Bayesuvius,
a small visual dictionary of Bayesian Networks

Robert R. Tucci
www.ar-tiste.xyz

August 15, 2020

Figure 1: View of Mount Vesuvius from Pompeii

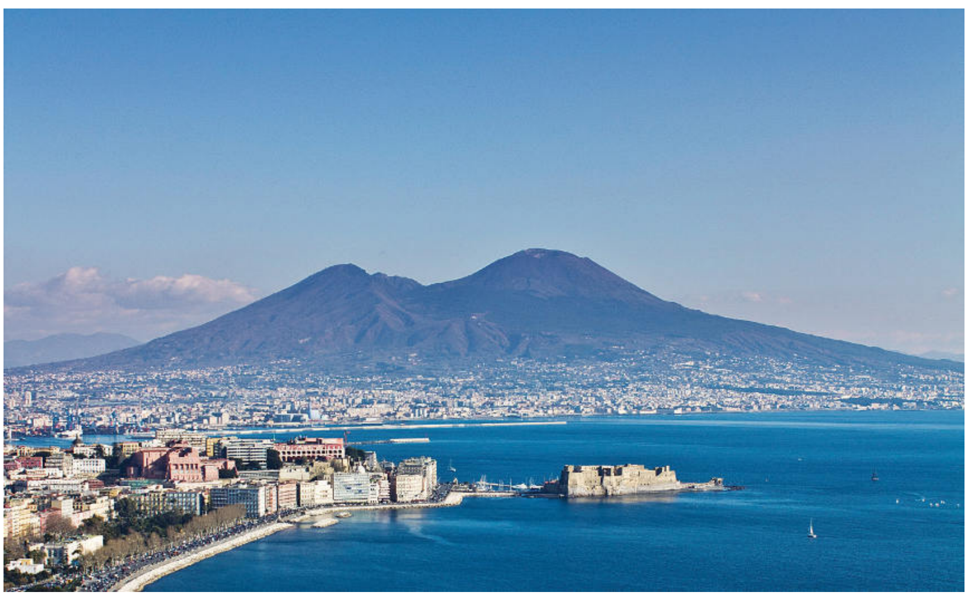


Figure 2: Mount Vesuvius and Bay of Naples

Contents

0.2 Notational Conventions

bnet=B net=Bayesian Network

Define $\mathbb{Z}, \mathbb{R}, \mathbb{C}$ to be the integers, real numbers and complex numbers, respectively.

For $a < b$, define $\mathbb{Z}_I$ to be the integers in the interval I, where $I = [a,b], [a,b), (a,b], (a,b)$ (i.e, I can be closed or open on either side).

$A_{>0} = \{k \in A : k > 0\}$ for $A = \mathbb{Z}, \mathbb{R}$.

Random Variables will be indicated by underlined letters and their values by non-underlined letters. Each node of a bnet will be labelled by a random variable. Thus, $\underline{x} = x$ means that node $\underline{x}$ is in state x.

$P_{\underline{x}}(x) = P(\underline{x} = x) = P(x)$ is the probability that random variable $\underline{x}$ equals $x \in S_{\underline{x}}$. $S_{\underline{x}}$ is the set of states (i.e., values) that $\underline{x}$ can assume and $n_{\underline{x}} = |S_{\underline{x}}|$ is the size (aka cardinality) of that set. Hence,

$$\sum_{x \in S_{\underline{x}}} P_{\underline{x}}(x) = 1 \tag{1}$$

$$P_{\underline{x},\underline{y}}(x,y) = P(\underline{x} = x, \underline{y} = y) = P(x,y) \tag{2}$$

$$P_{\underline{x}|\underline{y}}(x|y) = P(\underline{x} = x | \underline{y} = y) = P(x|y) = \frac{P(x,y)}{P(y)} \tag{3}$$

Kronecker delta function: For x, y in discrete set S,

$$\delta(x,y) = \begin{cases} 1 \text{ if } x = y \\ 0 \text{ if } x \neq y \end{cases} \tag{4}$$

Dirac delta function: For $x, y \in \mathbb{R}$,

$$\int_{-\infty}^{+\infty} dx\ \delta(x-y) f(x) = f(y) \tag{5}$$

Transition probability matrix of a node of a bnet can be either a discrete or a continuous probability distribution. To go from continuous to discrete, one replaces integrals over states of node by sums over new states, and Dirac delta functions by Kronecker delta functions. More precisely, consider a function $f : S \to \mathbb{R}$. Let $S_{\underline{x}} \subset S$ and $S \to S_{\underline{x}}$ upon discretization (binning). Then

$$\int_S dx\ P_{\underline{x}}(x) f(x) \to \frac{1}{n_{\underline{x}}} \sum_{x \in S_{\underline{x}}} f(x)\ . \tag{6}$$

Both sides of last equation are 1 when $f(x) = 1$. Furthermore, if $y \in S_{\underline{x}}$, then

$$\int_S dx\ \delta(x-y) f(x) = f(y) \to \sum_{x \in S_{\underline{x}}} \delta(x,y) f(x) = f(y)\ . \tag{7}$$

Indicator function (aka Truth function):

$$\mathbb{1}(\mathcal{S}) = \begin{cases} 1 \text{ if } \mathcal{S} \text{ is true} \\ 0 \text{ if } \mathcal{S} \text{ is false} \end{cases} \tag{8}$$

For example, $\delta(x,y) = \mathbb{1}(x=y)$.

$$\vec{x} = (x[0], x[1], x[2] \ldots, x[nsam(\vec{x})-1]) = x[:] \tag{9}$$

$nsam(\vec{x})$ is the number of samples of $\vec{x}$. $\underline{x}[i]$ are i.d.d. (independent identically distributed) samples with

$$x[i] \sim P_{\underline{x}} \;\; \text{(i.e. } P_{\underline{x[i]}} = P_{\underline{x}}) \tag{10}$$

$$P(\underline{x} = x) = \frac{1}{nsam(\vec{x})} \sum_i \mathbb{1}(x[i] = x) \tag{11}$$

If we use two sampled variables, say $\vec{x}$ and $\vec{y}$, in a given bnet, their number of samples $nsam(\vec{x})$ and $nsam(\vec{y})$ need not be equal.

$$P(\vec{x}) = \prod_i P(x[i]) \tag{12}$$

$$\sum_{\vec{x}} = \prod_i \sum_{x[i]} \tag{13}$$

$$\partial_{\vec{x}} = [\partial_{x[0]}, \partial_{x[1]}, \partial_{x[2]}, \ldots, \partial_{x[nsam(\vec{x})-1]}] \tag{14}$$

$$\begin{aligned} P(\vec{x}) \;\; &\approx \;\; [\prod_x P(x)^{P(x)}]^{nsam(\vec{x})} && (15) \\ &= \;\; e^{nsam(\vec{x}) \sum_x P(x) \ln P(x)} && (16) \\ &= \;\; e^{-nsam(\vec{x}) H(P_{\underline{x}})} && (17) \end{aligned}$$

$$f^{[1,\partial_x,\partial_y]}(x,y) = [f, \partial_x f, \partial_y f] \tag{18}$$

$$f^+ = f^{[1,\partial_x,\partial_y]} \tag{19}$$

For probabilty distributions $p(x), q(x)$ of $x \in S_{\underline{x}}$

- Entropy:

$$H(p) = -\sum_x p(x) \ln p(x) \geq 0 \tag{20}$$

- Kullback-Liebler divergence:

$$D_{KL}(p \parallel q) = \sum_x p(x) \ln \frac{p(x)}{q(x)} \geq 0 \tag{21}$$

- Cross entropy:

$$\begin{aligned} CE(p \to q) \;\; &= \;\; -\sum_x p(x) \ln q(x) && (22) \\ &= \;\; H(p) + D_{KL}(p \parallel q) && (23) \end{aligned}$$

Normal Distribution: $x, \mu, \sigma \in \mathbb{R}$, $\sigma > 0$

$$\mathcal{N}(\mu, \sigma^2)(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{24}$$

Uniform Distribution: $a < b$, $x \in [a, b]$

$$\mathcal{U}(a, b)(x) = \frac{1}{b-a} \tag{25}$$

Expected Value

Given a random variable $\underline{x}$ with states $S_{\underline{x}}$ and a function $f : S_{\underline{x}} \to \mathbb{R}$, define

$$E_{\underline{x}}[f(\underline{x})] = E_{x\sim P(x)}[f(x)] = \sum_x P(x) f(x) \tag{26}$$

Conditional Expected Value

Given a random variable $\underline{x}$ with states $S_{\underline{x}}$, a random variable $\underline{y}$ with states $S_{\underline{y}}$, and a function $f : S_{\underline{x}} \times S_{\underline{y}} \to \mathbb{R}$, define

$$E_{\underline{x}|\underline{y}}[f(\underline{x}, \underline{y})] = \sum_x P(x|\underline{y}) f(x, \underline{y}) \ , \tag{27}$$

$$E_{\underline{x}|\underline{y}=y}[f(\underline{x}, y)] = E_{\underline{x}|y}[f(\underline{x}, y)] = \sum_x P(x|y) f(x, y) \ . \tag{28}$$

Note that

$$\begin{aligned} E_{\underline{y}}[E_{\underline{x}|\underline{y}}[f(\underline{x}, \underline{y})]] &= \sum_{x,y} P(x|y)P(y)f(x,y) & (29) \\ &= \sum_{x,y} P(x,y)f(x,y) & (30) \\ &= E_{\underline{x},\underline{y}}[f(\underline{x}, \underline{y})] \ . & (31) \end{aligned}$$

Sigmoid function: For $x \in \mathbb{R}$,

$$\text{sig}(x) = \frac{1}{1+e^{-x}} \tag{32}$$

$\mathcal{N}(!a)$ will denote a normalization constant that does not depend on a. For example, $P(x) = \mathcal{N}(!x)e^{-x}$ where $\int_0^\infty dx\, P(x) = 1$.

A **one hot** vector of zeros and ones is a vector with all entries zero with the exception of a single entry which is one. A **one cold** vector has all entries equal to one with the exception of a single entry which is zero. For example, if $x^n = (x_0, x_1, \ldots, x_{n-1})$ and $x_i = \delta(i, 0)$ then x^n is one hot.

Short Summary of Boolean Algebra.

See Ref.[1] for more info about this topic.

Suppose $x, y, z \in \{0, 1\}$. Define

$$x \text{ or } y = x \vee y = x + y - xy \ , \tag{33}$$

$$x \text{ and } y = x \wedge y = xy \ , \tag{34}$$

and

$$\text{not } x = \overline{x} = 1 - x \ , \tag{35}$$

where we are using normal addition and multiplication on the right hand sides.

Associativity	$x \vee (y \vee z) = (x \vee y) \vee z$ $x \wedge (y \wedge z) = (x \wedge y) \wedge z$
Commutativity	$x \vee y = y \vee x$ $x \wedge y = y \wedge x$
Distributivity	$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$
Identity	$x \vee 0 = x$ $x \wedge 1 = x$
Annihilator	$x \wedge 0 = 0$ $x \vee 1 = 1$
Idempotence	$x \vee x = x$ $x \wedge x = x$
Absorption	$x \wedge (x \vee y) = x$ $x \vee (x \wedge y) = x$
Complementation	$x \wedge \overline{x} = 0$ $x \vee \overline{x} = 1$
Double negation	$\overline{(\overline{x})} = x$
De Morgan Laws	$\overline{x} \wedge \overline{y} = \overline{(x \vee y)}$ $\overline{x} \vee \overline{y} = \overline{(x \wedge y)}$

Table 1: Boolean Algebra Identities

Actually, since $x \wedge y = xy$, we can omit writing the symbol $\wedge$. The symbol $\wedge$ is useful to exhibit the symmetry of the identities, and to remark about the analogous identities for sets, where $\wedge$ becomes intersection $\cap$ and $\vee$ becomes union $\cup$. However, for practical calculations, $\wedge$ is an unnecessary nuisance.

Since $x \in \{0, 1\}$,

$$P(\overline{x}) = 1 - P(x) \ . \tag{36}$$

Clearly, from analyzing the simple event space $(x, y) \in \{0, 1\}^2$,

$$P(x \vee y) = P(x) + P(y) - P(x \wedge y) \ . \tag{37}$$

Chapter 27

Reliability Box Diagrams and Fault Tree Diagrams

This chapter is based on Refs.[2] and [3].

In this chapter, we assume that reader is familiar with Boolean Algebra. See the Notational Conventions Chapter 0.2 for a quick review of what we recommend that you know about Boolean Algebra to fully appreciate this chapter.

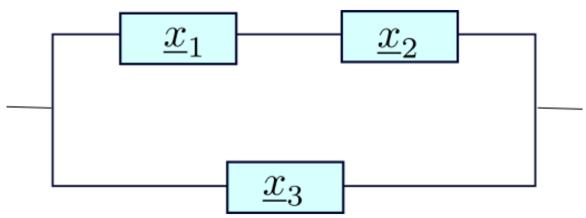


Figure 27.1: Example of rbox diagram.

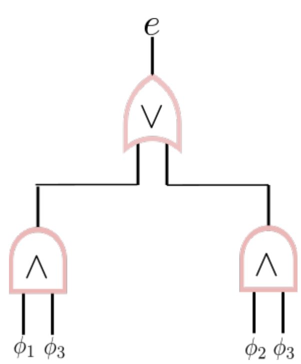


Figure 27.2: An ftree diagram equivalent to Fig.27.1. It represents $e = (\phi_1 \wedge \phi_3) \vee (\phi_2 \wedge \phi_3)$.

Complicated devices with a large number of components such as airplanes or rockets can fail in many ways. If their performance depends on some components working in series and one of the

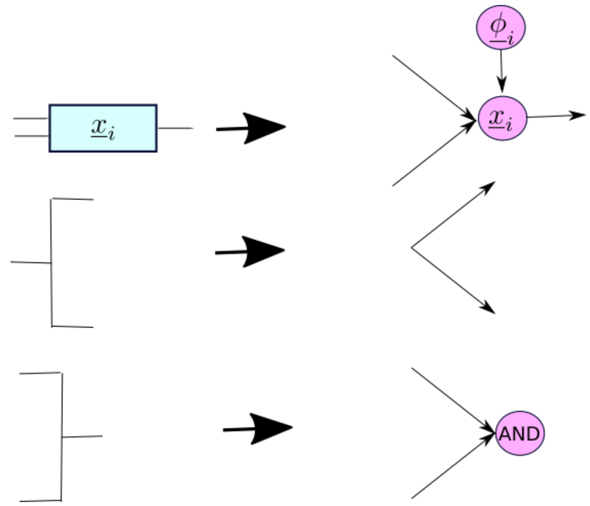


Figure 27.3: How to map an rbox diagram to a bnet.

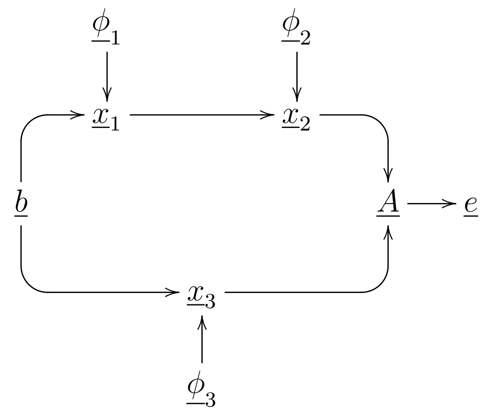


Figure 27.4: bnet corresponding the rbox diagram Fig.27.1.

components in the series fails, this may lead to catastrophic failure. To avert such disasters, engineers use equivalent components connected in parallel instead of in series, thus providing multiple backup systems.They analyze the device to find its weak points and add backup capabilities there. They also estimate the average time to failure for the device.

The two most popular diagrams for finding the failure modes and their rates for large complicated devices are

- rbox diagrams = Reliability Box diagrams. See Fig.27.1 for an example.
- ftree diagrams = Fault Tree Diagrams. See Fig.27.2 for an example.

In an ftree diagram, several nodes might stand for the same component of a physical device. In an rbox diagram, on the other hand, each node represents a distinct component in a device. Hence, rbox diagrams resemble the device they are addressing whereas ftree diagrams don't. Henceforth, we will refer to this desirable property as **physical resemblance**.

As we will show below with an example, it is pretty straightforward to translate an rbox to an ftree diagram. Going the other way, translating an ftree to an rbox diagram is much more difficult.

Next we will define a new kind of bnet that we will call a failure bnet that has physical resemblance. Then we will describe a simple method of translating (i.e., mapping) any rbox diagram to a failure bnet. Then we will show how a failure bnet can be used to do all the calculations that are normally done with an rbox or an ftree diagram. In that sense, failure bnets seem to afford all the benefits of both ftree and rbox diagrams.

A **failure bnet** contains nodes of 5 types, labeled $\underline{b}$, $\underline{e}$, $\underline{x}_i$, $\underline{\phi}_i$, and $\underline{A}_i$. All nodes have only two possible states $S = Success = 0$, $F = Failure = 1$.

1. The bnet has a beginning node labeled $\underline{b}$ which is always set to success. The $\underline{b}$ node and the $\underline{\phi}_i$ nodes are the only root nodes of the bnet.

2. The bnet has a single leaf node, the end node, labeled $\underline{e}$. $\underline{e}$ is fixed. In rbox diagrams, $\underline{e} = S$ whereas in ftree diagrams, $\underline{e} = F$.

3. $\underline{x}^{nx} = (\underline{x}_0, \underline{x}_1, \ldots, \underline{x}_{nx-1})$. $\underline{x}_i \in \{S, F\}$ for all i.

   Suppose $\underline{x}_i$ has parents $\underline{\phi}_i$ and $\underline{a}^{na} = (\underline{a}_0, \underline{a}_1, \ldots \underline{a}_{na-1})$. Then the transition prob matrix of node $\underline{x}_i$ is defined to be

$$P(x_i|\phi_i, a^{na}) = \delta(x_i, \phi_i \vee \vee_{i=0}^{na-1} a_i) \tag{27.1}$$

4. For each node $\underline{x}_i$, the bnet has a "performance" root node $\underline{\phi}_i \in \{0, 1\}$ with an arrow pointing from it to $\underline{x}_i$ (i.e, $\underline{\phi}_i \rightarrow \underline{x}_i$). For all i,

$$P(\phi_i) = \epsilon_i \delta(\phi_i, F) + \overline{\epsilon}_i \delta(\phi_i, S) \; . \tag{27.2}$$

   ϵ_i is the failure probability and $\overline{\epsilon}_i = 1 - \epsilon_i$ is the success probability. We name the failure probability ϵ_i because it is normally very small. It is usually set to $1 - e^{-\lambda_i t} \approx \lambda_i t$ when $\lambda_i t << 1$, where λ_i is the failure rate for node $\underline{x}_i$ and t stands for time. The rblock literature usually calls $\overline{\epsilon_i} = R_i$ the **reliability** of node $\underline{x}_i$, and $\epsilon_i = (1 - R_i) = F_i$ its **unreliability**.

5. The nodes $\underline{A}_i \in \{0, 1\}$ are simply AND gates. If $\underline{A}_i$ has inputs $\underline{y}^{ny} = (\underline{y}_0, \underline{y}_1, \ldots, \underline{y}_{ny-1})$, then the transition prob matrix of $\underline{A}_i$ is

$$P(A_i|y^{ny}) = \delta(A_i, \wedge_{i=0}^{ny-1} y_i) \; . \tag{27.3}$$

An instance (instantiation) of a bnet is the bnet with all nodes set to a specific state. A **realizable instance (r-instance)** of a bnet is one which has non-zero probability.

Fig.27.3 shows how to translate any rbox diagram to a failure bnet. To illustrate this procedure, we translated the rbox diagram Fig.27.1 into the failure bnet Fig.27.4.

For the failure bnet Fig.27.4, one has:

$$\begin{array}{l} P(b) = \mathbb{1}(b = 0) \\ P(x_1|\phi_1, b) = \mathbb{1}(x_1 = \phi_1 \vee b) \\ P(x_2|\phi_2, x_1) = \mathbb{1}(x_2 = \phi_2 \vee x_1) \\ P(x_3|\phi_3, b) = \mathbb{1}(x_3 = \phi_3 \vee b) \\ P(A|x_2, x_3)e = \mathbb{1}(x_2 \wedge x_3) \\ P(e|A) = \mathbb{1}(e = A) \end{array} . \tag{27.4}$$

Therefore, all r-instances of this bnet must satisfy

$$e = (\phi_1 \vee \phi_2) \wedge \phi_3 \tag{27.5}$$

$$= (\phi_1 \wedge \phi_3) \vee (\phi_2 \wedge \phi_3) . \tag{27.6}$$

Eq.(27.6) proves that Fig.27.2 is indeed a representation of Fig.27.1.

Next, we consider r-instances of this bnet for two cases: $e = S$ and $e = F$.

- **rblock analysis:** $e = S = 0$**.**
  Table 27.1 shows the probability of all possible r-instances that end in success for the failure bnet Fig.27.4. (These r-instances are the main focus of rblock analysis). The first 4 of those probabilities (those with $\phi_3 = 0$) sum to $\bar{\epsilon}_3$ so the sum $P(e = S)$ of all 5 is

  $$P(e = S) = \bar{\epsilon}_3 + \bar{\epsilon}_1\bar{\epsilon}_2\epsilon_3 , \tag{27.7}$$

  or, expressing it in reliability language in which $\bar{\epsilon} = R$,

  $$P(e = S) = R_3 + R_1R_2\overline{R}_3 . \tag{27.8}$$

- **ftree analysis:** $e = F = 1$**.**
  Table 27.2 shows the probability of all possible r-instances that end in failure for the failure bnet Fig.27.4. (These r-instances are the main focus of ftree analysis). If we set $\epsilon_i = \epsilon$ and $\bar{\epsilon}_i \approx 1$ for $i = 1, 2, 3$, then the first two of those r-instances have probabilities of $order(\epsilon^2)$ and the third has probability of $order(\epsilon^3)$. The two lowest order ($order(\epsilon^2)$) r-instances are called the “minimal cut sets” of the ftree. We will have more to say about minimal cut sets later on. For now, just note from Eq.(27.6) that the ftree Fig.27.2 is just the result of joining together with ORs two expressions, one for each of the two minimal cut sets.

instance	probability
$\begin{tikzcd} & 0 \arrow[d] & & 1 \arrow[d] & & \\ & \underline{x}_1 \arrow[rr] & & \underline{x}_2 \arrow[dr] & & \\ 0 \arrow[ur] \arrow[drr] & & & & \underline{A} \arrow[r] & 0 \\ & & \underline{x}_3 \arrow[urr] & & & \\ & & 0 \arrow[u] & & & \end{tikzcd}$	$\bar{\epsilon}_1 \epsilon_2 \bar{\epsilon}_3$
$\begin{tikzcd} & 1 \arrow[d] & & 0 \arrow[d] & & \\ & \underline{x}_1 \arrow[rr] & & \underline{x}_2 \arrow[dr] & & \\ 0 \arrow[ur] \arrow[drr] & & & & \underline{A} \arrow[r] & 0 \\ & & \underline{x}_3 \arrow[urr] & & & \\ & & 0 \arrow[u] & & & \end{tikzcd}$	$\epsilon_1 \bar{\epsilon}_2 \bar{\epsilon}_3$
$\begin{tikzcd} & 1 \arrow[d] & & 1 \arrow[d] & & \\ & \underline{x}_1 \arrow[rr] & & \underline{x}_2 \arrow[dr] & & \\ 0 \arrow[ur] \arrow[drr] & & & & \underline{A} \arrow[r] & 0 \\ & & \underline{x}_3 \arrow[urr] & & & \\ & & 0 \arrow[u] & & & \end{tikzcd}$	$\epsilon_1 \epsilon_2 \bar{\epsilon}_3$
$\begin{tikzcd} & 0 \arrow[d] & & 0 \arrow[d] & & \\ & \underline{x}_1 \arrow[rr] & & \underline{x}_2 \arrow[dr] & & \\ 0 \arrow[ur] \arrow[drr] & & & & \underline{A} \arrow[r] & 0 \\ & & \underline{x}_3 \arrow[urr] & & & \\ & & 0 \arrow[u] & & & \end{tikzcd}$	$\bar{\epsilon}_1 \bar{\epsilon}_2 \bar{\epsilon}_3$
$\begin{tikzcd} & 0 \arrow[d] & & 0 \arrow[d] & & \\ & \underline{x}_1 \arrow[rr] & & \underline{x}_2 \arrow[dr] & & \\ 0 \arrow[ur] \arrow[drr] & & & & \underline{A} \arrow[r] & 0 \\ & & \underline{x}_3 \arrow[urr] & & & \\ & & 1 \arrow[u] & & & \end{tikzcd}$	$\bar{\epsilon}_1 \bar{\epsilon}_2 \epsilon_3$

Table 27.1: Probabilities of all possible r-instances with $e = S = 0$ for failure bnet Fig.27.4.

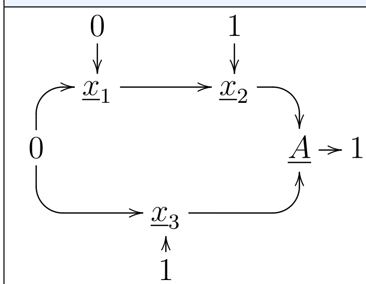
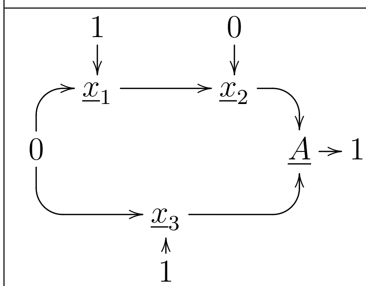
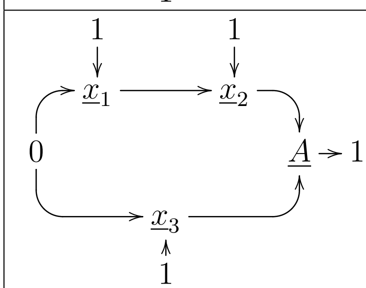
instance	probability
	$\overline{\epsilon}_1 \epsilon_2 \epsilon_3$
	$\epsilon_1 \overline{\epsilon}_2 \epsilon_3$
	$\epsilon_1 \epsilon_2 \epsilon_3$

Table 27.2: Probabilities of all possible r-instances with $e = F = 1$ for the failure bnet Fig.27.4.

More general $\underline{x}_i$.
Failure bnets can actually accommodate $\underline{x}_i$ nodes of a more general kind than what we first stipulated. Here are some possibilities:

For any $a^n \in \{0,1\}^n$, let

$$len(a^n) = \sum_i a_i \tag{27.9}$$

- **OR gate**

$$\begin{aligned} P(x_i|\phi_i, a^{na}) &= \delta(x_i, \phi_i \vee \vee_j a_j) &(27.10)\\ &= \delta(x_i, \phi_i \vee \mathbb{1}(len(a^{na}) > 0)) &(27.11)\end{aligned}$$

- **AND gate**

$$\begin{aligned} P(x_i|\phi_i, a^{na}) &= \delta(x_i, \phi_i \vee \wedge_j a_j) &(27.12)\\ &= \delta(x_i, \phi_i \vee \mathbb{1}(len(a^{na}) = na)) &(27.13)\end{aligned}$$

- **Fail if least K failures (less than K successes)**

$$P(x_i|\phi_i, a^{na}) = \delta(x_i, \phi_i \vee \mathbb{1}(len(a^{na}) \geq K)) \tag{27.14}$$

- **Fail if less than K failures (at least K successes)**

$$P(x_i|\phi_i, a^{na}) = \delta(x_i, \phi_i \vee \mathbb{1}(len(a^{na}) < K)) \tag{27.15}$$

- **Fail if exactly one failure**

$$P(x_i|\phi_i, a^{na}) = \delta(x_i, \phi_i \vee \mathbb{1}(len(a^{na}) = 1)) \tag{27.16}$$

  This equals an XOR (exclusive OR) gate when $na = 2$.

- **General gate**
  $f : \{0,1\}^{na} \to \{0,1\}$

$$P(x_i|\phi_i, a^{na}) = \delta(x_i, \phi_i \vee f(a^{na})) \tag{27.17}$$

Minimal Cut Sets

Suppose $x \in \{0,1\}$ and $f : \{0,1\} \to \{0,1\}$. Then by direct evaluation, we see that

$$f(x) = [\overline{x} f(0)] \vee [x f(1)] \ . \tag{27.18}$$

Let

$$\begin{aligned} !x &= 1 - x, \\ !^0 x &= x, \\ !^1 x &= !x \end{aligned} \tag{27.19}$$

Then Eq.27.18 can be rewritten as

$$f(x) = \vee_{a\in\{0,1\}}[(!^{\overline{a}} x) f(a)] \ . \tag{27.20}$$

Now suppose $x^n \in \{0,1\}^n$ and $f : \{0,1\}^n \to \{0,1\}$. Eq.(27.20) generalizes to

$$f(x^n) = \vee_{a^n\in\{0,1\}^n}[\prod_i (!^{\overline{a}_i} x_i) f(a^n)] \ . \tag{27.21}$$

Eq.(27.21) is called an **ors-of-ands** normal form expansion. There is also an **ands-of-ors** normal form expansion obtained by swapping multiplication and $\vee$ in Eq.(27.21), but we won't need it here.

A **minimal cut set** is the smallest set of ϕ_i's such that if they are all F, then $e = F$ for all the r-instances. From the failure bnet, we can always find a function $f : \{0,1\}^{nx} \to \{0,1\}$ such that $e = f(\phi^{nx})$ for all the r-instances. We did that for our example failure bnet and obtained Eq.(27.6). We can then express $f(\phi^{nx})$ as an ors-of-ands expansion to find the set of minimal cut sets. The ands terms in that ors-of-ands expansion each gives a different minimal cut set, after some simplification. The ors-of-ands expression is not unique and it may have redundancies that need to be simplified (using the Boolean Algebra identities given in Chapter 0.2) to achieve minimality in the cut sets.

Bibliography

[1] Wikipedia. Boolean algebra. https://en.wikipedia.org/wiki/Boolean_algebra.

[2] ReliaSoft. System analysis reference. http://reliawiki.org/index.php/System_Analysis_Reference.

[3] W. E. Vesely, F. F. Goldberg, N. H. Roberts, and D. F. Haasl. Fault tree handbook nureg-0492. https://www.nrc.gov/reading-rm/doc-collections/nuregs/staff/sr0492/.

[4] Robert R. Tucci. Bell's inequalities for Bayesian statisticians. blog post in blog Quantum Bayesian Networks, https://qbnets.wordpress.com/2008/09/19/bells-inequaties-for-bayesian-statistician/.

[5] Wikipedia. Binary decision diagram. https://en.wikipedia.org/wiki/Binary_decision_diagram.

[6] Wikipedia. Expectation maximization. https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm.

[7] Wikipedia. k-means clustering. https://en.wikipedia.org/wiki/K-means_clustering.

[8] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, David Warde-Farley Bing Xu, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. https://arxiv.org/abs/1406.2661.

[9] Wikipedia. Hidden Markov model. https://en.wikipedia.org/wiki/Hidden_Markov_model.

[10] Gregory Nuel. Tutorial on exact belief propagation in Bayesian networks: from messages to algorithms. https://arxiv.org/abs/1201.4724.

[11] Wikipedia. Kalman filter. https://en.wikipedia.org/wiki/Kalman_filter.

[12] Wikipedia. Markov blanket. https://en.wikipedia.org/wiki/Markov_blanket.

[13] Judea Pearl. *Probabilistic Inference in Intelligent Systems*. Morgan Kaufmann, 1988.

[14] Nitish Srivastava, G E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. http://www.cs.toronto.edu/~rsalakhu/papers/srivastava14a.pdf.

[15] Wikipedia. Non-negative matrix factorization. https://en.wikipedia.org/wiki/Non-negative_matrix_factorization.

[16] theinvestorsbook.com. Pert analysis. https://theinvestorsbook.com/pert-analysis.html.

[17] Wikipedia. Program evaluation and review technique. https://en.wikipedia.org/wiki/Program_evaluation_and_review_technique.

[18] Andrew Ng. Lecture at deeplearning.ai on recurrent neural networks. http://www.ar-tiste.com/ng-lec-rnn.pdf.

[19] Wikipedia. Long short term memory. https://en.wikipedia.org/wiki/Long_short-term_memory.

[20] Wikipedia. Gated recurrent unit. https://en.wikipedia.org/wiki/Gated_recurrent_unit.

[21] Charles Fox, Neil Girdhar, and Kevin Gurney. A causal bayesian network view of reinforcement learning. https://www.aaai.org/Papers/FLAIRS/2008/FLAIRS08-030.pdf".

[22] Sergey Levine. Course CS 285 at UC Berkeley, Deep reinforcement learning. http://rail.eecs.berkeley.edu/deeprlcourse/.

[23] Robert R. Tucci. Simpson's paradox, the bane of clinical trials. blog post in blog Quantum Bayesian Networks https://qbnets.wordpress.com/2020/07/09/simpsons-paradox-the-bane-of-clinical-trials/.

[24] Robert J. McEliece, David J. C. MacKay, and Jung-Fu Cheng. Turbo decoding as an instance of Pearls belief propagation algorithm. http://authors.library.caltech.edu/6938/1/MCEieeejstc98.pdf.