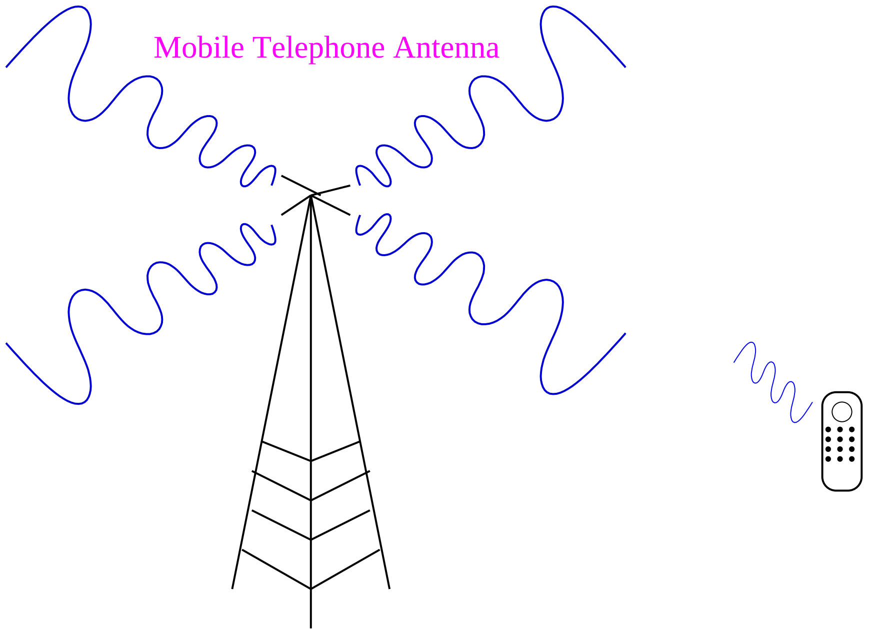
PEPA : Une Approche Formelle pour la Modélisation des Performances

Leïla Kloul,

20 Janvier 2004

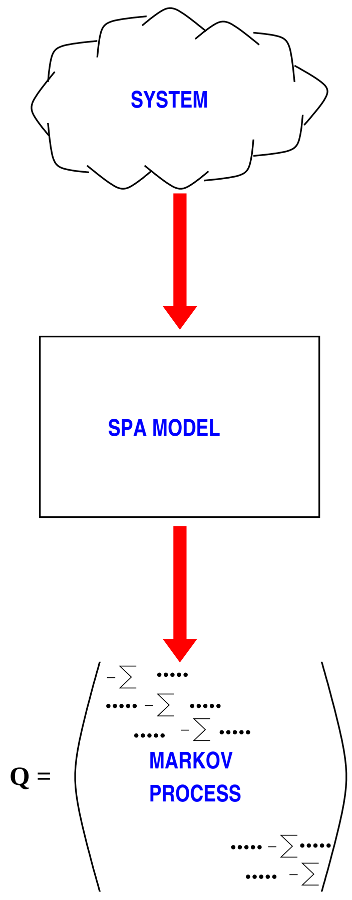
Modélisation des Performances



- Combien de fréquences doivent être utilisées pour avoir des probabilités de blocage/rejet faibles ?

- Comment doivent être réparties les fréquences dans un modèle hiérarchique pour avoir des probabilités de blocage/rejet faibles ?

Modélisation des Performances à l'aide des CMTC



Construction du Modèle

- décrire le système en utilisant un formalisme de haut niveau (SPN, SAN, PEPA)

Manipulation du Modèle

- simplifier le modèle
- agréger le modèle

Résolution du Modèle

- générer la Chaîne de Markov à Temps Continu (CMTC)
- résoudre la CMTC pour trouver la distribution stationnaire
- calculer les mesures de performance

Sommaire

- Les Algèbres de Processus Stochastiques (PEPA)
- Les avantages du formel
  - Construction du Modèle
  - Manipulation du Modèle
  - Résolution du Modèle
- Outils et Applications
- Conclusions

Les Algèbres de Processus Stochastiques

- Les algèbres de processus sont des langages compositionnels abstraits pour la spécification et l'analyse de systèmes concurrents.

- *"Méthodologie de conception qui permet systèmatiquement la construction de systèmes complexes à partir de systèmes plus petits."* [Milner]

Les Algèbres de Processus Stochastiques

Ingrédients

- Description textuelle
- Compositionalité
- Sémantique formelle
- Équivalences

Quantificateurs

- Actions avec des durées
- Branchements probabilistes

Performance Evaluation Process Algebra (PEPA)

- Modèles construits à partir de *composantes* qui exécutent des *activitiés*.

$$(\alpha, r).P$$

type d'action
ou nom

taux de l'activité
(paramètre d'une
distribution exponential)

composante/
dérivée

- Le langage est utilisé pour générer une CMTC pour modéliser les performances.

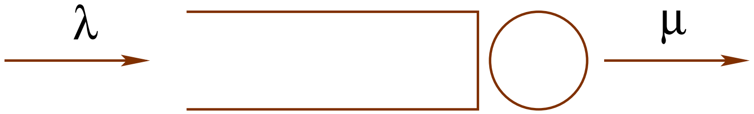
MODÈLE PEPA $\xrightarrow{\text{Règles SOS}}$ GRAPHE DE TRANSITIONS MULTI-LABEL $\xrightarrow{\text{Diagramme de transition des états}}$ CMTC $\boldsymbol{Q}$

La syntaxe de PEPA

$$S \quad ::= \quad (\alpha, r).S \mid P + Q \mid P \underset{L}{\bowtie} Q \mid P/L \mid A$$

PREFIXE:	$(\alpha, r).S$	désigne la première action
CHOIX:	$P + Q$	competition entre composantes (race policy)
COOPERATION:	$P \underset{L}{\bowtie} Q$	$\alpha \notin L$ actions concurrentes (*actions individuelles*) $\alpha \in L$ actions de coopération (*actions partagées*)
HIDING:	P/L	$\alpha \in L \Rightarrow \alpha \rightarrow \tau$ (*abstraction*)
CONSTANTE:	$S \overset{\mathit{def}}{=} A$	affecte un nom à une composante

Exemple



$$Source \stackrel{def}{=} (arrivée, \lambda).Source$$

$$Buffer_0 \stackrel{def}{=} (arrivée, \top).Buffer_1$$

$$Buffer_i \stackrel{def}{=} (arrivée, \top).Buffer_{i+1} + (départ, \mu).Buffer_{i-1}$$

$$Buffer_N \stackrel{def}{=} (départ, \mu).Buffer_{N-1}$$

$$\mathbf{Queue}_{\mathbf{i}} \stackrel{def}{=} Source \underset{\{arrivée\}}{\bowtie} Buffer_i$$

La Sémantique Formelle de PEPA

- Donne un sens non-ambiguë au langage.
- Pour chaque opérateur du langage, on associe une règle qui définit toutes les transitions possibles.

$$\text{MODÈLE AP} \xrightarrow{\text{Règles SOS}} \text{GRAPHE DE TRANSITIONS}$$

- SOS : Sémantique Opérationnelle Structurée

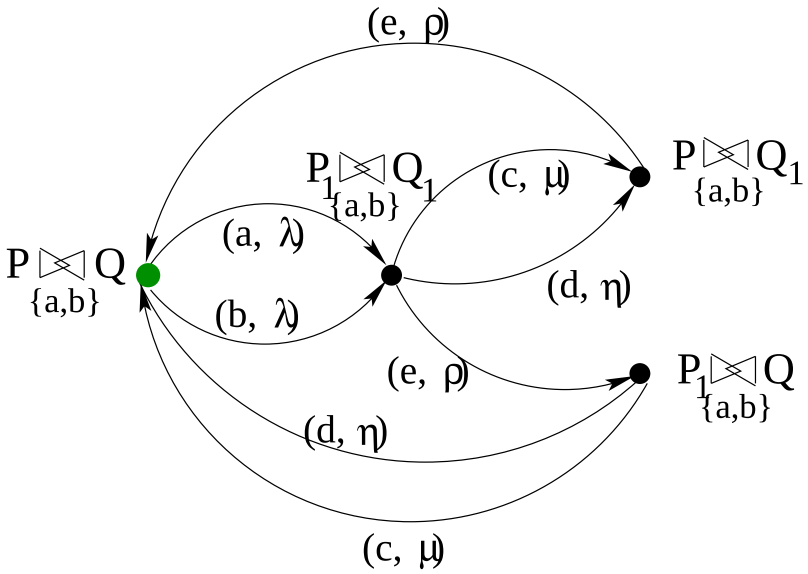
- Le système de transitions (labélisé) décrit tous les comportements possibles d'une expression du langage.

Exemple 1

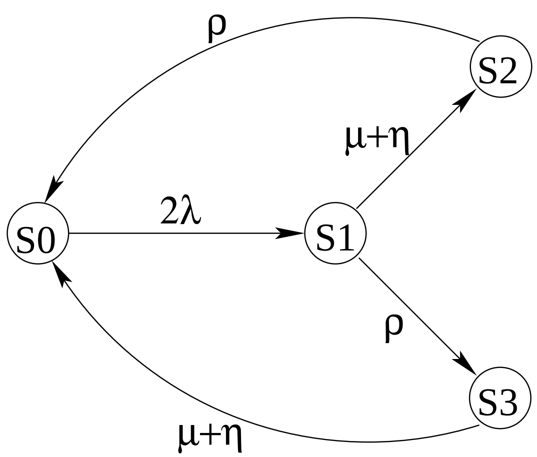
$$P \stackrel{def}{=} (a, \lambda).P_1 + (b, \lambda).P_1 \qquad Q \stackrel{def}{=} (a, \top).Q_1 + (b, \top).Q_1$$

$$P_1 \stackrel{def}{=} (c, \mu).P + (d, \eta).P \qquad Q_1 \stackrel{def}{=} (e, \rho).Q$$

$$\textbf{Système} = P \underset{\{a,b\}}{\bowtie} Q$$



Graphe de transition



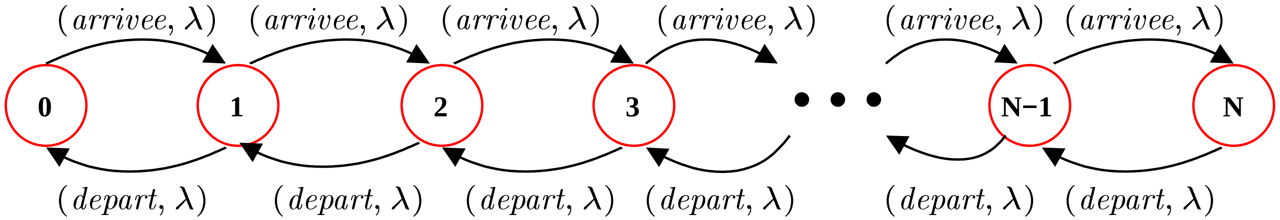
CMTC

Exemple 2

$$Source \stackrel{def}{=} (arrivée, \lambda).Source$$

$$\begin{aligned} Buffer_0 &\stackrel{def}{=} (arrivée, \top).Buffer_1 \\ Buffer_i &\stackrel{def}{=} (arrivée, \top).Buffer_{i+1} + (départ, \mu).Buffer_{i-1} \\ Buffer_N &\stackrel{def}{=} (départ, \mu).Buffer_{N-1} \end{aligned}$$

$$\mathbf{Queue_i} \stackrel{def}{=} Source \underset{\{arrivée\}}{\bowtie} Buffer_i$$



Relations d'Équivalence

Simplification: isomorphisme faible permet déliminer les états intermédiaires dans une séquence d'actions de type τ.

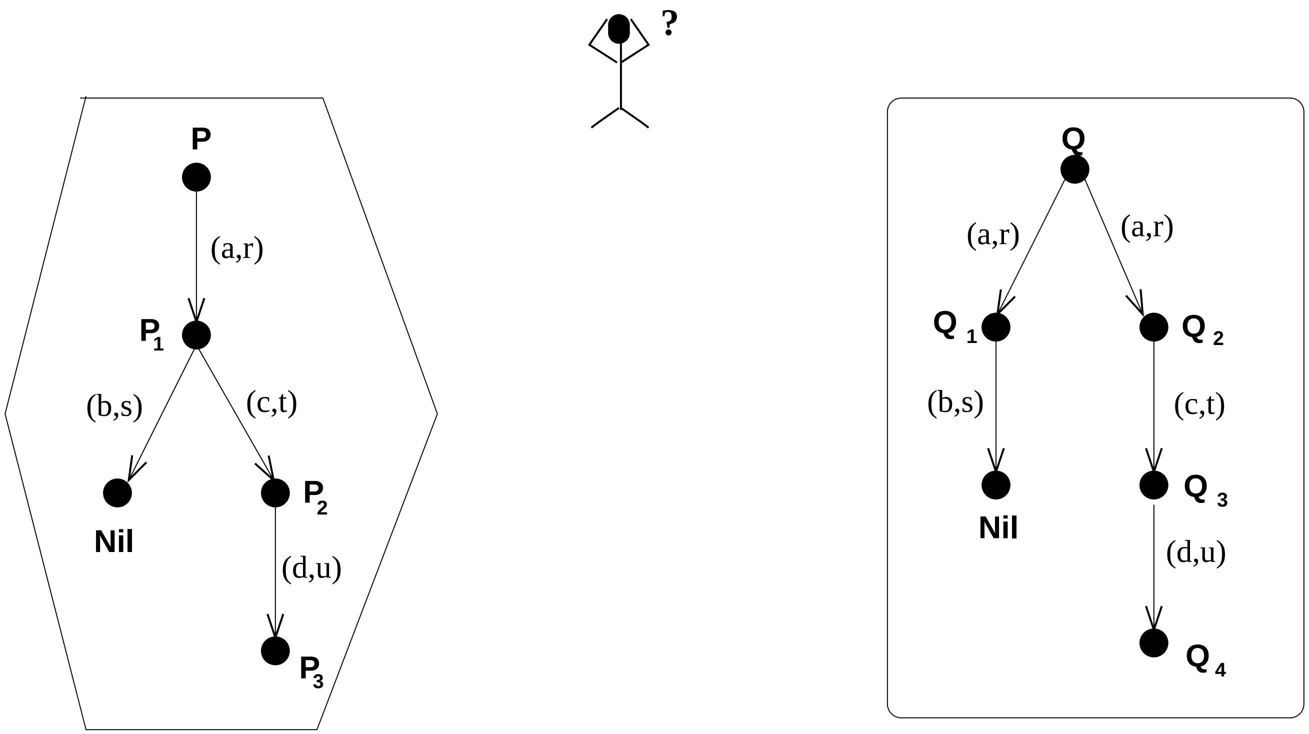
Agrégation: équivalence forte détecte des patterns de comportement qui se répétent et permet de choisir un représentant de chacun de ces patterns.

Ces relations d'équivalences sont des congruences car elles sont conservées par tous les opérateurs du langages.

$\Rightarrow$ les sous-modèles peuvent être simplifiés ou agrégés en isolation et être remplacés dans l'expression du modèle original.

Equivalence Forte

Deux composantes sont fortement équivalentes si elles affichent exactement le même comportement.



Equivalence Forte

Définition : Une relation d'équivalence $\mathcal{R}$ sur les composantes PEPA est une équivalence forte si chaque fois que $(P, Q) \in \mathcal{R}$ alors pour tous les types d'action α et toutes les classes d'équivalence $\mathcal{S}$,

$$q[P, \mathcal{S}, \alpha] = q[Q, \mathcal{S}, \alpha]$$

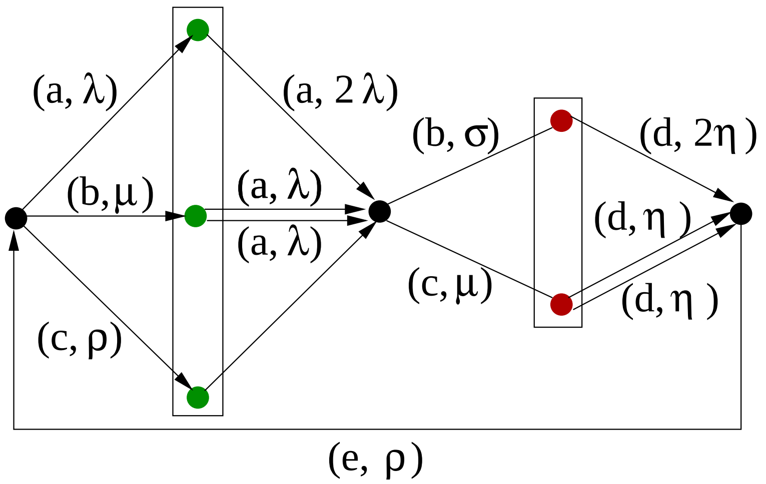
où $q[P, \mathcal{S}, \alpha]$ est le taux de transition conditionel total de P vers $\mathcal{S}$, c'est-à-dire $q[P, \mathcal{S}, \alpha] = \sum_{P' \in \mathcal{S}} q(P, P', \alpha)$.

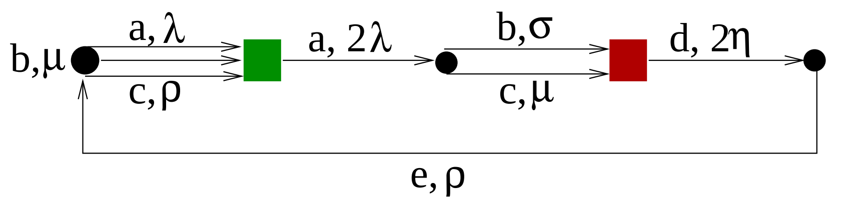
Equivalence Forte pour l'Agrégation

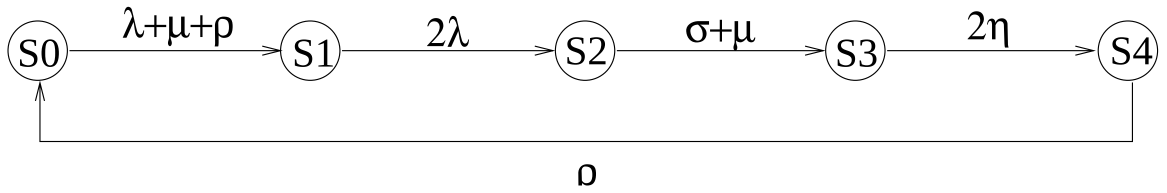
- De même qu'une équivalence peut être établie entre modèles, l'équivalence forte peut être utilisée pour trouver des états équivalents dans un modèle.
  - un partitionnement de l'espace d'états peut être trouvé qui donnerait une partition lumpable sur l'espace d'états du processus de Markov.

MODÈLE PEPA ⟶ GRAPHE DE TRANSITION $\xrightarrow{\text{Équivalence forte}}$ CMTC AGRÉGÉE

Equivalence Forte pour l'Agrégation







- Problème : génération de tout l'espace d'états . . .

Agrégation Compositionnelle

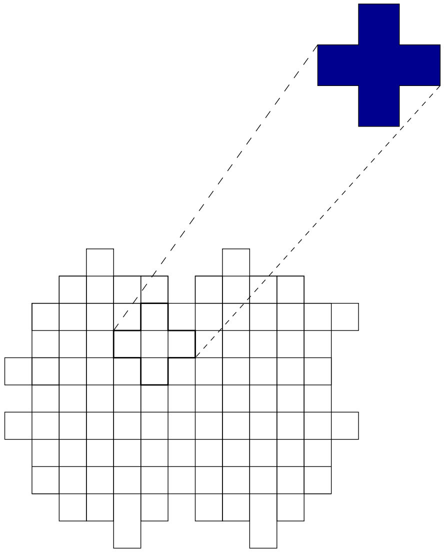
- . . . l'équivalence forte est une congruence

- l'agrégation peut donc être réalisée de manière compositionnelle

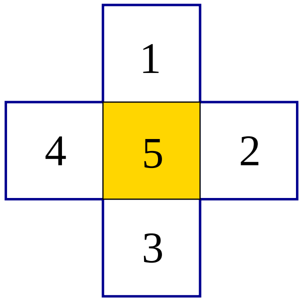
  – dans un modèle complexe, les composantes peuvent être remplacées par des composantes fortement équivalentes (lumpability), le nouveau modèle étant équivalent à l'original, avec probablement un espace d'états réduit.

$$\text{MODÈLE} \xrightarrow{\text{Équivalence forte}} \text{MODÈLE AGRÉGÉ} \longrightarrow \text{GRAPHE DE TRANSITION} \longrightarrow \text{CMTC}$$

Exemple



Le Modèle de Manhattan



Le Cluster

$$Cluster \stackrel{def}{=} (micro_1 \underset{\emptyset}{\bowtie} micro_2 \underset{\emptyset}{\bowtie} micro_3 \underset{\emptyset}{\bowtie} micro_4) \underset{N}{\bowtie} micro_5$$

$$Cluster \stackrel{def}{=} micro_{1234} \underset{N}{\bowtie} micro_5$$

Avantages du Formel

Construction du Modèle : Compositionalité

- construction aisée du modèle
- réutilisabilité des sous-modèles
- compréhension aisée du modèle

Manipulation du Modèle : Relations d'Équivalence

- techniques de réduction de l'espace d'états/réécriture des termes
- techniques d'agrégation basées sur la "lumpability"

Résolution du Modèle : Caractérisation des types "idéals"

- identifier les classes de modèles sensibles à des techniques de résolution efficaces basées sur la structure compositionnelle.

Quelques Applications

- Réseaux cellulaires hiérarchiques
- Systèmes éléctroniques de ventes aux enchères
- Caractérisation du trafic multimédia
- Systèmes de control de robots
- Algorithmes distribués
- . . .

Outils PEPA

Plusieurs outils complémentaires ont été développés à Edimbourgh.

- L'outil principal est le PEPA Workbench qui transforme les modèles PEPA en generator (matrice). Un solveur interne peut être utilisé ou un fichier peut être produit pour un outil externe tel que Maple, Matlab ou Mathematica.

- Il y a un simulateur à événement discret PEPAroni.

- Le PEPA State Finder est utilisé lorsqu'un modèle a été résolu pour identifier les états qui satisfont certaines propriétés et permet de calculer les mesures de performance.

PEPA dans d'autres Outils

Plusieurs autres outils utilisent PEPA comme formalisme de description.

- L'outil de modélisation Möbius développé par le groupe de recherche PERFORM à l'Université d'Illinois.
- PEPA est également utilisé comme langage dans le model-checker PRISM développé à l'Université de Birmingham.
- L'outil DNAmaca d'Imperial College à Londres accepte en entrée les systèmes spécifés en PEPA.

Conclusions

- Les algèbres de processus stochastiques présentent beaucoup d'attraits pour la modélisation des performances et sont de plus en plus populaires dans la communauté d'évaluation de performances.

- Notre expérience avec PEPA est que la combinaison d'une *sémantique formelle* bien définie avec des *outils* permet une utilisation efficace du formalisme.