Une extension des BDMP pour la modélisation des politiques de redondances passives

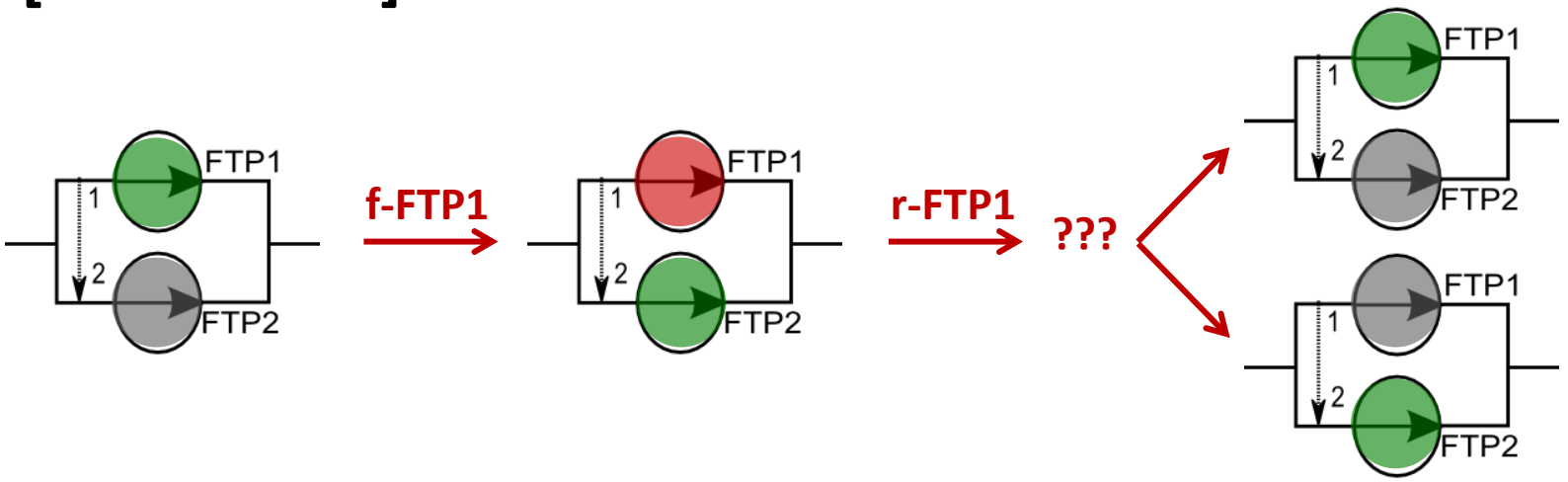
Pierre-Yves Piriou
(Doctorant au LURPA, ENS Cachan)
Directeurs de Thèse :
Jean-Marc Faure, Jean-Jacques Lesage
Financement : **cluster CONNEXION**
Connexion
Contrôle-commande numérique pour le nucléaire

Contexte industriel

- **Cluster CONNEXION** (COntrôle commande Nucléaire Numérique pour l'EXport et la rénovatION)
- **Objectif :** définir et valider une architecture innovante de plateformes de contrôle-commande (CC) adaptée aux centrales nucléaires en France et à l'International.
- **Partenaires**
  - **Opérateur de la filière nucléaire :** EDF
  - **Intégrateurs :** AREVA, ALSTOM
  - **Fournisseurs de logiciels embarqués :** Atos Worldgrid, Rolls-Royce, Civil Nuclear, CORYS T.E.S.S., Esterel Technologies, All4Tec, Predict
  - **Laboratoire académiques :** CEA, INRIA, CNRS/CRAN, ENS Cachan, LIG, Telecom ParisTech

Problématique

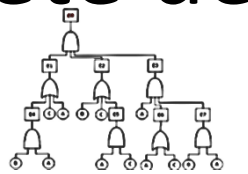
- Comment prendre en compte des **stratégies de redondances** complexes dans les études de SdF?
- Contraintes technologiques des systèmes considérés :
  - Systèmes dynamiques
  - Composants réparables
  - Composants faillibles à la sollicitation
- *Exemple :* commutation « au plus tôt » / « au plus tard » **[Piriou 2015]**



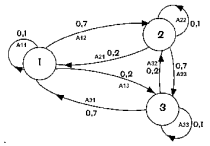
Commutateurs imparfaits

- Dans le cas des **redondances passives**, l'activation des composants dépend de leur état dysfonctionnel.
- La réalisation d'une redondance passive implique la mise en place de **commutateurs**.
- Ces commutateurs peuvent défaillir.
- Ces défaillances et leurs effets doivent pouvoir être modélisés. **[Piriou 2014]**
- *Exemple :* les sorties d'un Automate Programmable Industriel peuvent se figer.

Approche de modélisation pour la Sûreté de Fonctionnement



- **Booleen**
  - Représentatif de la structure du système
    - => Comportement du système à interpréter
  - Formalismes dédiés à la sûreté de fonctionnement
  - Faible pouvoir d'expression
  - « Popularité » industrielle
  - Exemples : arbre de défaillance, RBD, ADD … **[Ruijters 2014]**



- **Basé sur espace d'état**
  - Représentatif du comportement du système
    - => difficile à construire manuellement sans erreur
  - Formalismes utilisés pour la sûreté de fonctionnement
  - Grand pouvoir d'expression
  - « Popularité » académique
  - Exemples : chaîne de Markov, Réseaux de Petri, systèmes de transitions… **[Ajmone Marsan 1994]**

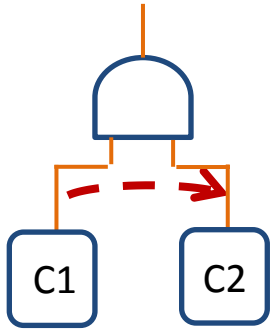
- Ces deux types de formalismes sont complémentaires; Les approches MBSA visent à intégrer les avantages des deux.

Boolean logic Driven Markov Processes

- Définit chez EDF R&D **[Bouissou 2003]**
- Formalisme combinant les avantages des arbres de défaillance et des chaînes de Markov
  - Permet de modéliser finement le comportement dysfonctionnel des composants (défaillance à la sollicitation et en fonctionnement, réparations…)
  - La primitive de gâchette permet de modéliser un commutateur parfait.
- Un modèle de BDMP est un **arbre de défaillance** dont la défaillance des feuilles est déterminée par des **processus de Markov**, pilotés par des **gâchettes**.

Modélisation d'une redondance passive

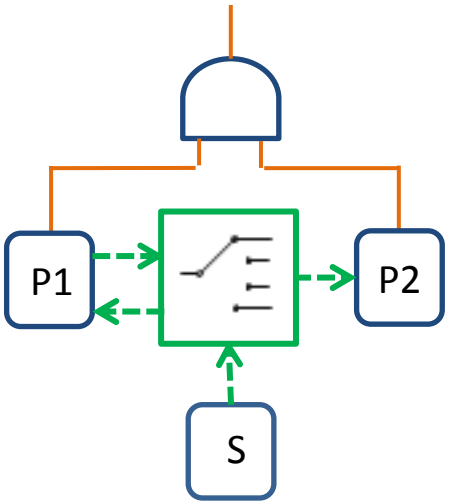
- Deux mécanismes de commutation
  - **Remplacement** : désactiver C1; activer C2
  - **Rétablissement :** désactiver C2; activer C1



- Une gâchette de BDMP traduit une unique politique de redondance :
  - **Remplacement** : dès que C1 défaille
  - **Rétablissement** : dès que C1 est réparé

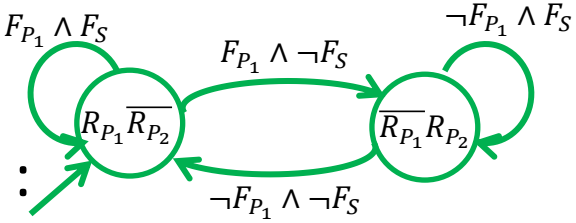
- Déclenchements inconditionnels des mécanismes
  - Pas de prise en compte des défauts du commutateur.

Modélisation de la SdF par BDMP Généralisé



- GBDMP = Arbre de défaillance + Processus de Markov + Commutateurs

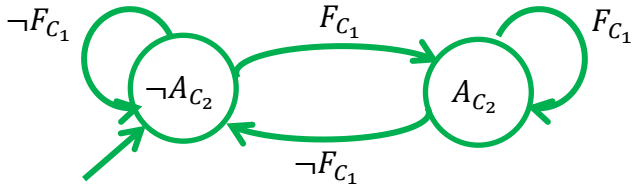
- Avec la notion de commutateur, on développe l'idée des gâchettes :
  - Un commutateur peut avoir plusieurs entrées/sorties
  - Il se réfère à une politique de redondance spécifiée par une machine de Moore

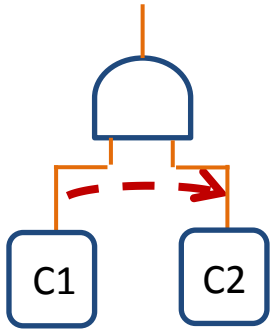


- Améliore le pouvoir d'expression des BDMP :
  - La politique de redondance passive implémentée est formellement spécifiable.
  - La politique peut considérer l'emploi de commutateurs imparfaits.

Machine de Moore

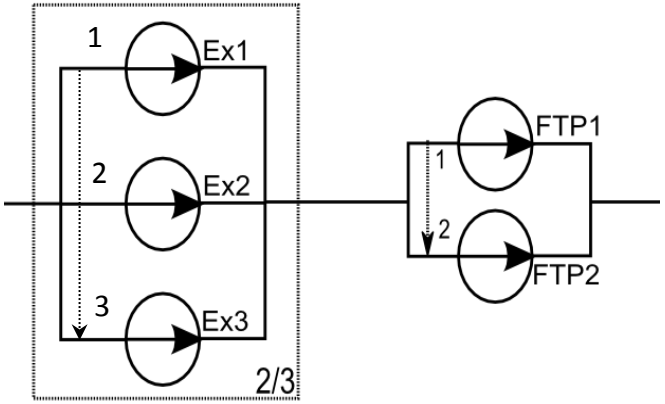
- Une machine de Moore est un automate à entrées/sorties :
  - La transition d'un état à un autre est provoquée par la réception d'un certain signal d'entrée
  - Chaque état correspond à l'émission d'un certain signal de sortie
- Idéal pour spécifier formellement une logique de commande. **[Moore 1956]**
- *Exemple :* machine de Moore traduisant la politique de redondance associée aux gâchettes de BDMP





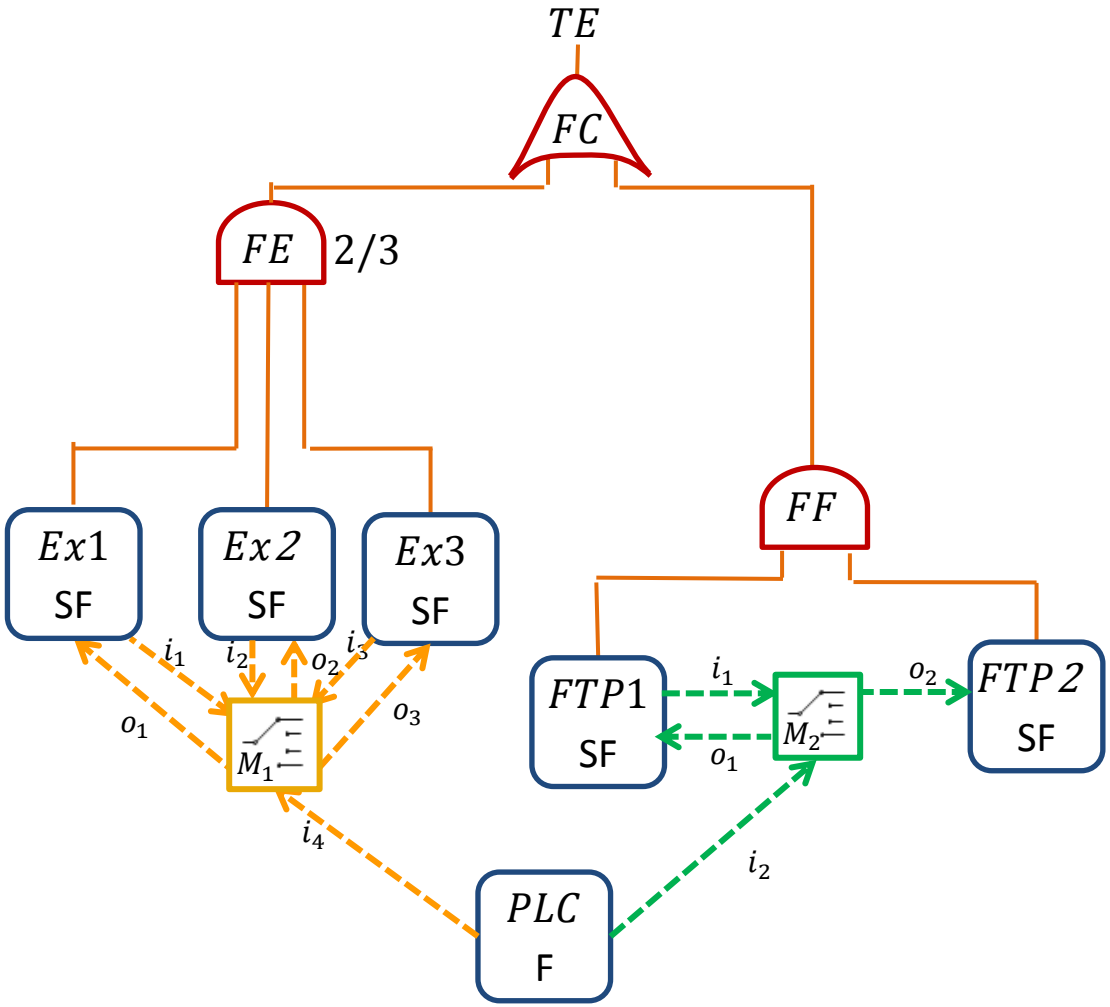
Exemple : description

- Le service est assuré par 2 pompes d'extraction (Ex) et 1 pompe de régulation (FTP).
- 2 politiques de redondances :
  - « FTP1 doit être utilisé dès que possible »
  - « Les commutations entre les pompes d'extraction doivent être limitées au maximum »
- Ces politiques sont implémentées sur un automate qui peut défaillir, tel que ses sorties sont figées.

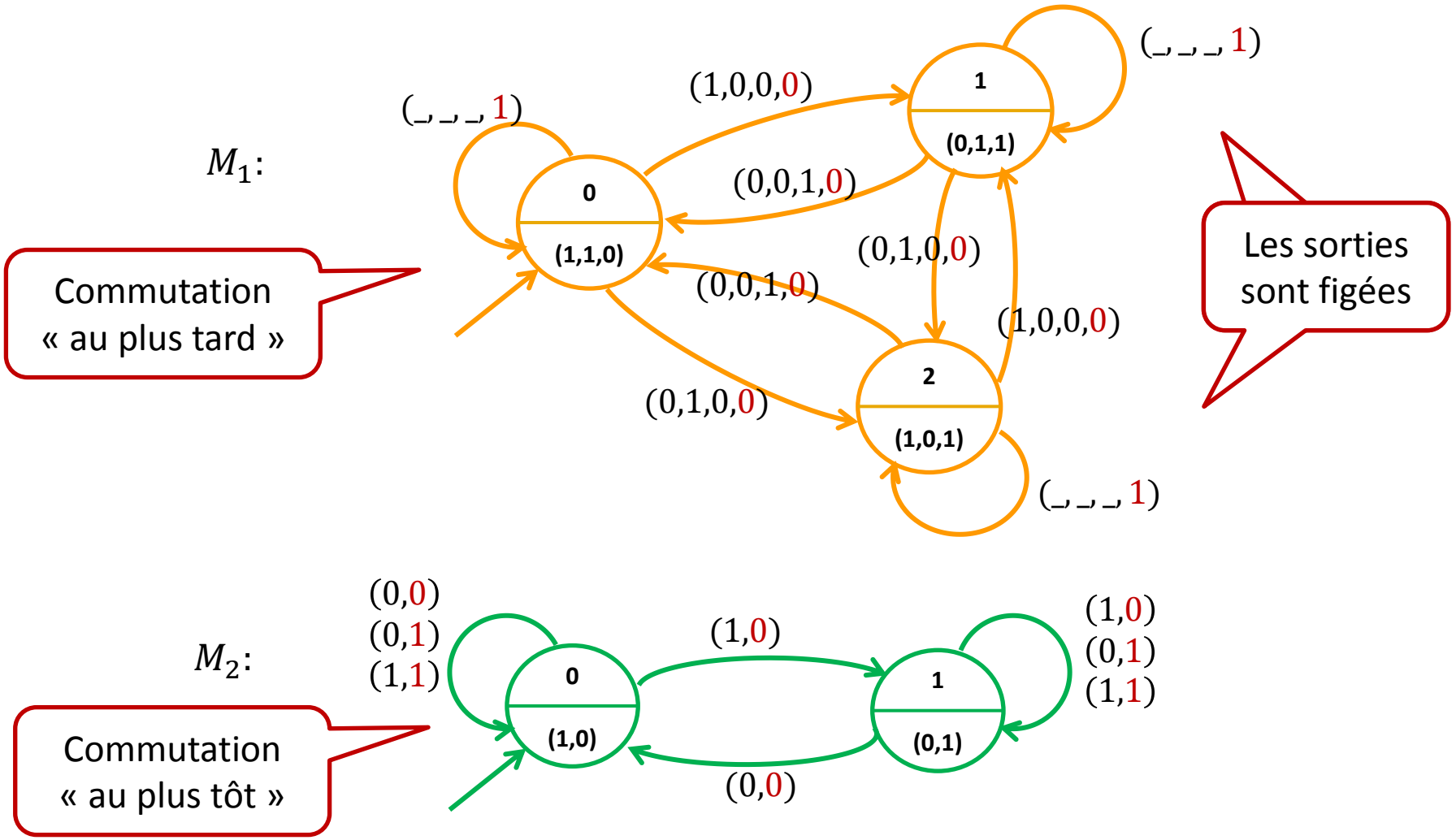


Traitement d'un exemple

- Les variables d'entrées traduisent le statut dysfonctionnel du nœud (Booléen)
- Les variables de sorties traduisent le statut d'activation du nœud (Booléen)



Exemple : Machine de Moore



Conclusion

- Contribution à la modélisation de la sûreté de fonctionnement des systèmes complexes
  - Prise en compte de stratégies de redondances complexes
  - Prise en compte des échecs de réalisation des redondances
  - Conservation du format arbre de défaillance
- Travail en cours et à faire
  - Généralisation pour considérer les systèmes multi-phasés et multi-états
  - Transcription de techniques d'analyse à ce formalisme
  - Outils support (méthodologie de construction, éditeur graphique, règles de cohérence, BDC, simulateur…)

Merci pour votre attention!

Une extension des BDMP pour la modélisation des politiques de redondances passives

Pierre-Yves Piriou

Connexion
Contrôle-commande numérique pour le nucléaire

Références

- **[Piriou 2015]** : P.-Y. Piriou, J.-M. Faure and J.-J. Lesage, *Modeling standby redundancies in repairable systems as guarded preemption mechanisms*, Dependable Control on Discrete Event Systems (DCDS'15), Cancun (Mexique), 7 pages, 2015, (*expected)*
- **[Piriou 2014] :** P.-Y. Piriou, J.-M. Faure and J.-J. Lesage, *Control-in-the-loop Model Based Safety Analysis,* 23th European Safety & Reliability Conf. (ESREL'14), Wroclaw (Poland), 8 pages, September 11-14, 2014
- **[Ruijters 2014] :** E. Ruijters and M. Stoelinga, *Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools*, Tech. Rep. TR CTIT-14-14, Centre for Telematics and Information Technology, University of Twente, Enschede (December 2014).
- **[Ajmone Marsan 1994] :** M. Ajmone Marsan, G. Balbo, S. Donatelli, G. Franceschinis and G. Conte, *Modelling with generalized stochastic Petri nets*, John Wiley & Sons, Inc., 1994.
- **[Bouissou 2003]** : M. Bouissou and J.-L. Bon. *A new formalism that combines advantages of fault trees and markov models: Boolean logic driven markov processes*. Reliability Engineering and Systems Safety, 82(2):149-163, 2003.
- **[Moore 1956]** : E.-F. Moore, *Gedanken-experiments on sequential machines*, Annals of Mathematical Studies 34 (1956) 129–153.

Définition formelle (1)

- Un processus de Markov commuté $\boldsymbol{P}$ à k phases est un 3-uplet $< \left(\boldsymbol{\mathcal{Z}_i^P}\right)_{\boldsymbol{0\leq i<k}}, \boldsymbol{A_F^P}, \left(\boldsymbol{f_{i\to j}^P}\right)_{\substack{(\boldsymbol{i,j})\in[\![\boldsymbol{0},\boldsymbol{k-1}]\!]^2 \\ \boldsymbol{i\neq j}}} >$, tel que :
  - $\left(\boldsymbol{\mathcal{Z}_i^P}\right)_{\boldsymbol{0\leq i<k}}$ est une famille de chaîne de Markov homogènes : $\boldsymbol{\mathcal{Z}_i^P} =< A_i^P, p0_i^P, M_i^P >$
    - A_i^P est un ensemble fini d'états;
    - $p0_i^P : A_i^P \longrightarrow [0,1]$ tel que $\sum_{x\in A_i^P} p0_i^P(x) = 1$ est une distribution de probabilité initiale;
    - M_i^P est la matrice des taux transitions;
  - $\boldsymbol{A_F^P \subseteq A^P = \bigcup_{i=0}^{k-1} A_i^P}$
  - $\left(\boldsymbol{f_{i\to j}^P}\right)_{\substack{(\boldsymbol{i,j})\in[\![\boldsymbol{0},\boldsymbol{k-1}]\!]^2 \\ \boldsymbol{i\neq j}}}$ est une famille de fonctions de commutation probabilistes, entre les états de deux chaînes de Markov :

$$\boldsymbol{f_{i\to j}^P}: A_i^P \times A_j^P \longrightarrow [0,1] \text{ tel que } \begin{cases} \forall x \in A_i^P, \sum_{y\in A_j^P} \boldsymbol{f_{i\to j}^P}(x,y) = 1 \\ \forall x \in A_i^P \cap \boldsymbol{A_F^P}, \sum_{y\in A_j^P \cap \boldsymbol{A_F^P}} \boldsymbol{f_{i\to j}^P}(x,y) = 1 \end{cases}$$

Permet d'identifier les états défaillants

On ne peut pas réparer un composant en le faisant commuter

Définition formelle (2)

- Une machine de Moore $\boldsymbol{M}$ est un 6-uplet $< \boldsymbol{Q^M}, \boldsymbol{{Q_0}^M}, \boldsymbol{{\Sigma_I}^M}, \boldsymbol{{\Sigma_O}^M}, \boldsymbol{trans^M}, \boldsymbol{out^M} >$ tel que :
  - $\boldsymbol{Q^M}$ est un ensemble fini d'états
  - $\boldsymbol{{Q_0}^M} \in \boldsymbol{Q^M}$ est un état initial
  - $\boldsymbol{{\Sigma_I}^M}$ est un alphabet dit d'entrée
  - $\boldsymbol{{\Sigma_O}^M}$ est un alphabet dit de sortie
  - $\boldsymbol{trans^M} : \boldsymbol{Q^M} \times \boldsymbol{{\Sigma_I}^M} \longrightarrow \boldsymbol{Q^M}$ est une fonction dite de transition
  - $\boldsymbol{out^M} : \boldsymbol{Q^M} \longrightarrow \boldsymbol{{\Sigma_O}^M}$ est une fonction dite de sortie

Définition formelle (3)

- Un GBDMP est un 6-uplet : $< \boldsymbol{V}, \boldsymbol{E}, \boldsymbol{\kappa}, \boldsymbol{v}, \boldsymbol{pol}, \boldsymbol{SMP} >$, tel que :
  - $\boldsymbol{V} = \boldsymbol{N} \cup \boldsymbol{S} = \boldsymbol{G} \cup \boldsymbol{C} \cup \boldsymbol{S}$ est un ensemble d'éléments, les uns sont appelés nœuds (node), les autres commutateurs (switch). Parmi les nœuds, certains sont appelés portes (gate), les autres composants (component).
  - $\boldsymbol{E} = \boldsymbol{E_F} \cup \boldsymbol{E_S} \subseteq \boldsymbol{V} \times \boldsymbol{V}$ est un ensemble d'arcs reliant les éléments tel que : $\boldsymbol{E_F} \subseteq \boldsymbol{N} \times \boldsymbol{N}$ et $\boldsymbol{E_S} \subseteq (\boldsymbol{N} \times \boldsymbol{S}) \cup (\boldsymbol{S} \times \boldsymbol{N})$.

  Notation : on dénomme les graphes suivant $\mathcal{G}_F =< N, E_F >$, $\mathcal{G}_S =< V, E_S >$ et $\mathcal{G} = \mathcal{G}_F \cup \mathcal{G}_S$

  - $\boldsymbol{\kappa}: \boldsymbol{G} \longrightarrow \mathbb{N}^*$ une fonction qui associe un entier non nul aux portes.
  - $\boldsymbol{v}: \boldsymbol{E} \longrightarrow \mathbb{N}$ une fonction qui associe un entier aux arcs.
  - $\boldsymbol{pol}: \boldsymbol{S} \longrightarrow \mathfrak{M}$ est une fonction qui associe une machine de Moore (politique de reconfiguration) aux commutateurs.
  - $\boldsymbol{SMP}: \boldsymbol{C} \longrightarrow \mathfrak{P}$ est une fonction qui associe un processus Markovien piloté aux composants.

Interprétation des sommets du graphe

- Soit $\boldsymbol{\mathcal{V}}$ l'ensemble des variables d'un GBDMP :
  - Pour chaque nœud $\boldsymbol{n}$, 2 variables Booléennes : $\boldsymbol{F_n}$ et $\boldsymbol{R_n}$, et une variable entière positive $\boldsymbol{M_n}$;
  - Pour chaque composant $\boldsymbol{c}$, 1 variable d'état : $\boldsymbol{X_c}$, dont le domaine de définition est l'ensemble des états définis pour les chaînes de Markov de $\boldsymbol{SMP(c)}$;
  - Pour chaque commutateur $\boldsymbol{s}$, 1 variable d'état : $\boldsymbol{U_s}$, dont le domaine de définition est l'ensemble des états définis pour $\boldsymbol{pol(s)}$.

Règles d'évolution

1) $\forall \boldsymbol{k} \in \mathbb{N}^*, \forall \boldsymbol{c} \in \boldsymbol{C}, \quad \boldsymbol{F_c(k)} \coloneqq \left(\boldsymbol{X_c(k)} \in A_F^{SMP(\boldsymbol{c})}\right)$

2) $\forall \boldsymbol{k} \in \mathbb{N}^*, \forall \boldsymbol{g} \in \boldsymbol{G},$

$$\boldsymbol{F_g(k)} \coloneqq \boldsymbol{Card}(\{\boldsymbol{n} \in \boldsymbol{Sons(g)} | \boldsymbol{F_n(k)} \vee \neg \boldsymbol{R_n(k)}\}) \geq \boldsymbol{\kappa(g)}$$

3) $\forall \boldsymbol{k} \in \mathbb{N}^*, \forall \boldsymbol{s} \in \boldsymbol{S}, \quad \boldsymbol{U_s(k)} \coloneqq \boldsymbol{trans}^{\boldsymbol{pol(s)}}(\boldsymbol{U_s(k-1)}, IN_s(k))$

4) $\forall \boldsymbol{k} \in \mathbb{N}, \forall \boldsymbol{n} \in \boldsymbol{N},$

$$\begin{cases} \textbf{si } \exists s \in S | s = Trig(\boldsymbol{n}): & \boldsymbol{R_n(k)} \coloneqq \left(\boldsymbol{out}^{\boldsymbol{pol(s)}}(\boldsymbol{U_s(k)})\right)_{\upsilon((s,n))} \\ \textbf{sinon}: & \boldsymbol{R_n(k)} \coloneqq \boldsymbol{True} \end{cases}$$

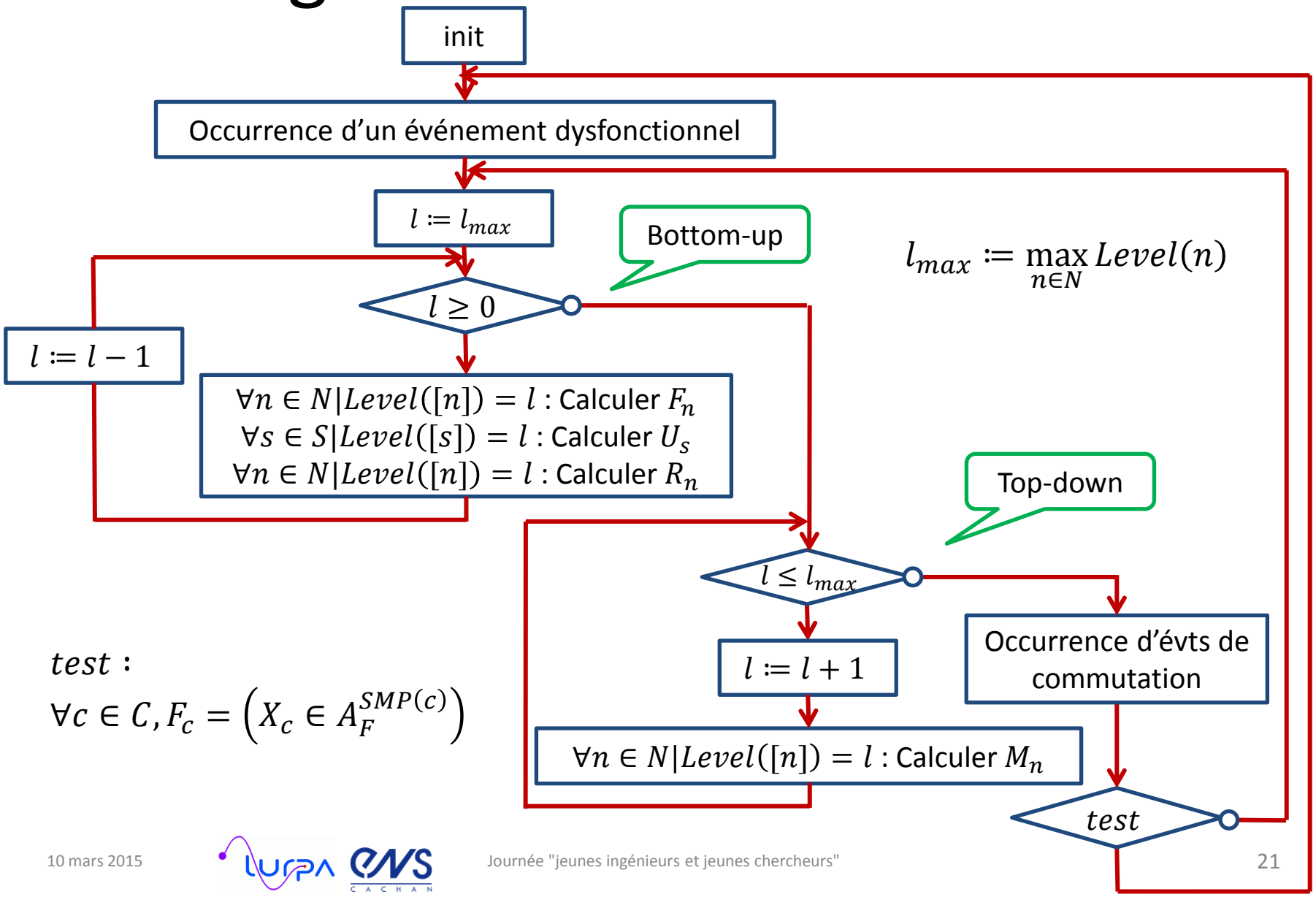
5) $\forall \boldsymbol{k} \in \mathbb{N}, \forall \boldsymbol{n} \in \boldsymbol{N},$

$$\begin{cases} \textbf{si } \boldsymbol{Fathers(n)} = \emptyset: & \boldsymbol{M_n(k)} \coloneqq \boldsymbol{R_n(k)} \\ \textbf{sinon}: & \boldsymbol{M_n(k)} \coloneqq \boldsymbol{R_n(k)} * \max_{g \in \boldsymbol{Fathers(n)}} \boldsymbol{M_g(k)} * \upsilon\big((g,n)\big) \end{cases}$$

6) $\forall \boldsymbol{k} \in \mathbb{N}, \forall \boldsymbol{c} \in \boldsymbol{C},$

$$\begin{cases} \textbf{si } \mathbf{e}(k) \cap \boldsymbol{\mathcal{E}_s^c} = \emptyset: & \boldsymbol{X_c(k+1)} \coloneqq \boldsymbol{X_c(k)} \\ \textbf{sinon}: & \boldsymbol{X_c(k+1)} \coloneqq \boldsymbol{y} \,|\mathbf{e}(k) \cap \boldsymbol{\mathcal{E}_s^c} = \left\{e^c_{X(k) \to y}\right\} \end{cases}$$

Algorithme du simulateur



$test$:

$\forall c \in C, F_c = \left(X_c \in A_F^{SMP(c)}\right)$