Diagramme de Fiabilité (Reliability Bloc Diagram (RBD))

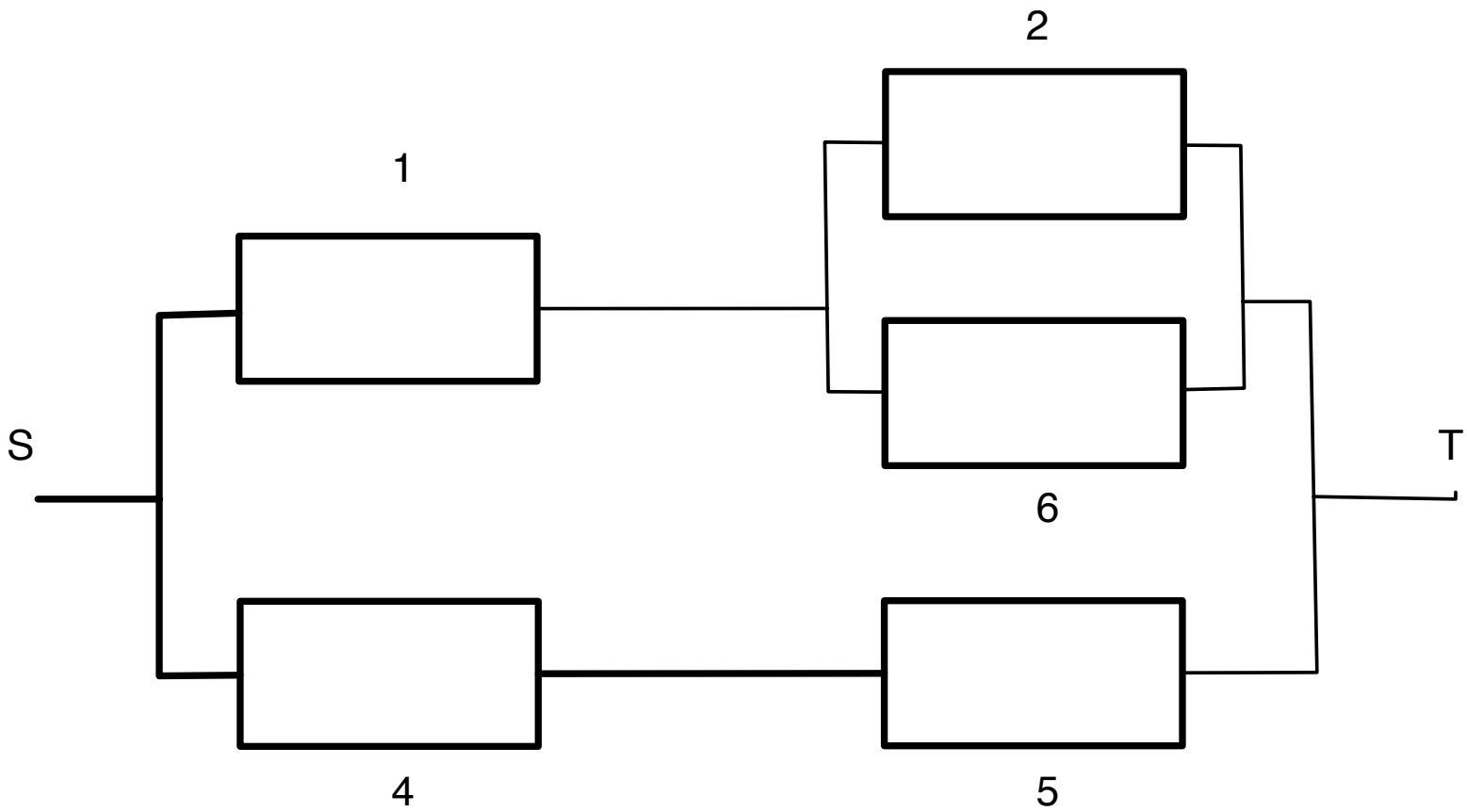
- Une représentation graphique du système et de la fiabilité.
- Chaque composant est représenté par un bloc.
- Sert à déterminer si le système est UP ou DOWN en fonction des états des composants.
- Idée intuitive : un bloc peut être vu comme un switch qui est fermé quand le composant est UP et ouvert quand le composant est DOWN.
- Il y a une entrée dans le diagramme S et une sortie T.

RBD

- C'est un modèle reposant sur la logique et non pas sur les états.
- Modèle Statique : pas de représentation du temps ni de l'ordre entre des événements successifs.
- Hypothèse d'Indépendence des pannes des différents composants.
- Pas de pannes arrivant conjointement ou de pannes provoquées par la panne d'un autre composant.

RBD

- Le système est UP si il y a au moins un chemin passant par des éléments UP et reliant S à T.



Logique

- Le comportement du système par rapport à la panne est modélisé par les connexions entre blocs.
- Si tous les composants sont nécéssaires, les modéliser en série
- Si un seul des composants est nécéssaire, les modéliser en parallèle.
- Si il en faut au moins K parmi N, utiliser la structure "K out of N"

Blocs en Série

- n composants indépendants en série.
- E_i le composant i fonctionne.
- $R_s = P(E_1 \cap E_2 \cap \ldots \cap E_n)$
- A cause de l'indépendance :

$$P(E_1 \cap E_2 \cap \ldots \cap E_n) = \prod_{i=1}^{n} P(E_i)$$

- En notant $R_i = P(E_i)$, on obtient :

$$R_s = \prod_{i=1}^{n} R_i$$

- On remarque que $R_s < min(R_i)$. Le système est moins fiable que sa composante la moins fiable.

Blocs en Parallèle

- n composants indépendants en prallèle.
- E_i le composant i fonctionne.
- $R_p = P(E_1 \cup E_2 \cup \ldots \cup E_n)$
- Le système est en panne si tous les composants sont en panne :

$$1 - R_p = \prod_{i=1}^{n} (1 - R_i)$$

Systèmes Série-Parallèles

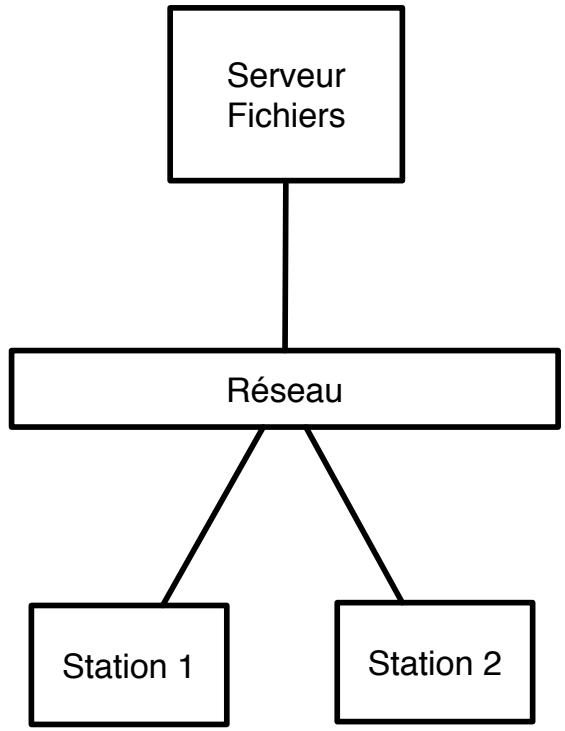
- Décomposition récursive : un système série-parallèle (SP) est soit :
  - un bloc isolé
  - plusieurs sous-systèmes SP en série
  - plusieurs sous-systèmes SP en parallèle
- Utilise la décomposition récursive de la construction pour obtenir la fiabilité.
- Exemple simple : n étages en série, chaque étage composé de m composants en parallèle tous identiques :

$$R_{sp} = (1 - (1 - R)^m)^n$$

Exemple : Station de Travail/Serveur de Fichiers

- Un serveur de fichiers,
- Deux stations de Travail identiques
- Un réseau pour les connecter. On suppose que le réseau est fiable.
- Le système est opérationnel si le serveur de fichiers est opérationnel et au moins une des deux stations de travail est opérationnelle.

Représentation de l'exemple



Etude de la fiabilité de l'exemple

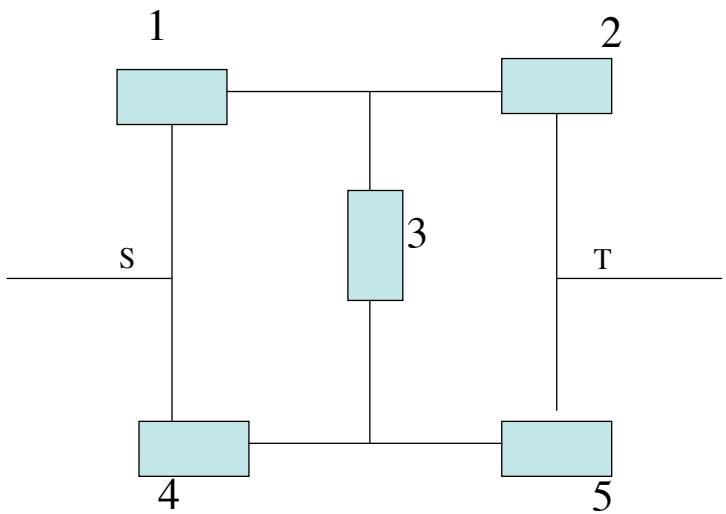
- R_w fiabilité d'une station de travail
- R_f fiabilité du serveur de fichiers
- R_{fsw} fiabilité du système

$$R_{fsw} = (1 - (1 - R_w)^2)R_f$$

Systèmes NON Série-Parallèles

- Un graphe est SP ssi il ne contient ni structure N ni structure W.
- Si on ne peut plus utiliser la décomposition SP, on peut enumérer et construire la table Booléenne.

réseau avec bridge :



Utilisation de la table Booléenne

- Examiner tous les cas UP et DOWN pour tous les composants
- Dans chaque cas, évaluer si le système est UP ou DOWN
- Calculer les probabilités de chaque cas (facile, c'est le produit des probas élémentaires à cause de l'hypothèse d'indépendance).
- Exemple : Si $E1 = E2 = E4 = 1$ et $E3 = E5 = 0$ le système est UP et la probabilité de cette configuration est $R_1 R_2 R_4 (1 - R_3)(1 - R_5)$.
- Sommer les probabilités que le système soit UP

Première partie de la table de l'exemple

1	2	3	4	5	Bridge	Probability
0	0	0	0	0	0	
0	0	0	0	1	0	
0	0	0	1	0	0	
0	0	0	1	1	1	(1-R1)(1-R2)(1-R3)R4R5
0	0	1	0	0	0	
0	0	1	0	1	0	
0	0	1	1	0	0	
0	0	1	1	1	1	(1-R1)(1-R2)R3R4R5
0	1	0	0	0	0	
0	1	0	0	1	0	
0	1	0	1	0	0	
0	1	0	1	1	1	(1-R1)R2(1-R3)R4R5
0	1	1	0	0	0	
0	1	1	0	1	0	
0	1	1	1	0	1	(1-R1)R2R3R4(1-R5)
0	1	1	1	1	1	(1-R1)R2R3R4R5

Seconde partie de la table de l'exemple

1	0	0	0	0	0	
1	0	0	0	1	0	
1	0	0	1	0	0	
1	0	0	1	1	1	R1(1-R2)(1-R3)R4R5
1	0	1	0	0	0	
1	0	1	0	1	1	R1(1-R2)R3(1-R4)R5
1	0	1	1	0	0	
1	0	1	1	1	1	R1(1-R2)R3R4R5
1	1	0	0	0	1	R1R2(1-R3)(1-R4)(1-R5)
1	1	0	0	1	1	R1R2(1-R3)(1-R4)R5
1	1	0	1	0	1	R1R2(1-R3)R4(1-R5)
1	1	0	1	1	1	R1R2(1-R3)R4R5
1	1	1	0	0	1	R1R2R3(1-R4)(1-R5)
1	1	1	0	1	1	R1R2R3(1-R4)R5
1	1	1	1	0	1	R1R2R3R4(1-R5)
1	1	1	1	1	1	R1R2R3R4R5

Résultat pour l'exemple

- En agrégeant la table précédente et en factorisant on trouve que :

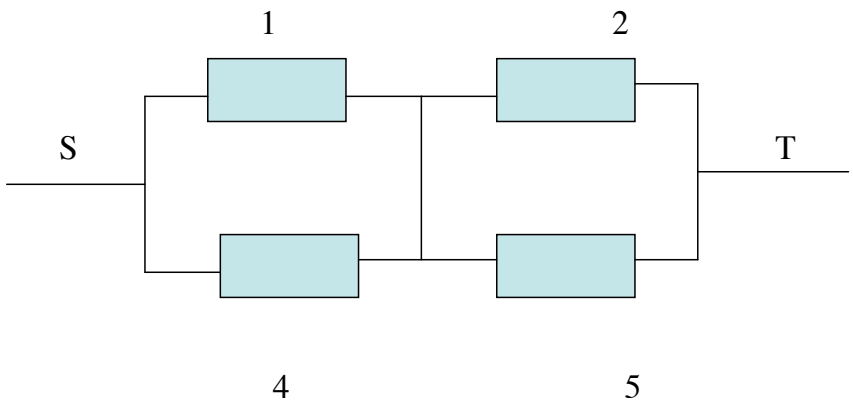
$$\begin{aligned} R_{bridge} &= R_1 R_2 \\ &+ R_1(1-R_2)(R_4R_5 + R_3(1-R_4)R_5) \\ &+ (1-R_1)R_4(R_5 + (1-R_5)R_2R_3) \end{aligned}$$

Conditionnement et Factorisation

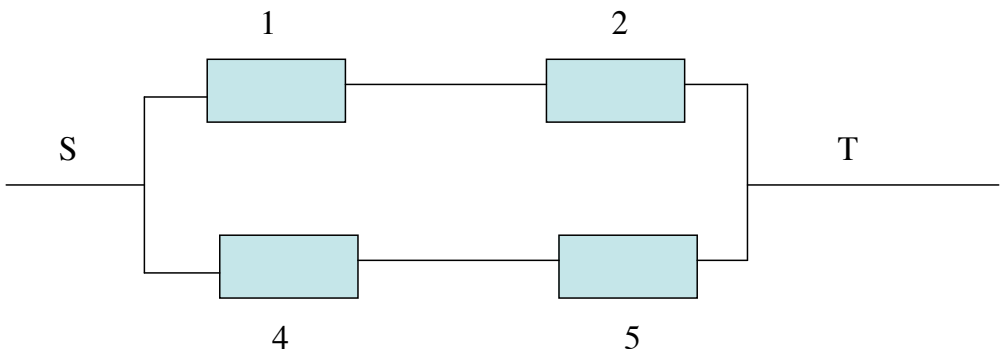
- Mais il faut considérer les 2^n configurations si il y a n objets.
- On conditionne sur le l'état d'un composant (ou de plusieurs composants) pour se ramener à des structures déjà étudiées ou faciles (SP)
- Sur l'exemple du bridge, on conditionne sur l'état du bloc 3.

Conditionnement sur 3

3 fonctionne



3 ne fonctionne pas



Conditionnement

- Si le composant 3 est DOWN, on obtient un modèle SP
- Si le composant 3 est UP, on obtient également un modèle SP
- On applique le théorème de conditionnement et les formules pour les modèles série-parallèles.

- $R_{3down} = 1 - (1 - R_1R_2)(1 - R_4R_5)$
- $R_{3up} = (1 - (1 - R_1)(1 - R_2))(1 - (1 - R_4)(1 - R_5))$
- On applique le théorème de conditionnement

$$R_{bridge} = R_3R_{3up} + (1 - R_3)R_{3down}$$

Composant K parmi N

- Système consistant en N composants indépendants.
- Le système est UP quand K ou plus de ces composants sont UP.
- Cas Identique : tous les composants ont le même taux de panne et de réparation.
- Cas Non Identique : Les composants ont des taux de panne et de réparation distincts par composant.

Avec sous-composants identiques

- Soit R la fiabilité d'un composant.
- On additionne les probabilités de toutes les configurations avec au moins K composants opérationnels.

$$R(K,N) = \sum_{j=K}^{N} R^j (1-R)^{N-j} \frac{N!}{j!(N-j)!}$$

- Somme partielle d'une distribution binomiale.
- Algorithme classique (attention aux approximations numériques)

Récurrence dans le cas général

- Système dont les composants sont distincts. La fiabilité du composant i est R_i.
- 3 remarques simples :
  - Un système avec $K = 0$ (sans contraintes) est toujours UP.

$$R(0, N) = 1$$

  - Un système trop contraint est toujours DOWN.

$$R(j, N) = 0 \quad si\ j > N$$

  - En conditionnant sur l'état du composant N :

$$R(i, N) = (1 - R_N)R(i, N - 1) + R_N R(i - 1, N - 1)$$

- Algorithme récursif.

Arbres de Fautes

- Une représentation graphique de la combinaison d'événements qui peut provoquer l'occurence d'une panne.
- Modèle combinatoire (i.e. ne representant ni le temps, ni les états)
- Objets élémentaires : feuilles, noeud interne d'un arbre, racine.
- Feuille : composant du système
- Noeud interne : porte logique
- racine : un boolean qui porte l'état du système. Il passe à TRUE quand le système est en PANNE...

Arbres de Fautes -2

- Portes :
  - OR : pour connecter des sous-systèmes en série
  - AND : pour connecter des sous-systèmes en parallèle
  - Porte $(N - K + 1)$ of N : pour connecter des composants qui sont dans un système K parmi N
- La panne d'un composant met à TRUE l'entrée de la porte où il est connecté.
- Dans les autres cas, elle est à FALSE.
- Evaluation de l'arbre...
- Si la racine de l'arbre est à TRUE, le système est en panne.
- Fiabilité : Probabilité que la racine soit FALSE.

Arbres de Fautes -3

- On peut ajouter de nombreuses autres portes : NOT, XOR, Priority AND, des dépendances fonctionnelles.
- Deux types de Fault Trees : avec ou sans événements répétés.
- La complexité de la résolution est fonction de la (non) répétition des événements.
- Répétition : formellement ce n'est plus un arbre si on regroupe et ce n'est plus indépendent si on sépare.

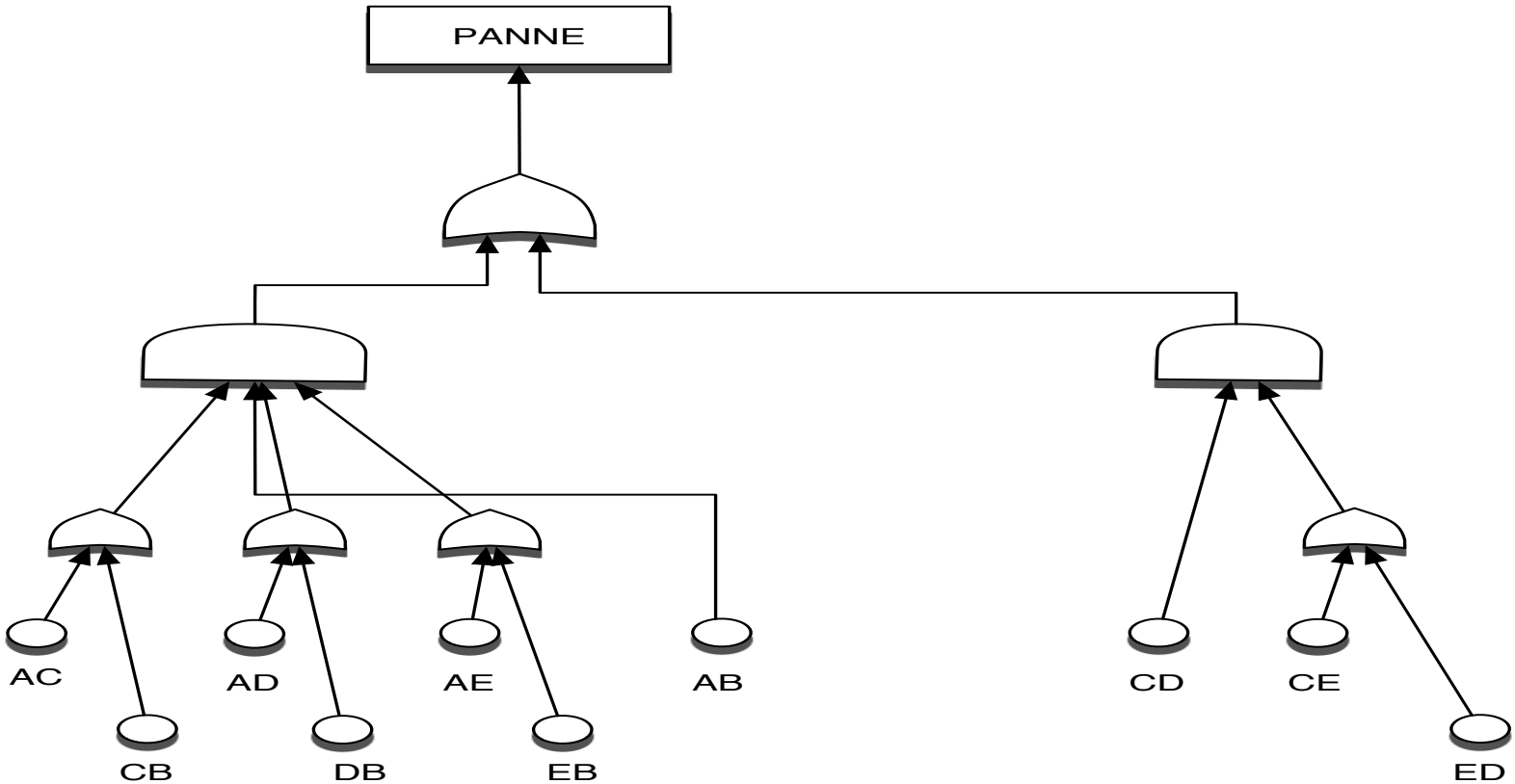
Exemple sans répétition : alternate routing

- Un réseau avec 5 noeuds : A, B, C, D, E et 10 liens.
- Le système est UP lorsque A est connecté à B et C est connecté à D par une route directe : AB, et CD ou une route supplémentaire : CED, ACB, ADB, AEB.

Arbre de Faute pour le routage

- Panne de liens. Routeurs Fiables.
- Feuilles : liens du réseau.
- Portes : de type OR ou AND.
- Pas de répétition à cause des chemins considérés.

Arbre de Faute pour le routage



Analyse des portes simples

- Panne sur une porte AND à deux entrées a et b :

$$(1 - R_p) = (1 - R_a)(1 - R_b)$$

- Porte sur une porte OR à deux entrées a et b :

$$(1 - R_s) = (1 - R_a) + (1 - R_b) - (1 - R_a)(1 - R_b)$$

- Car il faut ne pas compter deux fois le cas ou les composants a et b sont en panne.
- Cette équation est équivalente à $R_s = R_a R_b$.

Analyse de l'arbre de Faute pour le routage

- Ce qui donne pour le modèle du routage :

$$R = R_1 R_2$$

où

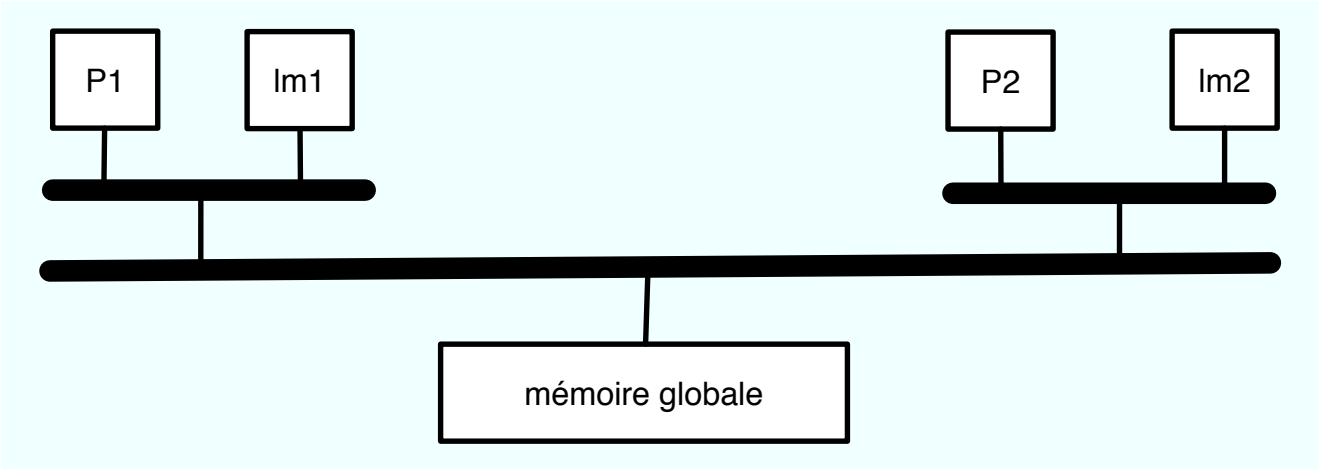
$$R_1 = 1 - (1 - R_{ab})(1 - R_{ac}R_{cb})(1 - R_{ad}R_{db})(1 - R_{ae}R_{eb})$$

et

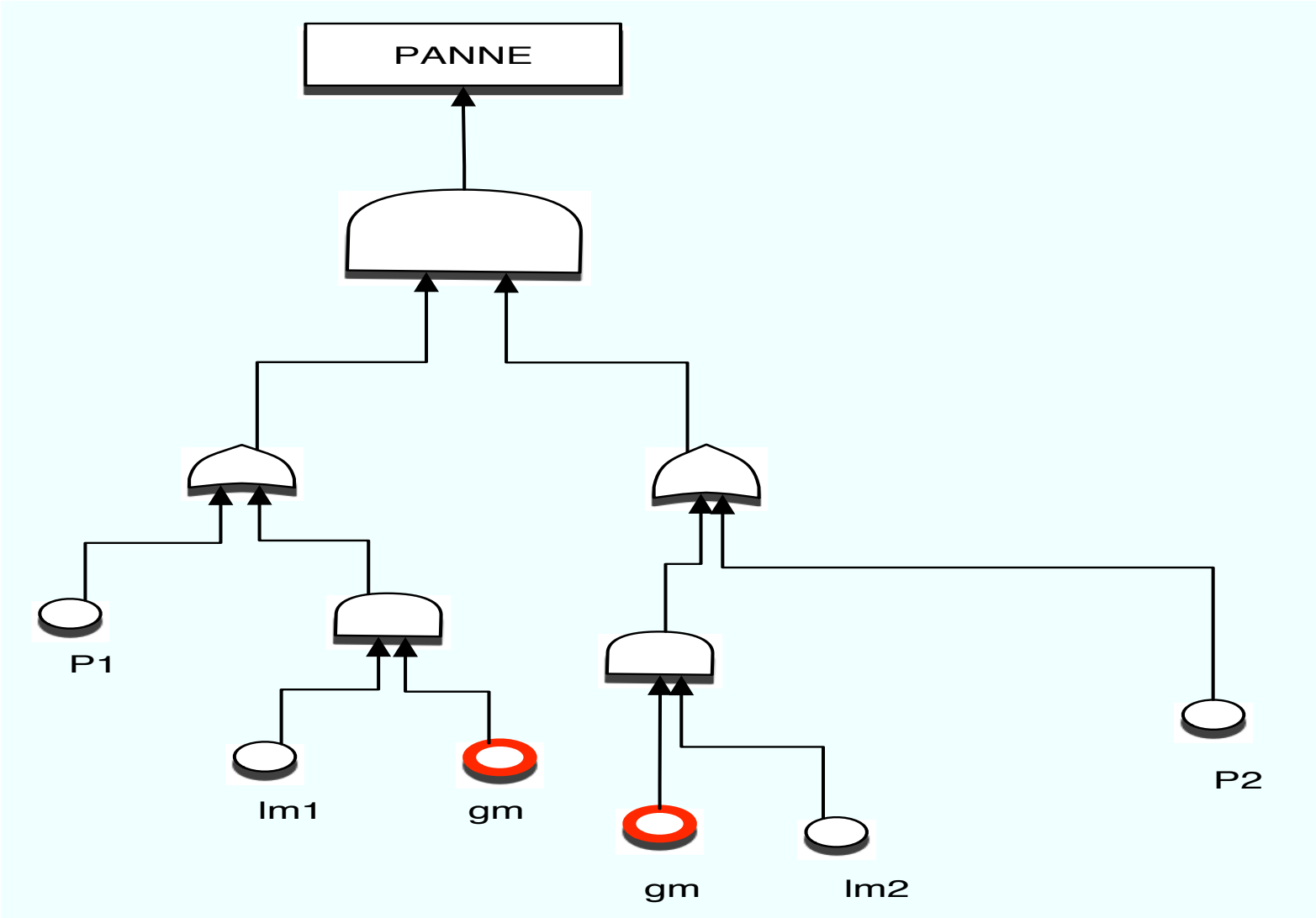
$$R_2 = 1 - (1 - R_{cd})(1 - R_{ce}R_{ed})$$

Arbre de Fautes avec répétition

- Des feuilles sont dupliquées.
- Exemple : 2 processeurs $p1$ et $p2$, deux mémoires locales $lm1$ et $lm2$ et une mémoire globale gm.
- Le système est UP si il existe au moins un processeur UP connecté à une mémoire (locale ou globale) UP.

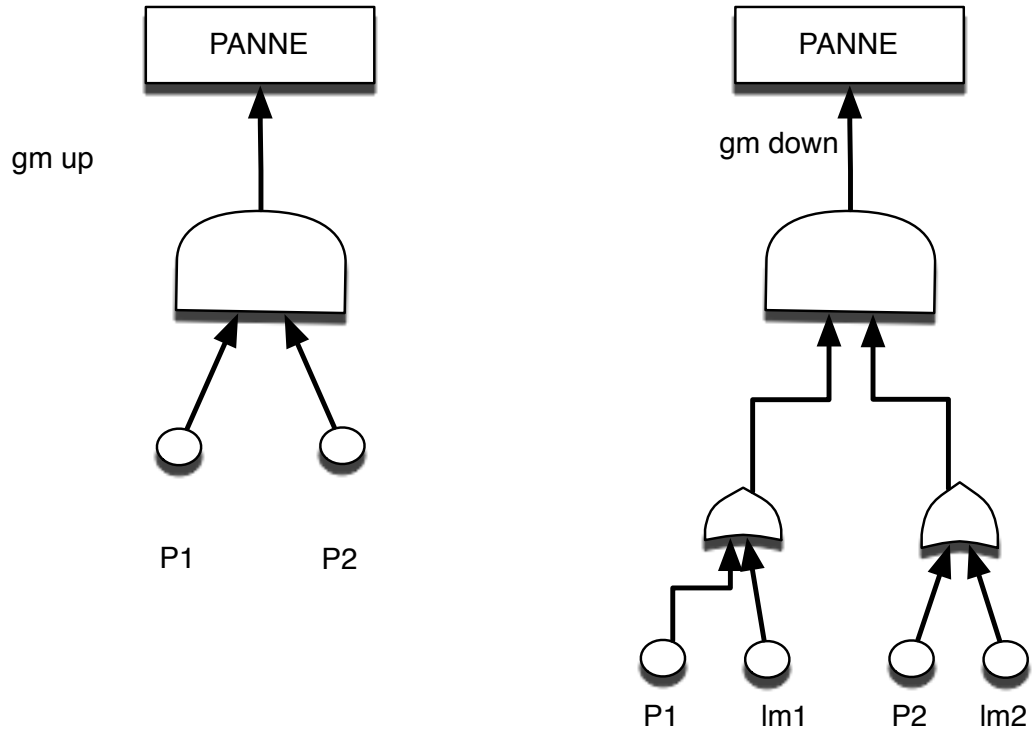


Arbre de Fautes avec répétition



Analyse Exemple

- On conditionne sur l'état de gm et le théorème de conditionnement.



Formule Logique

- On peut aussi écrire la panne comme une formule logique à partir de l'arbre de fautes.
- Dans l'exemple précédent, on note $P1$ la proposition "le processeur $p1$ fonctionne", (notations identiques pour les autres propositions)
- $$Panne = GM(\bar{P}1\bar{P}2) + G\bar{M}[(\bar{P}1 + L\bar{M}1)(\bar{P}2 + L\bar{M}2)]$$
- après développement :
  $Panne = GM\bar{P}1\bar{P}2 + G\bar{M}\bar{P}1\bar{P}2 + \bar{P}2L\bar{M}1G\bar{M} + \bar{P}1L\bar{M}2G\bar{M} + L\bar{M}1L\bar{M}2G\bar{M}$
- On obtient :

$$Panne = \bar{P}1\bar{P}2 + \bar{P}2L\bar{M}1G\bar{M} + \bar{P}1L\bar{M}2G\bar{M} + L\bar{M}1L\bar{M}2G\bar{M}$$

Coupes minimales

COUPE : Ensemble d'événements de base qui entraînent l'Événement Indésirable (EI).
COUPE MINIMALE : Plus petite combinaison d'événements de base qui entraînent l'Événement Indésirable.

- Plusieurs coupes minimales peuvent entraîner l'événement indésirable.
- L'occurrence de tous les événements de base d'une coupe minimale est nécessaire pour qu'apparaisse l'événement indésirable.
- La recherche des coupes minimales se fait à partir dune transformation de l'arbre de défaillance en expression booléenne.

$EI = \cup_i \{Coupes\ Minimales\}$.

Méthodes pour le calcul de la probabilité de EI

- Méthode directe si les événements de base ne sont pas répétés
- Méthode utilisant les coupes minimales :

Soit $EI = x \cup y \cup z$ avec x, y et z des coupes minimales
$P(EI) = P(x \cup y \cup z)$
$P(EI) =$
$P(x) + P(y) + P(z) - P(y)P(z) - P(x)P(y) - P(x)P(z) + P(x)P(y)P(z)$

Généralisation (Formule de Poincaré) :

$P(EI) = \sum_i P(x_i) - \sum_{ij} P(x_i)P(y_j) + \sum_{ijk} P(x_i)P(y_j)P(z_k) - \cdots + \cdots - \cdots$

Approximation :
$\sum_i P(C_i) - \sum_i \sum_{k \neq i} P(C_i \cap C_k) \leq P(EI) \leq \sum_i P(C_i)$

où $P(C_i \cap C_k) = P(C_i)P(C_k)$

Complexité de l'Analyse

- Arbre de Fautes sans répétition : résolution en temps linéaire.
- Arbre de Fautes avec répétition : résolution en temps exponentiel.
- En pratique : avec BDD et Factorisation, les outils logiciels peuvent résoudre des problèmes avec des centaines de composants.