

LANGAGES DE SCÉNARIOS

Utiliser des ordres partiels pour modéliser, vérifier
et superviser des systèmes parallèles et répartis

Thomas GAZAGNAIRE

**projet DistribCom, IRISA
Université de Rennes 1
ENS Cachan, antenne de Bretagne**

Soutenance de Thèse

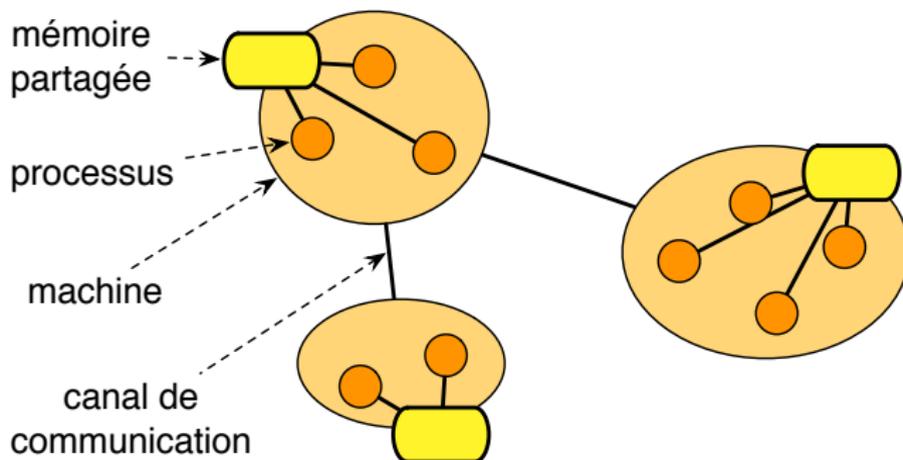
27 mars 2008

Philippe	DARONDEAU	Président du jury
Jean-Michel	COUVREUR	Rapporteur
Marc	ZEITOUN	Rapporteur
Thierry	MASSART	Examineur
Loïc	HÉLOUËT	Co-encadrant
Claude	JARD	Directeur de thèse

- Cadre : systèmes **parallèles** et **répartis** ;
- Modes d'interactions
 - ▶ proches : par mémoire partagée ;
 - ▶ à distance : par échange de messages.

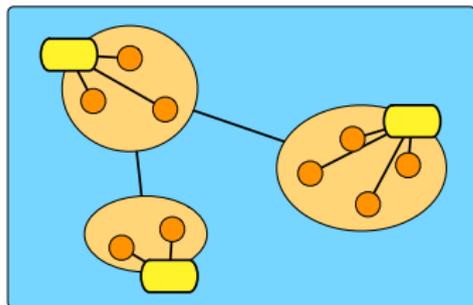
Contexte

- Cadre : systèmes parallèles et répartis ;
- Modes d'interactions
 - ▶ **proches** : par mémoire partagée ;
 - ▶ **à distance** : par échange de messages.



Problématique

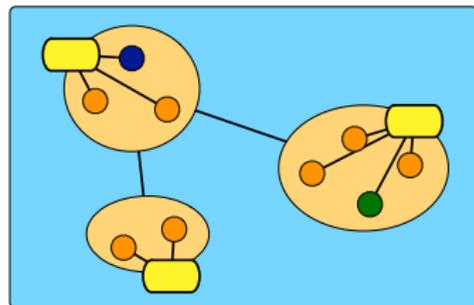
- Modéliser ;
- Vérifier ;
- Superviser.



modèle *M*

Problématique

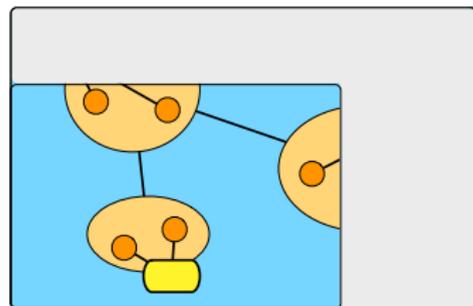
- Modéliser ;
- Vérifier ;
- Superviser.



modèle M

- ▶ **Propriété φ**
- ▶ “Est-ce qu’une action qu’effectue le processus **bleu** est toujours suivie d’une action sur le processus **vert** ?”
- ▶ $M \models \varphi$

- Modéliser ;
- Vérifier ;
- Superviser.



modèle M

- ▶ Propriété φ
observation partielle O limitée à Σ_o
- ▶ “Est-ce que l’observation permet de déduire que le système a toujours fait une action sur le processus **bleu** avant de faire une action sur le processus **vert** ?”
- ▶ $Diag(O, M) \models \varphi$

■ Approche locale :

- ▶ modèles séquentiels où l'on ajoute des actions d'interaction ;
 - ▶ algèbres de processus : CCS [Milner80], π -calcul [Milner92] ...
 - ▶ automates finis :
 - asynchrones [Zielonka87], communicants [Brand81].
- ▶ nécessite de calculer les interactions entre les modèles locaux ;
- ▶ la vérification et la supervision sont **indécidables**.

■ Approche globale :

- ▶ modélisation globale des interactions ;
 - ▶ ordres partiels [Lamport78], pomsets [Pratt86,Gischer88] ;
 - ▶ structures d'événements [Winskel81].
- ▶ langages normalisés de spécification :
 - High-level Message Sequence Charts (HMSC) [ITU-TS,UML2.0] ;
- ▶ permet d'**identifier** des familles de systèmes **non bornés** où vérification et supervision sont **décidables** [GMSZ02,Genest05].

■ Approche locale :

- ▶ modèles séquentiels où l'on ajoute des actions d'interaction ;
 - ▶ algèbres de processus : CCS [Milner80], π -calcul [Milner92] ...
 - ▶ automates finis :
 - asynchrones [Zielonka87], communicants [Brand81].
- ▶ nécessite de calculer les interactions entre les modèles locaux ;
- ▶ la vérification et la supervision sont **indécidables**.

■ Approche globale :

- ▶ modélisation globale des interactions ;
 - ▶ ordres partiels [Lamport78], pomsets [Pratt86,Gischer88] ;
 - ▶ structures d'événements [Winskel81].
- ▶ langages normalisés de spécification :
High-level Message Sequence Charts (HMSC) [ITU-TS,UML2.0] ;
- ▶ permet d'**identifier** des familles de systèmes **non bornés** où
vérification et supervision sont **décidables** [GMSZ02,Genest05].

■ Approche locale :

- ▶ modèles séquentiels où l'on ajoute des actions d'interaction ;
 - ▶ algèbres de processus : CCS [Milner80], π -calcul [Milner92] ...
 - ▶ automates finis :
asynchrones [Zielonka87], communicants [Brand81].
- ▶ nécessite de calculer les interactions entre les modèles locaux ;
- ▶ la vérification et la supervision sont **indécidables**.

■ Approche globale :

- ▶ modélisation globale des interactions ;
 - ▶ ordres partiels [Lamport78], pomsets [Pratt86,Gischer88] ;
 - ▶ structures d'événements [Winskel81].
- ▶ langages normalisés de spécification :
High-level Message Sequence Charts (HMSC) [ITU-TS,UML2.0] ;
- ▶ permet d'**identifier** des familles de systèmes **non bornés** où
vérification et supervision sont **décidables** [GMSZ02,Genest05].

■ Approche locale :

- ▶ modèles séquentiels où l'on ajoute des actions d'interaction ;
 - ▶ algèbres de processus : CCS [Milner80], π -calcul [Milner92] ...
 - ▶ automates finis :
asynchrones [Zielonka87], communicants [Brand81].
- ▶ nécessite de calculer les interactions entre les modèles locaux ;
- ▶ la vérification et la supervision sont **indécidables**.

■ Approche globale :

- ▶ modélisation globale des interactions ;
 - ▶ ordres partiels [Lamport78], pomsets [Pratt86,Gischer88] ;
 - ▶ structures d'événements [Winskel81].
- ▶ langages normalisés de spécification :
High-level Message Sequence Charts (HMSC) [ITU-TS,UML2.0] ;
- ▶ permet d'**identifier** des familles de systèmes **non bornés** où
vérification et supervision sont **décidables** [GMSZ02,Genest05].

■ Approche locale :

- ▶ modèles séquentiels où l'on ajoute des actions d'interaction ;
 - ▶ algèbres de processus : CCS [Milner80], π -calcul [Milner92] ...
 - ▶ automates finis :
asynchrones [Zielonka87], communicants [Brand81].
- ▶ nécessite de calculer les interactions entre les modèles locaux ;
- ▶ la vérification et la supervision sont **indécidables**.

■ Approche globale :

- ▶ modélisation globale des interactions ;
 - ▶ ordres partiels [Lamport78], pomsets [Pratt86,Gischer88] ;
 - ▶ structures d'événements [Winskel81].
- ▶ langages normalisés de spécification :
High-level Message Sequence Charts (HMSC) [ITU-TS,UML2.0] ;
- ▶ permet d'identifier des familles de systèmes non bornés où
vérification et supervision sont **décidables** [GMSZ02,Genest05].

■ Approche locale :

- ▶ modèles séquentiels où l'on ajoute des actions d'interaction ;
 - ▶ algèbres de processus : CCS [Milner80], π -calcul [Milner92] ...
 - ▶ automates finis :
asynchrones [Zielonka87], communicants [Brand81].
- ▶ nécessite de calculer les interactions entre les modèles locaux ;
- ▶ la vérification et la supervision sont **indécidables**.

■ Approche globale :

- ▶ modélisation globale des interactions ;
 - ▶ ordres partiels [Lamport78], pomsets [Pratt86,Gischer88] ;
 - ▶ structures d'événements [Winskel81].
- ▶ langages normalisés de spécification :
High-level Message Sequence Charts (HMSC) [ITU-TS,UML2.0] ;
- ▶ permet d'identifier des familles de systèmes non bornés où
vérification et supervision sont **décidables** [GMSZ02,Genest05].

■ Approche locale :

- ▶ modèles séquentiels où l'on ajoute des actions d'interaction ;
 - ▶ algèbres de processus : CCS [Milner80], π -calcul [Milner92] ...
 - ▶ automates finis :
asynchrones [Zielonka87], communicants [Brand81].
- ▶ nécessite de calculer les interactions entre les modèles locaux ;
- ▶ la vérification et la supervision sont **indécidables**.

■ Approche globale :

- ▶ modélisation globale des interactions ;
 - ▶ ordres partiels [Lamport78], pomsets [Pratt86,Gischer88] ;
 - ▶ structures d'événements [Winskel81].
- ▶ langages normalisés de spécification :
High-level Message Sequence Charts (HMSC) [ITU-TS,UML2.0] ;
- ▶ permet d'identifier des familles de systèmes non bornés où
vérification et supervision sont **décidables** [GMSZ02,Genest05].

■ Approche locale :

- ▶ modèles séquentiels où l'on ajoute des actions d'interaction ;
 - ▶ algèbres de processus : CCS [Milner80], π -calcul [Milner92] ...
 - ▶ automates finis :
asynchrones [Zielonka87], communicants [Brand81].
- ▶ nécessite de calculer les interactions entre les modèles locaux ;
- ▶ la vérification et la supervision sont **indécidables**.

■ Approche globale :

- ▶ modélisation globale des interactions ;
 - ▶ ordres partiels [Lamport78], pomsets [Pratt86,Gischer88] ;
 - ▶ structures d'événements [Winskel81].
- ▶ langages normalisés de spécification :
High-level Message Sequence Charts (HMSC) [ITU-TS,UML2.0] ;
- ▶ permet d'**identifier** des familles de systèmes **non bornés** où
vérification et supervision sont **décidables** [GMSZ02,Genest05].

■ Approche locale :

- ▶ modèles séquentiels où l'on ajoute des actions d'interaction ;
 - ▶ algèbres de processus : CCS [Milner80], π -calcul [Milner92] ...
 - ▶ automates finis :
asynchrones [Zielonka87], communicants [Brand81].
- ▶ nécessite de calculer les interactions entre les modèles locaux ;
- ▶ la vérification et la supervision sont **indécidables**.

■ Approche globale :

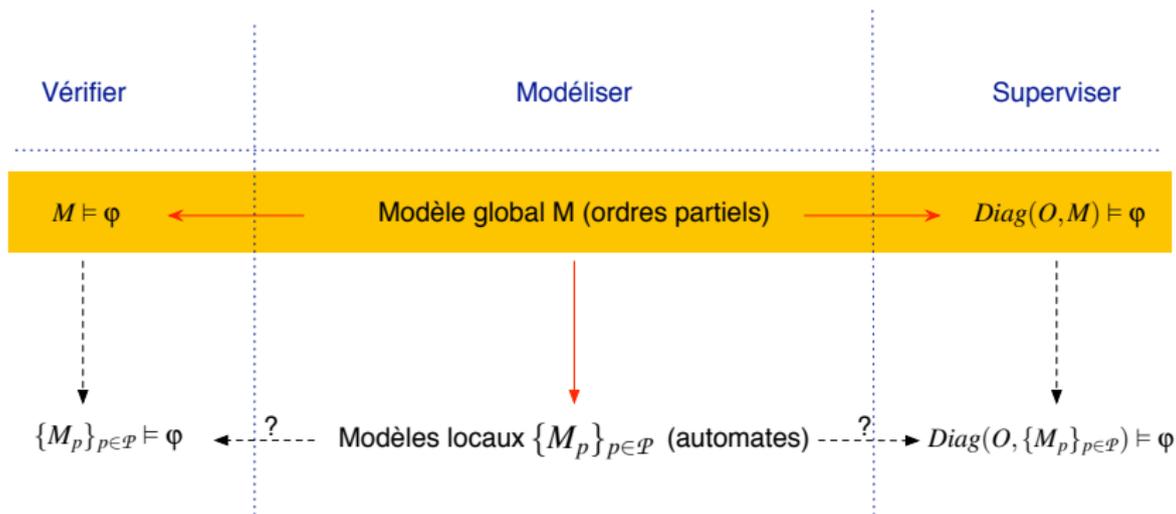
- ▶ modélisation globale des interactions ;
 - ▶ ordres partiels [Lamport78], pomsets [Pratt86,Gischer88] ;
 - ▶ structures d'événements [Winskel81].
- ▶ langages normalisés de spécification :
High-level Message Sequence Charts (HMSC) [ITU-TS,UML2.0] ;
- ▶ permet d'identifier des familles de systèmes non bornés où
vérification et supervision sont **décidables** [GMSZ02,Genest05].

■ Approche locale :

- ▶ modèles séquentiels où l'on ajoute des actions d'interaction ;
 - ▶ algèbres de processus : CCS [Milner80], π -calcul [Milner92] ...
 - ▶ automates finis :
asynchrones [Zielonka87], communicants [Brand81].
- ▶ nécessite de calculer les interactions entre les modèles locaux ;
- ▶ la vérification et la supervision sont **indécidables**.

■ Approche globale :

- ▶ modélisation globale des interactions ;
 - ▶ ordres partiels [Lamport78], pomsets [Pratt86,Gischer88] ;
 - ▶ structures d'événements [Winskel81].
- ▶ langages normalisés de spécification :
High-level Message Sequence Charts (HMSC) [ITU-TS,UML2.0] ;
- ▶ permet d'**identifier** des familles de systèmes **non bornés** où
vérification et supervision sont **décidables** [GMSZ02,Genest05].



Plan de l'exposé

Modèles locaux et leurs interactions globales

- Interactions par mémoire partagée

- Interactions par échange de messages

- Interactions mixtes

Modéliser

- HMSC causaux

- Expressivité

- Critiques et perspectives

Vérifier

- Vérification complète

- Vérification partielle

- Critiques et perspectives

Superviser

- Diagnostic

- Critiques et perspectives

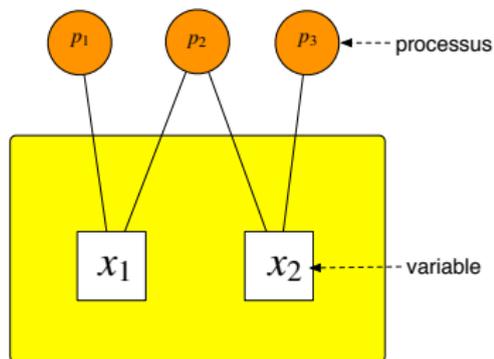
Conclusion

Modèles d'interactions (1/3) : par mémoire partagée

Systèmes parallèles qui interagissent par **mémoire partagée**

modèle local : automates asynchrones [Zielonka87]

modèle global : traces de Mazurkiewicz [Mazurkiewicz77]

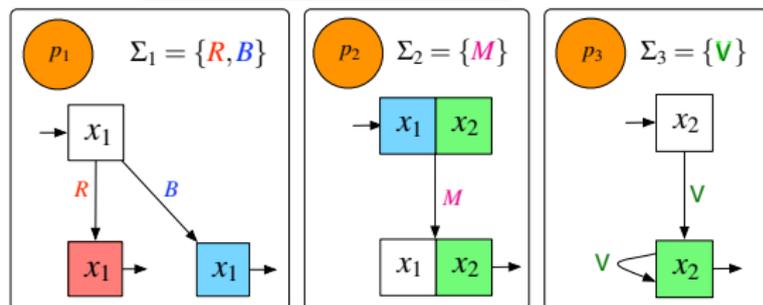
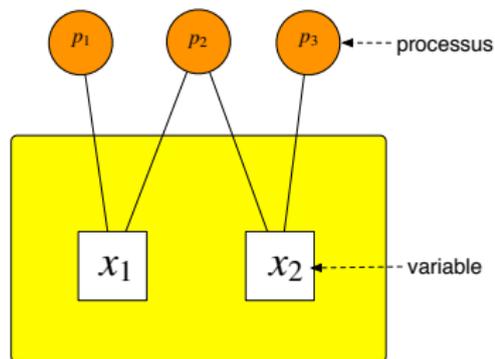


Modèles d'interactions (1/3) : par mémoire partagée

Systèmes parallèles qui interagissent par **mémoire partagée**

modèle local : automates asynchrones [Zielonka87]

modèle global : traces de Mazurkiewicz [Mazurkiewicz77]

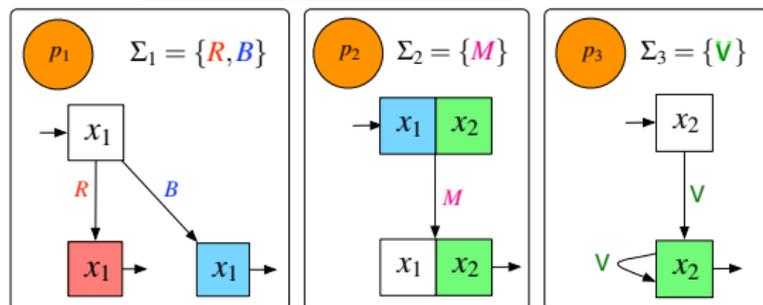
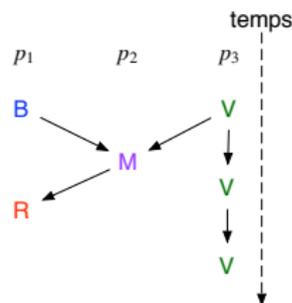
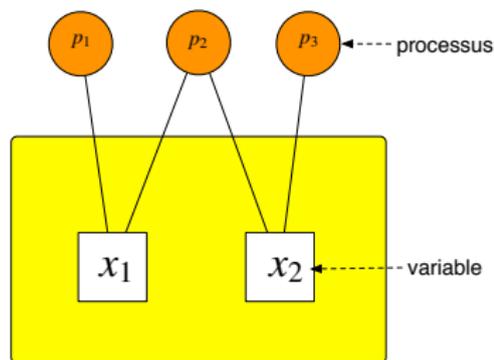


Modèles d'interactions (1/3) : par mémoire partagée

Systèmes parallèles qui interagissent par **mémoire partagée**

modèle local : automates asynchrones [Zielonka87]

modèle global : traces de Mazurkiewicz [Mazurkiewicz77]

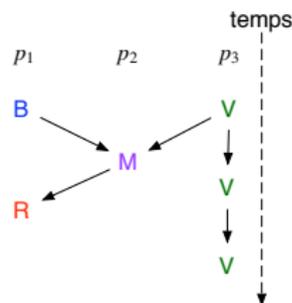
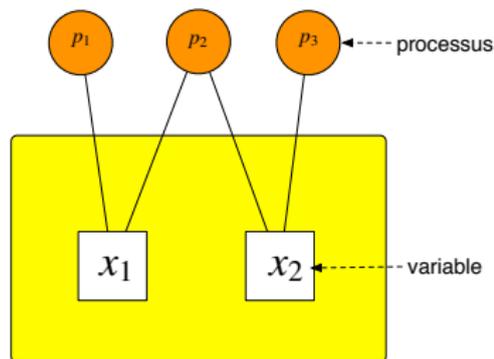


Modèles d'interactions (1/3) : par mémoire partagée

Systèmes parallèles qui interagissent par **mémoire partagée**

modèle local : automates asynchrones [Zielonka87]

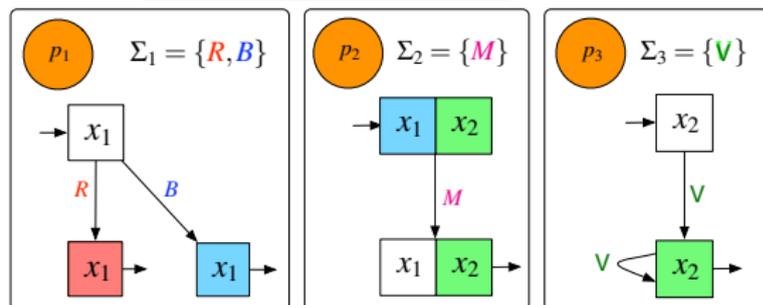
modèle global : traces de Mazurkiewicz [Mazurkiewicz77]



relation de dépendance **statique**

entre actions effectuées par des processus qui modifient une même variable partagée : $D =$

$\{\dots (B, M), (M, R), (V, M), (V, V) \dots\}$



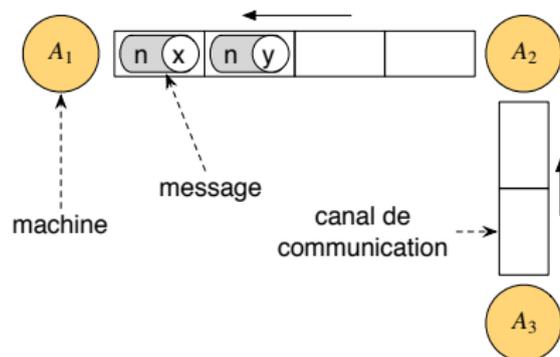
Modèles d'interactions (2/3) : par échange de messages

Systèmes parallèles qui interagissent par **échange de messages**

modèle local : automates communicants [Brand81]

modèle global : "Message Sequence Charts" (MSC)

[Reniers98,ITU-TS,UML2.0]



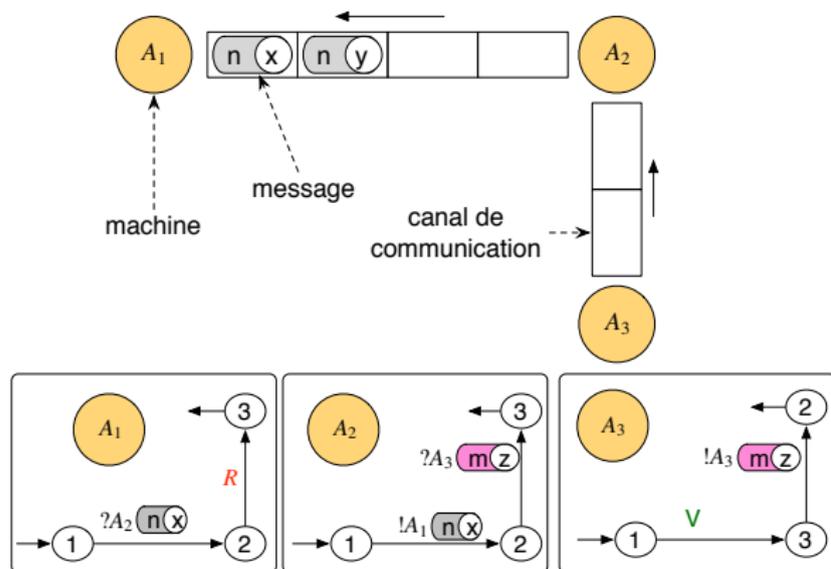
Modèles d'interactions (2/3) : par échange de messages

Systèmes parallèles qui interagissent par **échange de messages**

modèle local : automates communicants [Brand81]

modèle global : "Message Sequence Charts" (MSC)

[Reniers98,ITU-TS,UML2.0]



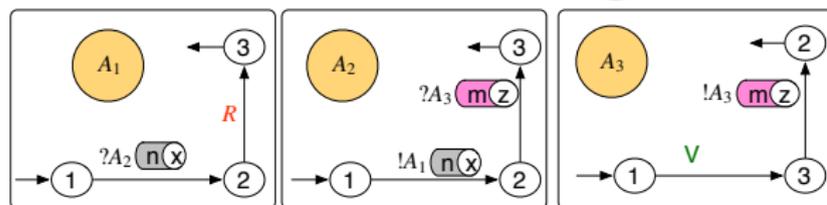
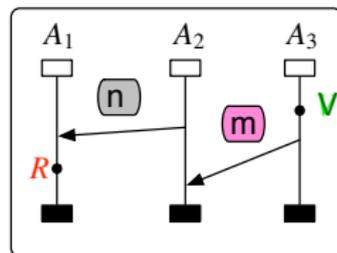
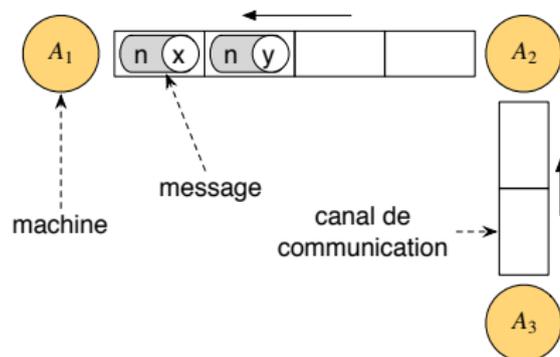
Modèles d'interactions (2/3) : par échange de messages

Systèmes parallèles qui interagissent par **échange de messages**

modèle local : automates communicants [Brand81]

modèle global : "Message Sequence Charts" (MSC)

[Reniers98,ITU-TS,UML2.0]



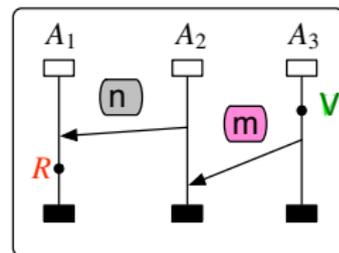
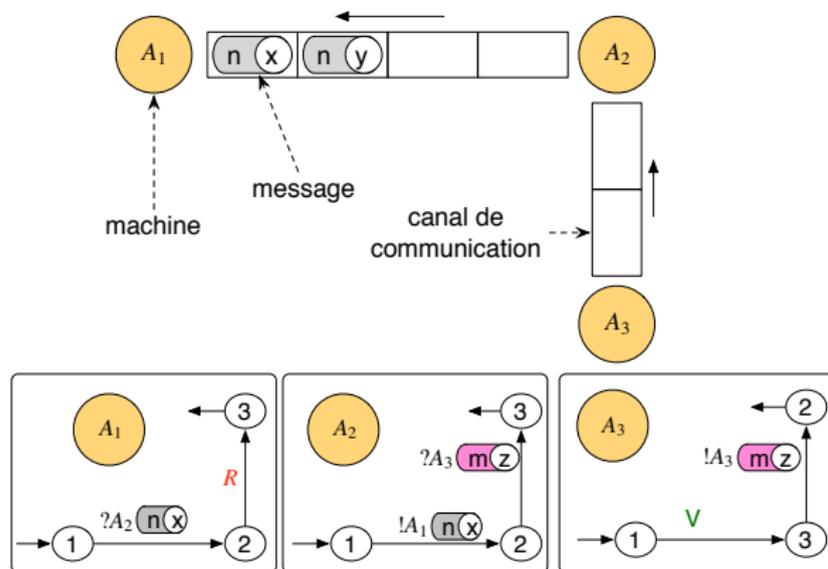
Modèles d'interactions (2/3) : par échange de messages

Systèmes parallèles qui interagissent par **échange de messages**

modèle local : automates communicants [Brand81]

modèle global : "Message Sequence Charts" (MSC)

[Reniers98,ITU-TS,UML2.0]



- sur chaque machine les événements sont **totalment** ordonnés

Modèles d'interactions (3/3) : mixtes

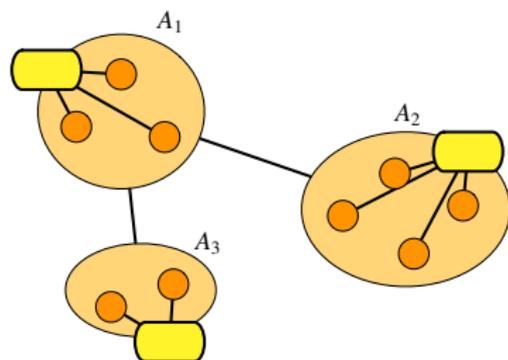
Systèmes parallèles et répartis avec :

- ▶ interactions proches : par **mémoire partagée** ;
- ▶ interactions à distance : par **échange de messages**.

modèle local : automates mixtes

modèle global : MSC causaux

- MSC avec co-régions généralisées [ITU-TS]
- ordre causal pour les MSC [AHP96,MP98]



Modèles d'interactions (3/3) : mixtes

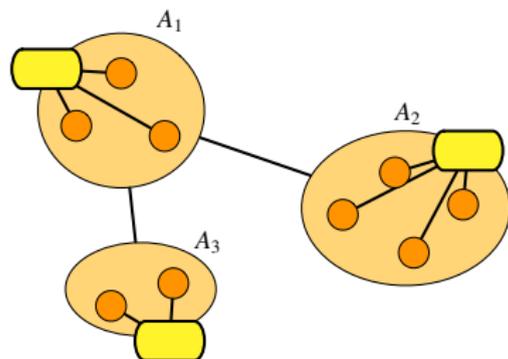
Systèmes parallèles et répartis avec :

- ▶ interactions proches : par **mémoire partagée** ;
- ▶ interactions à distance : par **échange de messages**.

modèle local : automates mixtes

modèle global : MSC causaux

- MSC avec co-régions généralisées [ITU-TS]
- ordre causal pour les MSC [AHP96,MP98]



Les actions des automates locaux :

- modifie des variables partagées ;
- modifie l'état des canaux de communication.

Modèles d'interactions (3/3) : mixtes

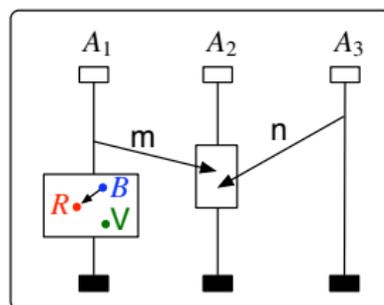
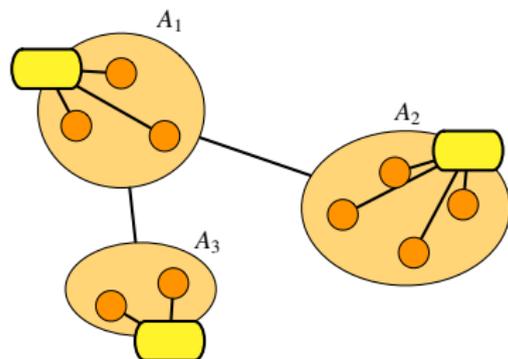
Systèmes parallèles et répartis avec :

- ▶ interactions proches : par **mémoire partagée** ;
- ▶ interactions à distance : par **échange de messages**.

modèle local : automates mixtes

modèle global : MSC causaux

- MSC avec co-régions généralisées [ITU-TS]
- ordre causal pour les MSC [AHP96,MP98]



Plan de l'exposé

Modèles locaux et leurs interactions globales

Interactions par mémoire partagée

Interactions par échange de messages

Interactions mixtes

Modéliser

HMSC causaux

Expressivité

Critiques et perspectives

Vérifier

Vérification complète

Vérification partielle

Critiques et perspectives

Superviser

Diagnostic

Critiques et perspectives

Conclusion

Modéliser : HMSC causaux (1/2)

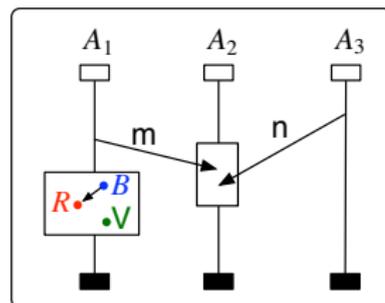
Définition (HMSC causaux [CONCUR'07])

- Relation de dépendance D_i pour chaque machine A_i ;
- Briques de base : MSC causaux ;
- Opérateurs de composition :
 - + choix ; · composition séquentielle ; * itération.

HMSC causal : expression
rationnelle de MSC causaux

Exemple : $u \cdot (v + w)^* \cdot u$

MSC causal u



Définition (HMSC causaux [CONCUR'07])

- Relation de dépendance D_i pour chaque machine A_i ;
- Briques de base : MSC causaux ;
- Opérateurs de composition :
 - + choix ; \cdot composition séquentielle ; \star itération.

Définition (langage causal d'un HMSC causal)

$\mathcal{L}(M)$: ensemble des MSC causaux engendrés par M

- $\mathcal{L}(u) = \{u\}$ (u MSC causal) ;
- $\mathcal{L}(M + N) = \mathcal{L}(M) \cup \mathcal{L}(N)$ (M, N HMSC causaux) ;
- $\mathcal{L}(M \cdot N) = \{u \cdot v \mid u \in \mathcal{L}(M), v \in \mathcal{L}(N)\}$ (M, N HMSC causaux) ;
- $\mathcal{L}(M^*) = \bigcup_{n \geq 0} \mathcal{L}(M)^n$ (M HMSC causal).

Définition (HMSC causaux [CONCUR'07])

- Relation de dépendance D_i pour chaque machine A_i ;
- Briques de base : MSC causaux ;
- Opérateurs de composition :
 - + choix ; \cdot composition séquentielle ; \star itération.

Définition (langage causal d'un HMSC causal)

$\mathcal{L}(M)$: ensemble des MSC causaux engendrés par M

- $\mathcal{L}(u) = \{u\}$ (u MSC causal) ;
- $\mathcal{L}(M + N) = \mathcal{L}(M) \cup \mathcal{L}(N)$ (M, N HMSC causaux) ;
- $\mathcal{L}(M \cdot N) = \{u \cdot v \mid u \in \mathcal{L}(M), v \in \mathcal{L}(N)\}$ (M, N HMSC causaux) ;
- $\mathcal{L}(M^*) = \bigcup_{n \geq 0} \mathcal{L}(M)^n$ (M HMSC causal).

Modéliser : HMSC causaux (1/2)

Définition (HMSC causaux [CONCUR'07])

- Relation de dépendance D_i pour chaque machine A_i ;
- Briques de base : MSC causaux ;
- Opérateurs de composition :
 - + choix ; · composition séquentielle ; * itération.

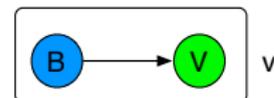
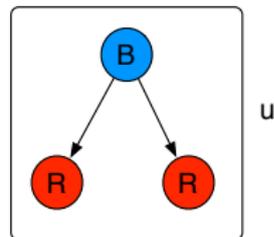
Définition (composition [Pratt86])

Paramétrée par $\cup D_i$

Exemple :

$D_1 = \{\dots, (B, B), (R, V), \dots\}$ sur A_1

$u \cdot v = w$



Modéliser : HMSC causaux (1/2)

Définition (HMSC causaux [CONCUR'07])

- Relation de dépendance D_i pour chaque machine A_i ;
- Briques de base : MSC causaux ;
- Opérateurs de composition :
 - + choix ;
 - composition séquentielle ;
 - ★ itération.

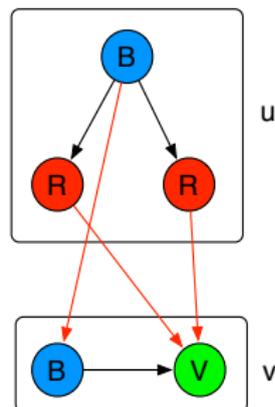
Définition (composition [Pratt86])

Paramétrée par $\cup D_i$

Exemple :

$D_1 = \{\dots, (B, B), (R, V), \dots\}$ sur A_1

$u \cdot v = w$



Définition (HMSC causaux [CONCUR'07])

- Relation de dépendance D_i pour chaque machine A_i ;
- Briques de base : MSC causaux ;
- Opérateurs de composition :
 - + choix ;
 - composition séquentielle ;
 - ★ itération.

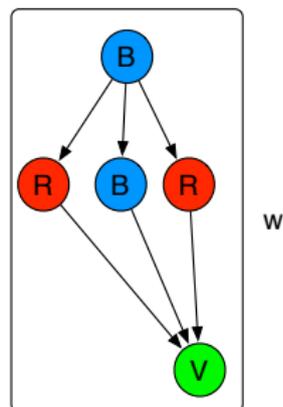
Définition (composition [Pratt86])

Paramétrée par $\cup D_i$

Exemple :

$D_1 = \{\dots, (B, B), (R, V), \dots\}$ sur A_1

$u \cdot v = w$



Définition (HMSC causaux [CONCUR'07])

- Relation de dépendance D_i pour chaque machine A_i ;
- Briques de base : MSC causaux ;
- Opérateurs de composition :
 - + choix ; \cdot composition séquentielle ; \star itération.

Définition (langage causal d'un HMSC causal)

$\mathcal{L}(M)$: ensemble des MSC causaux engendrés par M

- $\mathcal{L}(u) = \{u\}$ (u MSC causal) ;
- $\mathcal{L}(M + N) = \mathcal{L}(M) \cup \mathcal{L}(N)$ (M, N HMSC causaux) ;
- $\mathcal{L}(M \cdot N) = \{u \cdot v \mid u \in \mathcal{L}(M), v \in \mathcal{L}(N)\}$ (M, N HMSC causaux) ;
- $\mathcal{L}(M^*) = \bigcup_{n \geq 0} \mathcal{L}(M)^n$ (M HMSC causal).

Définition (HMSC causaux [CONCUR'07])

- Relation de dépendance D_i pour chaque machine A_i ;
- Briques de base : MSC causaux ;
- Opérateurs de composition :
+ choix ; \cdot composition séquentielle ; \star itération.

Définition (langage causal d'un HMSC causal)

$\mathcal{L}(M)$: ensemble des MSC causaux engendrés par M

- $\mathcal{L}(u) = \{u\}$ (u MSC causal) ;
- $\mathcal{L}(M + N) = \mathcal{L}(M) \cup \mathcal{L}(N)$ (M, N HMSC causaux) ;
- $\mathcal{L}(M \cdot N) = \{u \cdot v \mid u \in \mathcal{L}(M), v \in \mathcal{L}(N)\}$ (M, N HMSC causaux) ;
- $\mathcal{L}(M^*) = \bigcup_{n \geq 0} \mathcal{L}(M)^n$ (M HMSC causal).

Définition (cohérence)

Un MSC causal est cohérent si :

- *deux événements sont ordonnés \Rightarrow il existe une chaîne d'interactions qui les relie ;*
- *deux événements ne sont pas ordonnés \Rightarrow les processus qui les exécutent ne modifient pas une même variable partagée.*

Pour les HMSC causaux, il suffit de regarder les briques de bases.

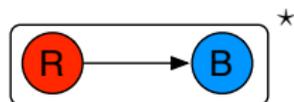
Définition (cohérence)

Un MSC causal est cohérent si :

- deux événements sont ordonnés \Rightarrow il existe une chaîne d'interactions qui les relie ;
- deux événements ne sont pas ordonnés \Rightarrow les processus qui les exécutent ne modifient pas une même variable partagée.

Pour les HMSC causaux, il suffit de regarder les briques de bases.

Exemple : $D_1 = \{(R, R), (B, B), (R, B), (B, R)\}$



cohérent



non cohérent

Définition (cohérence)

Un MSC causal est cohérent si :

- deux événements sont ordonnés \Rightarrow il existe une chaîne d'interactions qui les relie ;
- deux événements ne sont pas ordonnés \Rightarrow les processus qui les exécutent ne modifient pas une même variable partagée.

Pour les HMSC causaux, il suffit de regarder les briques de bases.

Exemple : $D_1 = \{(R, R), (B, B)\}$



non cohérent

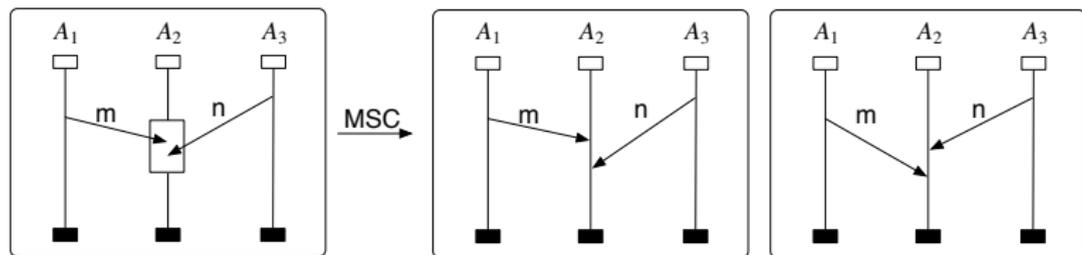


cohérent

Modéliser : Expressivité (1/2)

On s'intéresse aux exécutions des machines :

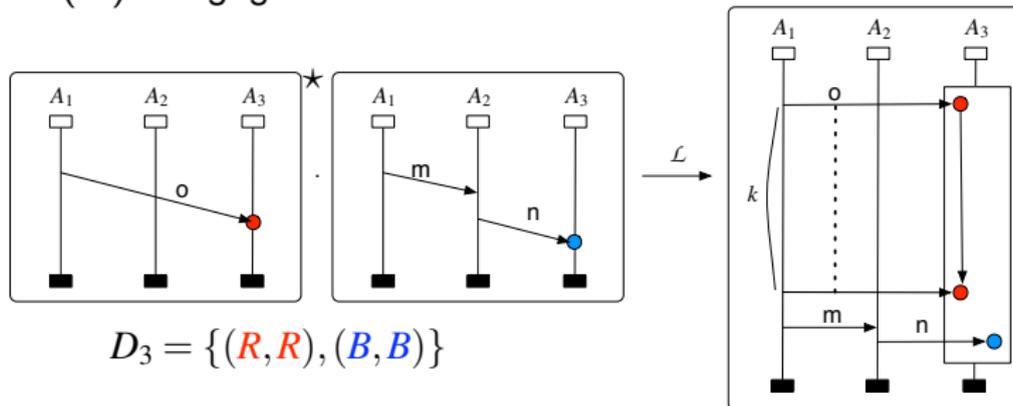
1. $MSC(u)$: langage de MSC d'un MSC causal u



Modéliser : Expressivité (1/2)

On s'intéresse aux exécutions des machines :

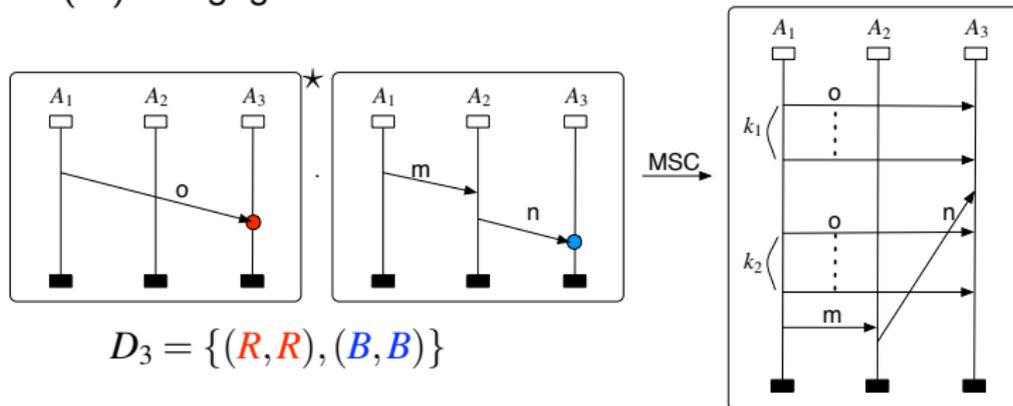
1. $MSC(u)$: langage de MSC d'un MSC causal u
2. $MSC(M)$: langage de MSC d'un HMSC causal M



Modéliser : Expressivité (1/2)

On s'intéresse aux exécutions des machines :

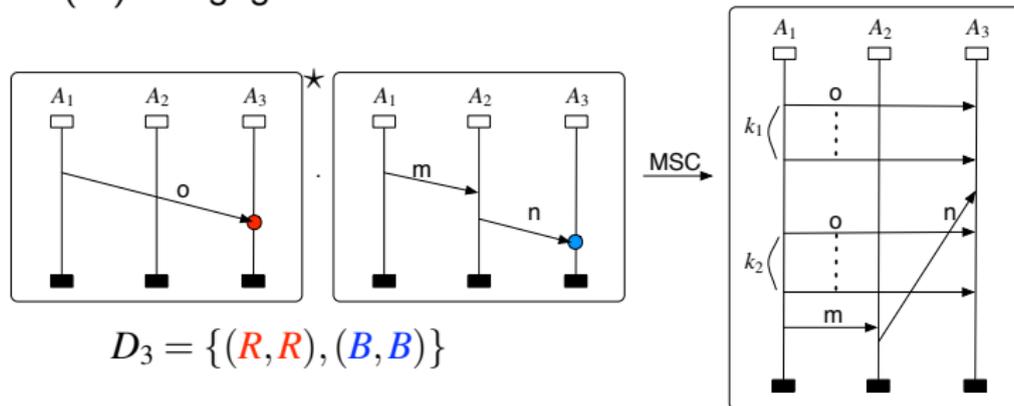
1. $MSC(u)$: langage de MSC d'un MSC causal u
2. $MSC(M)$: langage de MSC d'un HMSC causal M



Modéliser : Expressivité (1/2)

On s'intéresse aux exécutions des machines :

1. $MSC(u)$: langage de MSC d'un MSC causal u
2. $MSC(M)$: langage de MSC d'un HMSC causal M



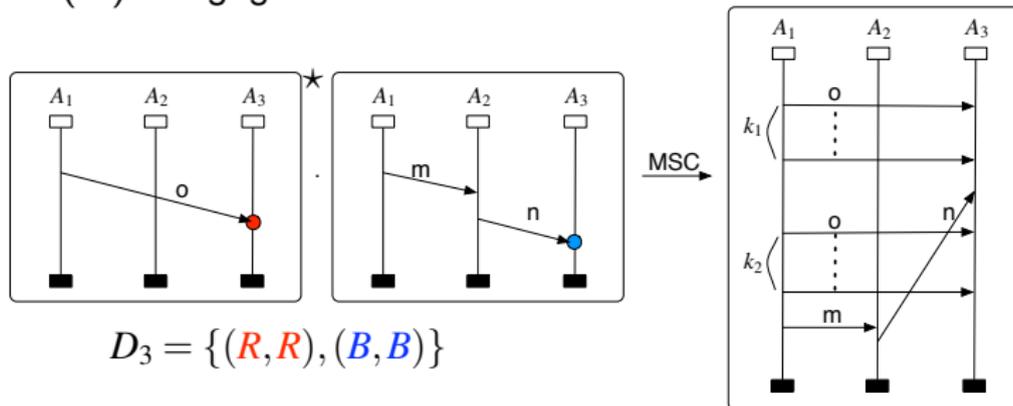
Expressivité des HMSC causaux cohérents

- plus expressif que HMSC [Rudolph96, Reniers98]
- incomparable avec HMSC compositionnels sûrs [Genest04]
- incomparable avec $EMSO(\prec_m, (\leq_i)_{A_i})$ [Bollig05]

Modéliser : Expressivité (1/2)

On s'intéresse aux exécutions des machines :

1. $MSC(u)$: langage de MSC d'un MSC causal u
2. $MSC(M)$: langage de MSC d'un HMSC causal M



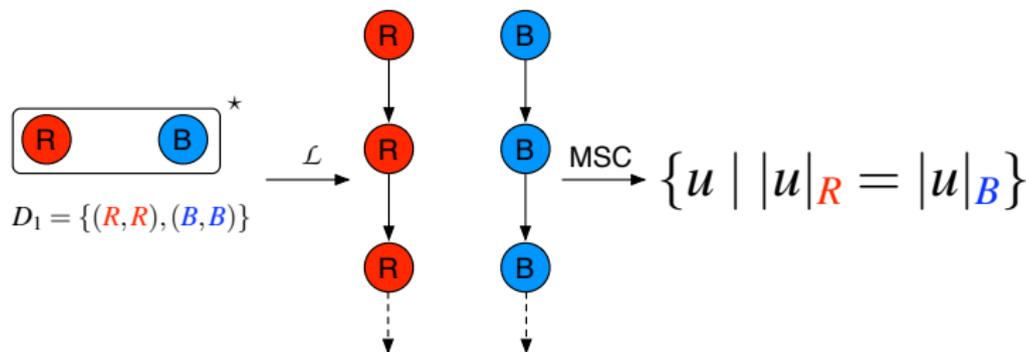
Expressivité des HMSC causaux cohérents

- plus expressif que HMSC [Rudolph96, Reniers98]
- incomparable avec HMSC compositionnels sûrs [Genest04]
- incomparable avec EMSO($\prec_m, (\leq_i)_{A_i}$) [Bollig05]

Modéliser : Expressivité (1/2)

On s'intéresse aux exécutions des machines :

1. $MSC(u)$: langage de MSC d'un MSC causal u
2. $MSC(M)$: langage de MSC d'un HMSC causal M



Expressivité des HMSC causaux cohérents

- plus expressif que HMSC [Rudolph96, Reniers98]
- incomparable avec HMSC compositionnels sûrs [Genest04]
- incomparable avec $EMSO(\prec_m, (\leq_i)_{A_i})$ [Bollig05]

Modéliser : Expressivité (2/2)

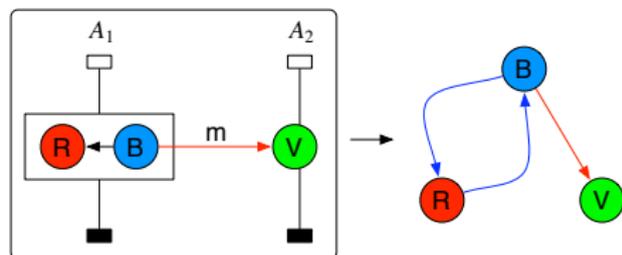
Définition (GI)

$GI(u, \cup D_i)$: Graphe d'Interactions :

sommets : actions de u ;

arêtes : interactions possibles entre les processus qui font ces actions.

$$D_1 = \{(R, R), (B, B), \underline{(R, B)}, \underline{(B, R)}\}$$



Définition (régularité faible)

Soient $(D_i)_{A_i}$ fixées. Un HMSC causal M est faiblement régulier si, pour toutes les sous-expressions N^* de M , pour tous les MSC causaux u de $\mathcal{L}(N)$: $GI(u, \cup D_i)$ est fortement connexe.

Théorème (mise en œuvre)

pour $(D_i)_{A_i}$ fixées :

\mathcal{L} est le langage d'un HMSC causal cohérent et faiblement régulier \Rightarrow

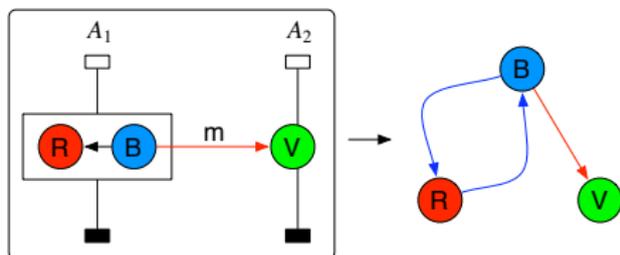
\mathcal{L} est le langage d'interactions d'un automate mixte à canaux bornés .

Modéliser : Expressivité (2/2)

Définition (GI)

$GI(u, \cup D_i)$: Graphe d'Interactions :
sommets : actions de u ;
arêtes : interactions possibles entre les processus qui font ces actions.

$$D_1 = \{(R, R), (B, B), \underline{(R, B)}, \underline{(B, R)}\}$$



Définition (régularité faible)

Soient $(D_i)_{A_i}$ fixées. Un HMSC causal M est **faiblement régulier** si, pour toutes les sous-expressions N^* de M , pour tous les MSC causaux u de $\mathcal{L}(N)$: $GI(u, \cup D_i)$ est **fortement connexe**.

Théorème (mise en œuvre)

pour $(D_i)_{A_i}$ fixées :

\mathcal{L} est le langage d'un HMSC causal cohérent et faiblement régulier \Rightarrow

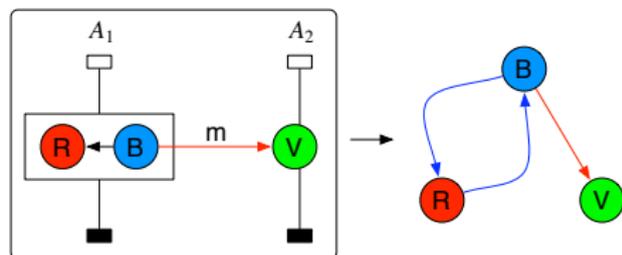
\mathcal{L} est le langage d'interactions d'un automate mixte à canaux bornés .

Modéliser : Expressivité (2/2)

Définition (GI)

$GI(u, \cup D_i)$: *Graphe d'Interactions* :
sommets : actions de u ;
arêtes : interactions possibles entre les processus qui font ces actions.

$$D_1 = \{(R, R), (B, B), \underline{(R, B)}, \underline{(B, R)}\}$$



Définition (régularité faible)

Soient $(D_i)_{A_i}$ fixées. Un HMSC causal M est faiblement régulier si, pour toutes les sous-expressions N^* de M , pour tous les MSC causaux u de $\mathcal{L}(N)$: $GI(u, \cup D_i)$ est fortement connexe.

Théorème (mise en œuvre)

pour $(D_i)_{A_i}$ fixées :

\mathcal{L} est le langage d'un HMSC causal cohérent et faiblement régulier \Rightarrow

\mathcal{L} est le langage d'interactions d'un automate mixte à canaux bornés .

Résumé :

- les **HMSC causaux** permettent d'exprimer de nouveaux langages de MSC qui correspondent à des modèles locaux mixtes ;
- les HMSC causaux cohérents et faiblement réguliers peuvent être mis en œuvres par des automates mixtes à canaux bornés ;
 - ▶ adaptation des preuves de [Ochmanski85,MP99,Genest04]
- + Les langage de linéarisations de HMSC causaux cohérents et faiblement réguliers sont reconnaissables par des automates finis :
 - ▶ la vérification et la supervision sont décidables.

Perspectives :

- Etendre les résultats de mise en œuvre à des systèmes à canaux non bornés.

Résumé :

- les HMSC causaux permettent d'exprimer de nouveaux langages de MSC qui correspondent à des modèles locaux mixtes ;
- les **HMSC causaux cohérents et faiblement réguliers** peuvent être mis en œuvre par des automates mixtes à canaux bornés ;
 - ▶ adaptation des preuves de [Ochmanski85,MP99,Genest04]
- + Les langage de linéarisations de HMSC causaux cohérents et faiblement réguliers sont reconnaissables par des automates finis :
 - ▶ la vérification et la supervision sont décidables.

Perspectives :

- Etendre les résultats de mise en œuvre à des systèmes à canaux non bornés.

Résumé :

- les HMSC causaux permettent d'exprimer de nouveaux langages de MSC qui correspondent à des modèles locaux mixtes ;
- les HMSC causaux cohérents et faiblement réguliers peuvent être mis en œuvres par des automates mixtes à canaux bornés ;
 - ▶ adaptation des preuves de [Ochmanski85,MP99,Genest04]
- + Les langage de linéarisations de **HMSC causaux cohérents et faiblement réguliers** sont reconnaissables par des automates finis :
 - ▶ la vérification et la supervision sont décidables.

Perspectives :

- Etendre les résultats de mise en œuvre à des systèmes à canaux non bornés.

Résumé :

- les HMSC causaux permettent d'exprimer de nouveaux langages de MSC qui correspondent à des modèles locaux mixtes ;
- les HMSC causaux cohérents et faiblement réguliers peuvent être mis en œuvres par des automates mixtes à canaux bornés ;
 - ▶ adaptation des preuves de [Ochmanski85,MP99,Genest04]
- + Les langage de linéarisations de HMSC causaux cohérents et faiblement réguliers sont reconnaissables par des automates finis :
 - ▶ la vérification et la supervision sont décidables.

Perspectives :

- Etendre les résultats de mise en œuvre à des systèmes à canaux **non bornés**.

Plan de l'exposé

Modèles locaux et leurs interactions globales

Interactions par mémoire partagée

Interactions par échange de messages

Interactions mixtes

Modéliser

HMSC causaux

Expressivité

Critiques et perspectives

Vérifier

Vérification complète

Vérification partielle

Critiques et perspectives

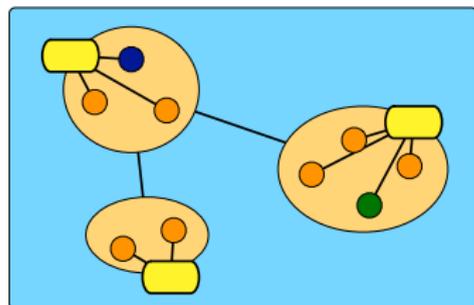
Superviser

Diagnostic

Critiques et perspectives

Conclusion

modèle M



- **Propriété φ**
- “Est-ce qu’une action qu’effectue le processus **bleu** est toujours suivie d’une action sur le processus **vert** ?”
- $M \models \varphi$

modèle M : ■ HMSC causal cohérent ;

■ projection d’un HMSC causal cohérent sur Σ_φ .

propriété φ : HMSC causal cohérent (pas de négation)

Questions : ■ si φ décrit de mauvaises interactions :

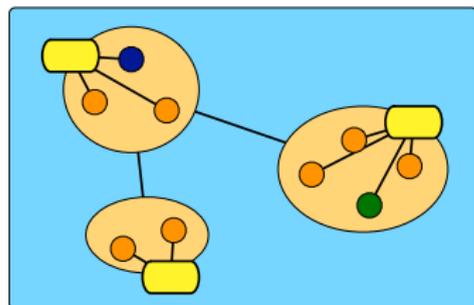
$$MSC(M) \cap MSC(\varphi) = \emptyset$$

■ si φ décrit de bonnes interactions :

$$MSC(M) \subseteq MSC(\varphi)$$

Vérifier : contexte

modèle M



- **Propriété φ**
- “Est-ce qu’une action qu’effectue le processus **bleu** est toujours suivie d’une action sur le processus **vert** ?”
- $M \models \varphi$

modèle M : ■ HMSC causal cohérent ;
■ projection d’un HMSC causal cohérent sur Σ_φ .

propriété φ : HMSC causal cohérent (pas de négation)

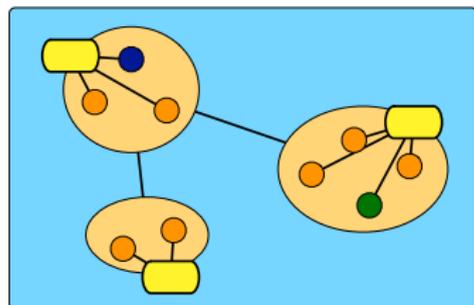
Questions : ■ si φ décrit de mauvaises interactions :

$$MSC(M) \cap MSC(\varphi) = \emptyset$$

■ si φ décrit de bonnes interactions :

$$MSC(M) \subseteq MSC(\varphi)$$

modèle M



- Propriété φ
- “Est-ce qu’une action qu’effectue le processus **bleu** est toujours suivie d’une action sur le processus **vert** ?”
- $M \models \varphi$

modèle M : ■ HMSC causal cohérent ;

■ projection d’un HMSC causal cohérent sur Σ_φ .

propriété φ : HMSC causal cohérent (pas de négation)

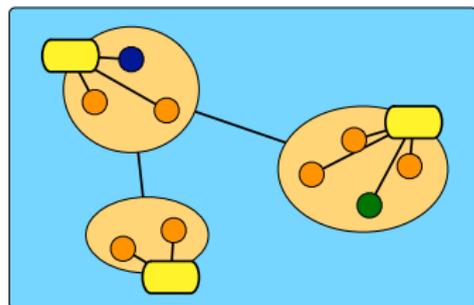
Questions : ■ si φ décrit de mauvaises interactions :

$$MSC(M) \cap MSC(\varphi) = \emptyset$$

■ si φ décrit de bonnes interactions :

$$MSC(M) \subseteq MSC(\varphi)$$

modèle M



- **Propriété φ**
- “Est-ce qu’une action qu’effectue le processus **bleu** est toujours suivie d’une action sur le processus **vert** ?”
- $M \models \varphi$

modèle M : ■ HMSC causal cohérent ;

■ projection d’un HMSC causal cohérent sur Σ_{φ} .

propriété φ : HMSC causal cohérent (pas de négation)

Questions : ■ si φ décrit de mauvaises interactions :

$$MSC(M) \cap MSC(\varphi) = \emptyset$$

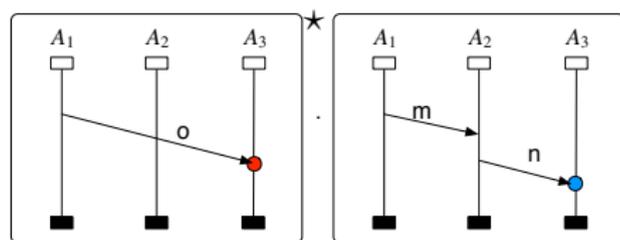
■ si φ décrit de bonnes interactions :

$$MSC(M) \subseteq MSC(\varphi)$$

Vérification complète (1/2)

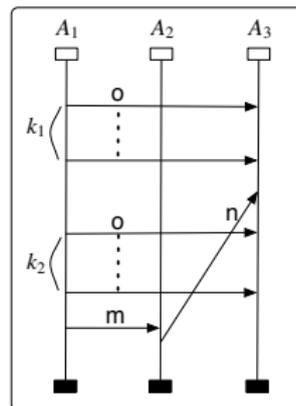
Définition (globalement coopératifs)

Soient $(D_i)_{A_i}$ fixées. Un HMSC causal M est *globalement coopératif* si, pour toutes les sous-expressions N^* de M , pour tous les MSC causaux u de $\mathcal{L}(N)$: $Gl(u, \cup D_i)$ est *connexe*.



$$D_3 = \{(R, R), (B, B)\}$$

MSC



Vérification complète (1/2)

Définition (globalement coopératifs)

Soient $(D_i)_{A_i}$ fixées. Un HMSC causal M est globalement coopératif si, pour toutes les sous-expressions N^* de M , pour tous les MSC causaux u de $\mathcal{L}(N)$: $Gl(u, \bigcup D_i)$ est connexe.

Théorème (vérification) [CONCUR'07]

Soient $(D_i)_{A_i}$ fixées et M, φ HMSC causaux cohérents.

Si φ est globalement coopératif alors :

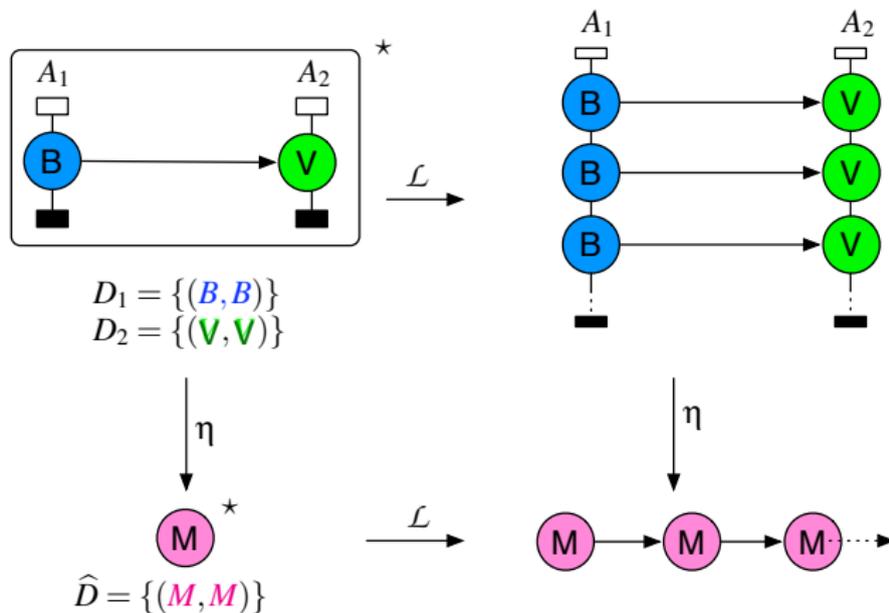
- Savoir si $MSC(M) \cap MSC(\varphi) = \emptyset$ est PSPACE-complet ;
- Savoir si $MSC(M) \subseteq MSC(\varphi)$ est EXPSPACE-complet.

complexité comparable à la vérification de modèles séquentiels mais avec des modèles exponentiellement plus concis

Vérification complète (2/2)

il existe un **isomorphisme** η :

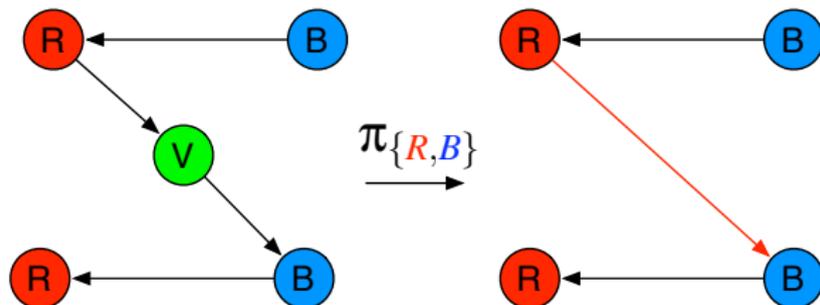
HMSC causaux \rightarrow langages rationnels de traces



Définition (Projection)

$\pi_{\Sigma_\varphi}(u)$ projection d'un MSC causal u sur Σ_φ :

- restriction des événements à ceux étiquetés par Σ_φ ;
- l'ordre est conservé.

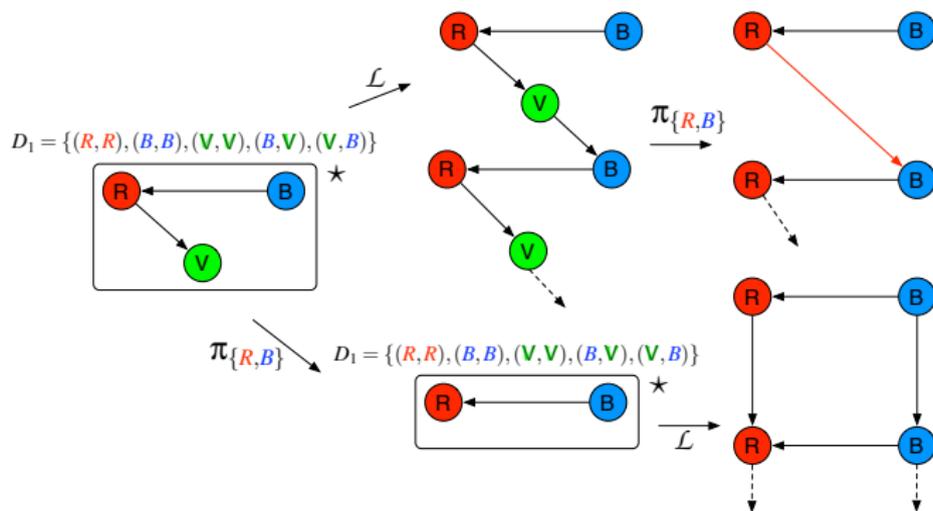


Définition (Projection)

$\pi_{\Sigma_\phi}(u)$ projection d'un MSC causal u sur Σ_ϕ :

- restriction des événements à ceux étiquetés par Σ_ϕ ;
- l'ordre est conservé.

Problème : les HMSC causaux ne sont pas stables par projection



Proposition

Soient M un HMSC causal cohérent et Σ_φ un ensemble d'actions.

On sait décider s'il existe un HMSC causal (pas forcément cohérent) M' tel que :

$$\pi_{\Sigma_\varphi}(MSC(M)) = MSC(M')$$

Théorème (vérification partielle)

Soient $(D_i)_{A_i}$ fixées et M, φ HMSC causaux cohérents.

Si φ est globalement coopératif alors :

- Savoir si $\pi_{\Sigma_\varphi}(MSC(M)) \subseteq MSC(\varphi)$ est décidable.

Proposition

Soient M un HMSC causal cohérent et Σ_φ un ensemble d'actions.

On sait décider s'il existe un HMSC causal (pas forcément cohérent) M' tel que :

$$\pi_{\Sigma_\varphi}(MSC(M)) = MSC(M')$$

Théorème (vérification partielle)

Soient $(D_i)_{A_i}$ fixées et M, φ HMSC causaux cohérents.

Si φ est globalement coopératif alors :

- Savoir si $\pi_{\Sigma_\varphi}(MSC(M)) \subseteq MSC(\varphi)$ est décidable.

Résumé :

- les **HMSC causaux cohérents** permettent de faire de la vérification de systèmes non bornés, vis-à-vis de propriétés données sous la forme de **HMSC causaux cohérents et globalement coopératifs**
 - ▶ Adaptation des preuves de [HM00,Morin02,GMSZ02]
- si φ n'est pas cohérent :
 $\mathcal{L}(M) = \mathcal{L}(\varphi) \Rightarrow MSC(M) = MSC(\varphi)$
et donc : $\mathcal{L}(M) \neq \mathcal{L}(\varphi) \not\Rightarrow MSC(M) \neq MSC(\varphi)$
- + complexité comparable à la vérification classique mais avec des modèles exponentiellement plus concis

Perspectives :

- Faire de la vérification sur les langages de MSC dans le cas non cohérent ;
- Faire de la vérification de HMSC causaux où l'on s'autorise à séparer l'envoi et la réception de certains messages (et à les recoller lors de la composition).

Résumé :

- les HMSC causaux cohérents permettent de faire de la vérification de systèmes non bornés, vis-à-vis de propriétés données sous la forme de HMSC causaux cohérents et globalement coopératifs
 - ▶ Adaptation des preuves de [HM00,Morin02,GMSZ02]
- si φ n'est pas cohérent :
 $\mathcal{L}(M) = \mathcal{L}(\varphi) \Rightarrow \text{MSC}(M) = \text{MSC}(\varphi)$
et donc : $\mathcal{L}(M) \neq \mathcal{L}(\varphi) \not\Rightarrow \text{MSC}(M) \neq \text{MSC}(\varphi)$
- + complexité comparable à la vérification classique mais avec des modèles exponentiellement plus concis

Perspectives :

- Faire de la vérification sur les langages de MSC dans le cas non cohérent ;
- Faire de la vérification de HMSC causaux où l'on s'autorise à séparer l'envoi et la réception de certains messages (et à les recoller lors de la composition).

Résumé :

- les HMSC causaux cohérents permettent de faire de la vérification de systèmes non bornés, vis-à-vis de propriétés données sous la forme de HMSC causaux cohérents et globalement coopératifs
 - ▶ Adaptation des preuves de [HM00,Morin02,GMSZ02]
- si φ n'est pas cohérent :
 $\mathcal{L}(M) = \mathcal{L}(\varphi) \Rightarrow MSC(M) = MSC(\varphi)$
et donc : $\mathcal{L}(M) \neq \mathcal{L}(\varphi) \not\Rightarrow MSC(M) \neq MSC(\varphi)$
- + complexité comparable à la vérification classique mais avec des modèles exponentiellement plus concis

Perspectives :

- Faire de la vérification sur les langages de MSC dans le cas non cohérent ;
- Faire de la vérification de HMSC causaux où l'on s'autorise à séparer l'envoi et la réception de certains messages (et à les recoller lors de la composition).

Résumé :

- les HMSC causaux cohérents permettent de faire de la vérification de systèmes non bornés, vis-à-vis de propriétés données sous la forme de HMSC causaux cohérents et globalement coopératifs
 - ▶ Adaptation des preuves de [HM00,Morin02,GMSZ02]
- si φ n'est pas cohérent :
 $\mathcal{L}(M) = \mathcal{L}(\varphi) \Rightarrow \text{MSC}(M) = \text{MSC}(\varphi)$
et donc : $\mathcal{L}(M) \neq \mathcal{L}(\varphi) \not\Rightarrow \text{MSC}(M) \neq \text{MSC}(\varphi)$
- + complexité comparable à la vérification classique mais avec des modèles exponentiellement plus concis

Perspectives :

- Faire de la vérification sur les langages de MSC dans le cas non cohérent ;
- Faire de la vérification de HMSC causaux où l'on s'autorise à séparer l'envoi et la réception de certains messages (et à les recoller lors de la composition).

Résumé :

- les HMSC causaux cohérents permettent de faire de la vérification de systèmes non bornés, vis-à-vis de propriétés données sous la forme de HMSC causaux cohérents et globalement coopératifs
 - ▶ Adaptation des preuves de [HM00,Morin02,GMSZ02]
- si φ n'est pas cohérent :
 $\mathcal{L}(M) = \mathcal{L}(\varphi) \Rightarrow MSC(M) = MSC(\varphi)$
et donc : $\mathcal{L}(M) \neq \mathcal{L}(\varphi) \not\Rightarrow MSC(M) \neq MSC(\varphi)$
- + complexité comparable à la vérification classique mais avec des modèles exponentiellement plus concis

Perspectives :

- Faire de la vérification sur les langages de MSC dans le cas non cohérent ;
- Faire de la vérification de HMSC causaux où l'on s'autorise à séparer l'envoi et la réception de certains messages (et à les recoller lors de la composition).

Plan de l'exposé

Modèles locaux et leurs interactions globales

Interactions par mémoire partagée

Interactions par échange de messages

Interactions mixtes

Modéliser

HMSC causaux

Expressivité

Critiques et perspectives

Vérifier

Vérification complète

Vérification partielle

Critiques et perspectives

Superviser

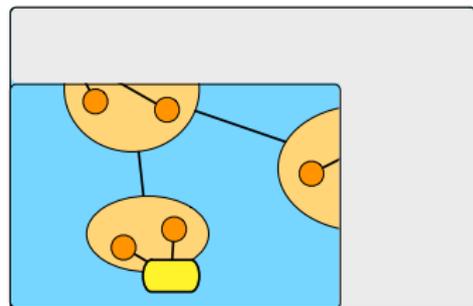
Diagnostic

Critiques et perspectives

Conclusion

Superviser : contexte

modèle M



Architecture

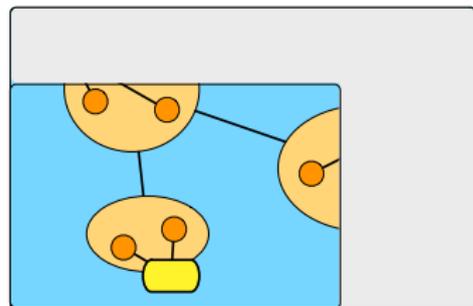
Certains processus sont des capteurs
qui envoient à un superviseur :

- les actions effectuées ;
- les causalités connues.

Le superviseur calcule un diagnostic
 $Diag(O, M)$ et vérifie φ

- Propriété φ
observation partielle O limitée à Σ_o
- “Est-ce que l’observation permet de déduire
que le système a toujours fait une action sur
le processus **bleu** avant de faire une action
sur le processus **vert** ?”
- $Diag(O, M) \models \varphi$

modèle M



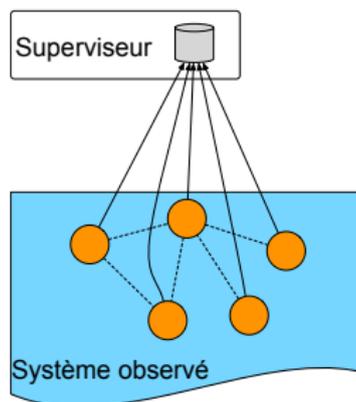
Architecture

Certains processus sont des capteurs qui envoient à un superviseur :

- les **actions effectuées** ;
- les **causalités connues**.

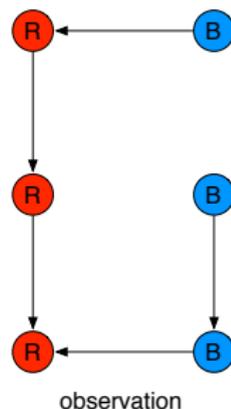
Le superviseur calcule un diagnostic $Diag(O, M)$ et vérifie φ

- **Propriété φ**
observation partielle O limitée à Σ_o
- “Est-ce que l’observation permet de déduire que le système a toujours fait une action sur le processus **bleu** avant de faire une action sur le processus **vert** ?”
- $Diag(O, M) \models \varphi$



Définition (observation)

une observation est un MSC causal



Définition (M -diagnostic)

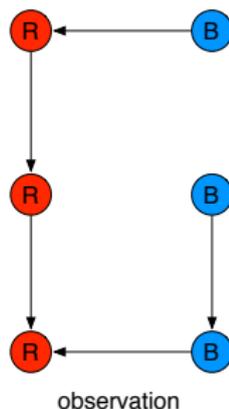
u est un M -diagnostic de O si :

- (i) $u \in \mathcal{L}(M)$;
- (ii) $\pi_{\Sigma_o}(u)$ est une extension d'ordre de O .

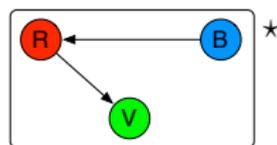
Diagnostic (1/3)

Définition (observation)

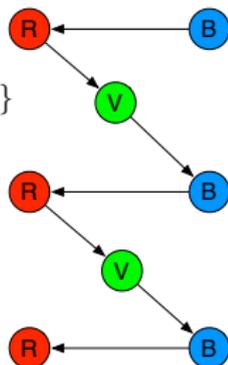
une observation est un MSC causal



$$D_1 = \{\dots(B, V), (V, B)\dots\}$$



modèle M



$u \in \mathcal{L}(M)$

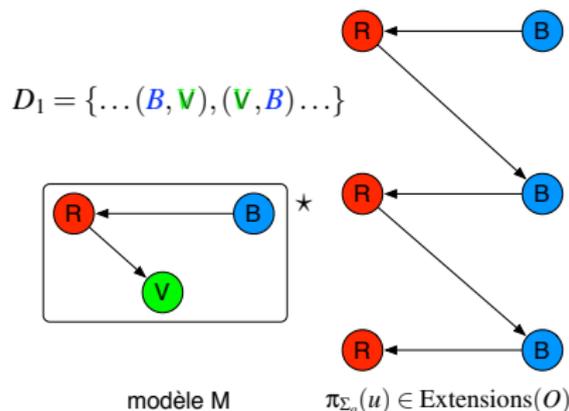
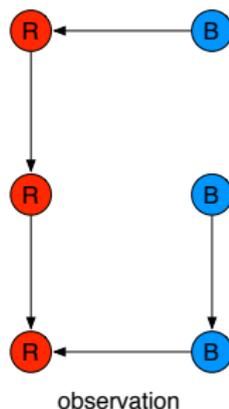
Définition (M-diagnostic)

u est un M-diagnostic de O si :

- (i) $u \in \mathcal{L}(M)$;
- (ii) $\pi_{\Sigma_o}(u)$ est une extension d'ordre de O.

Définition (observation)

une observation est un MSC causal



Définition (M-diagnostic)

u est un M-diagnostic de O si :

- (i) $u \in \mathcal{L}(M)$;
- (ii) $\pi_{\Sigma_o}(u)$ est une extension d'ordre de O .

Diagnostic (2/3)

$$\begin{aligned}\mathcal{L}(\text{Diag}(O, M)) &= \{u \mid u \text{ est un } M\text{-diagnostic de } O\} \\ &= \mathcal{L}(M) \cap \pi_{\Sigma_o}^{-1}(\text{Extensions}(O)) \\ &\simeq \hat{\pi}_{\Sigma_o}^{-1}(\mathcal{L}(\hat{\pi}_{\Sigma_o}(M)) \cap \text{Extensions}(O))\end{aligned}$$

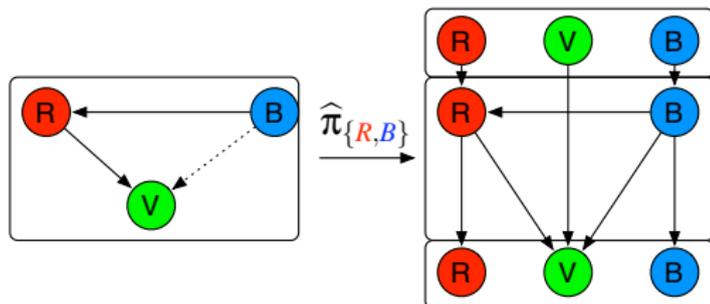
Diagnostic (2/3)

$$\begin{aligned}\mathcal{L}(\text{Diag}(O, M)) &= \{u \mid u \text{ est un } M\text{-diagnostic de } O\} \\ &= \mathcal{L}(M) \cap \pi_{\Sigma_o}^{-1}(\text{Extensions}(O)) \\ &\simeq \hat{\pi}_{\Sigma_o}^{-1}(\mathcal{L}(\hat{\pi}_{\Sigma_o}(M)) \cap \text{Extensions}(O))\end{aligned}$$

HMSC causaux encapsulés

Définition (projection)

$\hat{\pi}_{\Sigma_o}$: on garde en mémoire
la causalité pour chaque
action de Σ



Diagnostic (2/3)

$$\begin{aligned}\mathcal{L}(\text{Diag}(O, M)) &= \{u \mid u \text{ est un } M\text{-diagnostic de } O\} \\ &= \mathcal{L}(M) \cap \pi_{\Sigma_o}^{-1}(\text{Extensions}(O)) \\ &\simeq \hat{\pi}_{\Sigma_o}^{-1}(\mathcal{L}(\hat{\pi}_{\Sigma_o}(M)) \cap \text{Extensions}(O))\end{aligned}$$

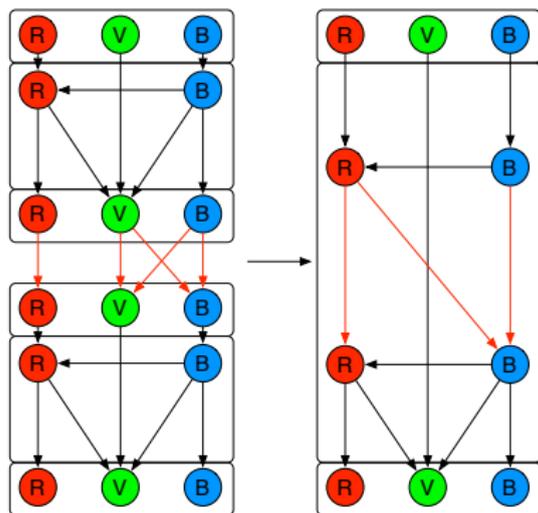
HMSC causaux encapsulés

Définition (composition)

*on compose les blocs
intermédiaires puis on les efface*

Exemple :

$$D_1 = \{\dots (V, B), (B, V) \dots\}$$



Diagnostic (2/3)

$$\begin{aligned}\mathcal{L}(\text{Diag}(O, M)) &= \{u \mid u \text{ est un } M\text{-diagnostic de } O\} \\ &= \mathcal{L}(M) \cap \pi_{\Sigma_o}^{-1}(\text{Extensions}(O)) \\ &\simeq \widehat{\pi}_{\Sigma_o}^{-1}(\mathcal{L}(\widehat{\pi}_{\Sigma_o}(M)) \cap \text{Extensions}(O))\end{aligned}$$

Théorème (stabilité) [FORTE'07]

Pour tout HMSC causal cohérent M :

$$\mathcal{L}(\widehat{\pi}_{\Sigma_o}(M)) = \widehat{\pi}_{\Sigma_o}(\mathcal{L}(M))$$

Diagnostic (2/3)

$$\begin{aligned}\mathcal{L}(\text{Diag}(O, M)) &= \{u \mid u \text{ est un } M\text{-diagnostic de } O \} \\ &= \mathcal{L}(M) \cap \pi_{\Sigma_o}^{-1}(\text{Extensions}(O)) \\ &\simeq \widehat{\pi}_{\Sigma_o}^{-1}(\mathcal{L}(\widehat{\pi}_{\Sigma_o}(M)) \cap \text{Extensions}(O))\end{aligned}$$

Théorème (stabilité) [FORTE'07]

Pour tout HMSC causal cohérent M :

$$\mathcal{L}(\widehat{\pi}_{\Sigma_o}(M)) = \widehat{\pi}_{\Sigma_o}(\mathcal{L}(M))$$

Théorème (supervision) [WODES'06]

Soient $(D_i)_{A_i}$ fixées et M, φ HMSC causaux cohérents.

Si φ est globalement coopératif alors on peut décider si :

- $MSC(Diag(O, M)) \cap MSC(\varphi) = \emptyset$;
- $MSC(Diag(O, M)) \subseteq MSC(\varphi)$.

En utilisant le même genre de techniques :

Théorème (corrélation d'événements) [FORTE'07]

Savoir si deux événements observés sont ordonnés dans toutes les exécutions de M est décidable (CoNP-complet).

Théorème (supervision) [WODES'06]

Soient $(D_i)_{A_i}$ fixées et M, φ HMSC causaux cohérents.

Si φ est globalement coopératif alors on peut décider si :

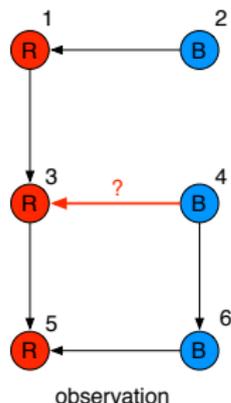
- $MSC(Diag(O, M)) \cap MSC(\varphi) = \emptyset$;
- $MSC(Diag(O, M)) \subseteq MSC(\varphi)$.

En utilisant le même genre de techniques :

Théorème (corrélation d'événements)

[FORTE'07]

Savoir si deux événements observés sont ordonnés dans **toutes** les exécutions de M est décidable (CoNP-complet).



Résumé :

- HMSC causaux *encapsulés* [FORTE'07]
 - ▶ stables par projection
 - ▶ structure qui permet d'utiliser des techniques de dépliage pour le diagnostic [BFJH03]
- opérations qui peuvent être précédées par un phase d'abstraction ou de filtrage [NOTERE'07] ;
- la construction a été mise en œuvre dans un prototype ;

Perspectives :

- faire de la corrélation d'événement probabilisée ;
- faire de la supervision à la volée.

Résumé :

- HMSC causaux *encapsulés* [FORTE'07]
 - ▶ stables par projection
 - ▶ structure qui permet d'utiliser des techniques de dépliage pour le diagnostic [BFJH03]
- opérations qui peuvent être précédées par un phase d'abstraction ou de filtrage [NOTERE'07] ;
- la construction a été mise en œuvre dans un prototype ;

Perspectives :

- faire de la corrélation d'événement probabilisée ;
- faire de la supervision à la volée.

Résumé :

- HMSC causaux *encapsulés* [FORTE'07]
 - ▶ stables par projection
 - ▶ structure qui permet d'utiliser des techniques de dépliage pour le diagnostic [BFJH03]
- opérations qui peuvent être précédées par un phase d'abstraction ou de filtrage [NOTERE'07] ;
- la construction a été mise en œuvre dans un prototype ;

Perspectives :

- faire de la corrélation d'événement probabilisée ;
- faire de la supervision à la volée.

Résumé :

- HMSC causaux *encapsulés* [FORTE'07]
 - ▶ stables par projection
 - ▶ structure qui permet d'utiliser des techniques de dépliage pour le diagnostic [BFJH03]
- opérations qui peuvent être précédées par un phase d'abstraction ou de filtrage [NOTERE'07] ;
- la construction a été mise en œuvre dans un prototype ;

Perspectives :

- faire de la corrélation d'événement probabilisée ;
- faire de la supervision à la volée.

Résumé :

- HMSC causaux *encapsulés* [FORTE'07]
 - ▶ stables par projection
 - ▶ structure qui permet d'utiliser des techniques de dépliage pour le diagnostic [BFJH03]
- opérations qui peuvent être précédées par un phase d'abstraction ou de filtrage [NOTERE'07] ;
- la construction a été mise en œuvre dans un prototype ;

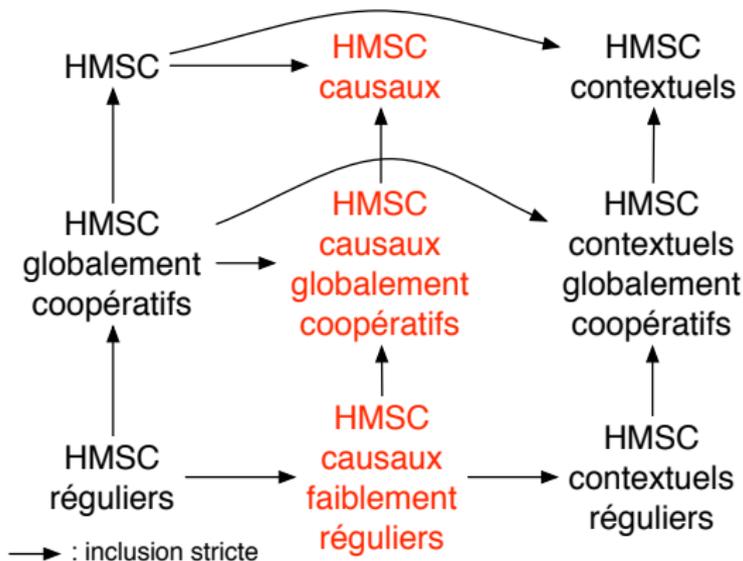
Perspectives :

- faire de la corrélation d'événement probabilisée ;
- faire de la supervision à la volée.

Résumé

HMSC causaux cohérents et faiblement réguliers : mise en œuvre par des automates mixtes à canaux bornés ;

Tous les HMSC causaux cohérents : vérification et supervision pour des propriétés décrites sous la forme de HMSC causaux cohérents globalement coopératifs.



- modéliser : mise en œuvre de systèmes non bornés ;
- vérifier : vérification de HMSC causaux contextuels ;
- superviser : construire un diagnostic à la volée.

Perspectives générales :

- augmenter l'expressivité des modèles en gardant la vérification et la supervision décidables dans le but d'obtenir un modèle qui servirait de sémantique à des programmes parallèles et répartis ;
- construction de modèles globaux à partir de programmes existants en utilisant des techniques d'interprétation abstraite pour le comportement des canaux de communication.

Merci de votre attention !