitch is closed.

Let  $Pr(C)$  represent the probability that some switching circuit  $C$  is closed. A circuit is closed iff its terminals are connected; otherwise, it is open. Some probability  $x$  is *realized* by  $C$  iff  $x = Pr(C)$ . Connecting a single terminal of  $A$  with one terminal of  $B$  places them in series such that the new circuit is closed only when both  $A$  and  $B$  are closed (see Figure 1b). Connecting both terminals of two switching circuits  $A$  and  $B$  places them in parallel, such that the new circuit is closed only when at least one of  $A$  and  $B$  are closed (see Figure 1c).

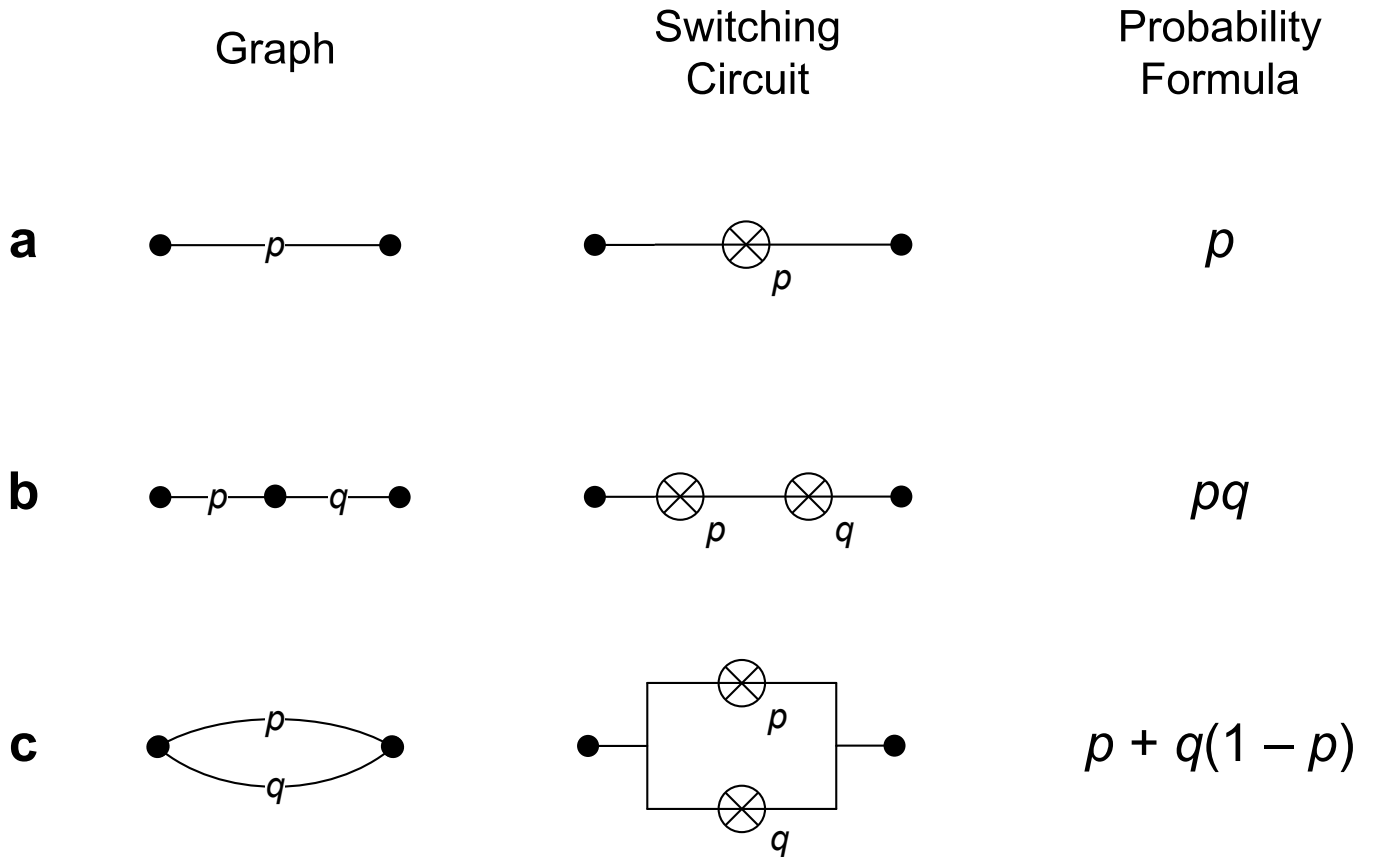


Fig. 1. **Series and Parallel Constructions.** The relationship between graphs and switching circuits is shown. There are only two ways to combine two single-pswitch circuits, both shown here. (a) Only one single-pswitch circuit exists. (b) Series. Only when both pswitches are closed is a series circuit closed. (c) Parallel. Only when both pswitches are open is a parallel circuit open.

Now, we will add a single pswitch closed with probability  $x$  to an established circuit  $C$ . Let  $F = Pr(C)$ . First, note that only one switching circuit exists with a single pswitch (see Figure 1a). If the pswitch is added in series, then the pswitch and  $C$  must both be closed; hence, the new circuit is closed with probability  $F' = Fx$ . If the pswitch is added in parallel, then the circuit is only open if both the pswitch and  $C$  are open; hence, the new circuit is closed with probability  $F' = 1 - (1 - x)(1 - F) = (1 - x)F + x$ .

In this paper, we shall construct two-terminal stochastic switching circuits where each pswitch is closed with some rational probability. The set of possible pswitch closure probabilities from which a circuit is constructed will be referred to as the *pswitch set*  $S$ . We will call a circuit which realizes a Bernoulli distribution using the fewest possible pswitches an *optimal size circuit*. For example, given a pswitch set  $S = \{\frac{1}{2}\}$ , we can use four pswitches to construct a stochastic switching circuit with probability  $P = \frac{11}{16}$  (see Figure 2). No other circuit can be constructed which realizes it with fewer pswitches (proven in Section II), and so it is also an optimal size circuit.

We are now ready to state the main results in the paper:

- 1) *Synthesizing optimal size switching circuits* which realize Bernoulli distributions. (Sections II, III)
- 2) *A duality property* allowing for optimality and existence proofs. (Section III)
- 3) *A universal probability generator* (UPG) which maps  $n$  deterministic input bits to all  $n$ -bit binary fractions (in increasing order) using only  $4n - 2$  switches. The UPG can be used to synthesize a circuit realizing any arbitrary deterministic to probabilistic mapping. (Section IV)

In the final two sections, we will discuss additional relationships between Boolean algebra and stochastic switching circuits. In Section V, we discuss two additional representations of stochastic switching circuits

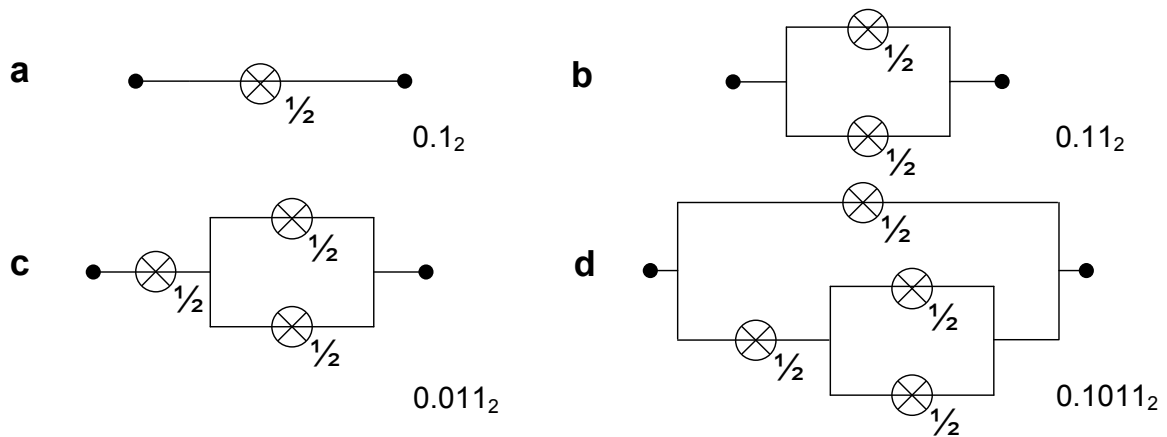


Fig. 2. **Realizing**  $F = \frac{1}{16} = 0.1011_2$ . Progressing from the least-significant to the most-significant bit in the binary representation of  $F$ , a pswitch is added in series if '0' and in parallel if '1'. The probability that each circuit is closed as a binary fraction is printed beneath each circuit. (a) Begin with a single pswitch; the least significant bit is always one, since we can remove any trailing zeros. (b) To add a pswitch in parallel, we shift the binary fraction representing the circuit's probability of closure right with replacement. (c) To add a pswitch in series, we shift the binary fraction right without replacement. (d) Using this algorithm, the generated switching circuit is proven to use the least possible number of pswitches to realize the binary fraction, across all switching circuits.

– as probabilistic Binary Decision Diagrams (BDDs) [Mei98] and as Boolean polynomials [Lec71]. We use probabilistic BDDs to prove the correctness of the B-algorithm, a major synthesis result discussed in Section II. We also show that the symbolic probability of any Boolean switching circuit being closed is equivalent to its Boolean polynomial.

Before we continue with the details of our results, we describe some of the related literature. Series-parallel circuits, a subset of switching circuits, have been rigorously analyzed, including their enumeration [RS42], duality properties [Mac92], and other interesting topological properties [Duf65]. For instance, Duffin found that the presence of a Wheatstone bridge is necessary and sufficient to make a circuit non-series-parallel [Duf65].

Many duality properties for series-parallel circuits have been studied, particularly with resistor networks [Mac92], logic gate networks (e.g. De Morgan's Law) [Whi61], and electrical networks [Tel40].

Circuit elements have been traditionally modeled stochastically to assess reliability of components [Col87]. To produce a system failure, a series connection of components only requires a single failure, whereas a parallel connection of components requires all components to fail. Several physical circuits have also been proposed for designing stochastic systems. For example, Gill suggested how to generate a probability transformation element using sequential memory logic [Gil63].

## II. REALIZING BINARY FRACTIONS.

We first present a simple algorithm that constructs an  $n$ -pswitch circuit for any probability expressible as an  $n$ -bit binary fraction. We prove that the resulting circuit is optimal in size, for any switching circuit.

*The B-algorithm:* an algorithm for generating circuits that realize binary fractions with pswitch set  $S = \{\frac{1}{2}\}$ . Let  $F_i$  be the  $i$ th least significant bit of  $F$ , an  $n$ -bit binary fraction with (WLOG) a nonzero least-significant bit.

- 1) Let circuit  $C_1$  be the single-pswitch circuit.
- 2) For bit  $F_i$ ,  $i = 2$  to  $n$ , let circuit  $C_i$  be:
  - a) If  $F_i = 0$ ,  $C_1$  in series with  $C_{i-1}$ , or
  - b) If  $F_i = 1$ ,  $C_1$  in parallel with  $C_{i-1}$ .

See Figure 2 for an example which realizes  $11/16$ ; namely, we use the B-algorithm with  $F = 1011_2$ .

*Theorem 1:* The B-algorithm synthesizes a stochastic switching circuit closed with probability  $F$ .

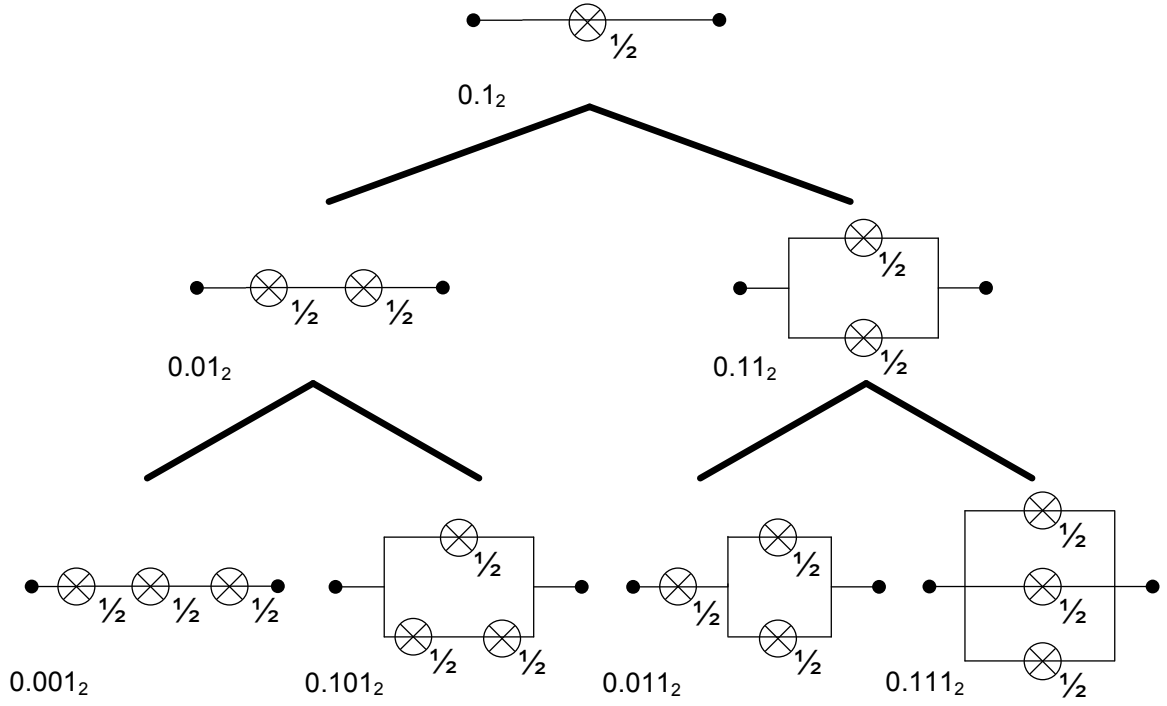


Fig. 3. **Tree of ssp Circuits.** The binary tree depicted here shows how to generate all possible ssp circuits. Each left branch depicts inserting a pswitch in series, and each right branch depicts inserting a pswitch in parallel. Note that adding a series pswitch shifts '0' into the most significant bit of the binary probability fraction; the addition of a parallel pswitch shifts in '1'.

*Proof:* The proof is by induction on the number of bits in  $F$ . For the base case, we begin with  $C_1$  as the single-pswitch circuit closed with probability  $1/2$ ; in other words,  $Pr(C_1) = 0.1_2$ .

Now, suppose that some circuit  $C_i$  is closed with probability  $Pr(C_i) = 0.\bar{x}$ , where  $\bar{x}$  is a bit vector. Then, we will show that a pswitch added in series or parallel yields  $Pr(C_{i+1}) = 0.0\bar{x}$  or  $Pr(C_{i+1}) = 0.1\bar{x}$ , respectively. Adding a pswitch in series yields  $Pr(C_{i+1}) = Pr(C_i)/2$ , namely, a right shift with an addition of '0' in the most significant bit of  $Pr(C_i)$ . Adding a pswitch in parallel yields  $Pr(C_{i+1}) = 1/2 + Pr(C_i)/2$ , namely, a right shift with an addition of '1' in the most significant bit of  $Pr(C_i)$ .

Hence, by using this construction for each of the  $n$  bits in  $F$ , an  $n$ -pswitch circuit that realizes  $F$  is synthesized. ■

Note that the B-algorithm only produces a subset of all switching circuits – those synthesizable by adding single pswitches in series or in parallel. To see how the B-algorithm works graphically, see Figure 3.

We will now prove that even when *all* switching circuits are considered, the B-algorithm is optimal in the number of pswitches. To do this, we will first present an equation that holds for all switching circuits. Suppose that we add a new pswitch closed with probability  $x$  to an existing circuit. Then, where  $C$  and  $O$  are the probabilities of the circuit formed by adding a wire and a short in place of the new pswitch, respectively:

$$P' = xC + (1 - x)O \quad (1)$$

Using this equation, we can prove statements about all stochastic switching circuits (beyond those which are enumerably series-parallel). Note that  $C > O$ . Intuitively, by adding a new pswitch we create a new potential path between the terminals. Hence, for the new circuit  $M$ ,  $Pr(M) > O$ . Since closing the new path makes the terminals more likely to be connected, then  $C > Pr(M)$ . This demarkation of  $C > O$

actually stems from a sign convention –  $x$  represents the probability that a circuit is closed rather than open.

In the following theorem, we shall use the previous equation to prove a general lower bound for pswitch sets. Here, the optimality of the B-algorithm corresponds to the case  $q = 2$ .

*Theorem 2:* Let  $q$  be an arbitrary positive integer. Using the pswitch set  $\mathbf{S} = \{1/q, 2/q, \dots, (q-1)/q\}$ , an optimal size circuit  $C$  that realizes a rational  $F = a/q^n$ ,  $0 < a < q^n$ , requires at least  $n$  pswitches.

*Proof:* We will assume that every optimal size circuit  $C$  as defined in the claim of the theorem has size at most  $n-1$ . The idea is to show that if  $C$  (which realizes  $F \in \{0, q-1\}^n$ , the base- $q$  representation of the desired probability) of size  $n-1$  exists, then eventually we must realize a non-integral probability with zero pswitches, a contradiction.

Here is the idea in the proof. Suppose we have a circuit  $C_i$  such that  $F^i$  associated with  $Pr(C_i)$  has  $i$  digits in the alphabet  $\{0, \dots, q-1\}$ . We can assume that  $F^i$  has a nonzero value in the least significant digit. Then, we will choose some pswitch  $x$  in  $C_i$ , create two new circuits by opening or closing  $x$ , and prove that one of the circuits has probability represented by  $F^{i-1}$ , with  $i-1$  digits and a nonzero value in its least significant digit. Namely, we can reduce the number of pswitches by one and get a probability value that uses one less digit. This process will lead to a contradiction.

By the laws of probability,

$$\begin{aligned} Pr(C_i) &= Pr(C_i|x \text{ open})Pr(x \text{ open}) + \\ &\quad Pr(C_i|x \text{ closed})Pr(x \text{ closed}) \\ &= a/q^i. \end{aligned} \tag{2}$$

We know that  $Pr(x \text{ closed}) \in \mathbf{S}$  and  $Pr(x \text{ open}) \in \mathbf{S}$ , since  $Pr(x \text{ open}) = 1 - Pr(x \text{ closed})$ . Both denominators are  $q$ , so for some integers  $b$  and  $c$  where  $0 < b, c < q-1$ ,

$$\frac{a}{q^{i-1}} = bPr(C_i|x \text{ open}) + cPr(C_i|x \text{ closed}). \tag{3}$$

Let  $F^i$  be the string associated with circuit  $C_i$ , where  $Pr(C_i) = a/q^i$ , for integers  $a$  and  $q$ .  $F^i$  has length  $i$  and a non-zero least significant digit. Then, by opening or closing some pswitch  $x$  in  $C_i$ , one of the new circuits is at least  $i-1$  digits long. Why? Suppose that both of the new circuit probabilities are  $r, s < i-1$  bits long. Then, there exists  $b', c' \in \mathbf{N}$  such that  $a/q^{i-1} = b'/q^r + c'/q^s$ , where each fraction has a nonzero digit in its least significant digit. Then,  $a = b'q^{i-r-1} + c'q^{i-s-1} = q^{i-r-1}(b' + c'q^{r-s})$ . However,  $q|a$ , so  $a$  has a 0 in its least significant digit, producing a contradiction.

If we assume that there exists  $C$  as defined in the claim of the theorem with at most  $n-1$  pswitches realizing an  $F$  with  $n$  digits, then we can apply the above process  $n-1$  times and eliminate all the pswitches. However, the resulting probability will still be a fraction, and we reach a contradiction. ■

### III. DUALITY.

Duality is an important property integral to the study of circuits. The *dual* of a circuit is a systematic change of the structure of the circuit at each stage in its construction to yield some inverse global property. This concept of duality appears in many domains, three of which are illustrated in Figure 4.

To take the inverse (the NOT) of a Boolean function composed of AND and OR gates, we can apply De Morgan's duality law; namely,  $\overline{A \wedge B} = \overline{A} \vee \overline{B}$  (see Figure 4a) [Whi61].

Likewise, Macmahon showed that the dual of a series-parallel resistor network composed of  $r$ -ohm resistors with equivalent resistance  $(p/q)r$  has the equivalent resistance  $(q/p)r$  (the inverse) [Mac92] (see

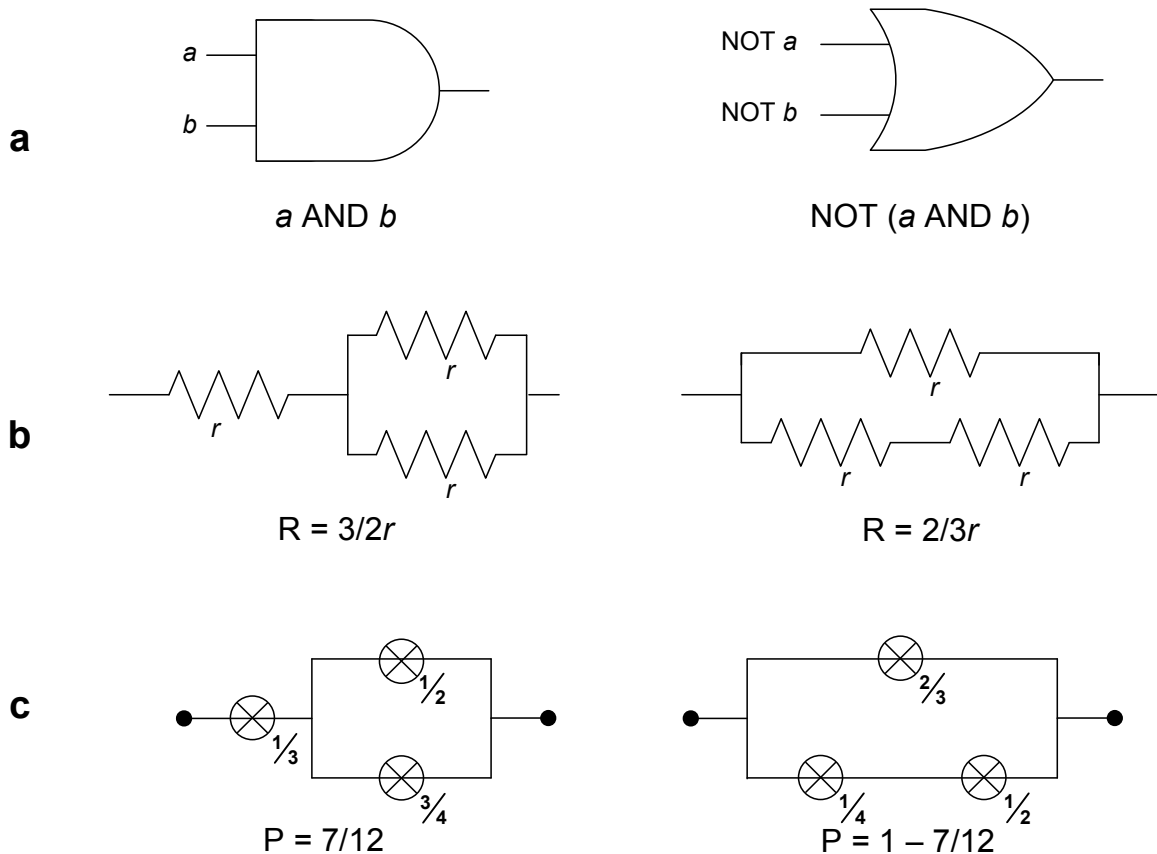


Fig. 4. **Duality.** Duality has played an important role in the analysis of circuits. (a) The dual of logic gates, by De Morgan's Law (in words,  $(a \text{ AND } b) = \text{NOT} ((\text{NOT } a) \text{ OR } (\text{NOT } b)) = \text{NOT} (a \text{ OR } b)$ ); (b) The dual of series-parallel resistor circuits of equal resistance, by Macmahon. Series connections are now parallel, and vice versa; the equivalent resistance coefficients are reciprocals. (c) The dual of stochastic series-parallel switching circuits. Each pswitch closed with probability  $p$  is closed with  $1 - p$  in the dual. Note the similarity to the dual of (b).

Figure 4b). The construction of a series-parallel circuit  $C$  is a series of stages involving inserting a series-parallel circuit in series or parallel to an existing series-parallel circuit. To achieve Macmahon's duality result, the same sequence of steps is followed, but each insertion is inverted, i.e. if a circuit is added in series in  $C$ , then in the dual of  $C$  it is inserted in parallel, and vice versa. Then, if some resistor circuit  $C$  comprised of  $r$ -ohm resistors has equivalent resistance  $pr$ , then the resulting dual circuit will have an equivalent resistance of  $(1/p)r$ .

We have found that duality exists in series-parallel stochastic switching circuits as well. Like Macmahon's algorithm, if a series-parallel circuit is added in series to a circuit, it is added in parallel to the dual (and vice versa). Like De Morgan, instead of adding a circuit closed with probability  $p$ , we add a circuit closed with probability  $1 - p$ . Then, if the original circuit is closed with probability  $P$ , the dual circuit is closed with probability  $(1 - P)$ , the inverse in probabilistic terms (see Figure 4c).

We will use duality in this section to find algorithms for realizing general probability classes (e.g. the probability class of all binary fractions).

First, note that the dual of  $C$  only exists if  $C$  is series-parallel, and if for every pswitch  $x$  used in  $C$ , the pswitch  $1 - x$  exists in  $S$ . We will now show an important property – the dual of  $C$  is closed with probability  $1 - Pr(C)$ .

**Theorem 3: Duality Theorem.** Given some stochastic series-parallel circuit  $C$  and its dual  $\overline{C}$ , then  $Pr(C) + Pr(\overline{C}) = 1$ .

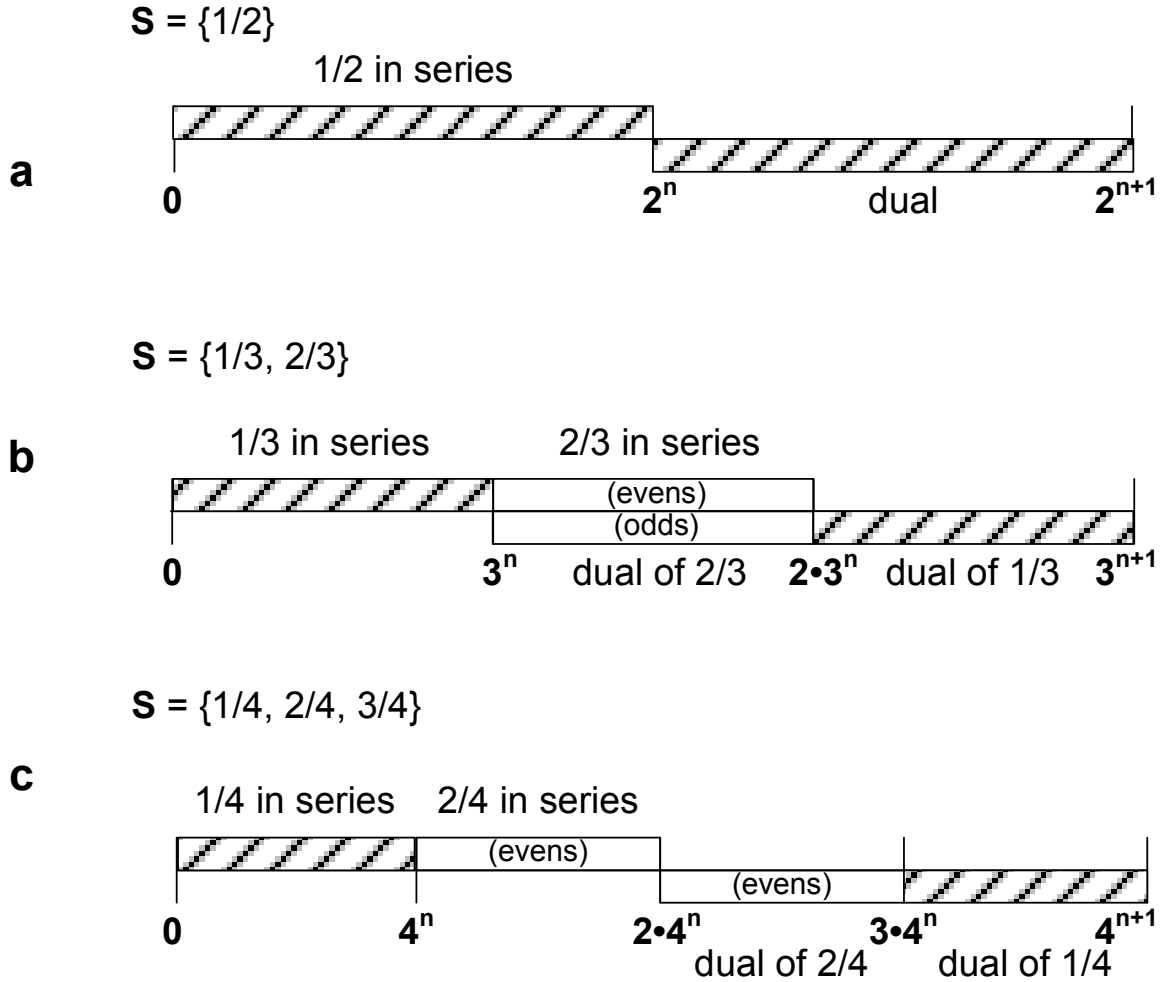


Fig. 5. **Expressive Power.** Here, we are given an initial pswitch set  $S$ . Further, we assume that all rationals  $a/q^n$ ,  $0 \leq a < q^n$ , for some positive integer  $q$ , can be realized. Using duality, we show how to realize all  $b/q^{n+1}$ ,  $0 < b < q^{n+1}$ , by adding a single pswitch. The numerator  $b$  is shown on each number line along with which pswitch to add to realize it. (a) All  $a/2^n$  can be synthesized using  $n$  pswitches. (b) All  $a/3^n$  can be synthesized using  $n$  pswitches. (c) All  $a/4^n$  can be synthesized using  $n + 1$  pswitches. Note that using  $n$  pswitches we can only generate the evens in the middle half. To generate some odd numerator  $o$ , place a  $1/2$  pswitch in series with  $2o$ , using  $n + 1$  pswitches (if  $o > 2 \cdot 4^n$ , then use duality).

*Proof:* This is shown using induction on the definition of series-parallel. For the base case, the dual of a single-pswitch circuit with pswitch  $x$  is the single-pswitch circuit with pswitch  $1 - x$ . Now, suppose that the dual of a stochastic circuit  $C$  is closed with probability  $1 - Pr(C)$ . Adding a second series-parallel circuit  $C'$  in series with  $C$  to form  $C_s$  yields  $Pr(C_s) = Pr(C)Pr(C')$ . Adding  $C'$  in parallel to the dual of  $C$  to form  $\overline{C_p}$  yields  $Pr(\overline{C_p}) = 1 - [1 - (1 - Pr(C'))][1 - (1 - Pr(C))] = 1 - Pr(C)Pr(C') = 1 - Pr(C_s)$ . The case of adding  $C'$  in parallel is identical. ■

This duality property is a powerful tool for analyzing stochastic switching circuits. Suppose that for every pswitch  $x$  in a circuit  $C$ , a pswitch  $1 - x$  is in  $S$ . As follows, the duality property can be used to prove which probability classes can be realized by all simple series-parallel circuits given  $S$ .

**Theorem 4:**  $S = \{1/2\}$ . All  $Pr(C^n) = \frac{a}{2^n}$ ,  $0 < a < 2^n$ , (i.e. all  $n$ -bit binary fractions) can be realized with at most  $n$  pswitches.

*Proof:* Suppose that all rational probabilities  $a/2^n$  can be realized. Now, add a pswitch in series with each circuit, realizing all  $(1/2)(a/2^n) = a/2^{n+1}$ ,  $0 < a < 2^n$ . By applying the Duality Theorem, every  $b/2^{n+1} = 1 - a/2^{n+1}$ , where  $2^n < b < 2^{n+1}$ , can be realized by synthesizing the dual of each original

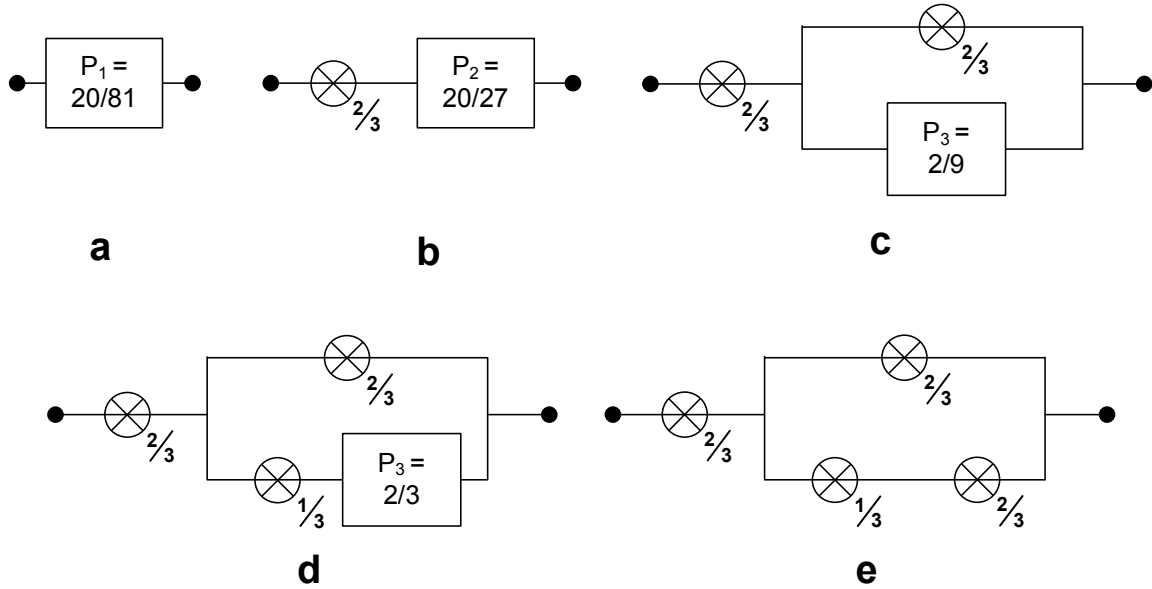


Fig. 6. **Realizing  $P = \frac{40}{81}$ , given  $S = \{1/3\}, \{2/3\}$ .** Each box represents an unknown circuit closed with the desired probability  $P$ . (a) This is an example of backwards construction, and we initially begin with an unknown circuit closed with probability  $P$ . (b) Initially,  $P_1 = 40/81$ , and here  $1 \cdot 3^3 < P_1 < 2 \cdot 3^3$ . Since the numerator 40 is additionally even, then we must add a  $2/3$  pswitch in series. The remainder of the circuit must be closed with probability  $P_2 = P_1/(2/3) = 20/27$ . (c) Now,  $P_2 = 20/27 > 2 \cdot 3^2/27$ , so the second-to-last pswitch is  $2/3$  in parallel. Hence, the probability the remaining circuit must be closed is  $P_3 = \frac{P_2 - 2/3}{1 - 2/3} = 2/9$ . (d)  $P_3 = 2/9 < 3^1/9$ , so the next pswitch is  $1/3$  in series. The remaining circuit must be closed with probability  $P_4 = P_3/(1/3) = 2/3$ . (e) Now,  $P_4 = 2/3$  is in our switch set. We replace the final missing circuit box with this, and we are finished.

circuit.

Hence, we can realize all  $n$ -bit binary fractions with  $n$  pswitches (see Figure 5a). ■

**Theorem 5:**  $S = \{\frac{1}{3}, \frac{2}{3}\}$ . All rational  $Pr(C^n) = \frac{a}{3^n}$ ,  $0 < a < 3^n$ , (i.e. all  $n$ -trit ternary fractions) can be realized with  $n$  pswitches.

*Proof:* Suppose that all rationals  $a/3^n$ ,  $0 < a < 3^n$ , can be realized. Now, add a  $1/3$  pswitch in series with each circuit, realizing all  $a/3^{n+1}$ ,  $0 < a < 3^n$ . By the Duality Theorem, all  $b/3^{n+1} = 1 - a/3^{n+1}$  can be realized by the dual of each original circuit, where  $2 \cdot 3^n < b < 3^{n+1}$ . Hence, we can synthesize the lower third and upper third of the new probabilities.

Now, we add a  $2/3$  pswitch in series with each circuit, realizing all even  $c/3^{n+1}$ ,  $3^n < c < 2 \cdot 3^{n+1}$ . By the Duality Theorem, we can realize all  $1 - c/3^{n+1}$ , i.e. all odds in the center range, by synthesizing the dual of each of the  $2/3$ -series circuits.

Hence, all  $n$ -trit ternary fractions can be realized with  $n$  pswitches (see Figure 5b). ■

Note that an optimal algorithm for realizing any rational fraction  $a/3^n$  can be obtained from the previous proof. From Theorem 2 ( $q = 3$ ), a minimum of  $n$  pswitches is required to realize all  $n$ -trit ternary fractions; hence, the strategy in the proof is optimal. Given a desired probability  $F$  and  $S = \{1/3, 2/3\}$ , then the algorithm is (see Figure 6 for an example):

- 1) Begin with an open circuit.
- 2) If  $F = 1/3$  or  $F = 2/3$ , then add the pswitch  $F$  and halt. Otherwise, let  $a$  be the numerator of  $F$ .
- 3) Add a pswitch:
  - a) If  $a < 3^n$ , add a  $1/3$  pswitch in series. (Let  $p = 1/3$ .)
  - b) If  $3^n < a < 2 \cdot 3^n$ , then:
    - i) If  $a$  is odd, add a  $1/3$  pswitch in parallel. (Let  $p = 1/3$ .)
    - ii) If  $a$  is even, add a  $2/3$  pswitch in series. (Let  $p = 2/3$ .)
  - c) If  $2 \cdot 3^n < a < 3^{n+1}$ , add a  $2/3$  pswitch in parallel. (Let  $p = 2/3$ .)



- 4) Find the new desired probability:
  - a) If a pswitch was added in series, let  $F' = F/p$ .
  - b) If a pswitch was added in parallel, let  $F' = \frac{F-p}{1-p}$ .
- 5) Let  $F = F'$ . Goto 2.

This trend of  $n$ -pswitch circuits realizing all rational numbers  $a/q^n$  cannot continue. For  $q > 3$ , already in the case of  $n = 2$  there are fractions  $a/q^2$  that cannot be realized by a size two circuit. We formalize this result in the following theorem.

*Theorem 6:* We are given a pswitch set containing only rational numbers with denominator  $q$ , where  $q > 3$ . Then, there exist rational probabilities  $a/q^2$  which cannot be realized by a pswitch circuit of size two.

*Proof:* The idea in the proof is to show that there exists a prime number  $b$  such that  $b/q^2$  cannot be realized by two pswitches.

A network with two pswitches in series results in a composite numerator (or a prime numerator that belongs to the pswitch set). Now notice that a circuit with two  $1/q$  pswitches in parallel results in the smallest fraction obtainable by adding two pswitches in parallel. The probability of this circuit is  $(2q - 1)/q^2$ .

Hence, the range of numerators for which only composite numerators can be generated is  $q < a \leq 2q - 2$ . By Bertrand's Postulate [Ram19] [Erd34], there exists at least one prime number between  $q$  and  $2q - 2$ , for  $q > 3$ ; hence, a prime numerator always exists within this range that cannot be realized. ■

As an example, suppose we have the pswitch set  $S = \{1/4, 2/4, 3/4\}$ , and we want to realize  $P = 5/16$ , where the numerator is prime, using two pswitches. Then, the largest prime-numerator fraction realized by placing the pswitches in series is  $3/16$ . The smallest fraction realized by placing the pswitches in parallel is  $7/16$  (placing  $1/4$  in parallel with  $1/4$ ). Hence,  $5/16$  cannot be realized using two pswitches.

Now, we provide an upper-bound on the number of pswitches necessary to realize any fraction  $a/4^n$  using the pswitch set of all fourths.

*Theorem 7:*  $S = \{\frac{1}{4}, \frac{2}{4}, \frac{3}{4}\}$ . All rational  $Pr(C^n) = \frac{a}{4^n}$ ,  $0 < a < 4^n$  can be realized with  $x$  pswitches, where  $n \leq x < 2n$ .

*Proof:* Suppose that all rational numbers of the form  $a/4^n$ ,  $0 < a < 4^n$ , can be realized. Now, by adding a  $1/4$  pswitch in series, the lower quadrant of  $a/4^{n+1}$  can be realized. By the Duality Theorem, the upper quadrant can be realized by synthesizing the dual of each  $1/4$ -series circuit.

By adding a  $2/4$  pswitch in series with each of the original circuits, the remaining even numerators can be synthesized. Now, any remaining odd numerator  $s$  can be generated by realizing  $2s/4^n$  and adding a  $1/2$  pswitch in series (or, if  $s > 2 \cdot 4^n$ , realizing twice the dual of  $s$ ).

Each stage requires at most two switches with the exception of the first, and so all  $a/4^n$ ,  $0 < a < 4^n$ , can be realized by at most  $2n - 1$  pswitches (see Figure 5c). ■

The previous proof suggests that many stages may require two switches. However, in practice often no more than  $n + 1$  pswitches are required, provided that certain prime numerators are not synthesized. As an example,  $5/16$  cannot be realized with two pswitches, but it can be realized with three.

Discovering the optimal number of pswitches required to realize any  $a/4^n$  (given q pswitch set  $S = \{1/4, 2/4, 3/4\}$ ) is still an open problem, as is an algorithm to generate the optimal size circuit.

#### IV. A UNIVERSAL PROBABILITY GENERATOR

Shannon showed how to synthesize any deterministic circuit from a truth table – an explicit mapping of  $n$ -bit inputs to output bits. This synthesis procedure was a fundamental building block in the evolution of modern-day computing. Here, we show an efficient synthesis procedure for a similar building block, the synthesis of any arbitrary probabilistic truth table – a mapping of  $n$ -bit deterministic inputs to output probabilities. In other words, deterministic inputs will change the structure of the probabilistic circuit to realize any desired binary fraction.

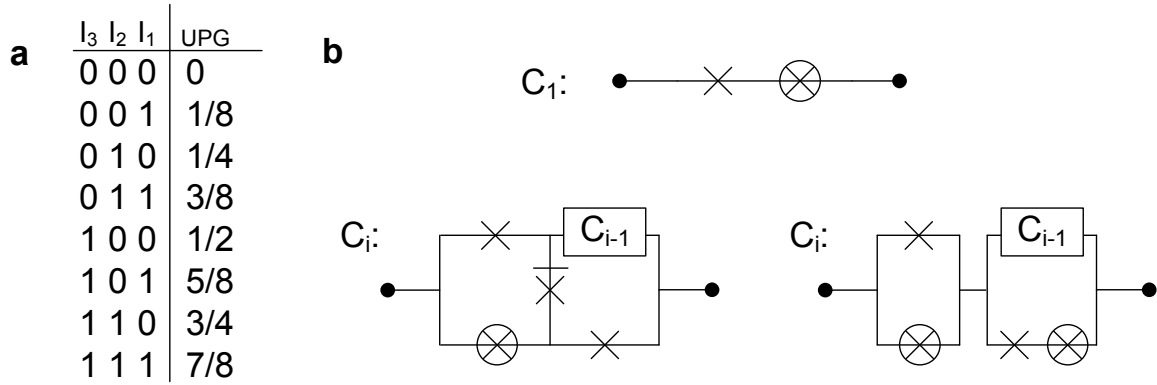


Fig. 7. **The Universal Probability Generator (UPG).** Here, we show the construction of the UPG, a circuit which maps deterministic inputs to probabilities. (a) The mappings for a UPG with three deterministic input bits. (b)  $C_1$  is two switches in series. Two possible constructions for  $C_i$  are shown – one which uses a single pswitch (left), which is minimal, and one which is monotonic in the deterministic switches (right). Each recursive circuit stage requires an additional four switches.

Most impressive, however, is that the underlying stochastic circuit powering this – the Universal Probability Generator (UPG) – can be synthesized using only a linear number of switches. To generate every  $n$ -bit binary fraction ( $2^n$  total probabilities) in increasing order, selectable using  $n$  deterministic bits, requires only  $4n - 2$  switches. Only  $n$  of these, the minimum possible number, need be probabilistic.

After we show how to construct a UPG, we will then return to Shannon’s deterministic synthesis results to build a combinational logic block which maps deterministic inputs to arbitrary probabilistic outputs. Then, we will have a synthesis procedure for constructing any probabilistic truth table (where the probabilities are binary fractions).

We define a *universal probability generator* (UPG) to be a circuit which maps  $n$  deterministic input bits to all  $2^n$   $n$ -bit binary fractions in increasing order (e.g. for  $n = 3$ , see Figure 7a).

This can be easily accomplished using an exponential number of switches; we simply construct each of the  $2^n$  probabilistic circuits separately then uniquely select them with deterministic switches. Using the B-algorithm from earlier, we can synthesize all  $2^n$  probabilistic circuits closed with rational probability  $a/2^n$ , where  $0 \leq a < 2^n$ , each requiring  $n$  pswitches. Then, we can add a deterministic selector which only selects one of the  $2^n$  probabilistic circuits, depending on the  $n$  deterministic input bits. This is an inefficient method, however, since it minimally requires an exponential number of pswitches.

Here, we propose two constructions which require only  $4n - 2$  total switches. The first construction requires  $n$  pswitches, the fewest possible; the second is monotonic in the value of its deterministic variables. In this section, all pswitches will be closed with probability  $1/2$ .

*Theorem 8:* The following recursive construction will synthesize an  $n$ -bit deterministic input UPG circuit  $C_n$  using  $4n - 2$  switches:

- 1) Given a deterministic input  $x$ , let  $C_1$  be a switch and pswitch in series as in Figure 7b.
- 2) To synthesize circuit  $C_i$ , substitute  $C_{i-1}$  into the template for  $C_i$  given in Figure 7b.
- 3) Let all deterministic switches added for circuit  $C_i$  be closed (unless negated) iff the  $i$ th bit from the least significant input bit is '0'.

*Proof:* For  $C_1$ , if the deterministic switch is open, we realize  $0/2$ ; if it is closed, we realize  $1/2$ . Hence, we realize all 1-bit binary fractions (See Figure 8a).

Suppose that we can generate all  $(i - 1)$ -bit binary fractions with circuit  $C_{i-1}$ . Then, we shall show we can generate all  $i$ -bit binary fractions with circuit  $C_i$ . If the  $i$ th least significant input bit is '0', then the deterministic switches added for  $C_i$  are open (unless negated). Hence, in both constructions in Figure 7b, a  $1/2$  pswitch is connected in series with  $C_{i-1}$ . This yields the first half of the new numerators, since  $a/2^n \cdot 1/2 = a/2^{n+1}$ . Similarly, if the  $i$ th bit is '1', then a  $1/2$  pswitch is connected in parallel with  $C_{i-1}$ ,

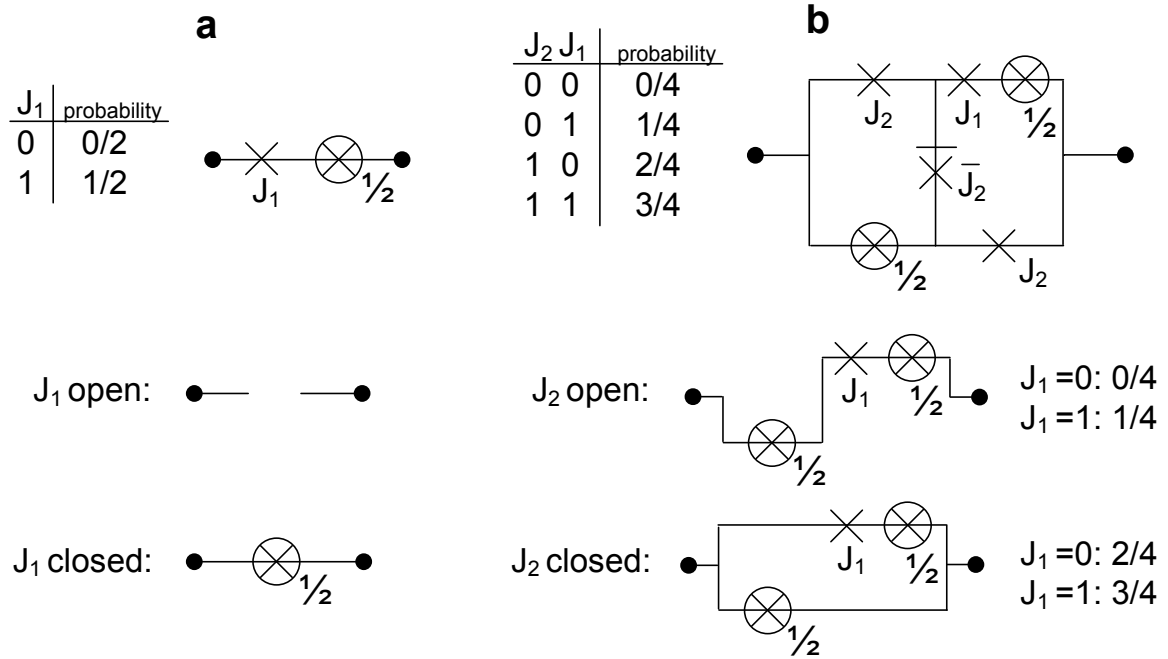


Fig. 8. **One-bit and Two-bit UPGs.** (a) For one deterministic bit  $J_1$ , a UPG can generate all probabilities  $0/2$  and  $1/2$ , in increasing order by the deterministic input. (b) For two deterministic bits  $J_2$  and  $J_1$ , a UPG generates all probabilities  $0/2^2$ ,  $1/2^2$ ,  $2/2^2$ , and  $3/2^2$ , again in increasing order.

yielding the second half of the new numerators, since  $1 - 1/2 \cdot (1 - a/2^n) = a/2^{n+1} + 1/2$  (see Figure 8b). Hence, every  $i$ -bit binary fraction is generated for each recursive step, mapping each deterministic input  $x$  to  $x/2^n$ .

In both constructions, four switches are required for each  $C_i$ , excluding  $C_1$  for which only two are required. Hence, after  $n - 1$  recursions and the base case, we can generate  $2^n$  probabilities with  $4n - 2$  switches. ■

In addition to these two constructions, also note that the dual of the series-parallel UPG construction is also a valid UPG circuit. By the Duality Theorem, this construction will realize all  $1 - Pr(C_n)$ , yielding again  $2^n$  unique  $n$ -bit binary fractions. Also, note that a parallel circuit can also be used for  $C_1$ ; this would map each deterministic input  $x$  to  $(x + 1)/2^n$ , generating an always-closed circuit ( $2^n/2^n$ ) rather than an always-open circuit ( $0/2^n$ ).

After generating the UPG, deterministic inputs yield probabilities in increasing order. To create an arbitrary mapping, a combinational logic block can be added (see Figure 9a) which maps the desired deterministic inputs to the UPG deterministic inputs using classical logic synthesis.

We will now provide an example of constructing a circuit which realizes the truth table in Figure 9b. First, since 3-bit binary fractions must be realized by the circuit, we build a 3-bit deterministic input UPG by following the construction rules (see Figure 9c). The UPG maps three deterministic inputs to probabilities as shown in Figure 7a. Hence, we will add combinational logic to map the circuit deterministic inputs  $I$  to the UPG deterministic inputs  $J$ .

Note that the mentioned probabilistic truth table also can be interpreted as a discrete probability distribution. By only using pswitches and deterministic switches, by the procedure above any discrete probability distribution of binary fractions can be generated.

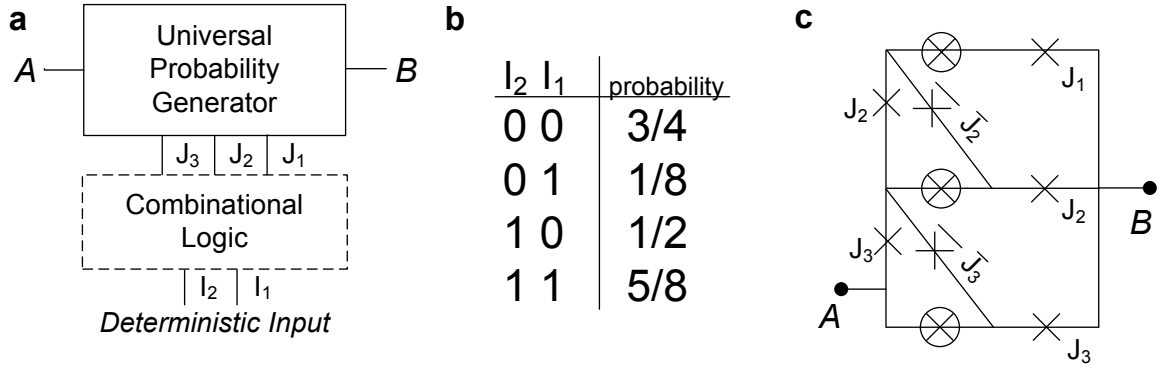


Fig. 9. **Realizing a Probabilistic Truth Table.** As an example, we will synthesize a UPG and combinational logic block which satisfies a given mapping between deterministic inputs and probabilities. (a) Combinational logic allows arbitrary mappings to be achieved. (b) The desired truth table, mapping two input bits to probabilities. (c) A UPG construction which can realize any 3-bit binary fraction. To achieve the mapping in the truth table, the following Boolean functions must be realized in the combinational logic –  $J_1 : I_1$ ,  $J_2 : \bar{I}_1 \wedge \bar{I}_2$ , and  $J_3 : \bar{I}_1 \vee I_2$ .

## V. REPRESENTATIONS.

In this section, we shall show alternative representations of stochastic switching circuits using binary decision diagrams and Boolean polynomials.

### A. Probabilistic Binary Decision Diagrams.

Binary Decision Diagrams (BDDs) describe Boolean formulas using binary trees [Mei98]. A probabilistic BDD associates a random variable with each non-sink node, indicating the probability that each child is selected [HTP06]. Using these, we will show an additional proof of the B-algorithm.

We define a *single-chain BDD* to be a BDD where, for each node, the '0' child leads to:

- 1) a 0-sink and the '1' child either leads to a one-sink or a single-chain BDD, or
- 2) a single-chain BDD and the '1' child leads to a one-sink.

**Theorem 9:** Simple series-parallel (ssp) circuits and single-chain BDDs are isomorphic.

**Proof: One-to-one.** We claim that for each ssp circuit there exists a unique single-chain BDD. We will show this by induction on the definition of the ssp circuit. Suppose the ssp circuit is a single switch. Then, it can be represented as a single leaf node branching to a zero-sink and a one-sink, which is a single-chain BDD by definition.

Now, suppose that some ssp circuit with  $n$  pswitches  $C_n$  can be represented by the single-chain BDD  $O_n$  and that a pswitch is added in parallel to  $C_n$ . Add a new node  $N_p$  to  $O_n$  representing the new pswitch. If the switch is closed, then the circuit is closed, so one child of  $N_p$  is a one-sink. If the pswitch is open, then the circuit is closed when  $C_n$  is closed, so the second child of  $N_p$  is the root node of  $O_n$ . Hence,  $O_{n+1}$  is a single-chain BDD. When a node  $N_s$  representing the added pswitch is added in series to  $O_n$ , then a single-chain BDD is formed in a similar manner.

Any given ssp circuit is a unique construction of series and parallel pswitches. All single-chain BDDs can be generated by adding left- and right-children. A single-chain BDD is added as the left-child when a pswitch is added in parallel and as a right-child when a pswitch is added in series. Hence, the BDD for each ssp circuit is unique. Therefore, by induction we will show that any ssp circuit can be represented by a unique single-chain BDD.

**Onto.** We claim that the number of ssp circuits is equivalent to the number of single-chain BDDs. We claim there are  $2^{n-1}$  single-chain BDDs given exactly  $n$  nodes. For a single node, there exists exactly one single-chain BDD, where both children are constants, and  $2^{1-1} = 1$ . As shown above, each node can either add the existing BDD as a left-child or as a right-child. Hence, since there are only two possibilities, then for  $k + 1$  nodes, there are  $2 \cdot 2^{k-1} = 2^k$  possibilities. By induction, then, for exactly  $n$  nodes there

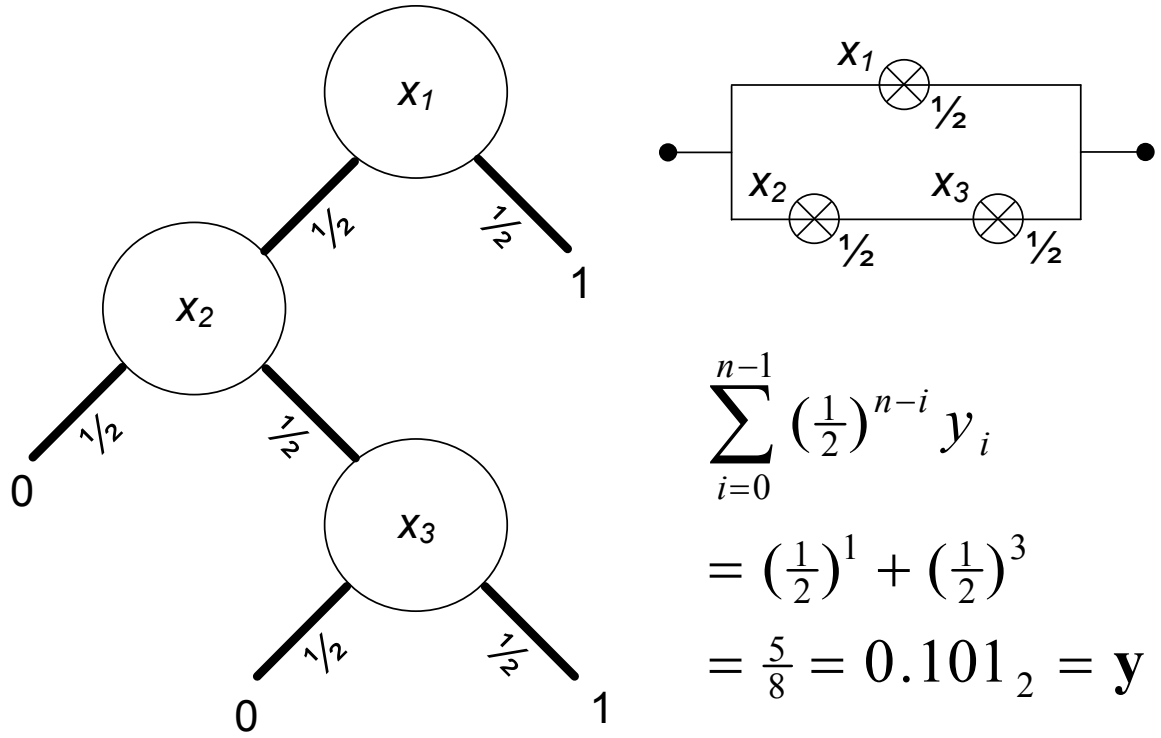


Fig. 10. **An ssp Circuit and Single-Chain BDD.** The isomorphic single-chain BDD of an ssp circuit is shown. The probability that the circuit is closed can be computed easily from the BDD. Each switch  $i$  is on or off, represented by  $x_i = 1$  and  $x_i = 0$ , respectively. By constructing a vector  $\mathbf{y}$  which for each  $i$ ,  $y_i = 1$  iff the BDD returns '1' when  $x_i = 1$ , then the circuit can also be represented compactly as a binary string.

exist  $2^{n-1}$  single-chain BDDs. We have shown previously that there exist  $2^{n-1}$  ssp circuits given exactly  $n$  switches. Then, because it was shown that the mapping is one-to-one, all possible single-chain BDDs must be mapped to.

As shown, the mapping between ssp circuits and simple BDDs is one-to-one and onto. Hence, they are isomorphic. ■

Now, we will prove the B-algorithm using probabilistic BDDs.

**Theorem 10: The B-algorithm.** Given a ssp circuit  $C_n$  with  $n$  pswitches, each closed with probability  $1/2$ , such that  $Pr(C_n)$  is a binary fraction (i.e. a rational number of the form  $a/2^n$ ), then  $C_n$  is exactly described by the binary representation of  $a$  with  $n$  bits.

*Proof:* By the previous theorem, there exists some single-chain BDD  $O_n$  isomorphic to  $C_n$ . Then, the probability of  $C_n$  being closed is equivalent to reaching a one-sink by a random walk down the BDD of  $O_n$ . Suppose that some binary vector  $\mathbf{y}$  describes the single-chain BDD in the following manner: for some  $i$ ,  $y_i = 1$  if node  $N_i$  has a child which is a one-sink (where node  $N_{n-1}$  is the root node), and  $y_i = 0$  otherwise. For any  $y_i$ ,  $i$  paths must be followed, each with probability  $1/2$ , to reach the one-sink present at node  $i$ . So, the probability of reaching the  $i$ th node one-sink is  $(1/2)^{n-i}$ . Then, the total probability of reaching a one-sink in  $O_n$  is the sum of the probabilities of reaching each one-sink, or:

$$P = \sum_{i=1}^n y_i \left(\frac{1}{2}\right)^{n-i} = \frac{\sum_{i=1}^n (2^i y_i)}{2^n}. \quad (4)$$

Now, the numerator is the decimal expansion of the binary vector  $\mathbf{y}$  with  $n$  bits, where  $n$  is the number of switches in  $C_n$ . Hence, the numerator is equivalent to  $a$  in the probability expression. So,  $\mathbf{y}$  exactly describes  $C_n$ , and it is the binary representation of  $a$ . ■

For an example of how a ssp circuit and single-chain BDD are related, see Figure 10. Note that, as discussed in [Mei98], the ordering of the Boolean variables in the BDD (i.e. the Boolean variable which each layer corresponds to) affects whether a single-chain BDD can be formed. Hence, to be more precise, above we proved that for each simple series-parallel circuit, there exists at least one ordering of variables such that a single-chain BDD can be formed.

### B. Boolean Polynomials.

The relationship between Boolean functions and switching circuits was explored by Shannon in 1938 [Sha38]. In this paper, we investigate the relationship between probability functions and stochastic switching circuits. However, it is also of interest to investigate the relationship between Boolean functions and probability functions. For example, the circuit  $A \text{ OR } B$  is equivalent to  $A + B$  in Boolean algebra, but it is  $Pr(A) + Pr(B) - Pr(AB)$  in probabilistic terms. Hence, we ask in this section whether there exists a single form that can express both Boolean expressions and probabilistic expressions. We answer this in the affirmative – for any switching circuit, the probability function is equivalent to the Boolean polynomial.

A *Boolean polynomial* is a sum of monomials, each the product of Boolean variables. In this context, for every Boolean variable  $x$ ,  $x \in \{0, 1\}$ . Polynomials of this form were referred to in an early paper of George Boole [Boo47]. In prior literature, it has been shown that a unique Boolean polynomial exists for every Boolean function [Lec71]. As an example, the Boolean formula  $A \wedge B$ , where  $A$  and  $B$  are Boolean variables, is equivalent to the Boolean polynomial  $A \cdot B$ . If  $A = B = 1$ , then  $A \cdot B = 1 \cdot 1 = 1$ . When neither  $A$  nor  $B$  is 1, then  $A \cdot B = 0$ . Note that these compute the *AND* function. As a second example, the Boolean formula  $A \vee B$  is isomorphic to the Boolean polynomial  $P = A + B - AB$ . Here, if  $A = B = 0$ , then  $P = 0 + 0 - 0 = 0$ . Otherwise,  $P = 1$ , just as the logical function *OR*.

For the below theorems, for some switch  $x$ , let  $c(x)$  and  $o(x)$  be Boolean polynomials describing the switching circuit formed when  $x$  is closed and open, respectively.

*Theorem 11:* Given any switching circuit, adding a switch  $x$  between any two terminals A and B yields the Boolean polynomial  $p' = x(c(x) - o(x)) + o(x)$ .

*Proof:* Suppose that  $x$  is closed. Then,  $x = 1$  and  $p' = c(x)$  since  $x$  directly connects terminals A and B. Suppose that  $x$  is open. Then,  $x = 0$  and  $p' = o(x)$  since  $x$  ensures that terminals A and B are not connected. Since the polynomial  $p' = x(c(x) - o(x)) + o(x)$  meets these two conditions, it accurately describes the new circuit. ■

*Theorem 12:* Given any switching circuit  $C$ , the Boolean polynomial  $p$  describing the circuit is equivalent to the probability  $P$  of the circuit being closed.

*Proof:* For any switch  $x$  in  $C$ , suppose that  $C$  and  $O$  express the probabilities that  $C$  is closed given that  $x$  is respectively closed or open. Then, by the laws of probability,

$$P = xC + (1 - x)O. \quad (5)$$

Note that when  $C$  is an open circuit,  $C = 0$  and  $O = 1$ . When  $C$  is a closed circuit,  $C = 1$  and  $O = 0$ . Note that  $C$  and  $O$  can be defined inductively by using these as base cases. By rearrangement,

$$P = x(C - O) + O. \quad (6)$$

From above, it was shown that for any switch  $x$  and polynomials  $c(x)$  and  $o(x)$  (for when the terminals connecting  $x$  are connected or not connected, respectively),

$$P = x(c(x) - o(x)) + o(x). \quad (7)$$

Note that when  $C$  is a closed circuit,  $c(x) = 1$  and  $c(o) = 0$ . When  $C$  is an open circuit,  $c(x) = 0$  and  $o(x) = 1$ .  $c$  and  $o$  can also be defined recursively.

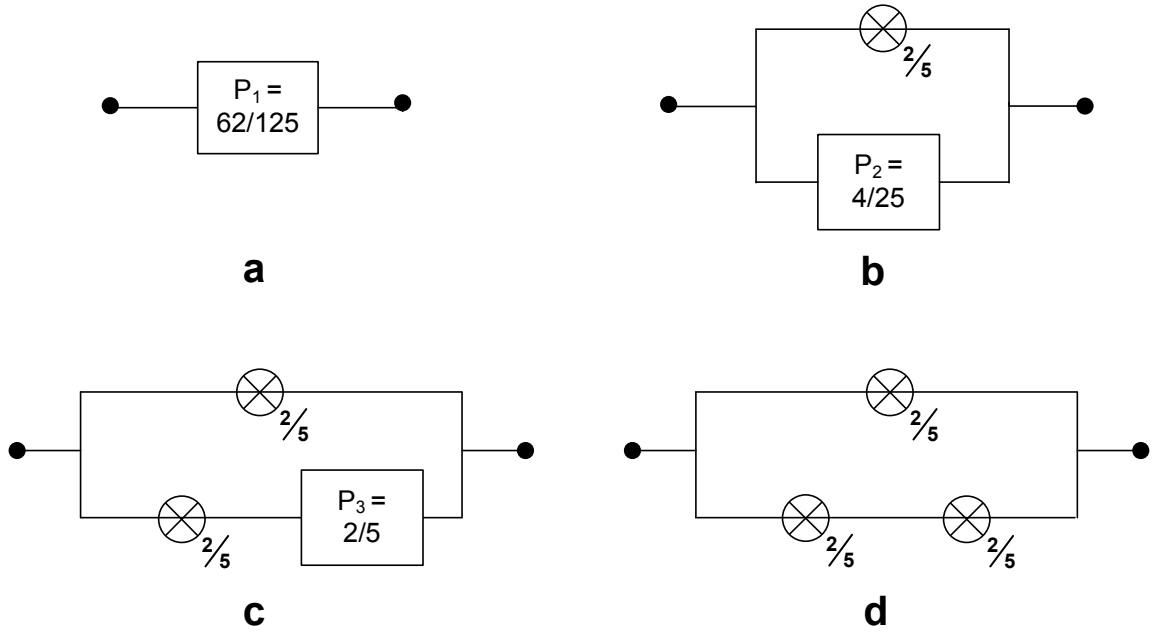


Fig. 11. **Realizing**  $P = \frac{62}{125}$ , **given**  $S = \{2/5\}$ . Each box represents the unknown circuit required to make the entire circuit closed with probability  $P$ . (a) This is an example of backwards construction, and we initially begin with an unknown circuit closed with probability  $P$ . (b) Initially,  $P_1 = 62/125 > 2/5$ , so the last-added pswitch is in parallel. To find the probability the remaining circuit must be closed by, we compute  $P_2 = \frac{P_1 - P}{1 - 2/5} = 4/25$ . (c) Now,  $P_2 = 4/25 < 2/5$ , so the second-to-last pswitch is in series. Hence, the probability the remaining circuit must be closed is  $P_3 = \frac{P_2 - P}{1 - 2/5} = 2/5$ . (d) Finally,  $P_3 = 2/5$ , so the remaining unknown circuit is a single pswitch.

Since all switching circuits  $C$  can be described recursively by the addition of a switch between any two terminals, all switching networks can be constructed by the above recursive formulas. Since both formulas have the same inductive and base conditions, the formulas for expressing the probability of closure and the resulting logic value are equivalent for all switching circuits. ■

## VI. A BACKWARD ALGORITHM.

Switching circuits can be synthesized using forward algorithms, where circuits are built first-pswitch first, or backward algorithms, where circuits are built last-pswitch first. For  $S = \{1/2\}$  above, a forward algorithm (the B-algorithm) was used to realize any binary fraction. For  $S = \{1/3, 2/3\}$ , a backward algorithm was used to realize any ternary fraction. Similarly to binary search, the backward algorithms presented in this paper recursively divide the rational space until the target probability is reached.

Given any single pswitch set  $S = \{p\}$  of a single rational probability, the following algorithm synthesizes an ssp circuit which realizes any rational  $F$ , but halts only if  $F$  can be realized:

- 1) Define circuit  $C_1$  such that  $Pr(C_1) = F$ .
- 2) While  $F \neq p$ :
  - a) Let circuit  $C_{i+1}$  be:
    - i) If  $F < p$ , a pswitch in series with  $C_i$  such that  $Pr(C_i) = F/p$ , or
    - ii) If  $F > p$ , a pswitch in parallel and  $C_i$  such that  $Pr(C_i) = \frac{F-p}{1-p}$ .
  - b) Let  $F = Pr(C^i)$ .
- 3) Define circuit  $C_i$  as two connected terminals.

See Figure 11 for an example of synthesizing  $11/27$  given  $S = \{1/3\}$ .

Now, we will show that at each stage it is only possible to insert the switch in series or parallel (but not both) to realize a given probability. Hence, if the probability can be realized, it must be realized using the specific circuit generated by the algorithm.

*Theorem 13:* Given a desired rational probability  $F$ , then to realize  $F$  using an ssp circuit, the last pswitch inserted must be in series if  $F < x$  and in parallel if  $F > x$ , given a pswitch set consisting of a single rational probability  $x$ .

*Proof:* We claim that when we add each pswitch, the decision whether to insert the pswitch in series or parallel is forced – in only one situation are both operations legal (when the desired probability is the same as a pswitch). For a circuit  $C$ , adding a pswitch in series yields  $C_s$  such that  $0 < Pr(C_s) \leq x$ , since  $Pr(C_s) = Pr(C)/2$  is increasing and  $0 < Pr(C) < 1$ . Similarly, adding a pswitch in parallel yields  $C_p$  such that  $x \leq Pr(C_p) < 1$ . Hence, the only way to realize some probability  $F < x$  is to place a pswitch in series. The only way to realize  $F > x$  is to place a pswitch in parallel. Then, the above algorithm is the only possible way to realize  $F$  with an ssp circuit. ■

## VII. CONCLUSIONS

In this paper, we introduced probabilistic switches as an extension to classical deterministic switches. We showed that a random variable, particularly a Bernoulli random variable, can be associated with each pswitch, allowing circuits to realize probability distributions. We found an algorithm to generate the optimal circuit for any binary or trinary fraction. We showed how duality extends from resistor networks and logic gates to the pswitch realm, using it to prove existence results. Finally, we constructed a linear-size universal probability generator, capable of mapping any set of deterministic inputs to unique probabilistic outputs.

Extending the work to find algorithms for more general random variables and multiple terminals would be useful for applied work, particularly in modeling stochastic events in biology and engineering.

## VIII. ACKNOWLEDGEMENT

Daniel Wilhelm was supported by the U.S. Department of Homeland Security (DHS) Scholarship and Fellowship Program, administered by the Oak Ridge Institute for Science and Education (ORISE).

## REFERENCES

- [Boo47] G. Boole. The calculus of logic. *Cambridge and Dublin Mathematical Journal*, 3:183–198, 1847.
- [Col87] C.J. Colbourn. The combinatorics of network reliability. *Oxford University Press*, 1987.
- [Duf65] R.J. Duffin. Topology of series-parallel networks. *J. Math. Anal. Appl.*, 10:303–318, 1965.
- [Erd34] P. Erdős. A theorem of sylvestre and schur. *J. London Math. Soc.*, 9:282–288, 1934.
- [Gil63] A. Gill. On a weight distribution problem, with applications to the design of stochastic generators. *J. of the ACM*, 10:110–121, 1963.
- [HTP06] O. Heling-Tveretina and G. Provan. Approximate compilation with weighted bdds. In *Constraint Solving and Constraint Programming*, 2006.
- [Lec71] R.J. Lechner. Harmonic analysis of switching functions. In A. Mukhopadhyay, editor, *Recent Developments in Switching Theory*. Academic Press, New York, 1971.
- [Mac92] P.A. MacMahon. The combinations of resistances. *The Electrician*, 28:601–602, 1892. (Reprinted in: *Discr. Appl. Math.*, 54:225–228, 1994.).
- [Mei98] C. Meinel. *Algorithms and Data Structures in VLSI Design: OBDD - Foundations and Applications*. Springer, New York, 1998.
- [Ram19] S. Ramanujan. A proof of bertrand’s postulate. *J. Indian Math. Soc.*, 11:181–182, 1919.
- [RS42] J. Riordan and C.E. Shannon. The number of two-terminal series-parallel networks. *J. Math. Physics*, 21:83–93, 1942.
- [Sha38] C.E. Shannon. A symbolic analysis of relay and switching circuits. *Trans. AIEE*, 57:713–723, 1938.
- [Tel40] B.H. Tellegen. Geometrical configurations and duality of electrical networks. *Philips Technical Review*, 5:324–330, 1940.
- [Whi61] J. E. Whitesitt. *Boolean Algebra and Its Applications*. Addison-Wesley, Massachusetts, 1961.