

Modélisation, Simulation et Vérification des Grands Réseaux de Régulation Biologique

Soutenance de thèse de doctorat - 6 octobre 2011

Loïc Paulevé

École Centrale de Nantes / IRCCyN (équipe MeForBio)



Rapporteurs

François Fages
Serge Haddad

Directeur de recherche à l'INRIA Paris-Rocquencourt
Professeur de l'École Normale Supérieure de Cachan/LSV

Examineurs

Jérôme Feret
Claude Jard
Jean-Phillipe Vert
Olivier Roux
Morgan Magnin

Chargé de recherche à l'INRIA Paris-Rocquencourt/École Normale Supérieure
Professeur de l'École Normale Supérieure de Cachan, Antenne de Bretagne/IRISA
Directeur du Centre de Bio-informatique à Mines ParisTech
Professeur de l'École Centrale de Nantes/IRCCyN
Maître de conférences à l'École Centrale de Nantes/IRCCyN

Biologie + Informatique ?

La science informatique pour la biologie systémique

But : comprendre, modifier, contrôler (guérir) le vivant



Hypothèses
Intuitions

Connaissances
(expériences)



Modèle du
système

comportements
dynamiques

Aide à la
réflexion



Biologie + Informatique ?

La science informatique pour la biologie systémique

But : comprendre, modifier, contrôler (guérir) le vivant



Hypothèses
Intuitions

Connaissances
(expériences)



Modèle du
système raffiné

comportements
dynamiques

Aide à la
réflexion



Biologie + Informatique ?

La science informatique pour la biologie systémique

But : comprendre, modifier, contrôler (guérir) le vivant



Hypothèses
Intuitions

Connaissances
(expériences)



Aide à la
réflexion



Modèle du
système raffiné

comportements
dynamiques



Raisonnements
automatiques

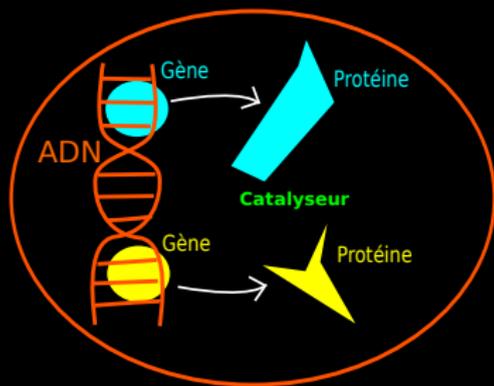


But : concevoir et prouver des méthodes de calcul

Défi : contenir la complexité des calculs
⇒ nécessaire pour l'analyse de grands modèles

Les Réseaux de Régulation Biologique (RRB)

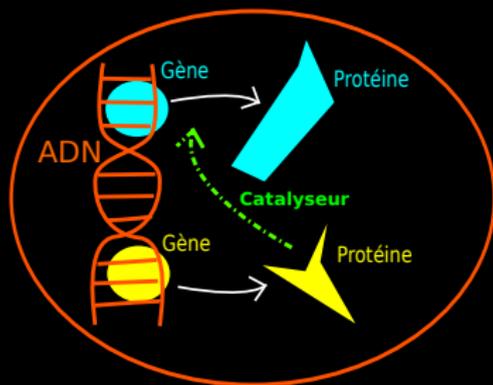
Exemple : la régulation génique



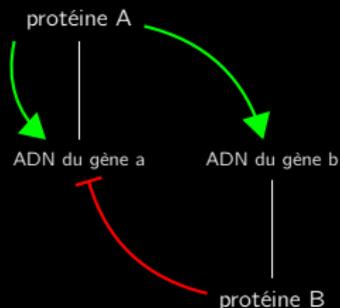
- Le comportement d'une cellule est déterminé par la concentration de protéines particulières.

Les Réseaux de Régulation Biologique (RRB)

Exemple : la régulation génique

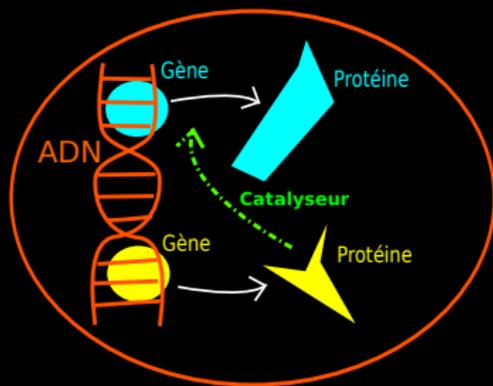


- Le comportement d'une cellule est déterminé par la concentration de protéines particulières.
- Certaines protéines **activent** ou **inhibent** l'expression d'un gène.

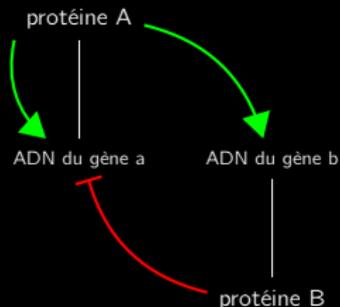


Les Réseaux de Régulation Biologique (RRB)

Exemple : la régulation génique



- Le comportement d'une cellule est déterminé par la concentration de protéines particulières.
- Certaines protéines **activent** ou **inhibent** l'expression d'un gène.



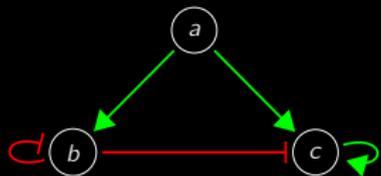
Graphe des interactions



Réseaux discrets

- Chaque composant possède un nombre fini de **niveaux qualitatifs** ($\{0, 1, 2\}$).
- Fonctions donnant le **niveau suivant selon l'état des régulateurs**.

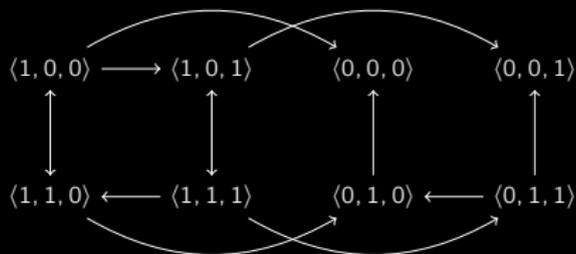
Exemple booléen :



$$f^a(x) = 0$$

$$+K \quad f^b(x) = x[a] \wedge \neg x[b]$$

$$f^c(x) = \neg x[b] \wedge (x[a] \vee x[c])$$



[René Thomas dans *Journal of Theoretical Biology*, 1973]

[Richard, Comet, Bernot dans *Modern Formal Methods and App.*, 2006]

Modélisations hybrides

Caractéristiques continues dirigent les transitions discrètes

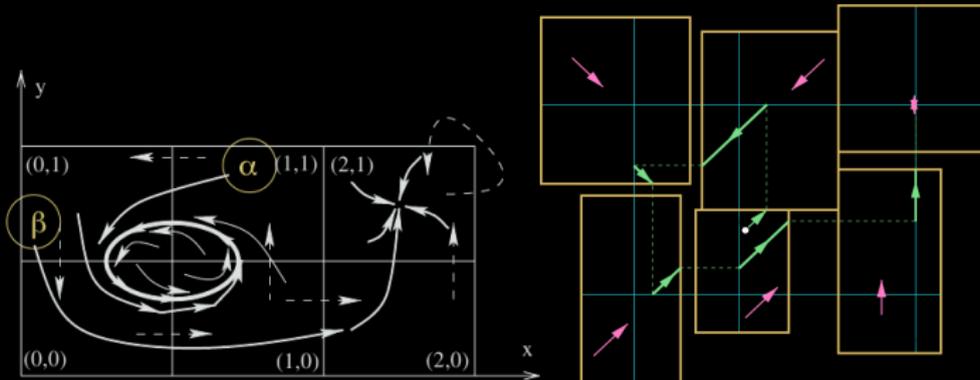
Introduction de délais pour les actions

Modèles stochastiques

- Les délais sont des **variables aléatoires** (généralement exponentielles).
- \Rightarrow Associer des probabilités aux comportements.

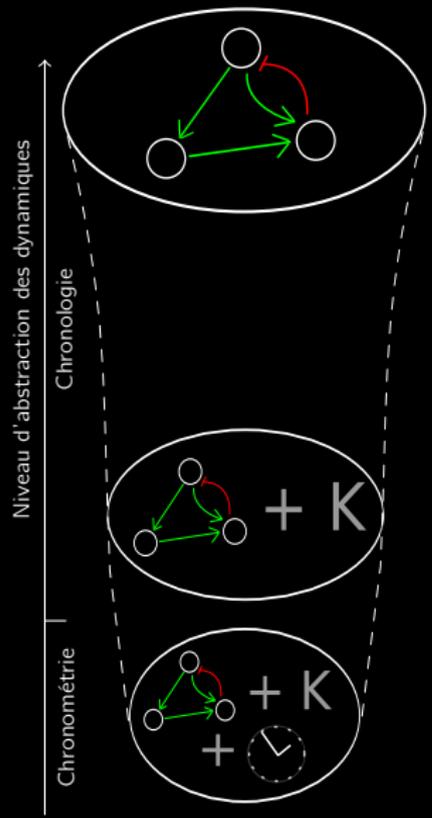
Réseaux de Petri stochastiques / π -calcul, etc. [Heiner, Regev, Priami, Phillips, etc.]

Modèles temporisés

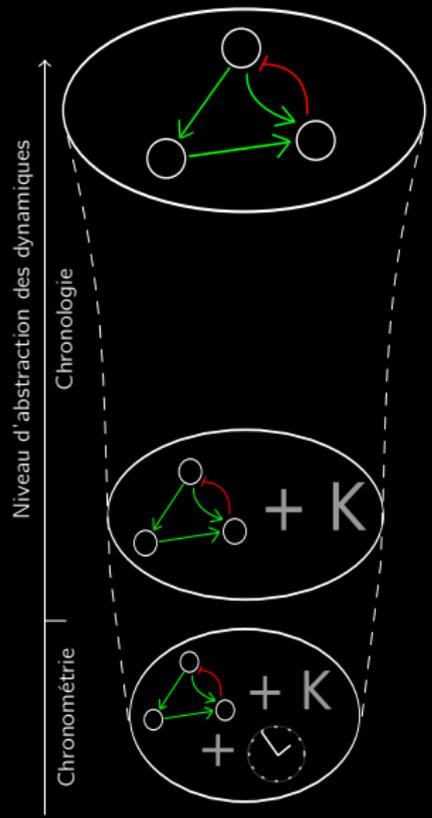


Automates temporels / hybrides [Ahmad, Roux, Batt, Bockmayr, etc.]

État de l'art



État de l'art



Propriétés générales : Bornes sur le nb d'attracteurs ; fonctionnalités ; ...

Points fixes (états stables)

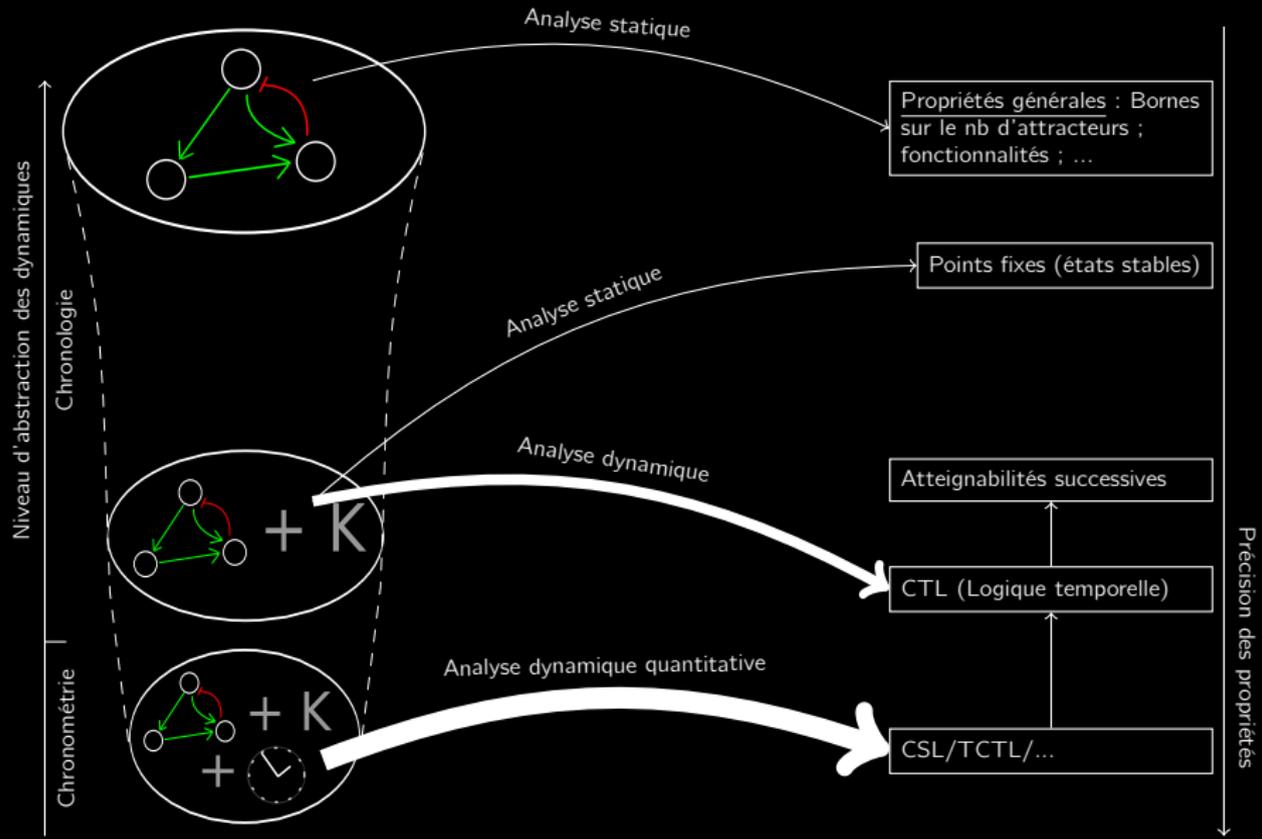
Atteignabilités successives

CTL (Logique temporelle)

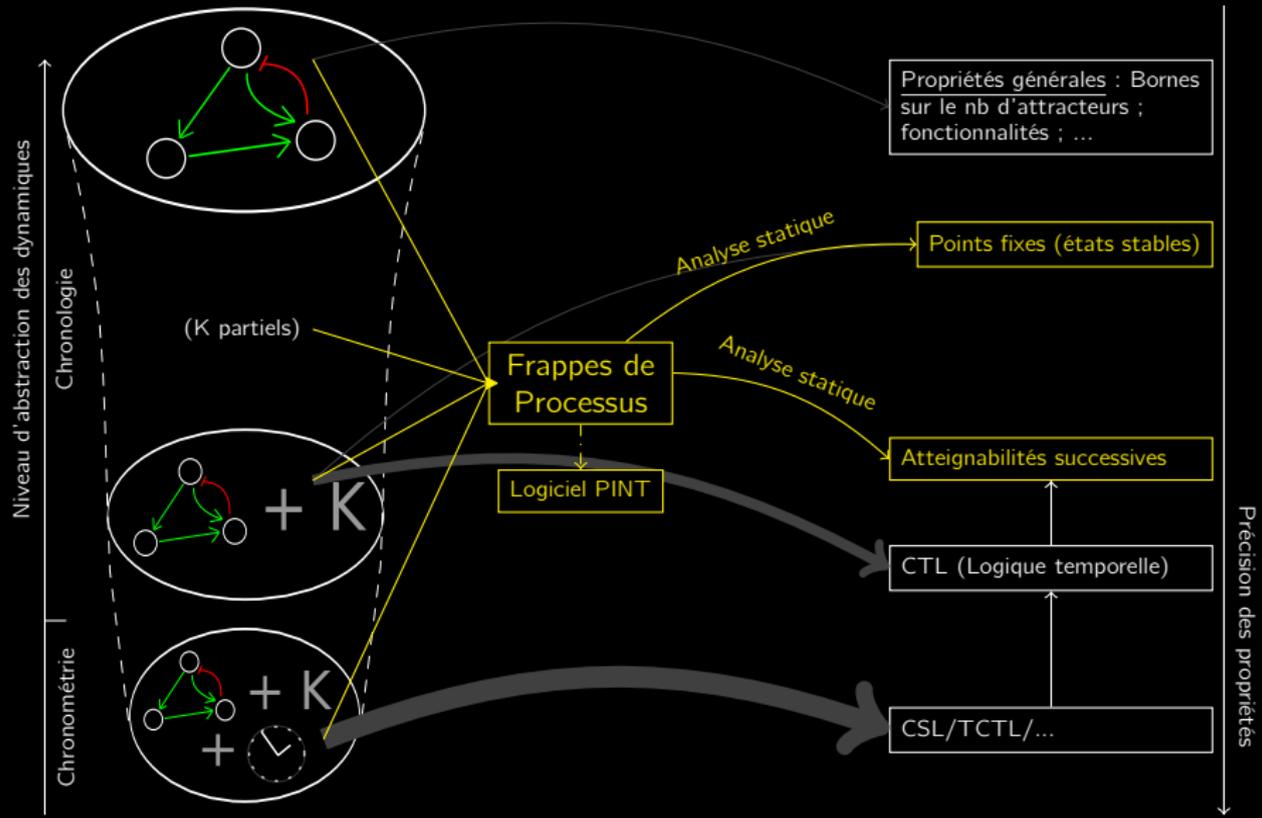
CSL/TCTL/...

Precision des propriétés

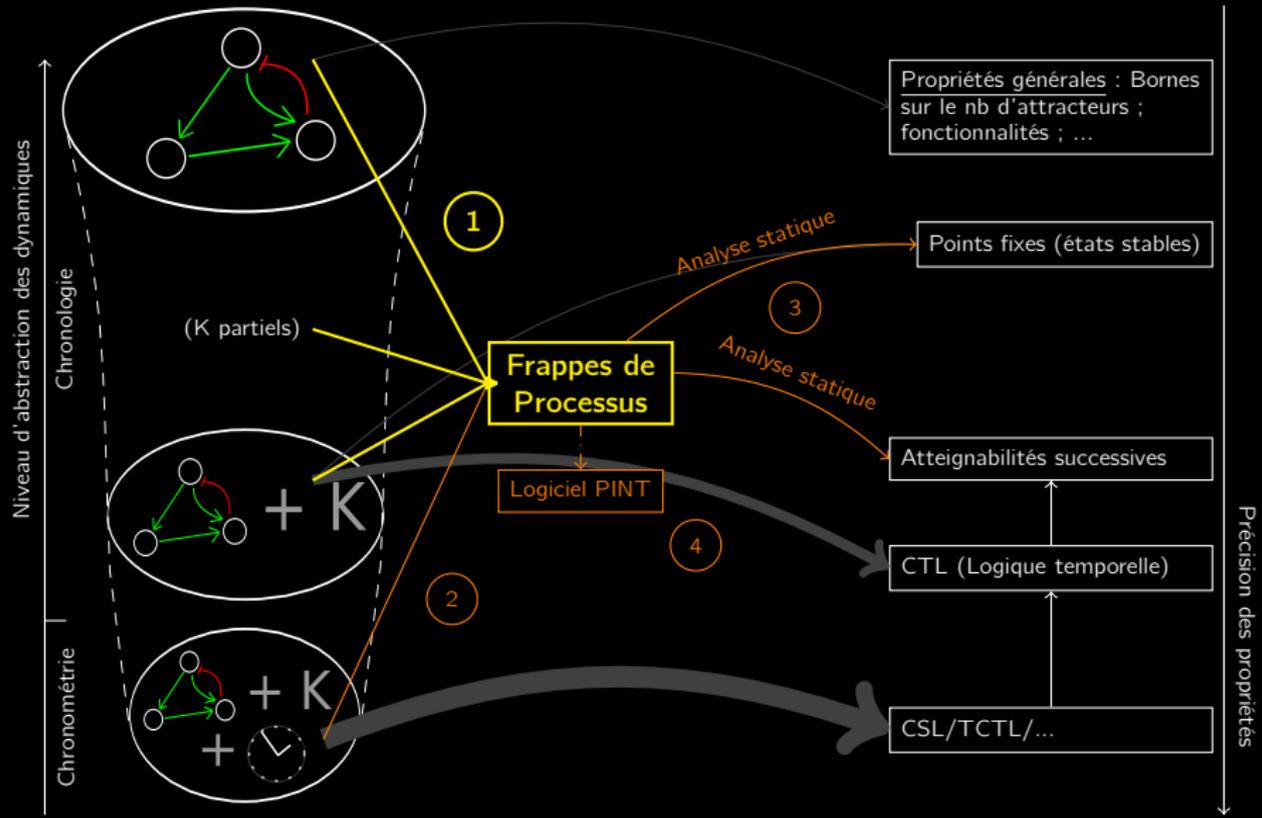
État de l'art



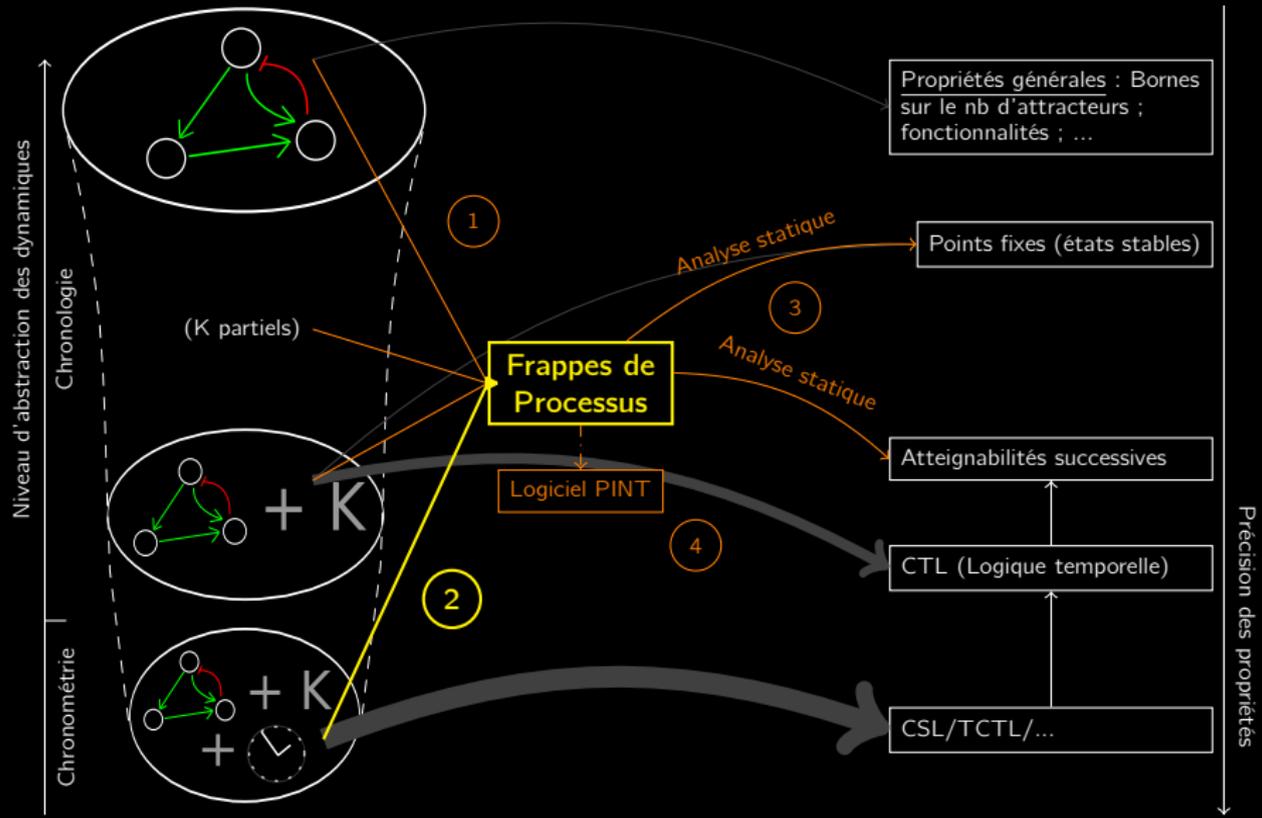
Contributions



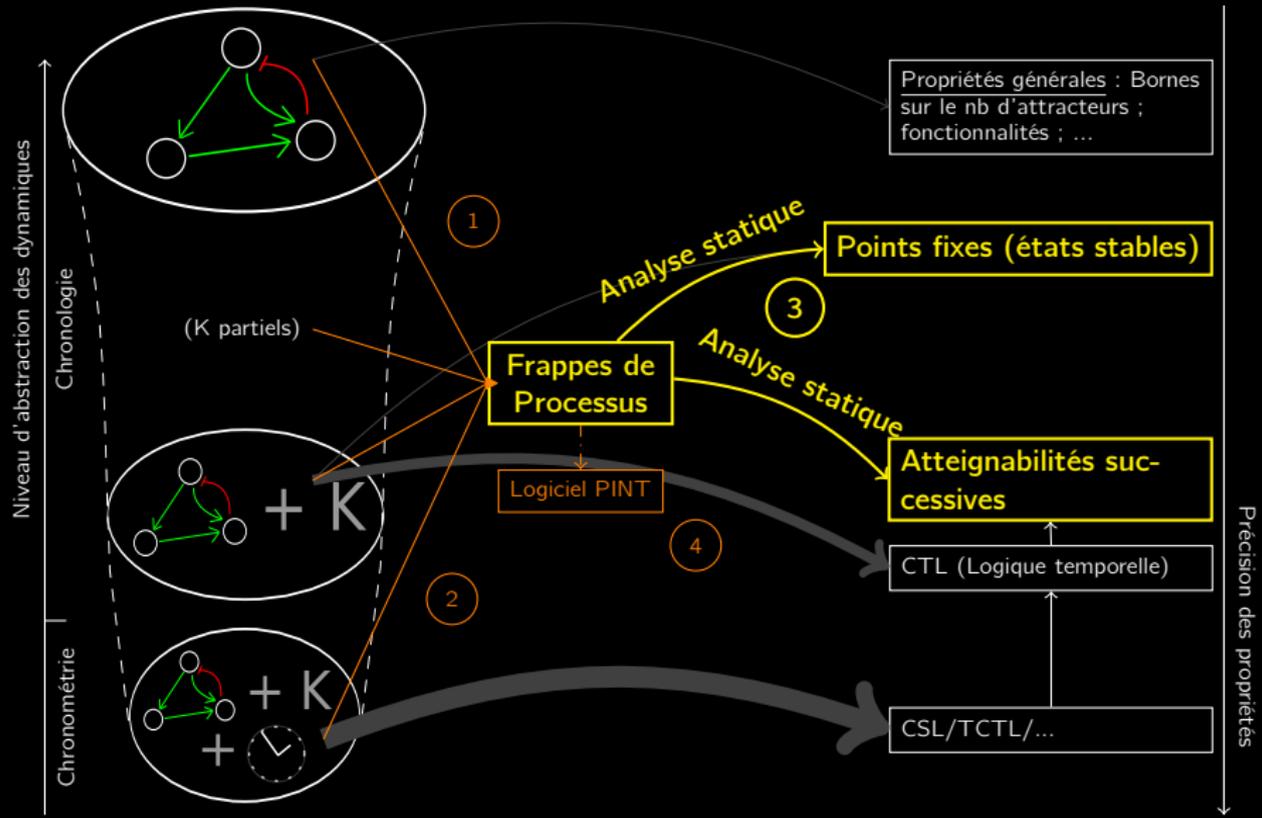
Plan



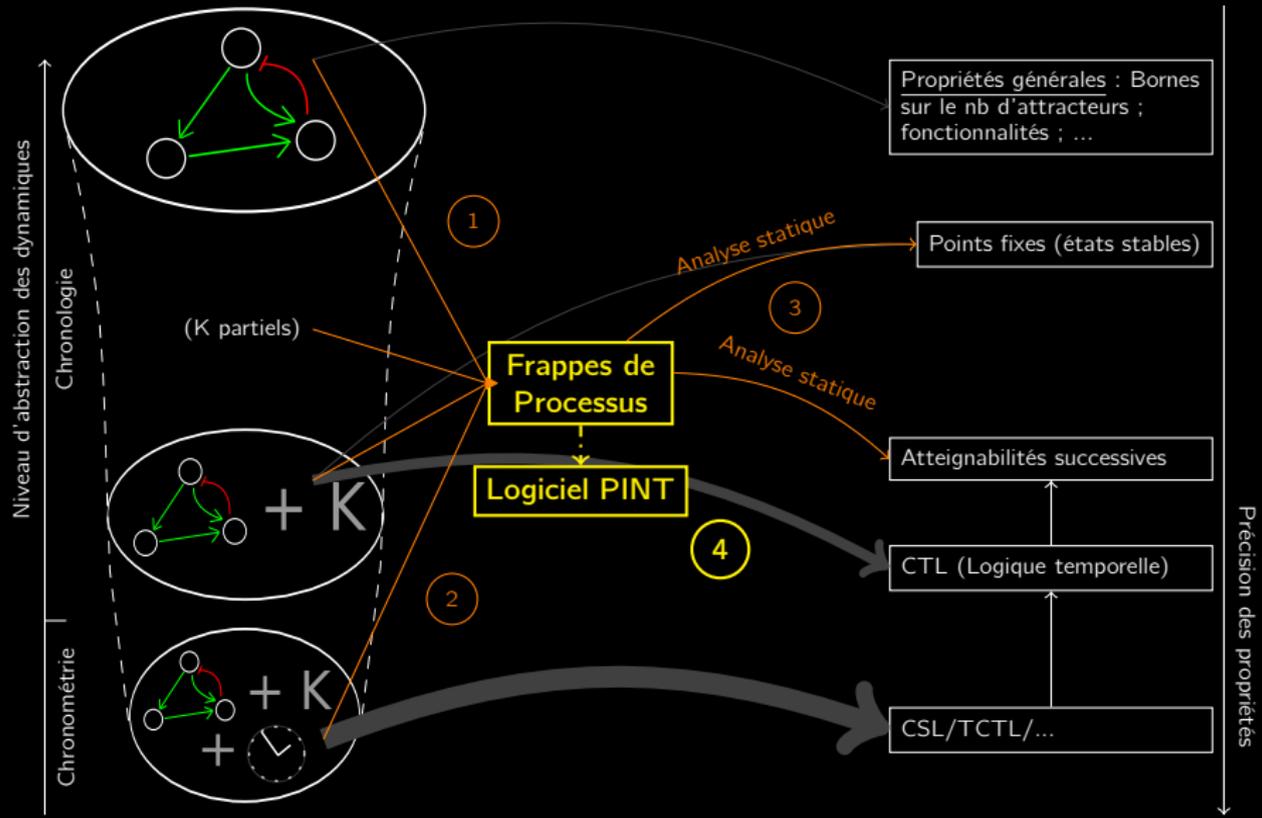
Plan



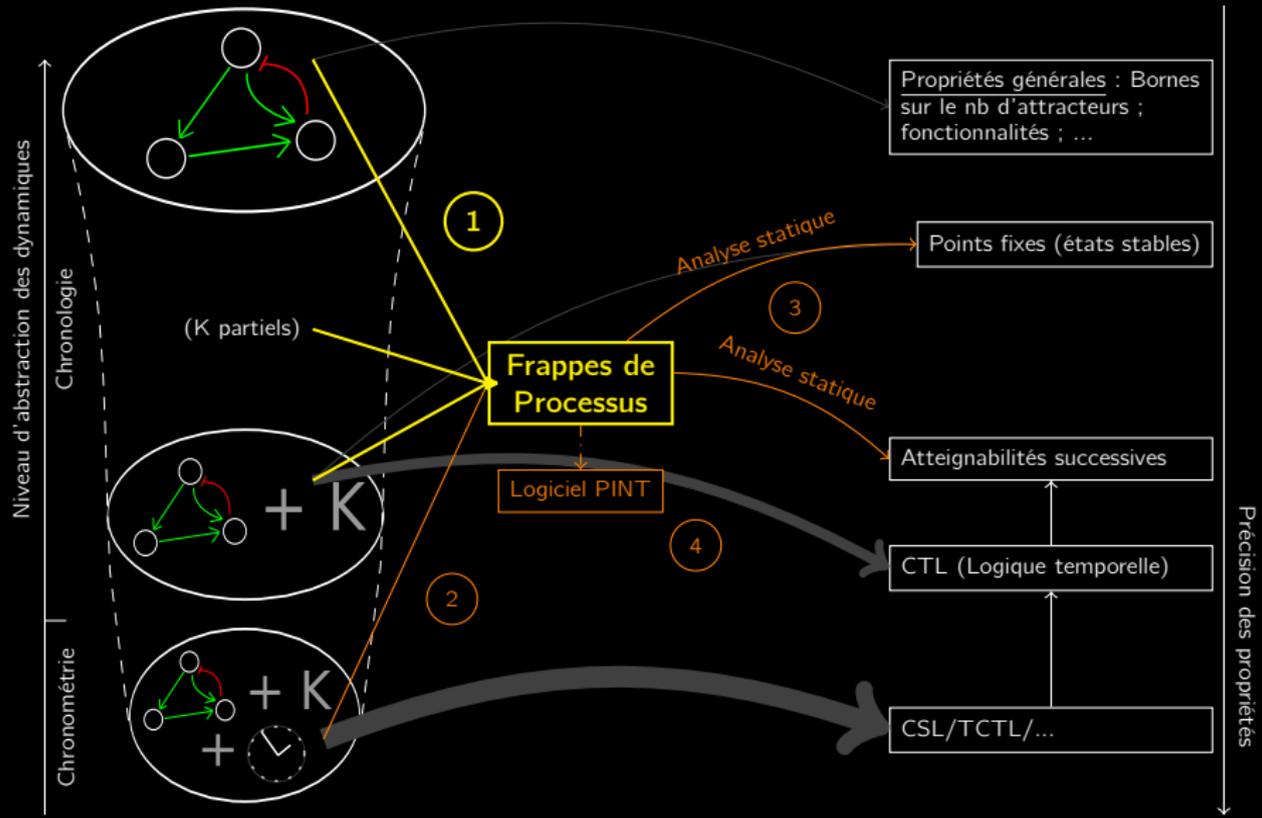
Plan



Plan

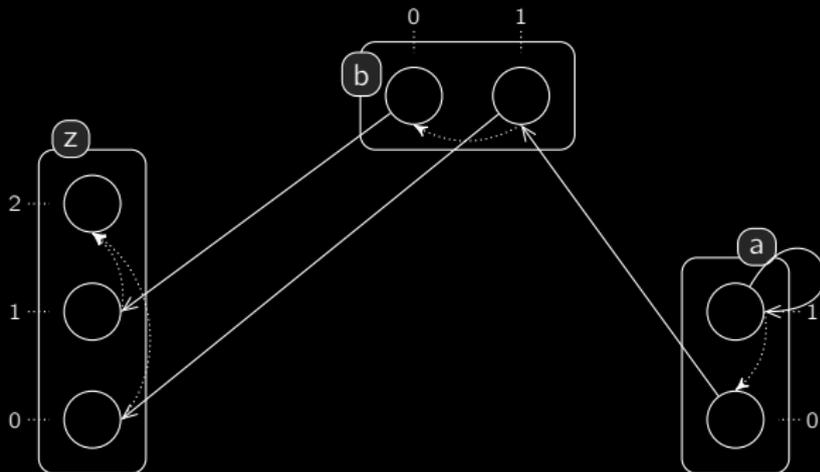


Plan



Les Frappes de Processus

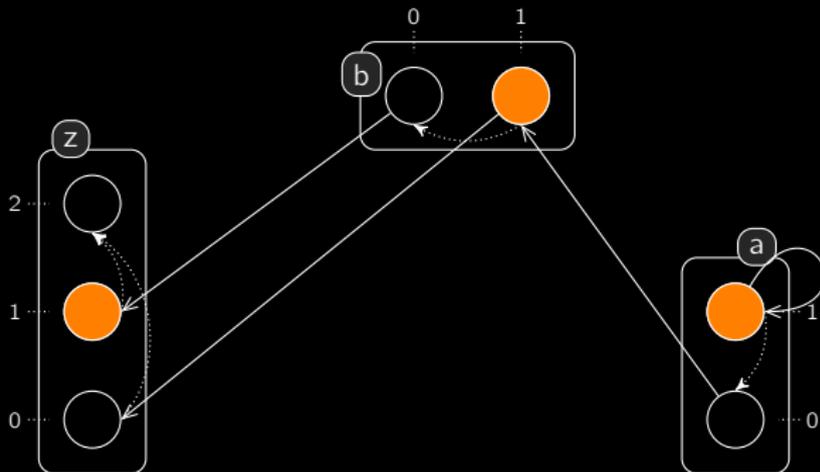
[Paulevé, Magnin, Roux dans TCSB 2011]



- **Sortes** : a,b,z ; **Processus** : $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- **Actions** : a_0 frappe b_1 pour le faire bondir en b_0, \dots ;
- **États** : $\langle a_1, b_1, z_2 \rangle, \langle a_0, b_1, z_1 \rangle, \langle a_0, b_0, z_1 \rangle, \dots$;
- Restriction des Automates Finis Communicants (entre autres).

Les Frappes de Processus

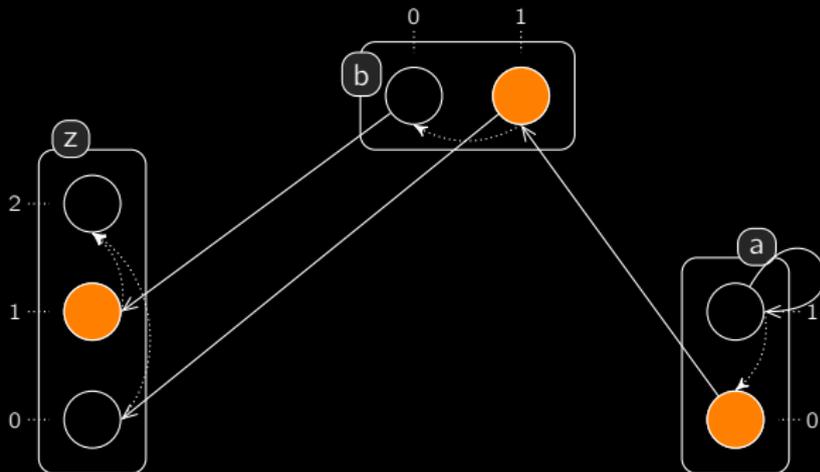
[Paulevé, Magnin, Roux dans TCSB 2011]



- **Sortes** : a, b, z ; **Processus** : $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- **Actions** : a_0 frappe b_1 pour le faire bondir en b_0, \dots ;
- **États** : $\langle a_1, b_1, z_2 \rangle, \langle a_0, b_1, z_1 \rangle, \langle a_0, b_0, z_1 \rangle, \dots$;
- Restriction des Automates Finis Communicants (entre autres).

Les Frappes de Processus

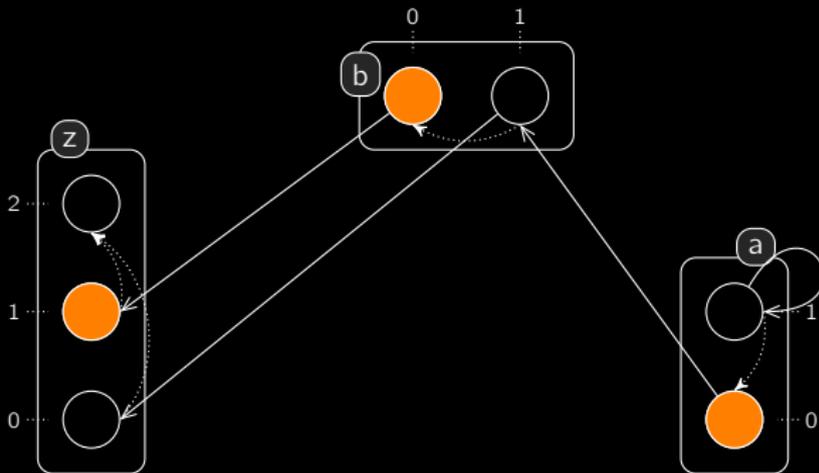
[Paulevé, Magnin, Roux dans TCSB 2011]



- **Sortes** : a, b, z ; **Processus** : $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- **Actions** : a_0 frappe b_1 pour le faire bondir en b_0, \dots ;
- **États** : $\langle a_1, b_1, z_2 \rangle, \langle a_0, b_1, z_1 \rangle, \langle a_0, b_0, z_1 \rangle, \dots$;
- Restriction des Automates Finis Communicants (entre autres).

Les Frappes de Processus

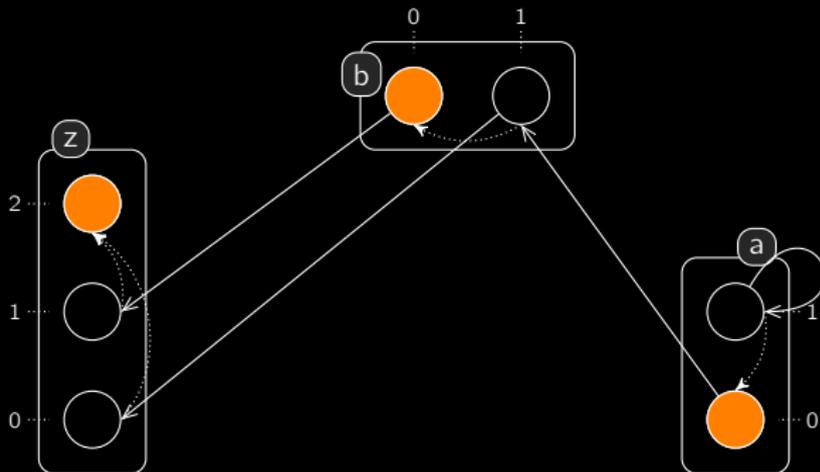
[Paulevé, Magnin, Roux dans TCSB 2011]



- **Sortes** : a, b, z ; **Processus** : $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- **Actions** : a_0 frappe b_1 pour le faire bondir en b_0, \dots ;
- **États** : $\langle a_1, b_1, z_2 \rangle, \langle a_0, b_1, z_1 \rangle, \langle a_0, b_0, z_1 \rangle, \dots$;
- Restriction des Automates Finis Communicants (entre autres).

Les Frappes de Processus

[Paulevé, Magnin, Roux dans TCSB 2011]

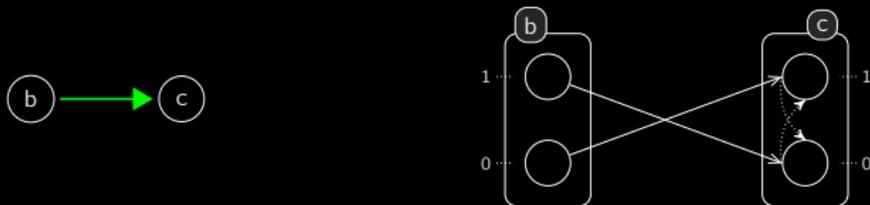


- **Sortes** : a, b, z ; **Processus** : $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- **Actions** : a_0 frappe b_1 pour le faire bondir en b_0, \dots ;
- **États** : $\langle a_1, b_1, z_2 \rangle, \langle a_0, b_1, z_1 \rangle, \langle a_0, b_0, z_1 \rangle, \dots$;
- Restriction des Automates Finis Communicants (entre autres).

Dynamique généralisée des RRB

- Idée : dynamique **la plus permissive** [Paulevé, Magnin, Roux dans TCSB 2011].
- **Pas besoin des fonctions précises** entre les composants.

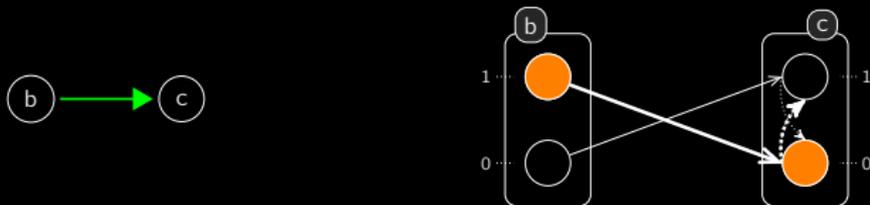
Illustration dans le cas booléen :



Dynamique généralisée des RRB

- Idée : dynamique **la plus permissive** [Paulevé, Magnin, Roux dans TCSB 2011].
- **Pas besoin des fonctions précises** entre les composants.

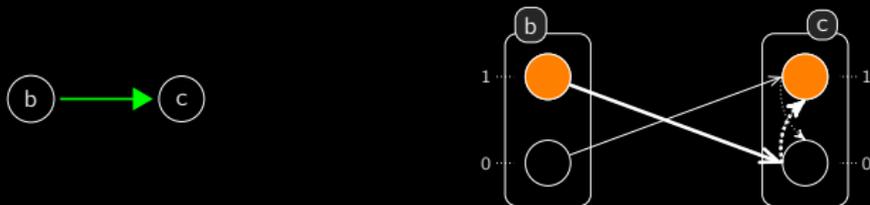
Illustration dans le cas booléen :



Dynamique généralisée des RRB

- Idée : dynamique **la plus permissive** [Paulevé, Magnin, Roux dans TCSB 2011].
- **Pas besoin des fonctions précises** entre les composants.

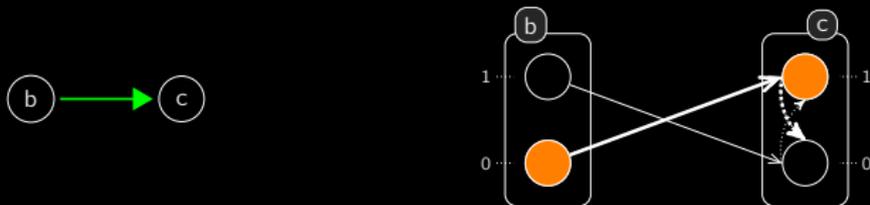
Illustration dans le cas booléen :



Dynamique généralisée des RRB

- Idée : dynamique **la plus permissive** [Paulevé, Magnin, Roux dans TCSB 2011].
- **Pas besoin des fonctions précises** entre les composants.

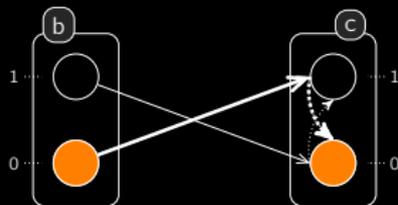
Illustration dans le cas booléen :



Dynamique généralisée des RRB

- Idée : dynamique **la plus permissive** [Paulevé, Magnin, Roux dans TCSB 2011].
- **Pas besoin des fonctions précises** entre les composants.

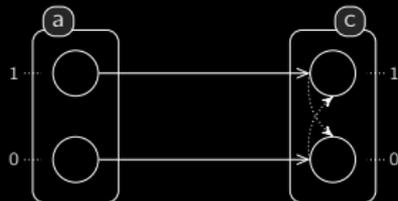
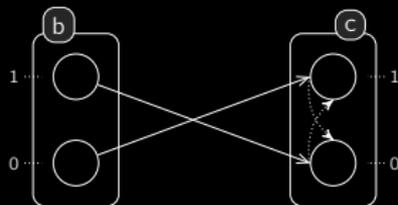
Illustration dans le cas booléen :



Dynamique généralisée des RRB

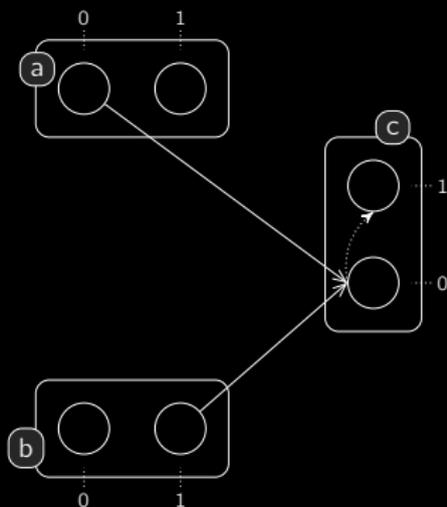
- Idée : dynamique **la plus permissive** [Paulevé, Magnin, Roux dans TCSB 2011].
- **Pas besoin des fonctions précises** entre les composants.

Illustration dans le cas booléen :



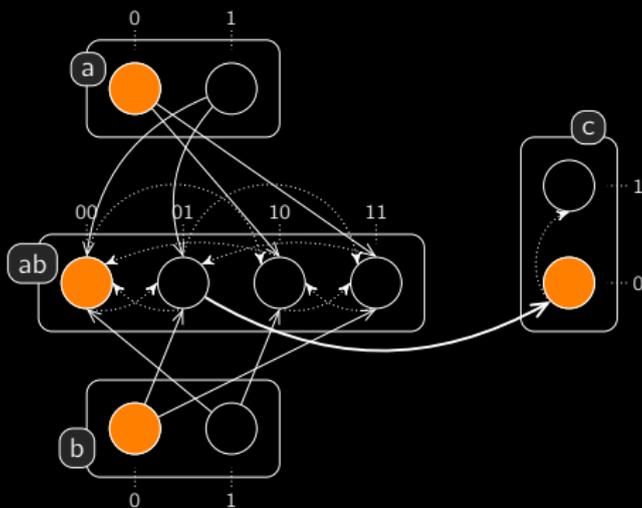
Raffinement par coopération

- Idée : $c_0 \rightarrow c_1$ quand a_0 et b_1 sont présents.
- Introduction d'une **sorte coopérative** reflétant l'état des sorties a et b .



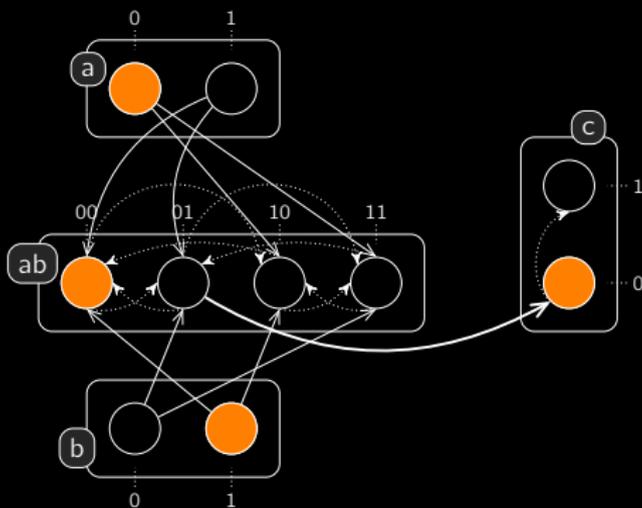
Raffinement par coopération

- Idée : $c_0 \rightarrow c_1$ quand a_0 et b_1 sont présents.
- Introduction d'une **sorte coopérative** reflétant l'état des sorties a et b .



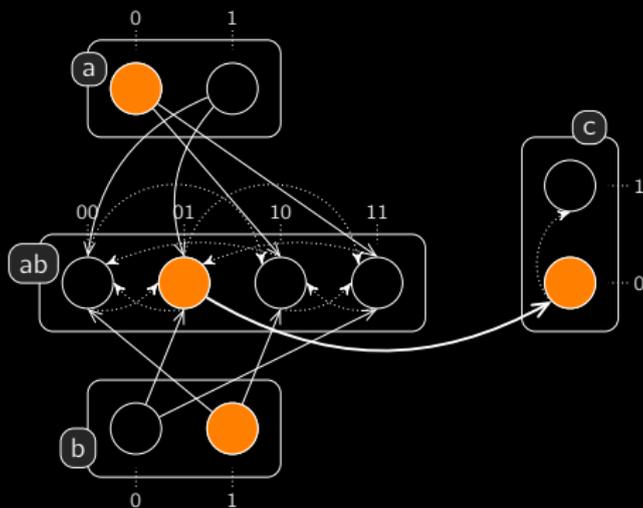
Raffinement par coopération

- Idée : $c_0 \rightarrow c_1$ quand a_0 et b_1 sont présents.
- Introduction d'une **sorte coopérative** reflétant l'état des sorties a et b .



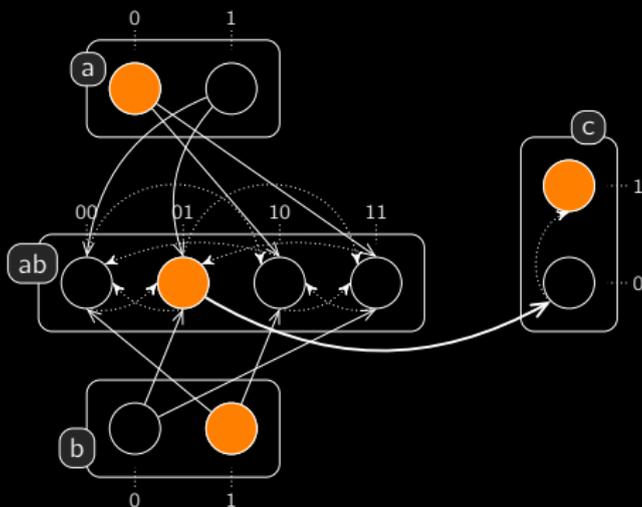
Raffinement par coopération

- Idée : $c_0 \rightarrow c_1$ quand a_0 et b_1 sont présents.
- Introduction d'une **sorte coopérative** reflétant l'état des sorties a et b .



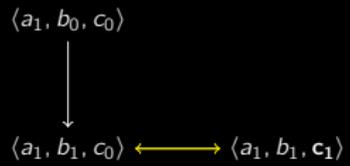
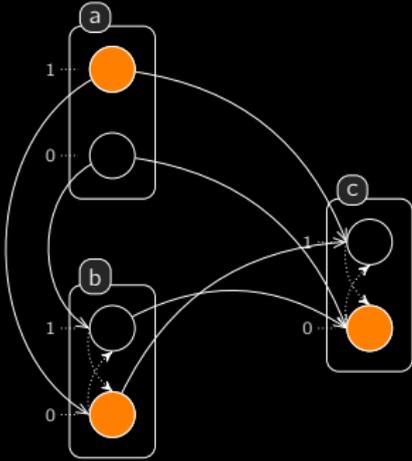
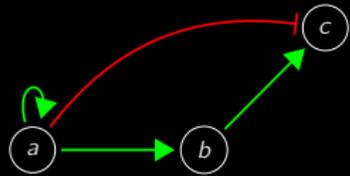
Raffinement par coopération

- Idée : $c_0 \rightarrow c_1$ quand a_0 et b_1 sont présents.
- Introduction d'une **sorte coopérative** reflétant l'état des sorties a et b .

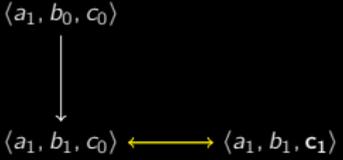
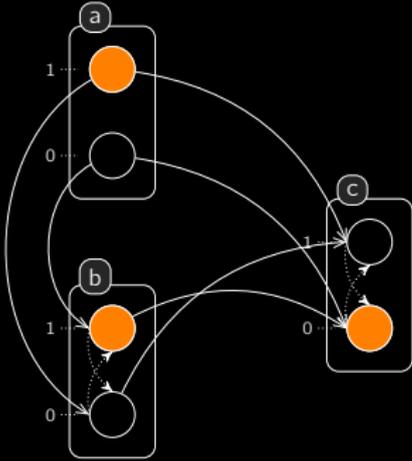
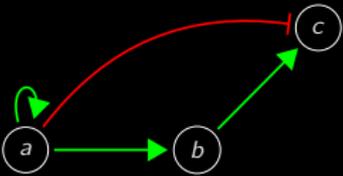


⇒ introduit un décalage temporel ; notion **proche d'un complexe**.

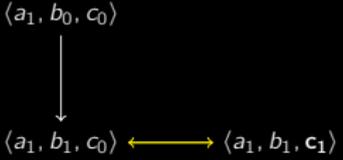
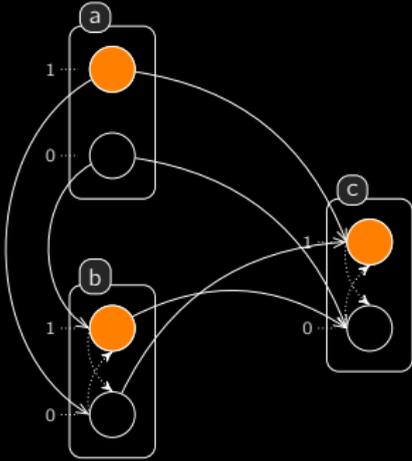
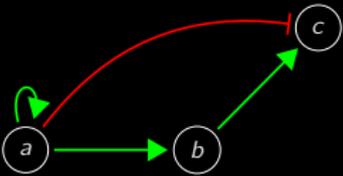
Exemple jouet



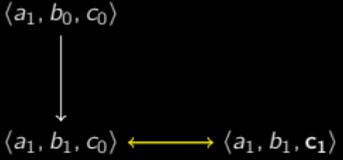
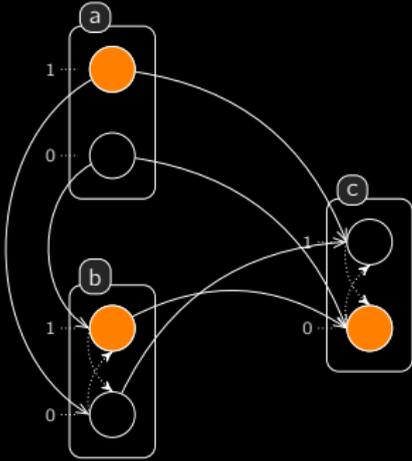
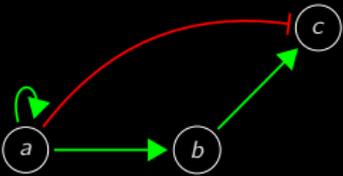
Exemple jouet



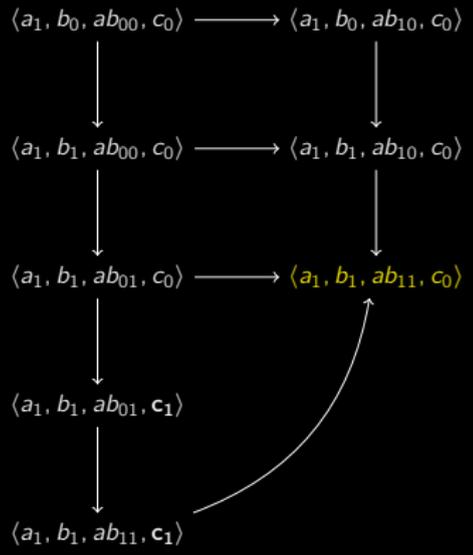
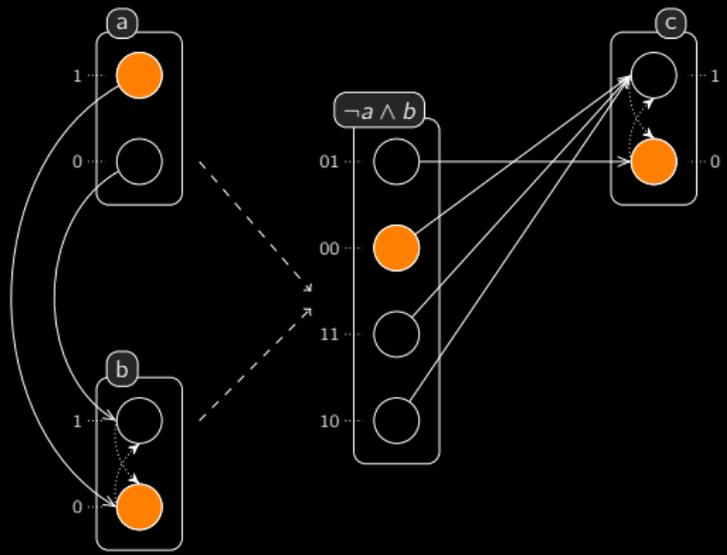
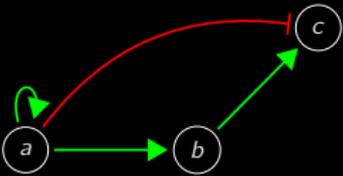
Exemple jouet



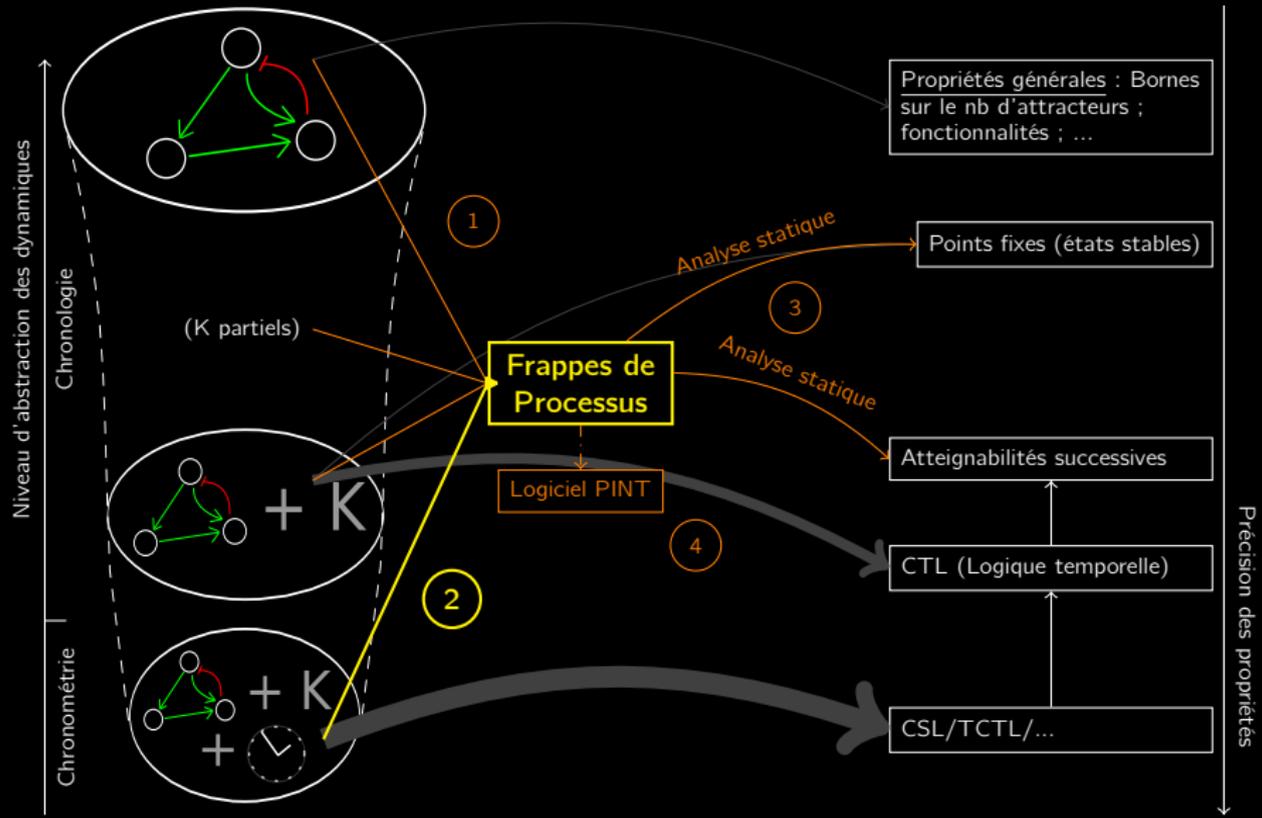
Exemple jouet



Exemple jouet



Plan



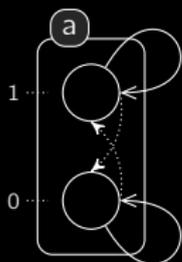
Introduction du temps et de l'aléatoire

[Paulevé, Magnin, Roux dans IEEE TSE, 2010]

But : **précision temporelle réglable** dans un cadre stochastique.

Paramétrage d'une action

- Taux r + **facteur d'absorption de stochasticité** sa .
- Durée de l'action \approx **somme de** sa exponentielles avec un **taux** $r.sa$.
- Durée moyenne : r^{-1} ; **variance** : $r^{-2}sa^{-1}$ (distribution d'Erlang).



(effectué dans le cadre général du π -calcul stochastique).

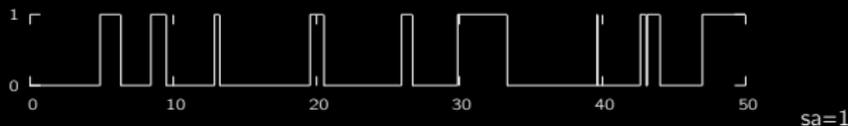
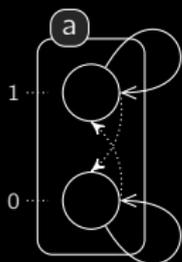
Introduction du temps et de l'aléatoire

[Paulevé, Magnin, Roux dans IEEE TSE, 2010]

But : **précision temporelle réglable** dans un cadre stochastique.

Paramétrage d'une action

- Taux r + **facteur d'absorption de stochasticité** sa .
- Durée de l'action \approx **somme de** sa exponentielles avec un **taux** $r.sa$.
- Durée moyenne : r^{-1} ; **variance** : $r^{-2}sa^{-1}$ (distribution d'Erlang).



(effectué dans le cadre général du π -calcul stochastique).

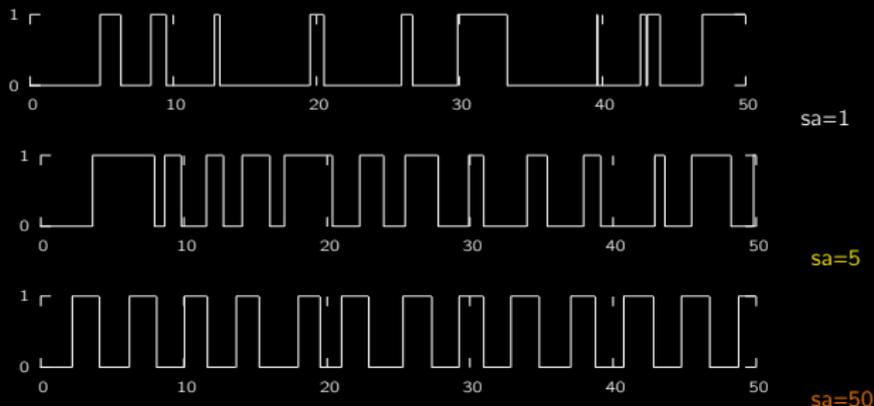
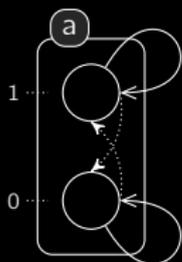
Introduction du temps et de l'aléatoire

[Paulevé, Magnin, Roux dans IEEE TSE, 2010]

But : **précision temporelle réglable** dans un cadre stochastique.

Paramétrage d'une action

- Taux r + **facteur d'absorption de stochasticité** sa .
- Durée de l'action \approx **somme de sa exponentielles** avec un **taux $r.sa$** .
- Durée moyenne : r^{-1} ; **variance** : $r^{-2}sa^{-1}$ (distribution d'Erlang).

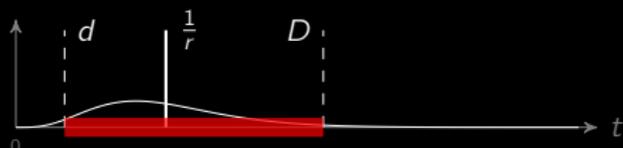


(effectué dans le cadre général du π -calcul stochastique).

Paramètres temporels et stochastiques

[Paulevé, Magnin, Roux dans IEEE TSE, 2010]

- Paramètres : soit (r, sa) , soit l'intervalle de tir $[d; D]$.
- Intervalle de confiance de la durée (à un niveau de confiance voulu).
- Nous avons des estimateurs pour traduire $[d; D]$ vers (r, sa) .

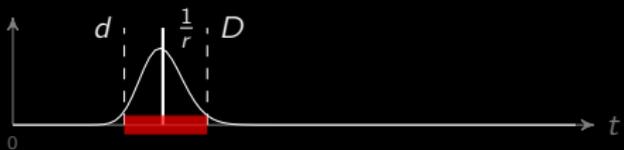


— durée de l'action

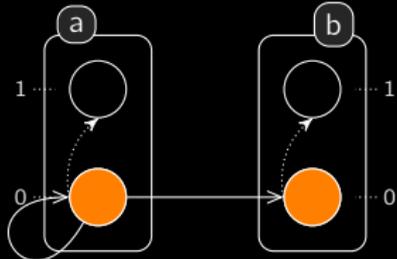
Paramètres temporels et stochastiques

[Paulevé, Magnin, Roux dans IEEE TSE, 2010]

- Paramètres : soit (r, sa) , soit l'intervalle de tir $[d; D]$.
- Intervalle de confiance de la durée (à un niveau de confiance voulu).
- Nous avons des estimateurs pour traduire $[d; D]$ vers (r, sa) .



durée de l'action



$\Rightarrow b_1$ atteint avec une probabilité très faible.

Simulation et vérification formelle

Simulation

- Simulation non-markovienne.
- Machine abstraite générique pour les calculs de processus stochastique [Paulevé, Youssef, Lakin, Phillips à CMSB 2010] :
- instanciation directe pour les Frappes de Processus.

Simulation et vérification formelle

Simulation

- Simulation non-markovienne.
- Machine abstraite générique pour les calculs de processus stochastique [Paulevé, Youssef, Lakin, Phillips à CMSB 2010] :
- instantiation directe pour les Frappes de Processus.

Vérification formelle quantitative

- Traduction depuis le π -calcul stochastique distribué selon Erlang vers PRISM [Paulevé, Magnin, Roux dans IEEE TSE, 2010].
- S'applique aux Frappes de Processus.
- Explosion combinatoire avec des grands facteurs d'absorption de stochasticité ;
- mais un espoir existe dans différentes méthodes de réductions/abstractions, etc.

Simulation et vérification formelle

Simulation

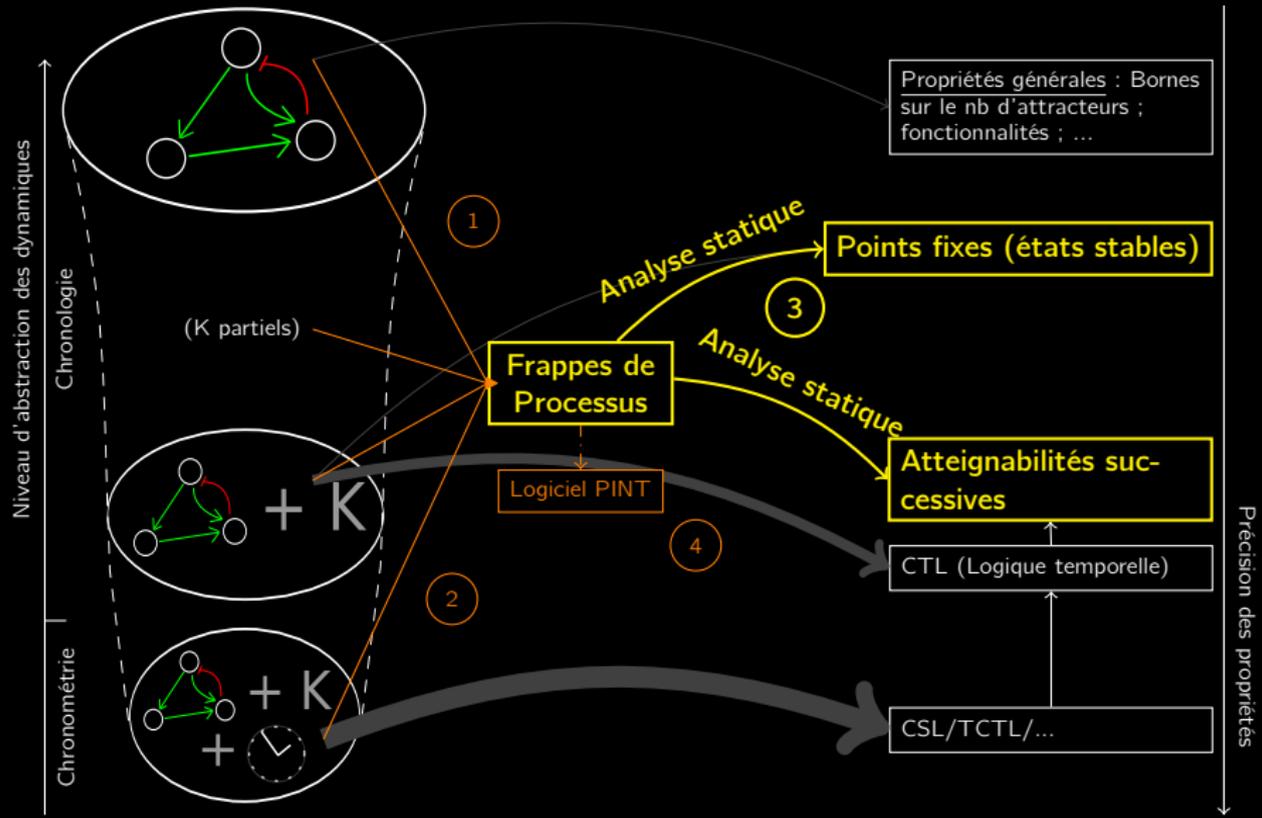
- Simulation non-markovienne.
- Machine abstraite générique pour les calculs de processus stochastique [Paulevé, Youssef, Lakin, Phillips à CMSB 2010] :
- instantiation directe pour les Frappes de Processus.

Vérification formelle quantitative

- Traduction depuis le π -calcul stochastique distribué selon Erlang vers PRISM [Paulevé, Magnin, Roux dans IEEE TSE, 2010].
- S'applique aux Frappes de Processus.
- Explosion combinatoire avec des grands facteurs d'absorption de stochasticité ;
- mais un espoir existe dans différentes méthodes de réductions/abstractions, etc.

Peut-on analyser efficacement des Frappes de Processus ?

Plan



Analyse statique des Frappes de Processus

Intuition

- Simplicité des Frappes de Processus \Rightarrow modèles avec **structures simples**.
- **Dérivation statique efficace** de propriétés dynamiques.

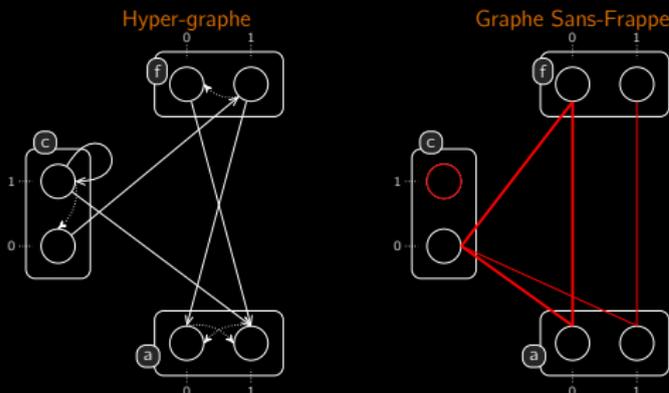
Analyse statique des Frappes de Processus

Intuition

- Simplicité des Frappes de Processus \Rightarrow modèles avec **structures simples**.
- **Dérivation statique efficace** de propriétés dynamiques.

Points Fixes

- Réduction aux n -cliques dans un graphe n -parti.
- Résultat connexe : **Points fixes topologiques** des Réseaux Booléens [Paulevé, Richard dans CRAS 2010].



Analyse statique des Frappes de Processus

Intuition

- Simplicité des Frappes de Processus \Rightarrow modèles avec **structures simples**.
- **Dérivation statique efficace** de propriétés dynamiques.

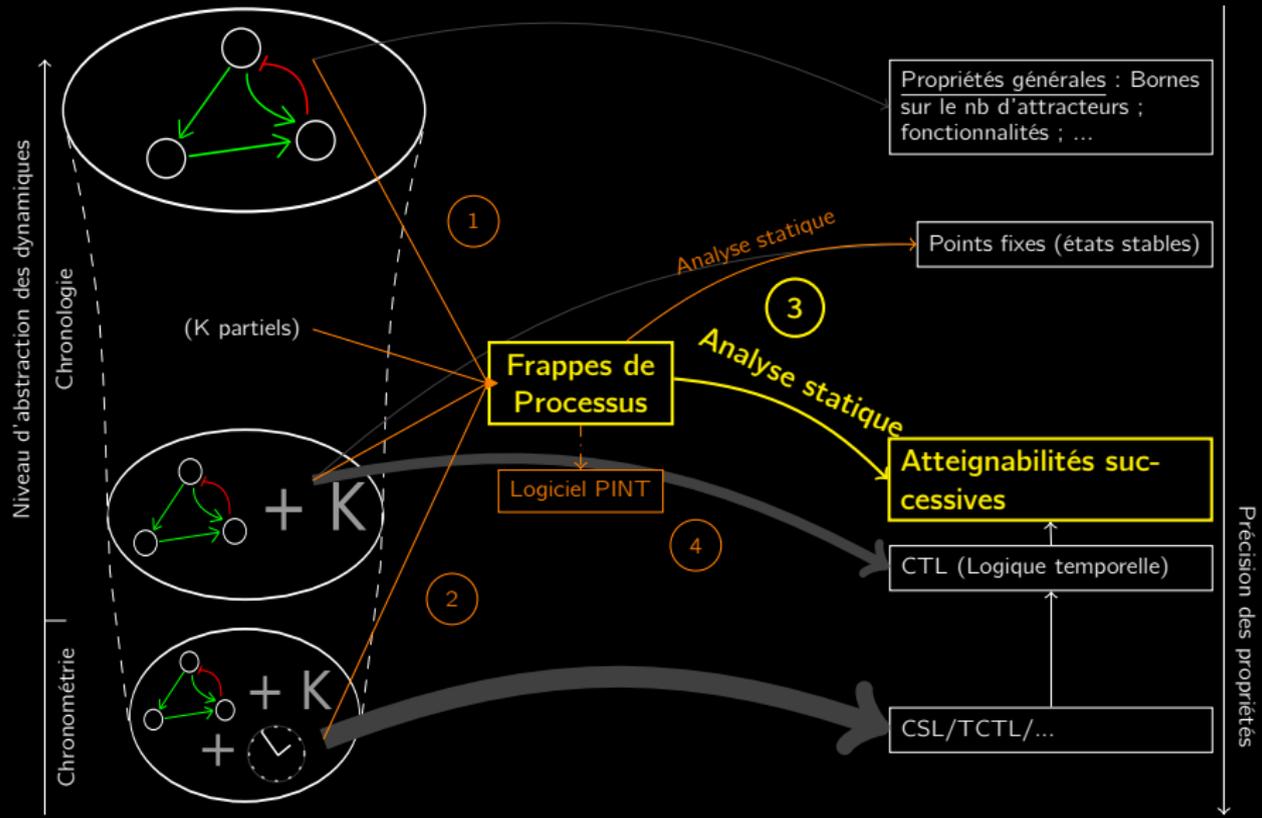
Points Fixes

- Réduction aux n -cliques dans un graphe n -parti.
- Résultat connexe : **Points fixes topologiques** des Réseaux Booléens [Paulevé, Richard dans CRAS 2010].

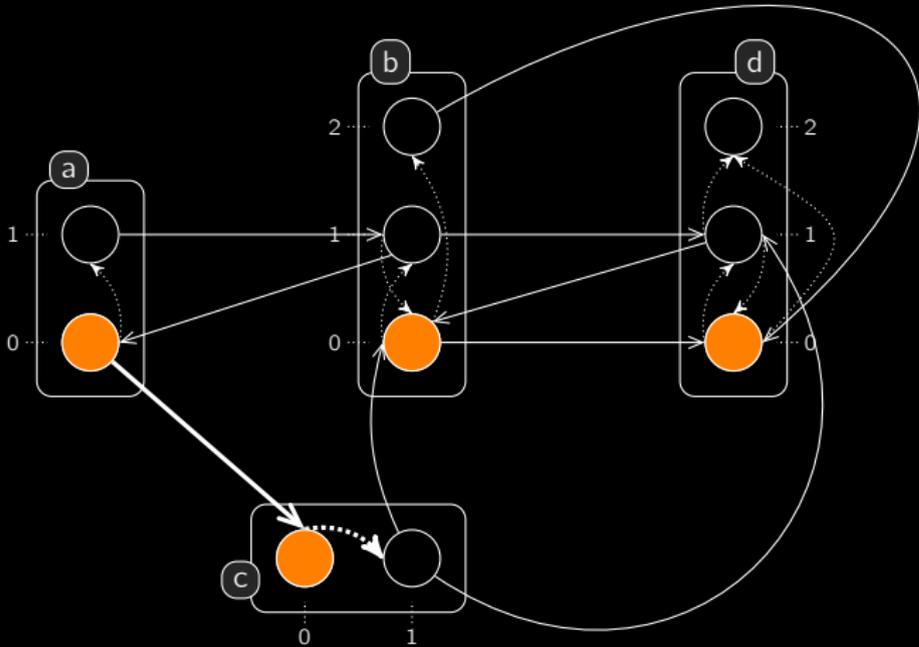
Propriétés d'atteignabilité successive $\text{EF } a_i \wedge (\text{EF } b_j \wedge \dots)$

- **Faible complexité** mais peut être non-concluant (**Oui/Non/Joker**).
- Techniques d'interprétation abstraite.
- Extraction des **processus clés** (vers le contrôle).

Plan

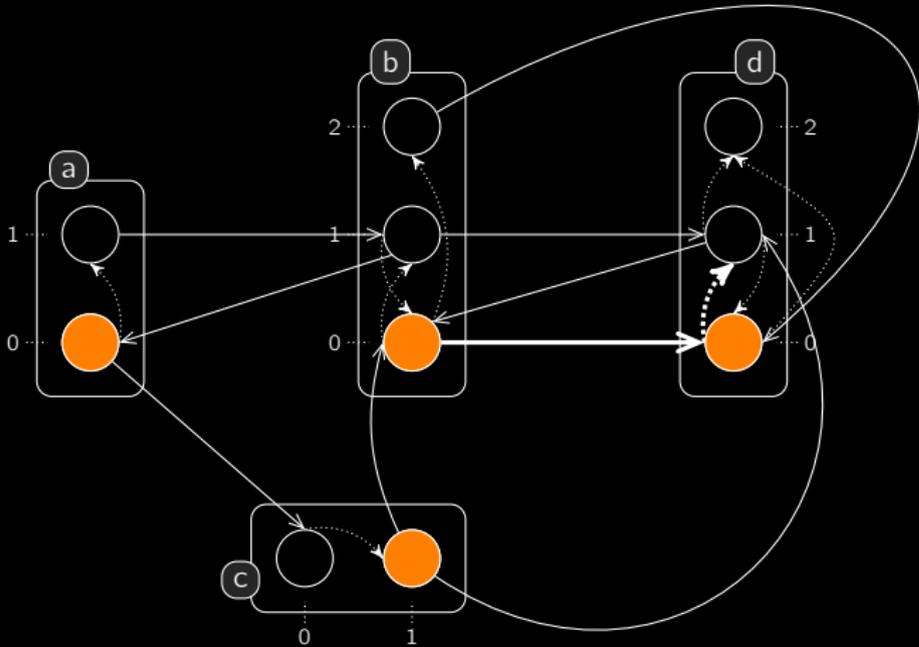


Scénarios



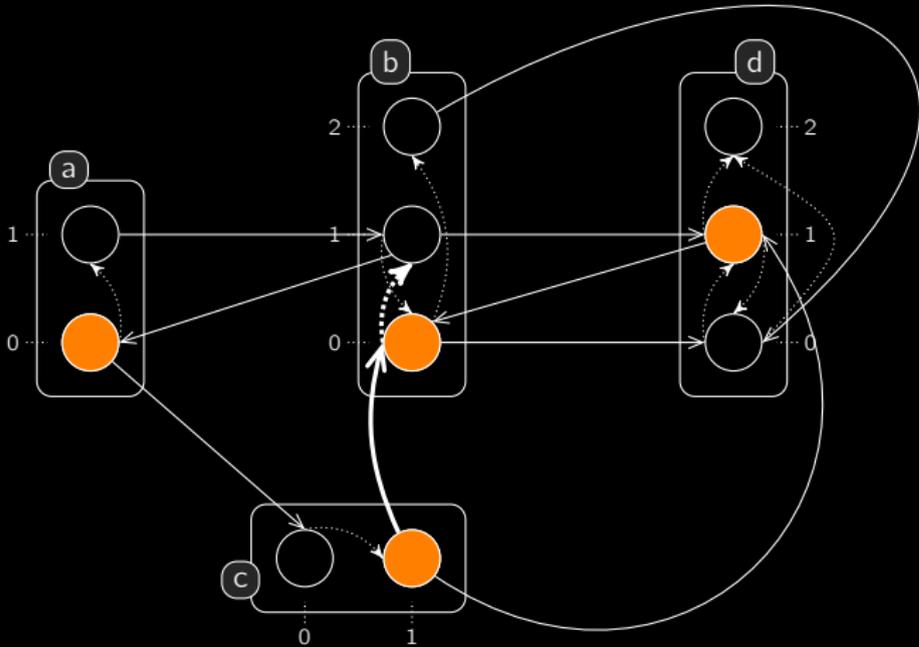
$$a_0 \rightarrow c_0 \uparrow c_1 :: b_0 \rightarrow d_0 \uparrow d_1 :: c_1 \rightarrow b_0 \uparrow b_1 :: b_1 \rightarrow d_1 \uparrow d_2$$

Scénarios



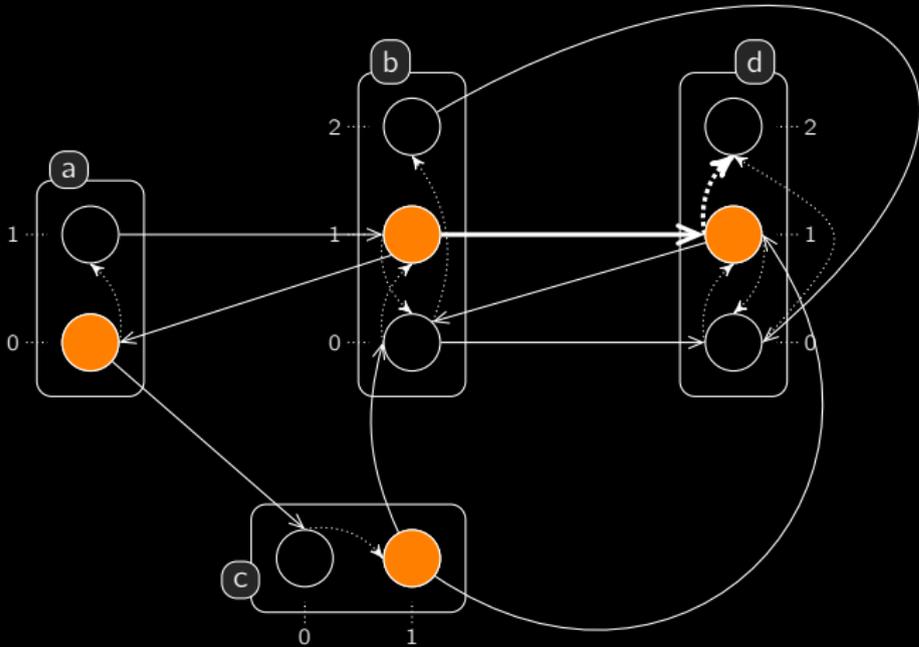
$$a_0 \rightarrow c_0 \uparrow c_1 :: b_0 \rightarrow d_0 \uparrow d_1 :: c_1 \rightarrow b_0 \uparrow b_1 :: b_1 \rightarrow d_1 \uparrow d_2$$

Scénarios



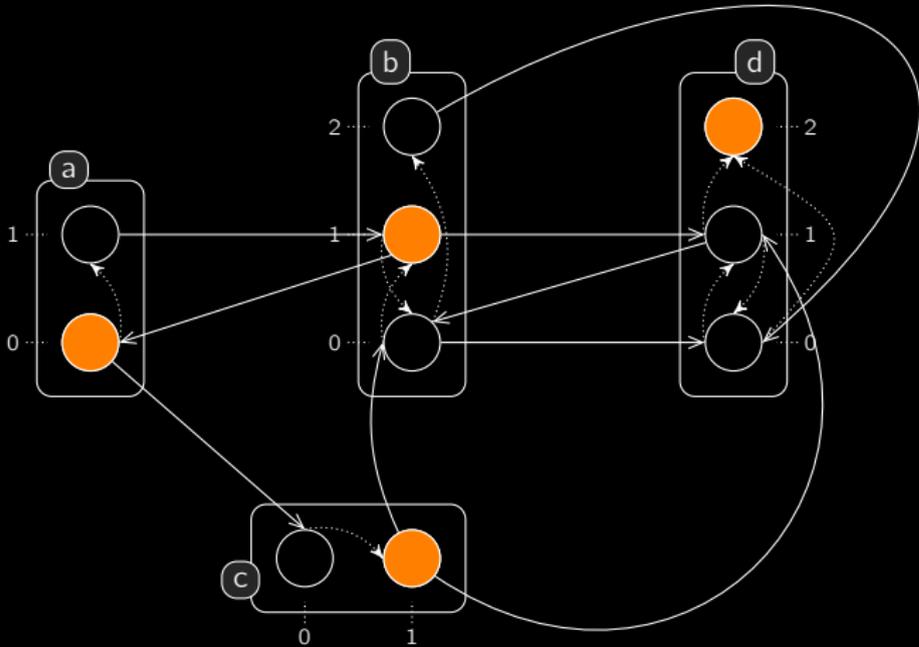
$$a_0 \rightarrow c_0 \uparrow c_1 :: b_0 \rightarrow d_0 \uparrow d_1 :: c_1 \rightarrow b_0 \uparrow b_1 :: b_1 \rightarrow d_1 \uparrow d_2$$

Scénarios



$$a_0 \rightarrow c_0 \uparrow c_1 :: b_0 \rightarrow d_0 \uparrow d_1 :: c_1 \rightarrow b_0 \uparrow b_1 :: b_1 \rightarrow d_1 \uparrow d_2$$

Scénarios



$$a_0 \rightarrow c_0 \uparrow c_1 :: b_0 \rightarrow d_0 \uparrow d_1 :: c_1 \rightarrow b_0 \uparrow b_1 :: b_1 \rightarrow d_1 \uparrow d_2$$

Analyse statique de propriétés d'atteignabilité

[Paulevé, Magnin, Roux à SASB 2010 + MSCS en révision]

Atteignabilités successives \mathcal{R}

- Étant donné un état initial des Frappes de Processus \mathcal{FP} ,
- est-il possible d'atteindre le processus a_i ? ...
- puis le processus b_j ? ... etc.

Analyse statique de propriétés d'atteignabilité

[Paulevé, Magnin, Roux à SASB 2010 + MSCS en révision]

Atteignabilités successives \mathcal{R}

- Étant donné un état initial des Frappes de Processus \mathcal{FP} ,
- est-il possible d'atteindre le processus a_i ? ...
- puis le processus b_j ? ... etc.

Problème : explosion combinatoire des dynamiques à explorer.

Analyse statique de propriétés d'atteignabilité

[Paulevé, Magnin, Roux à SASB 2010 + MSCS en révision]

Atteignabilités successives \mathcal{R}

- Étant donné un état initial des Frappes de Processus \mathcal{FP} ,
- est-il possible d'atteindre le processus a_i ? ...
- puis le processus b_j ? ... etc.

Problème : explosion combinatoire des dynamiques à explorer.

Approche adoptée

Approximations supérieures

\mathcal{FP} n'a pas la propriété $\mathcal{P} \implies \mathcal{R}$ est impossible.

Approximations inférieures

\mathcal{FP} a propriété $\mathcal{Q} \implies \mathcal{R}$ est possible.

Nécessité : vérifier \mathcal{P} (\mathcal{Q}) est rapide.

Interprétation abstraite des scénarios

Définitions

Scénarios Sce – Actions jouables successivement.

- P. ex. $\delta = a_0 \rightarrow c_0 \uparrow c_1 :: b_0 \rightarrow d_0 \uparrow d_1 :: c_1 \rightarrow b_0 \uparrow b_1 :: b_1 \rightarrow d_1 \uparrow d_2$.

Contexte – Pour chaque sorte, ensemble de processus initiaux.

- P. ex. $\varsigma = \langle a_1, \{b_1, b_2\}, c_1, d_0 \rangle$.

Interprétation abstraite des scénarios

Définitions

Scénarios Sca – Actions jouables successivement.

- P. ex. $\delta = a_0 \rightarrow c_0 \uparrow c_1 :: b_0 \rightarrow d_0 \uparrow d_1 :: c_1 \rightarrow b_0 \uparrow b_1 :: b_1 \rightarrow d_1 \uparrow d_2$.

Contexte – Pour chaque sorte, ensemble de processus initiaux.

- P. ex. $\varsigma = \langle a_1, \{b_1, b_2\}, c_1, d_0 \rangle$.

Séquences d'objectifs OS – abstraient un ensemble de scénarios (modélisent \mathcal{R}).

- P. ex. $\omega^1 = c_1 \uparrow^* c_0 :: d_0 \uparrow^* d_1 :: b_1 \uparrow^* b_0 :: d_1 \uparrow^* d_2$; $\omega^2 = b_1 \uparrow^* b_0 :: d_0 \uparrow^* d_2$; etc.
- $\gamma_\varsigma(\omega) = \{\delta \in \text{Sca} \mid \omega \text{ abstrait } \delta \wedge \text{support}(\delta) \subseteq \varsigma\}$.

Interprétation abstraite des scénarios

Définitions

Scénarios S_{ce} – Actions jouables successivement.

- P. ex. $\delta = a_0 \rightarrow c_0 \uparrow c_1 :: b_0 \rightarrow d_0 \uparrow d_1 :: c_1 \rightarrow b_0 \uparrow b_1 :: b_1 \rightarrow d_1 \uparrow d_2$.

Contexte – Pour chaque sorte, ensemble de processus initiaux.

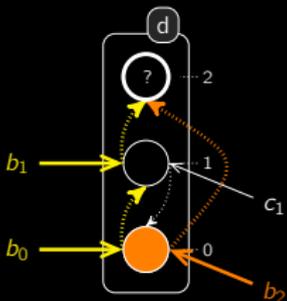
- P. ex. $\varsigma = \langle a_1, \{b_1, b_2\}, c_1, d_0 \rangle$.

Séquences d'objectifs OS – abstraient un ensemble de scénarios (modélisent \mathcal{R}).

- P. ex. $\omega^1 = c_1 \uparrow^* c_0 :: d_0 \uparrow^* d_1 :: b_1 \uparrow^* b_0 :: d_1 \uparrow^* d_2$; $\omega^2 = b_1 \uparrow^* b_0 :: d_0 \uparrow^* d_2$; etc.
- $\gamma_\varsigma(\omega) = \{\delta \in \text{Sce} \mid \omega \text{ abstrait } \delta \wedge \text{support}(\delta) \subseteq \varsigma\}$.

Séquences de bonds BS[^] – décrivent les solutions pour résoudre un objectif donné.

- P. ex. $\text{BS}^\wedge(d_0 \uparrow^* d_2) = \{\{b_0, b_1\}, \{b_2\}\}$.



Interprétation abstraite des scénarios

Approche générale

$$\gamma_{\mathcal{S}}(a_i \vec{\Gamma}^* a_j :: \dots :: z_k \vec{\Gamma}^* z_l) \neq \emptyset ?$$

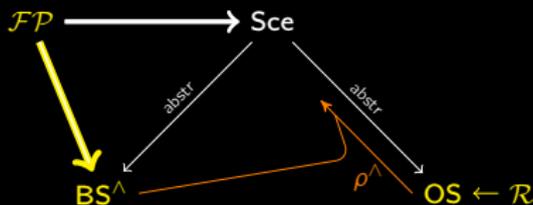
Interprétation abstraite des scénarios

Approche générale

$$\gamma_{\mathcal{S}}(a_i \mapsto^* a_j :: \dots :: z_k \mapsto^* z_l) \neq \emptyset ?$$

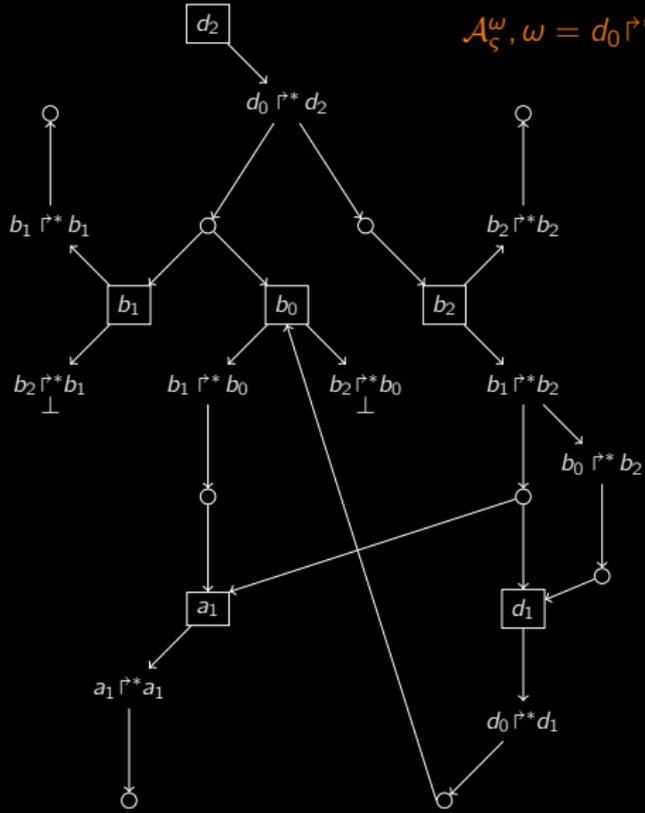
Raffinement d'un objectif

Obj $\times \wp(\text{BS}^\wedge)$	$\wp(\text{OS})$
$d_0 \mapsto^* d_2$	$\star \mapsto^* b_0 :: b_0 \mapsto^* b_1 :: d_0 \mapsto^* d_2,$
,	$\star \mapsto^* b_1 :: b_1 \mapsto^* b_0 :: d_0 \mapsto^* d_2,$
$\{\{b_0, b_1\}, \{b_2\}\}$	$\star \mapsto^* b_2 :: d_0 \mapsto^* d_2$
$\gamma_{\mathcal{S}}(d_0 \mapsto^* d_2)$	$= \gamma_{\mathcal{S}}(\rho^\wedge(d_0 \mapsto^* d_2, \text{BS}^\wedge(d_0 \mapsto^* d_2)))$



Structure abstraite des Frappes de Processus

$$\mathcal{A}_\zeta^\omega, \omega = d_0 \uparrow^* d_2, \zeta = \langle a_1, \{b_1, b_2\}, c_1, d_0 \rangle$$



Légende

Requis
 a_j \longrightarrow $a_i \uparrow^* a_j$

Solution
 $(\{b_i, c_j\} \in BS^\wedge(a_i \uparrow^* a_j))$

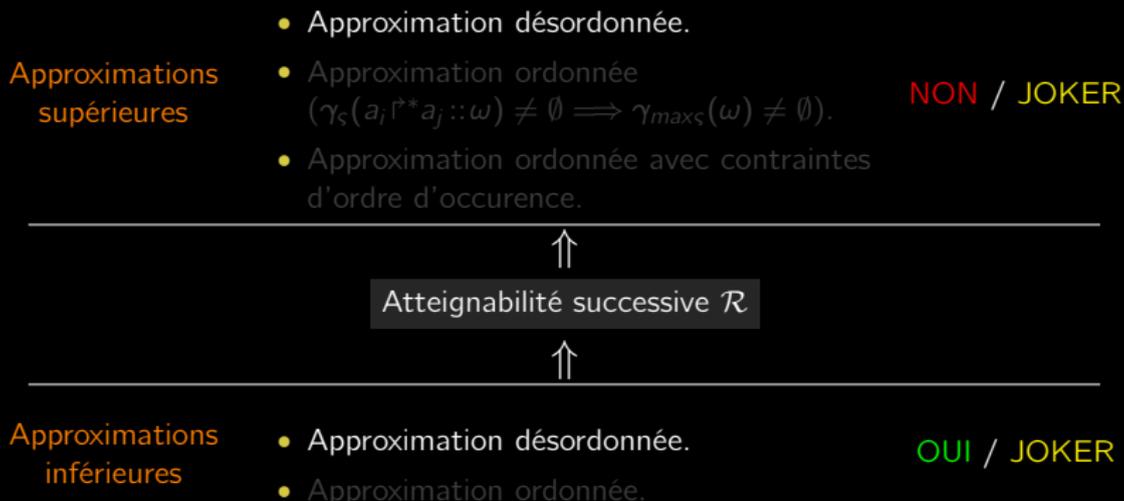
$a_i \uparrow^* a_j \longrightarrow$ $\begin{matrix} \circ \\ \swarrow \searrow \\ \boxed{b_i} \\ \boxed{c_j} \end{matrix}$

Continuité
 $a_i \uparrow^* a_j \longrightarrow a_k \uparrow^* a_j$

Solution triviale
 $a_i \uparrow^* a_j \longrightarrow \circ$

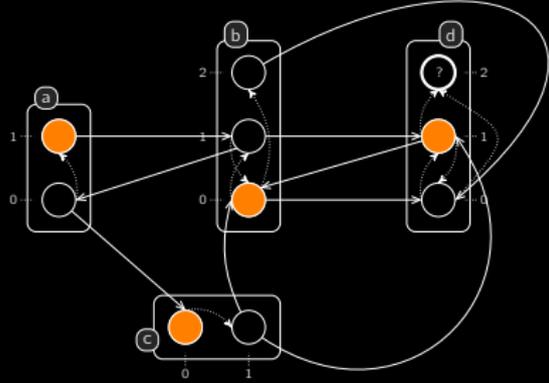
Pas de solution
 $a_i \uparrow^* a_j \longrightarrow \perp$

Analyse statique d'atteignabilités successives



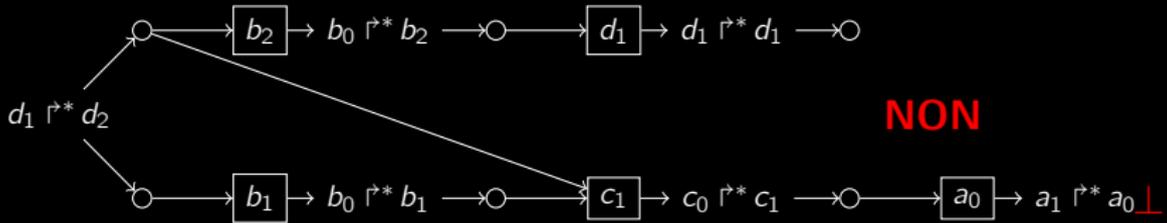
Approximation supérieure désordonnée

Exemple



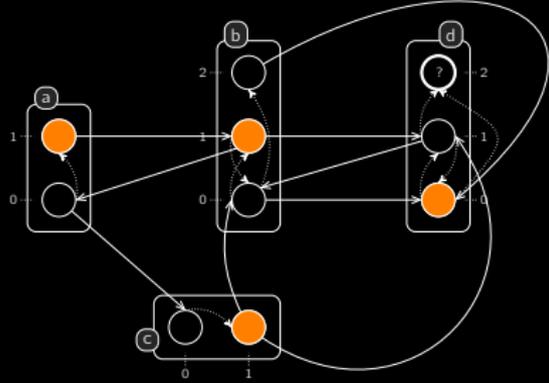
Condition nécessaire pour $\gamma_\zeta(\omega) \neq \emptyset$:
 Partant de chaque objectif dans ω , il existe un **parcours** dans \mathcal{A}_ζ^ω tel que :

- objectif \rightarrow sélectionne au moins une solution ;
- processus \rightarrow sélectionne tous les objectifs ;
- pas de cycle.



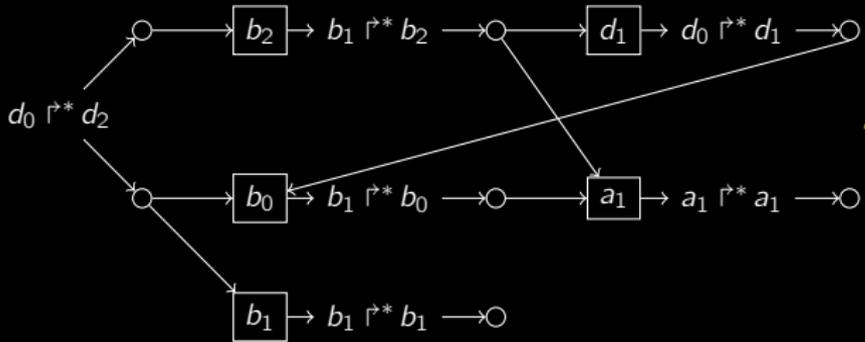
Approximation supérieure désordonnée

Exemple



Condition nécessaire pour $\gamma_\zeta(\omega) \neq \emptyset$:
 Partant de chaque objectif dans ω , il existe un **parcours** dans \mathcal{A}_ζ^ω tel que :

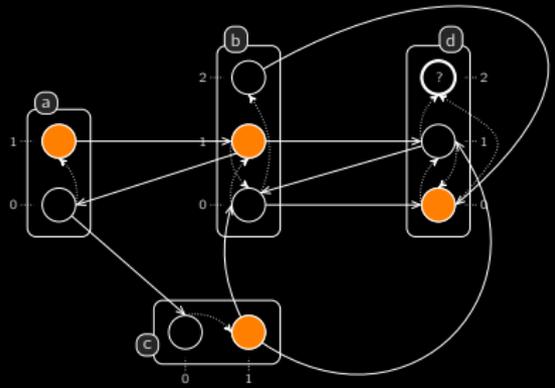
- objectif \rightarrow sélectionne au moins une solution ;
- processus \rightarrow sélectionne tous les objectifs ;
- pas de cycle.



JOKER

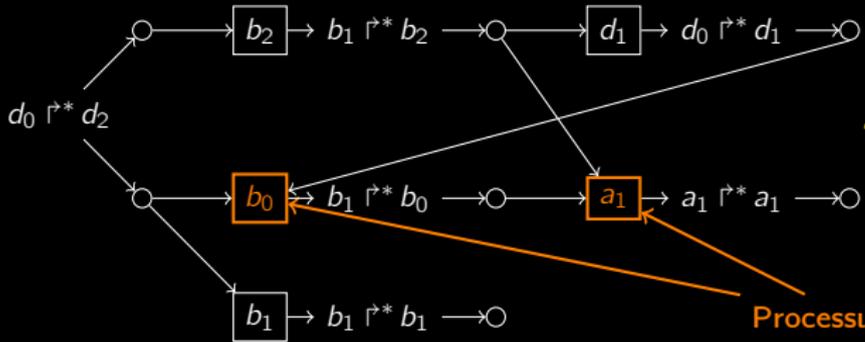
Approximation supérieure désordonnée

Exemple



Condition nécessaire pour $\gamma_\zeta(\omega) \neq \emptyset$:
 Partant de chaque objectif dans ω , il existe un parcours dans \mathcal{A}_ζ^ω tel que :

- objectif \rightarrow sélectionne au moins une solution ;
- processus \rightarrow sélectionne tous les objectifs ;
- pas de cycle.

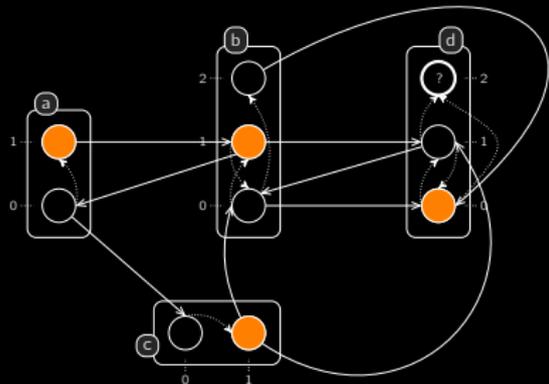


JOKER

Processus clés

Approximation inférieure désordonnée

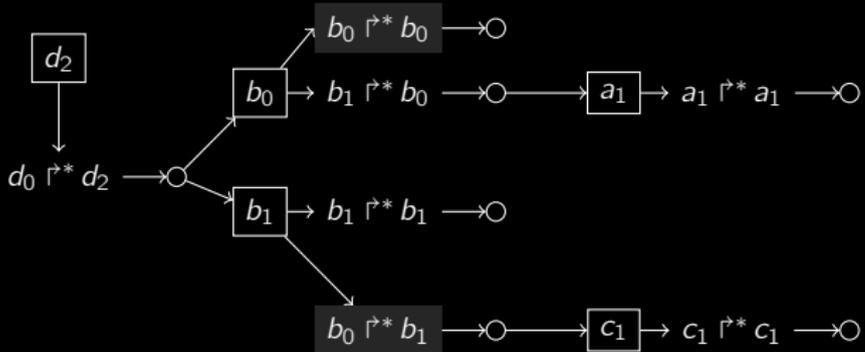
Exemple



Condition suffisante pour $\gamma_\zeta(\omega) \neq \emptyset$:

- $\lceil \mathcal{B}_\zeta^\omega \rceil$ est sans cycle ;
- tous les objectifs ont au moins une solution.

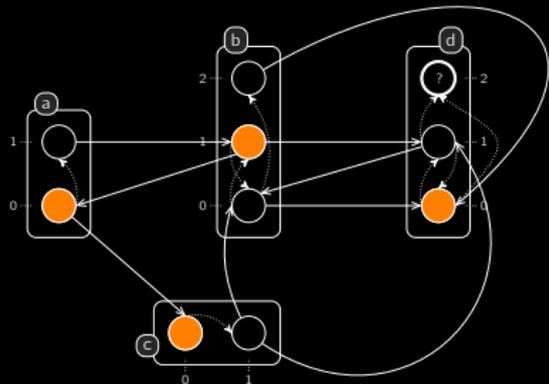
$\lceil \mathcal{B}_\zeta^\omega \rceil$: \mathcal{A}_ζ^ω saturée.



OUI

Approximation inférieure désordonnée

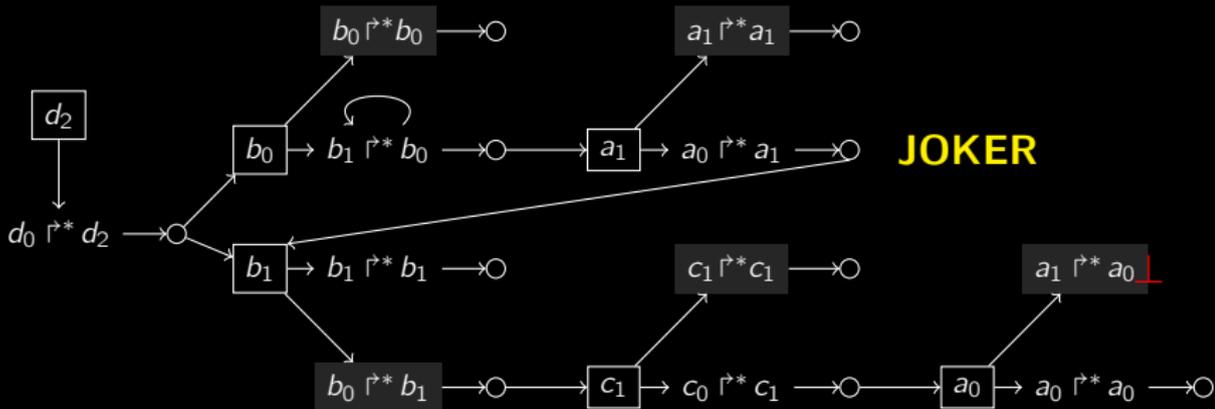
Exemple



Condition suffisante pour $\gamma_\zeta(\omega) \neq \emptyset$:

- $\lceil \mathcal{B}_\zeta^\omega \rceil$ est sans cycle ;
- tous les objectifs ont au moins une solution.

$\lceil \mathcal{B}_\zeta^\omega \rceil$: \mathcal{A}_ζ^ω saturée.



Analyse statique d'atteignabilités successives

Aller plus loin...

Approximations
supérieures

- Approximation désordonnée.
- Approximation ordonnée
($\gamma_{\zeta}(a_i \uparrow^* a_j :: \omega) \neq \emptyset \implies \gamma_{\max_{\zeta}}(\omega) \neq \emptyset$).
- Approximation ordonnée avec contraintes
d'ordre d'occurrence.

NON / JOKER

Atteignabilité successive \mathcal{R} Approximations
inférieures

- Approximation désordonnée.
- Approximation ordonnée.

OUI / JOKER

Analyse statique d'atteignabilités successives

Aller plus loin...

- Approximations supérieures
- Approximation désordonnée.
 - Approximation ordonnée
($\gamma_{\zeta}(a_i \uparrow^* a_j :: \omega) \neq \emptyset \implies \gamma_{\max_{\zeta}}(\omega) \neq \emptyset$).
 - Approximation ordonnée avec contraintes d'ordre d'occurrence.
- NON / JOKER

Atteignabilité successive \mathcal{R} 

- Approximations inférieures
- Approximation désordonnée.
 - Approximation ordonnée.
- OUI / JOKER

Toujours non-concluant ?

- Besoin de nouvelles analyses (plus approfondies) de la structure abstraite
- \Rightarrow guider les raffinements de \mathcal{R} .

Complexité

Structures abstraites

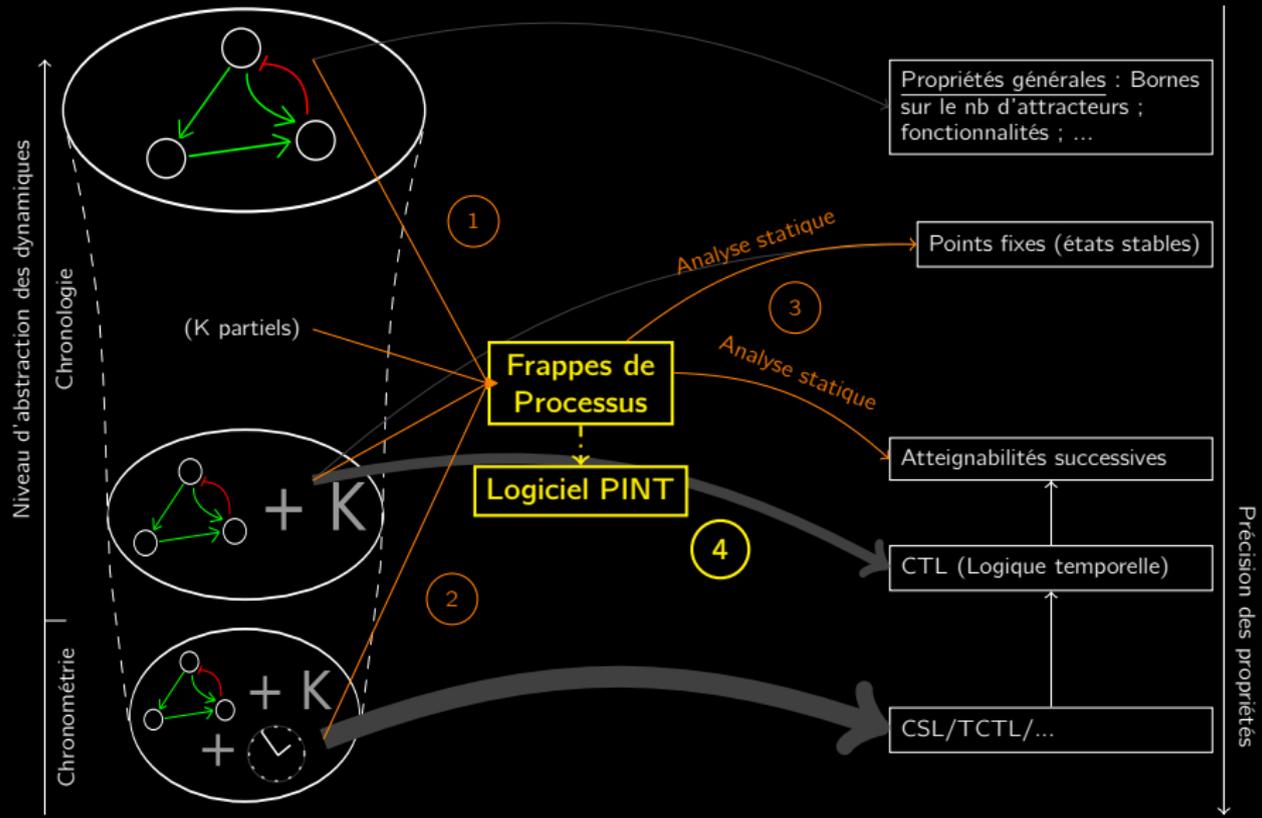
- Calcul de \mathbf{BS}^\wedge : **exponentiel** selon le nombre de **processus dans une seule sorte**.
- Taille de \mathbf{BS}^\wedge : combinaisons de $|\mathbf{Proc}_a|$ processus $\binom{|\mathbf{Proc}|}{|\mathbf{Proc}_a|}$.
- Taille de \mathcal{A}_ζ^ω (et $\lceil \mathcal{B}_\zeta^\omega \rceil$) : polynomiale \times taille de \mathbf{BS}^\wedge .

Analyses

- **Approximations sup. polynomiales** en la taille de \mathcal{A}_ζ^ω .
- Différentes stratégies pour les **approximations inf.** :
 - globale : **polynomiale** en la taille de $\lceil \mathcal{B}_\zeta^\omega \rceil$;
 - par solution : \times exponentielle selon la taille de \mathbf{BS}^\wedge .

\implies efficacité avec un **nombre réduit de processus par sorte**, alors qu'un **très grand nombre de sortes** peut être supporté.

Plan



Le logiciel PINT

Fonctionnalités

- **Language textuel** de spécification des Frappes de Processus.
- Simulation ;
- Liste des points fixes ;
- Analyse d'atteignabilités successives ;
- **Importations/exportations** depuis/vers de nombreux formats (dont logiciels orientés bio : GINsim, CellNetAnalyzer, Biocham).

Points techniques

- Implémentation en OCaml ; bibliothèque (réutilisable) ; logiciel libre.
- Disponible sur <http://process.hitting.free.fr> .
- Documentations + répertoire de modèles.

Temps d'exécution

Pour diverses analyses d'atteignabilité :

Modèle	sortes	procs	actions	états	Biocham ¹	libddd ²	PINT
egfr20	35	196	670	2^{64}	[3s-KO]	[1s-150s]	0.007s
tcrsig40	54	156	301	2^{73}	[1s-KO]	[0.6s-KO]	0.004s
tcrsig94	133	448	1124	2^{194}	KO	KO	0.030s
egfr104	193	748	2356	2^{320}	KO	KO	0.050s

¹ [Fages, Soliman]

² [LIP6/Move]

Résumé des contributions

Les Frappes de Processus [TCSB 11]

- **Nouveau formalisme** pour la modélisation abstraite des systèmes complexes.
- Simple (**un seul type d'action**) mais pas simpliste.
- Adapté à la modélisation des **Réseaux de Régulation Biologique**.

Résumé des contributions

Les Frappes de Processus [TCSB 11]

- **Nouveau formalisme** pour la modélisation abstraite des systèmes complexes.
- Simple (**un seul type d'action**) mais pas simpliste.
- Adapté à la modélisation des **Réseaux de Régulation Biologique**.

Paramètres temporels et stochastiques (dans le π -calcul) [IEEE TSE 10]

- Taux + **facteur d'absorption de stochasticité** (réducteur de variance).
- Spécification de la durée des actions par son **intervalle de confiance**.
- Simulation facile [CMSB 10]; **Vérification** formelle quantitative **difficile**.

Résumé des contributions

Les Frappes de Processus [TCSB 11]

- **Nouveau formalisme** pour la modélisation abstraite des systèmes complexes.
- Simple (**un seul type d'action**) mais pas simpliste.
- Adapté à la modélisation des **Réseaux de Régulation Biologique**.

Paramètres temporels et stochastiques (dans le π -calcul) [IEEE TSE 10]

- Taux + **facteur d'absorption de stochasticité** (réducteur de variance).
- Spécification de la durée des actions par son **intervalle de confiance**.
- Simulation facile [CMSB 10]; **Vérification** formelle quantitative **difficile**.

Analyse statique des Frappes de Processus [SASB 10, 11] [CRAS 10]

- Approximations très efficaces de propriétés d'**atteignabilité successive**.
- Extraction des **processus clés** nécessaires à une atteignabilité donnée (contrôle).
- **Rend possible** l'analyse formelle de **grands RRB**.

Ouvertures

Extension des analyses des Frappes de Processus

- Analyse de propriétés quantitatives (probabilité, temps, etc.).
- Autres propriétés biologiques intéressantes.
- Exemple : analyse des **attracteurs**.

Ouvertures

Extension des analyses des Frappes de Processus

- Analyse de propriétés quantitatives (probabilité, temps, etc.).
- Autres propriétés biologiques intéressantes.
- Exemple : analyse des **attracteurs**.

Inférence des paramètres discrets

- **Raffinement automatique** des dynamiques avec des données biologiques.
- Exemple : données d'états stables \Rightarrow **créer des points fixes**.

Ouvertures

Extension des analyses des Frappes de Processus

- Analyse de propriétés quantitatives (probabilité, temps, etc.).
- Autres propriétés biologiques intéressantes.
- Exemple : analyse des **attracteurs**.

Inférence des paramètres discrets

- **Raffinement automatique** des dynamiques avec des données biologiques.
- Exemple : données d'états stables \Rightarrow **créer des points fixes**.

Extension de l'applicabilité des Frappes de Processus

- Autres applications ?
- Fait : modélisation des **réseaux de réactions biochimiques** (p. ex. BIOCHAM).
- Étude des **Frappes de Processus avec Priorités** : encodage de divers formalismes existants.

Publications

Avec Olivier Roux et Morgan Magnin

- Journal *Transactions on Computational Systems Biology*, 2011.
- Journal *IEEE Transactions on Software Engineering*, en édition.
- Workshop *Static Analysis and Systems Biology*, 2010.
- En révision modérée : journal *Mathematical Structures for Computer Science*.

Avec Andrew Phillips et Matthew Lakin (Microsoft Research, Cambridge)

- Conférence *Computational Methods in Systems Biology*, 2010 (+ Simon Youssef, LMU, Munich).
- En révision modérée : journal *Transactions on Computer Science*.
- Article invité dans *Membrane Computing and Biologically Inspired Process Calculi*, 2010.

Avec Adrien Richard (I3S & CNRS, Nice)

- Journal *Comptes rendus de l'Académie des Sciences, Mathématiques*, 2010.
- Workshop *Static Analysis and Systems Biology*, 2011.