Contrôle décentralisé de systèmes symboliques infinis sous observation partielle

Gabriel Kalyon, Tristan Le Gall, Hervé Marchand,Thierry Massart

Département d'informatique, Service des systèmes distribués
Université Libre de Bruxelles (U.L.B.), campus de la Plaine, 1050 Bruxelles, Belgique
{gkalyon,tlegall,tmassart}@ulb.ac.be

Projet VerTeCs
INRIA Rennes, Bretagne Atlantique, campus de Beaulieu, 35 042 Rennes Cedex, France
herve.marchand@irisa.fr

RÉSUMÉ. *Nous proposons des algorithmes permettant de synthétiser des contrôleurs décentralisés n'ayant qu'une observation partielle du système à contrôler. Ces systèmes, dont le nombre d'états peut être infini, sont modélisés par des Systèmes à Transitions Symboliques. Nous présentons des modèles de contrôleurs (non-bloquants) valides permettant d'assurer l'interdiction d'un ensemble d'états dans un cadre décentralisé. Pour obtenir des algorithmes pour ces problèmes, nous utilisons des techniques d'interprétation abstraite, qui induisent une surapproximation de l'ensemble des transitions à interdire. Notre outil SMACS permet de valider empiriquement nos méthodes et de montrer leurs faisabilité et efficacité.*

ABSTRACT. *We propose algorithms for the synthesis of decentralized controllers with partial observation of infinite state systems modelled by Symbolic Transition Systems. We provide models of safe controllers both for potentially deadlock and deadlock free controlled systems. To obtain algorithms for these problems, we use abstract interpretation techniques which provide over-approximations of the transitions set to be disabled. Our tool SMACS allowed us to make an empirical validation of our methods and show their feasibility and efficiency.*

MOTS-CLÉS : *Systèmes à Transitions Symboliques, Synthèse de contrôleurs décentralisés, Observation Partielle, Interprétation Abstraite.*

KEYWORDS: *Symbolic Transition Systems, Decentralized Controller Synthesis, Partial Observation, Abstract Interpretation.*

JESA – 43/2009. MSR 2009, pages 805 à 819

1. Introduction

Nous nous intéressons au *problème d'interdiction d'états* (Takai *et al.*, 10 July 1998) dans le domaine de la synthèse de contrôleurs de Systèmes à Événements Discrets (Ramadge *et al.*, 1989). Le but est de synthétiser un contrôleur pour empêcher le système d'atteindre un ensemble donné d'états interdits ; nous nous plaçons dans le cadre où le système à contrôler a un nombre infini d'états et est partiellement observé par le(s) contrôleur(s). Nous utilisons le modèle des Systèmes à Transitions Symboliques (STS) (Henzinger *et al.*, 2005) pour décrire les systèmes à contrôler. Ces systèmes sont composés de variables, dont le domaine peut être infini, et de transitions gardées, qui mettent à jour les variables. Étant donné que les propriétés de contrôle sont définies sur les états du système, il semble plus naturel que le contrôleur observe le système à travers ses états (Ramadge *et al.*, 1989). De plus, le contrôleur ne dispose que d'une *observation partielle* du système due par exemple à l'imprécision du matériel d'observation. Cette observation partielle peut être modélisée par un *masque* (Kumar *et al.*, 1993) correspondant à une fonction de l'espace d'états vers un espace (infini) d'observations. Nous supposons ici que l'observation et le contrôle sont *décentralisés* ; cela signifie que n contrôleurs ont chacun une vue propre et partielle du système et ne peuvent chacun en contrôler qu'une partie. La qualité des contrôleurs peut être mesurée par les critères de *permissivité*, qui demandent, par exemple, que l'ensemble des transitions autorisées par les contrôleurs soit maximal (Takai *et al.*, 10 July 1998).

État de l'art. La *synthèse de contrôleurs centralisés de systèmes finis avec une observation partielle des actions* a été largement étudiée dans la littérature (cf. (Cassandras *et al.*, 2008) pour un aperçu des différents résultats). La synthèse pour les systèmes finis avec une observation partielle sur les états a été introduite par Kumar *et al.* dans (Kumar *et al.*, 1993). La notion de masque y est définie comme une partition de l'espace d'états (c'est-à-dire des ensembles disjoints d'états indistingables). Dans (Takai *et al.*, 10 July 1998), les auteurs définissent la notion de *M-contrôlabilité* et ils présentent une condition nécessaire et suffisante (basée sur cette notion) pour décider de l'existence d'un contrôleur, dont le système contrôlé résultant permet d'atteindre exactement un ensemble Q d'états autorisés. La *synthèse de contrôleurs centralisés de systèmes infinis avec une observation parfaite* a été étudiée à différents niveaux. Dans (Kumar *et al.*, 2005), Kumar et Garg étendent (Kumar *et al.*, 1993) pour traiter le cas de systèmes infinis. Ils prouvent que, dans ce cas, le problème de contrôle d'interdiction d'états est indécidable. Ils montrent également que le problème peut être résolu dans le cas des réseaux de Petri, lorsque l'ensemble Bad est clos par le haut. Dans (Le Gall *et al.*, 2005), des techniques symboliques sont utilisées pour contrôler des systèmes infinis modélisés par des STS. Des techniques d'interprétation abstraite (Cousot *et al.*, 1977; Jeannet, 2003) sont également utilisées pour assurer que le système contrôlé puisse être effectivement calculé. Ces techniques ont été étendues dans (Kalyon *et al.*, 2009a) en prenant en compte l'aspect d'observation partielle. La *synthèse de contrôleurs décentralisés de systèmes finis avec observation partielle des états* a été étudiée dans (Takai *et al.*, 1994). Dans ce travail, le contrôle est réalisé par

des contrôleurs locaux ayant chacun leur propre observation du système et le but est de satisfaire une spécification globale Q de contrôle. Les auteurs définissent la notion de n-*observabilité* et ils présentent une condition nécessaire et suffisante pour décider de l'existence de contrôleurs décentralisés, dont le système contrôlé résultant permet d'atteindre exactement l'ensemble Q des états autorisés.

Nous étendons ici les résultats de (Kalyon *et al.*, 2009a) pour traiter la synthèse de contrôleurs décentralisés de systèmes infinis avec observation partielle des états. Afin de traiter des espaces d'états infinis, les algorithmes présentés sont symboliques. Cela signifie qu'ils n'énumèrent pas les états, mais qu'ils manipulent les variables du système au moyen de calculs symboliques et de transformateurs de prédicats. Nous utilisons des techniques d'interprétation abstraite pour obtenir des algorithmes *effectifs* (qui terminent toujours). Nos algorithmes ne sont pas optimaux, puisque le problème de contrôle d'interdiction d'états pour des systèmes infinis est indécidable et que des abstractions sont utilisées pour assurer la terminaison des algorithmes. Notons que les *domaines abstraits et concrets* peuvent être infinis. Les algorithmes ont été implémentés dans notre outil SMACS, qui a permis de les valider expérimentalement.

Dans la section 2, nous introduisons notre modèle. Dans la section 3, nous définissons les mécanismes de contrôle utilisés et le problème de contrôle décentralisé d'interdiction d'états. Dans la section 4, nous présentons des algorithmes, qui résolvent nos problèmes, mais qui ne terminent pas toujours. Dans la section 5, nous expliquons comment obtenir des algorithmes effectifs en utilisant des techniques d'interprétation abstraite. Finalement, dans la section 6, nous comparons le contrôle centralisé et le contrôle décentralisé.

2. Système à transitions symboliques

Les Systèmes à Transitions Symboliques (STS) modélisent des systèmes à transitions avec variables dont le domaine est potentiellement infini, permettant ainsi de représenter des systèmes infinis.

Variables, Prédicats, Assignations. Nous supposons avoir un k-uple $V = \langle v_1, \ldots, v_k \rangle$ (k constante) de variables typées ; le domaine (infini) d'une variable v est noté $\mathcal{D}_v$. $\mathcal{D}_V$ dénote $\prod_{i \in [1,k]} \mathcal{D}_{v_i}$. Une *valuation* $\vec{\nu}$ de V est une k-uple $\langle \vec{\nu}_1, \ldots, \vec{\nu}_k \rangle \in \mathcal{D}_V$ et représente une assignation possible de valeurs pour les variables. Un prédicat sur le k-uple V est défini comme un sous-ensemble $P \subseteq \mathcal{D}_V$ (un ensemble d'états pour lesquels le prédicat est vrai). Le complément d'un ensemble $H \subseteq \mathcal{D}_V$ est noté par $\overline{H}$. La fonction préimage de $f : D_1 \mapsto D_2$ est notée $f^{-1} : D_2 \mapsto 2^{D_1}$ et est définie pour tout $d_2 \in D_2$ par $f^{-1}(d_2) = \{d_1 \in D_1 | f(d_1) = d_2\}$. Nous étendons naturellement une fonction $f : D_1 \mapsto D_2$ aux ensembles $H \subseteq D_1 : f(H) = \bigcup_{h \in H} f(h)$.

Définition de STS. Nos systèmes sont modélisés par des STS définis par :

Définition 1 (Système à Transitions Symboliques) *Un* système à transitions symboliques *(STS) est un tuple* $\mathcal{T} = \langle V, \Theta, \Sigma, \Delta \rangle$ *où* (i) $V = \langle v_1, \ldots, v_k \rangle$ *est un* k*-uple de*

variables (k constante), (ii) $\Theta \subseteq \mathcal{D}_V$ *est un prédicat sur* V *donnant la condition initiale sur les variables,* (iii) Σ *est une alphabet fini d'actions (ou événements) et* (iv) Δ *est un ensemble fini de transitions symboliques* $\delta = \langle \sigma_\delta, G_\delta, A_\delta \rangle$ *où :* $\sigma_\delta \in \Sigma$ *est l'action de* δ*,* $G_\delta \subseteq \mathcal{D}_V$ *est un prédicat sur* V *qui définit la garde de* δ*, et* $A_\delta : \mathcal{D}_V \mapsto \mathcal{D}_V$ *est la fonction d'assignation de* δ*.*

La sémantique d'un STS est un Système à Transitions Étiquetées (STE), qui peut être infini et dont les états sont des valuations des variables :

Définition 2 (Sémantique d'un STS) *La sémantique d'un STS* $\mathcal{T} = \langle V, \Theta, \Sigma, \Delta \rangle$ *est un STE* $[\![\mathcal{T}]\!] = \langle X, X_0, \Sigma, \rightarrow \rangle$ *où* (i) $X = \mathcal{D}_V$ *est l'ensemble des états,* (ii) $X_0 = \Theta$ *est l'ensemble des états initiaux,* (iii) Σ *est l'ensemble des étiquettes (ou actions) et* $(iv) \rightarrow \subseteq X \times \Sigma \times X$ *est le relation de transition définie par* $\rightarrow = \{\langle \vec{\nu}, \sigma, \vec{\nu}' \rangle \mid \exists \delta \in \Delta : (\sigma_\delta = \sigma) \wedge (\vec{\nu} \in G_\delta) \wedge (\vec{\nu}' = A_\delta(\vec{\nu}))\}$.

Initialement, un STS est dans un de ses états initiaux. Dans chaque état, une transition peut être tirée seulement si sa garde est satisfaite. Lorsqu'elle est tirée, les variables sont misent à jour suivant la fonction d'assignation. Un état $\vec{\nu} \in \mathcal{D}_V$ est *bloquant* si aucune transition ne peut être tirée à partir de cet état, c'est-à-dire que $\forall \delta \in \Delta : \vec{\nu} \notin G_\delta$. Notons que le STE $[\![\mathcal{T}]\!]$ peut être non déterministe.

Transformateurs de prédicats. Nous utilisons les notations suivantes pour tout $\delta \in \Delta$, $\sigma \in \Sigma$ et $B \subseteq \mathcal{D}_V$: $\mathrm{Trans}(\sigma) = \{\delta \in \Delta \mid \sigma_\delta = \sigma\}$: l'ensemble des transitions étiquetées par l'action σ ; $\mathsf{reachable}(\mathcal{T}) \subseteq \mathcal{D}_V$: l'ensemble des états qui sont accessibles à partir des états initiaux de $[\![\mathcal{T}]\!]$; $\mathrm{Pre}_\delta(B) = G_\delta \cap A_\delta^{-1}(B) = \{\vec{\nu} \in \mathcal{D}_V \mid \exists \vec{\nu}' \in \mathcal{D}_V : (\vec{\nu} \in G_\delta) \wedge (\vec{\nu}' = A_\delta(\vec{\nu})) \wedge (\vec{\nu}' \in B)\}$: l'ensemble des états menant à B par la transition δ ; $\mathrm{Pre}_\sigma(B) = \bigcup_{\delta \in \mathrm{Trans}(\sigma)} \mathrm{Pre}_\delta(B)$: l'ensemble des états menant à B par une transition étiquetée par σ ; et $\mathrm{Post}_\sigma(B) = \bigcup_{\delta \in \mathrm{Trans}(\sigma)} A_\delta(G_\delta \cap B)$: l'ensemble des états accessibles à partir de B par une transition étiquetée par σ. Tout au long de ce papier, nous travaillons sur des ensembles d'états et nous utilisons des opérations sur ces ensembles. Dans notre outil, les ensembles d'états sont symboliquement représentés par des prédicats et à chaque opération sur ces ensembles correspond un transformateur de prédicats (par exemple : $\mathrm{Pre}_\delta(B)$ est donné par l'ensemble des états $\vec{\nu}$, qui satisfont le transformateur de prédicats $\exists \vec{\nu}' \in \mathcal{D}_V : (\vec{\nu} \in G_\delta) \wedge (\vec{\nu}' = A_\delta(\vec{\nu})) \wedge (\vec{\nu}' \in B)$). De plus amples détails sont donnés dans (Le Gall *et al.*, 2005; Jeannet *et al.*, 2005). Un STS peut être défini avec des localités explicites, en ayant une variable finie de type énuméré, qui encode les localités. Nous utilisons cette facilité dans nos exemples.

Exemple 1 *Le STS de la figure 1 est un exemple assez simple d'un système de gestion de stocks. Cet exemple sera utilisé pour illustrer les concepts et méthodes présentés. Deux unités produisent et envoient (consomment) deux types de pièces* X *et* X'*.* Id*,* $\top$ *et* $\bot$ *dénotent resp. la fonction identité, et les prédicats vrai et faux. Le STS possède des localités explicites* ℓ *et quatre variables naturelles :* x *(resp.* x'*) indique le nombre de pièces* X *(resp.* X'*) et* y *(resp.* y'*) indique le nombre de pièces* X *(resp.* X'*) pouvant*

être produites. Un état du système correspond à un quintuplet $\langle \ell, x, x', y, y' \rangle$. L'état initial est $\langle Choix, 50, 50, 0, 0 \rangle$. Dans la localité CX (resp. CX'), l'action $Cons$ (resp. $Cons'$) permet de consommer une pièce X (resp. X') et $stop\_cons$ (resp. $stop\_cons'$) permet d'arrêter le processus de consommation. Dans la localités PX (resp. PX'), l'action $Prod$ (resp. $Prod'$) permet de produire une pièce X (resp. X') et $stop\_prod$ (resp. $stop\_prod'$) permet d'arrêter le processus de production. Le choix du type de pièces à produire est réalisé dans la localité $Choix$. La consommation des pièces est incontrôlable mais, comme nous le verrons, les variables y et y' assurent qu'il y a au plus 2 pièces consommées dans chaque cycle de consommation.

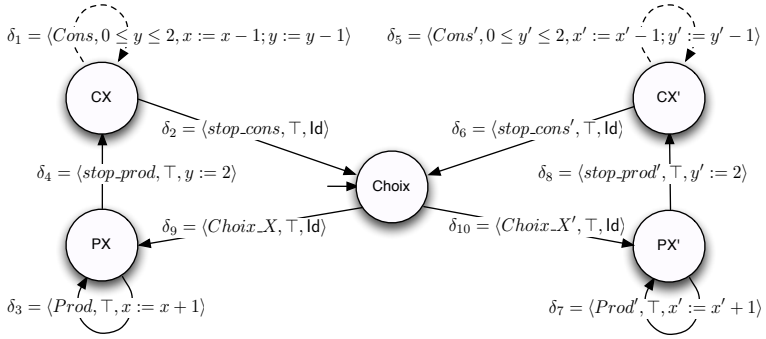

Figure 1. *Exemple de production et de consommation*

3. Le problème de contrôle d'interdiction d'états

Nous définissons ici le problème de contrôle d'interdiction d'états par rapport à l'observation du système et aux mécanismes de contrôle disponibles. L'observation et le contrôle sont *décentralisés* : n contrôleurs ont chacun une vue et un contrôle propres et partiels du système.

3.1. *Les moyens d'observation et de contrôle*

Définition 3 (Observateur) *Un observateur de l'espace d'états $\mathcal{D}_V$ est une paire $\langle Obs, M \rangle$, où Obs est une variable, dont le domaine est l'espace d'observations $\mathcal{D}_{Obs}$ (qui peut être infini), et le masque $M : \mathcal{D}_V \mapsto \mathcal{D}_{Obs}$ donne, pour chaque état $\vec{\nu} \in \mathcal{D}_V$, l'observation $M(\vec{\nu})$ que le contrôleur reçoit lorsque le système est dans cet état. De plus, on impose que $\forall \vec{\nu} \in \mathcal{D}_V : M(\vec{\nu}) \neq \emptyset$.*

L'espace d'observations est défini comme le domaine d'une variable afin de rester cohérent par rapport à la formalisation de l'espace d'états $\mathcal{D}_V$. Pour chaque observation $obs \in \mathcal{D}_{Obs}$, $M^{-1}(obs)$ donne l'ensemble des états, dont l'observation est obs.

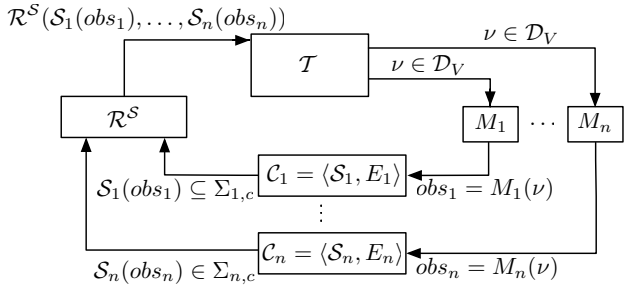

Figure 2. *Contrôle décentralisé sous observation partielle*

Notons que le masque M est une partition de l'espace d'états, ce qui signifie que $\forall obs, obs' \in \mathcal{D}_{Obs} : obs \neq obs' \Rightarrow M^{-1}(obs) \cap M^{-1}(obs') = \emptyset$.

Exemple 2 *Pour le système de la figure 1, un exemple d'observation partielle peut être le masque* $M : Loc \times \mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \mathbb{N} \mapsto Loc \times \mathbb{N} \times \mathbb{N}$*, où pour chaque état* $\langle \ell, x, x', y, y' \rangle \in \mathcal{D}_V$*, la valeur* $M(\langle \ell, x, x', y, y' \rangle) = \langle \ell, x, y \rangle$*. Cela signifie que les variables relatives aux pièces* X' *ne sont pas visibles.*

Nous basant sur la théorie de Ramadge & Wonham (Ramadge *et al.*, 1989; Cassandras *et al.*, 2008), nous voulons adjoindre n contrôleurs $\mathcal{C}_i$ $(\forall i \in [1, n])$, qui interagissent avec le système d'une manière rétro-active comme illustrée à la figure 2 : chaque contrôleur $\mathcal{C}_i$ $(\forall i \in [1, n])$ observe le système (à travers le masque M_i) et délivre, en fonction de cette observation, un ensemble d'événements, qui doivent être interdits pour assurer les propriétés désirées. Le contrôle est réalisé au moyen des *événements contrôlables* : l'alphabet Σ (resp. l'ensemble des transitions symboliques Δ) est partitionné en l'ensemble des événements contrôlables Σ_c et l'ensemble des événements incontrôlables Σ_{uc} (resp. les transitions contrôlables Δ_c et incontrôlables Δ_{uc}). De plus, à partir de Σ_c, n sous-ensembles $\Sigma_{1,c}, \ldots, \Sigma_{n,c}$ (non nécessairement disjoints) sont définis avec $\Sigma_c = \bigcup_{i=1}^{n} \Sigma_{i,c}$: chaque contrôleur $\mathcal{C}_i$ peut interdire les actions dans $\Sigma_{i,c}$ et ne peut pas inhiber les actions dans $\Sigma_{i,uc} = \Sigma \setminus \Sigma_{i,c}$. Dans ce qui suit, $\mathrm{In}(\sigma) = \{i | \sigma \in \Sigma_{i,c}\}$ dénote l'ensemble des indices des contrôleurs qui peuvent contrôler σ.

3.2. *Contrôleurs et système contrôlé*

Les contrôleurs ont pour but de restreindre le comportement du système de manière à assurer une propriété d'interdiction d'états (c'est-à-dire qu'ils doivent empêcher le système d'atteindre des états interdits). Un contrôleur avec observation partielle est formellement défini comme suit :

Définition 4 (Contrôleur) *Soient un STS $\mathcal{T} = \langle V, \Theta, \Sigma, \Delta \rangle$, une partition $\Sigma = \Sigma_c \cup \Sigma_{uc}$, et un observateur $\langle Obs, M \rangle$, un contrôleur pour $\mathcal{T}$ est une paire $\mathcal{C} = \langle \mathcal{S}, E \rangle$, où (i) $\mathcal{S} : \mathcal{D}_{Obs} \mapsto 2^{\Sigma_c}$ est une fonction de supervision qui définit, pour chaque observation $obs \in \mathcal{D}_{Obs}$, un ensemble $\mathcal{S}(obs)$ d'actions contrôlables à interdire lorsque obs est observé par le contrôleur $\mathcal{C}$ et (ii) $E \subseteq \mathcal{D}_V$ est un ensemble d'états à interdire, qui restreint l'ensemble des états initiaux.*

Par la suite, pour éviter les répétitions, nous supposerons toujours travailler avec un système $\mathcal{T} = \langle V, \Theta, \Sigma, \Delta \rangle$ à contrôler, n observateurs $\langle Obs_i, M_i \rangle$ $(\forall i \in [1, n])$, n contrôleurs $\mathcal{C}_i = \langle \mathcal{S}_i, E_i \rangle$ $(\forall i \in [1, n])$ et un prédicat Bad dénotant les états interdits.

Dans le cas décentralisé, n contrôleurs $\mathcal{C}_i$ $(\forall i \in [1, n])$ interagissent avec le système et chaque $\mathcal{C}_i$ peut interdire des actions dans $\Sigma_{i,c}$. Le contrôle global résultant de la synchronisation de ces n contrôleurs est alors défini par les règles de fusion pour les actions à interdire ($\mathcal{R}^{\mathcal{S}}$ dans la figure 2) et les états initiaux à interdire :

Définition 5 (Règles de fusion) *Les règles de fusion pour les actions à interdire et les états initiaux à interdire sont définies par :*

1) Soient $B_i \subseteq \Sigma_{i,c}$ $(\forall i \in [1, n])$ les actions interdites par chaque contrôleur $\mathcal{C}_i$. La règle de fusion $\mathcal{R}^{\mathcal{S}}$ donnant les actions à interdire globalement est définie par :

$$\mathcal{R}^{\mathcal{S}}(B_1, \ldots, B_n) = \{\sigma \,|\, \forall i \in \mathrm{In}(\sigma) : \sigma \in B_i\} \qquad [1]$$

Une action σ est interdite globalement si chaque contrôleur, qui a la possibilité d'inhiber cette action, l'interdit.

2) La règle de fusion $\mathcal{R}^E$ donnant les états initiaux à interdire globalement est définie par[1] *:*

$$\mathcal{R}^E(E_1, \ldots, E_n) = \bigcap_{i=1}^{n} E_i \qquad [2]$$

Lorsque le système est dans un état $\vec{\nu}$, chaque contrôleur $\mathcal{C}_i = \langle \mathcal{S}_i, E_i \rangle$ $(\forall i \in [1, n])$ reçoit l'observation $obs_i = M_i(\vec{\nu})$. Chaque contrôleur $\mathcal{C}_i$ calcule les actions $\mathcal{S}_i(obs_i)$ qu'il serait prudent d'interdire, parce qu'une transition avec une de ces actions pourrait mener à Bad (le calcul de $\mathcal{S}_i$ est défini dans la section 4). La règle de fusion (1) exprime que si un des contrôleurs, qui peut interdire σ, l'autorise, alors cette action sera globalement possible. Le système contrôlé est calculé comme suit.

Définition 6 (Système contrôlé) *Le système $\mathcal{T}$ contrôlé par le n-uple de contrôleurs $\langle \mathcal{C}_1, \ldots, \mathcal{C}_n \rangle$, est un STS $\mathcal{T}_{/\langle \mathcal{C}_1, \ldots, \mathcal{C}_n \rangle} = \langle V, \Theta_{/\langle \mathcal{C}_1, \ldots, \mathcal{C}_n \rangle}, \Sigma, \Delta_{/\langle \mathcal{C}_1, \ldots, \mathcal{C}_n \rangle} \rangle$, où*

1. Dans nos algorithmes, les contrôleurs $\mathcal{C}_i$ $(\forall i \in [1, n])$ interdiront le même ensemble E_i d'états initiaux. Donc, par exemple, $\mathcal{R}^E(E_1, \ldots, E_n) = E_1$.

(i) $\Theta_{/\langle \mathcal{C}_1,\ldots,\mathcal{C}_n\rangle} = \{\vec{\nu} | (\vec{\nu} \in (\Theta \setminus \mathcal{R}^E(E_1,\ldots,E_n)))\}$ *et (ii)* $\Delta_{/\langle \mathcal{C}_1,\ldots,\mathcal{C}_n\rangle}$ *est défini par la règle suivante :*

$$\frac{\langle \sigma, G, A\rangle \in \Delta \qquad G_{/\langle \mathcal{C}_1,\ldots,\mathcal{C}_n\rangle} = \{\vec{\nu} \mid (\vec{\nu} \in G) \wedge (\sigma \notin \mathcal{R}^{\mathcal{S}}(\mathcal{S}_1(M_1(\vec{\nu})),\ldots,\mathcal{S}_n(M_n(\vec{\nu}))))\}}{\langle \sigma, G_{/\langle \mathcal{C}_1,\ldots,\mathcal{C}_n\rangle}, A\rangle \in \Delta_{/\langle \mathcal{C}_1,\ldots,\mathcal{C}_n\rangle}}$$

Les fonctions $\mathcal{S}_i$ permettent de restreindre les gardes du système contrôlé. En effet, une transition δ ne peut plus être tirée dans $\mathcal{T}_{/\langle \mathcal{C}_1,\ldots,\mathcal{C}_n\rangle}$ à partir d'un état $\vec{\nu}$, si l'action σ_δ est interdite dans cet état par le contrôle global.

3.3. *Définition des problèmes de contrôle décentralisé*

Nous nous intéressons à deux variantes du problème de contrôle d'interdiction d'états :

Problème 1 (Contrôle décentralisé d'interdiction d'états de base) *Ce problème, appelé CDIEB, consiste à calculer n contrôleurs $\mathcal{C}_i$ $(\forall i \in [1,n])$ tels que* $\mathsf{reachable}(\mathcal{T}_{/\langle \mathcal{C}_1,\ldots,\mathcal{C}_n\rangle}) \cap Bad = \emptyset$.

Dans la suite, nous utilisons le terme *contrôleurs valides* pour dénoter un n-uple de contrôleurs $\langle \mathcal{C}_1,\ldots,\mathcal{C}_n\rangle$ tel que $\mathsf{reachable}(\mathcal{T}_{/\langle \mathcal{C}_1,\ldots,\mathcal{C}_n\rangle}) \cap Bad = \emptyset$. Une solution à ce problème n'assure pas que le système contrôlé obtenu ne contient pas d'états bloquants. Pour assurer cette propriété importante, nous définissons un second problème :

Problème 2 (Contrôle décentralisé d'interdiction d'états non-bloquant) *Ce problème, appelé CDIENb, consiste à calculer n contrôleurs $\mathcal{C}_i$ $(\forall i \in [1,n])$ tels que (i)* $\mathsf{reachable}(\mathcal{T}_{/\langle \mathcal{C}_1,\ldots,\mathcal{C}_n\rangle}) \cap Bad = \emptyset$ *et (ii)* $\mathcal{T}_{/\langle \mathcal{C}_1,\ldots,\mathcal{C}_n\rangle}$ *ne contient pas d'états bloquants atteignables.*

Nous pouvons immédiatement remarquer qu'une classe triviale de n-uples de contrôleurs $\langle \mathcal{C}_1,\ldots,\mathcal{C}_n\rangle$ corrects est celle où $E_i = \mathcal{D}_V$ pour tout $i \in [1,n]$ (c'est-à-dire que le système contrôlé ne contient plus aucun état). Par conséquent, la notion de permissivité a été introduite afin de comparer la qualité de différents n-uples de contrôleurs pour un STS donné.

Définition 7 (Permissivité) *Pour un système $\mathcal{T}$ et n observateurs donnés, un n-uple de contrôleurs $\langle \mathcal{C}_1,\ldots,\mathcal{C}_n\rangle$ est plus permissif qu'un n-uple de contrôleurs $\langle \mathcal{C}'_1,\ldots,\mathcal{C}'_n\rangle$ ssi* $\mathsf{reachable}(\mathcal{T}_{/\langle \mathcal{C}_1,\ldots,\mathcal{C}_n\rangle}) \supseteq \mathsf{reachable}(\mathcal{T}_{/\langle \mathcal{C}'_1,\ldots,\mathcal{C}'_n\rangle})$. *Lorsque l'inclusion est stricte, nous disons que le premier n-uple est strictement plus permissif que le second.*

En effet, puisque les observations et le contrôle du système sont basés sur les états, il semble plus cohérent de définir la permissivité par rapport aux états qui sont accessibles dans le système contrôlé, plutôt que par rapport au langage des actions qui

peuvent être tirées. Notons aussi que deux systèmes contrôlés, qui ont le même ensemble d'états accessibles, peuvent avoir des transitions autorisées différentes[2]. Nous pouvons prouver qu'en général, un n-uple de contrôleurs le plus permissif, résolvant CDIEB ou CDIENb, n'existe pas. Notons que le résultat aurait été le même pour d'autres types de permissivités telles que l'inclusion de langages, l'inclusion d'exécutions, ... Par conséquent, une solution maximale pour CDIEB (resp. CDIENb) est un n-uple de contrôleurs $\langle \mathcal{C}_1, \ldots, \mathcal{C}_n \rangle$ résolvant CDIEB (resp. CDIENb) tel qu'il n'existe aucun n-uple de contrôleurs $\langle \mathcal{C}'_1, \ldots, \mathcal{C}'_n \rangle$ strictement plus permissif, qui résout ce problème. Malheureusement, calculer une solution maximale pour CDIEB ou CDIENb est indécidable (Kalyon *et al.*, 2009a). Dès lors, notre but est de trouver des solutions qui soient correctes et aussi proches que possible d'une solution maximale pour être intéressantes en pratique. Nos expériences valideront nos méthodes.

4. Calcul symbolique des contrôleurs

Nous présentons un algorithme pour synthétiser des contrôleurs résolvant CDIEB ; ensuite nous étendons ce résultat pour traiter le cas non-bloquant. Les systèmes étant infinis, ces algorithmes, où aucune approximation n'est faite, n'assurent pas la terminaison des calculs. Dans la section 5, nous expliquons comment obtenir des algorithmes, basés sur les précédents, qui terminent toujours. L'idée générale du contrôle est de calculer, en utilisant un calcul de point fixe, l'ensemble des états $I(Bad)$, qui peuvent mener à Bad en ne tirant que des transitions incontrôlables ou qui peuvent être bloquants après le contrôle (pour le cas non-bloquant). Ensuite, en nous basant sur cet ensemble d'états, nous calculons les contrôleurs qui interdisent, pour chaque observation, toutes les transitions contrôlables pouvant mener à $I(Bad)$.

4.1. *Le contrôle décentralisé d'interdiction d'états de base*

Nous formalisons les deux étapes permettant de calculer les contrôleurs $\mathcal{C}_i = \langle \mathcal{S}_i, E_i \rangle$ ($\forall i \in [1, n]$), dont la synchronisation résout CDIEB.

Calcul de $I(Bad)$. Cet ensemble d'états (et plus généralement la fonction $I(.)$) est donné par la fonction $\mathrm{Coreach}_{uc} : 2^{\mathcal{D}_V} \mapsto 2^{\mathcal{D}_V}$ définie ci-dessous. Cet ensemble correspond à l'ensemble des états qui mènent à Bad en ne tirant que des transitions incontrôlables.

Nous définissons de manière classique la fonction $\mathrm{Pre}_{uc}(B)$, qui calcule l'ensemble des états à partir desquels un état de B est accessible en tirant exactement une transition incontrôlable : $\mathrm{Pre}_{uc}(B) = \bigcup_{\delta \in \Delta_{uc}} \mathrm{Pre}_{\delta}(B)$. $\mathrm{Coreach}_{uc}(Bad)$ est

2. Nous aurions pu utiliser une définition de permissivité différente dans laquelle, lorsque deux systèmes contrôlés ont le même espace d'états accessibles, l'inclusion des transitions qui peuvent être tirées à partir des états accessibles est aussi prise en compte.

dès lors obtenu par l'équation de point fixe suivante, où lfp dénote le plus petit point fixe :

$$\mathrm{Coreach}_{uc}(Bad) = \mathrm{lfp}(\lambda B.Bad \cup \mathrm{Pre}_{uc}(B)) \quad [3]$$

Par le théorème de Tarski (Tarski, 1955), puisque la fonction $\mathrm{Coreach}_{uc}$ est monotone, la limite du point fixe $\mathrm{Coreach}_{uc}(Bad)$ existe (mais peut être incalculable vu que l'espace d'états est infini). Notons que cette fonction est utilisée par tous les contrôleurs.

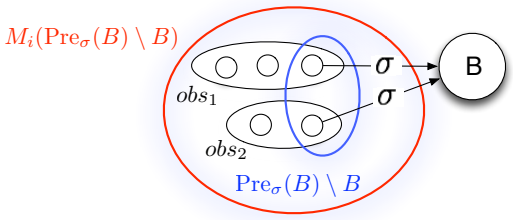

Figure 3. *Calcul de* $M_i(\mathrm{Pre}_\sigma(B) \setminus B)$

Calcul du contrôleur $\mathcal{C}_i$ $(\forall i \in [1,n])$ *et du système contrôlé* $\mathcal{T}_{/\langle \mathcal{C}_1,\ldots,\mathcal{C}_n \rangle}$. Nous définissons d'abord la fonction $\mathcal{F}_i : \Sigma \times 2^{\mathcal{D}_V} \mapsto 2^{\mathcal{D}_{Obs_i}}$. Pour une action $\sigma \in \Sigma$ et un ensemble $B \subseteq \mathcal{D}_V$ d'états à interdire, $\mathcal{F}_i(\sigma, B)$ spécifie l'ensemble des états d'observation, pour lesquels l'action σ doit être interdite par le contrôleur $\mathcal{C}_i$, c'est-à-dire l'ensemble $\mathcal{O}_i$ des observations telles qu'il existe un état $\vec{\nu} \in \mathcal{D}_V$ avec $M_i(\vec{\nu}) \in \mathcal{O}_i$, à partir duquel une transition étiquetée par σ mène à B (cf. figure 3).

$$\mathcal{F}_i(\sigma, B) = \begin{cases} M_i(\mathrm{Pre}_\sigma(B) \setminus B) & \text{si } \sigma \in \Sigma_{i,c} \\ \emptyset & \text{sinon} \end{cases} \quad [4]$$

Le contrôleur $\mathcal{C}_i$ $(\forall i \in [1,n])$ est alors défini comme suit :

$$\mathcal{C}_i = \langle \mathcal{S}_i, E_i \rangle \quad [5]$$

où (i) la fonction de supervision $\mathcal{S}_i$ est définie, pour chaque $obs \in \mathcal{D}_{Obs_i}$, par $\mathcal{S}_i(obs) = \{\sigma \in \Sigma \,|\, obs \in \mathcal{F}_i(\sigma, I(Bad))\}$, et (ii) l'ensemble $E_i = I(Bad)$.

Le calcul de la fonction $\mathcal{F}_i$ $(\forall i \in [1,n])$ est réalisé hors ligne et, étant donné n observations $\langle obs_1, \ldots, obs_n \rangle$, chaque contrôleur $\mathcal{C}_i$ $(\forall i \in [1,n])$ calcule en ligne l'ensemble $\mathcal{S}_i(obs_i)$ (qui utilise la fonction $\mathcal{F}_i$). Puisque Σ est fini, $\mathcal{S}_i(obs_i)$ est calculable. Finalement, les actions données par la règle de fusion $\mathcal{R}^{\mathcal{S}}$ paramétrée par les ensembles $\mathcal{S}_i(obs_i)$ $(\forall i \in [1,n])$ sont interdites.

Le système contrôlé est calculé selon la Définition 6 en utilisant les contrôleurs $\mathcal{C}_i$. Notons que les gardes restreintes $G_{/\langle \mathcal{C}_1,\ldots,\mathcal{C}_n \rangle}$ du système contrôlé peuvent être calculées par $G \setminus \{\bigcap_{i \in \mathrm{In}(\sigma)} M_i^{-1}(\mathcal{F}_i(\sigma, I(Bad)))\}$, puisque la règle de fusion donnant les actions à interdire globalement correspond à l'ensemble des actions σ interdites par chaque contrôleur pouvant inhiber cette action.

Proposition 1 *Le n-uple de contrôleurs $\mathcal{C}_i = \langle \mathcal{S}_i, E_i \rangle$ $(\forall i \in [1,n])$, définis par [5], résout CDIEB (Kalyon* et al.*, 2009b).*

Exemple 3 *Le système de la figure 1 est contrôlé par $\mathcal{C}_1$ et $\mathcal{C}_2$. $\mathcal{C}_1$ observe le système à travers le masque M_1 défini comme celui de l'Exemple 2. $\mathcal{C}_2$ observe le système à travers le masque $M_2 : Loc \times \mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \mathbb{N} \mapsto Loc \times \mathbb{N} \times \mathbb{N}$, où pour chaque état $\langle \ell, x, x', y, y' \rangle \in \mathcal{D}_V$, $M(\langle \ell, x, x', y, y' \rangle) = \langle \ell, x', y' \rangle$. Les transitions contrôlables (resp. incontrôlables) par les 2 contrôleurs, sont celles dessinées en lignes continues (resp. discontinues). Les contrôleurs doivent assurer qu'il y ait plus de dix pièces de chaque type : $Bad = \{\langle$CX$, x, x', y, y' \rangle | (x \leq 10) \wedge (x', y, y' \in \mathbb{N})\} \cup \{\langle$CX'$, x, x', y, y' \rangle | (x' \leq 10) \wedge (x, y, y' \in \mathbb{N})\}$. Cela implique que $I(Bad) = Bad \cup \{\langle$CX$, x, x', y, y' \rangle | [(x \leq 11) \wedge (y \in [1,2]) \wedge (x', y' \in \mathbb{N})] \vee [(x \leq 12) \wedge (y = 2) \wedge (x', y' \in \mathbb{N})]\} \cup \{\langle$CX'$, x, x', y, y' \rangle | [(x' \leq 11) \wedge (y' \in [1,2]) \wedge (x, y \in \mathbb{N})] \vee [(x' \leq 12) \wedge (y' = 2) \wedge (x, y \in \mathbb{N})]\}$. Le calcul de $\mathcal{F}_1$ donne : $\mathcal{F}_1(\sigma, I(Bad)) =$*

$$\begin{cases} M_1(\{\langle \textit{PX}, x, x', y, y' \rangle | (x \leq 12) \wedge (x', y, y' \in \mathbb{N})\}) & \\ \quad = \{\langle \textit{PX}, x, y \rangle | (x \leq 12) \wedge (y \in \mathbb{N})\} & \textit{si } \sigma = stop\_prod \\ M_1(\{\langle \textit{PX'}, x, x', y, y' \rangle | (x' \leq 12) \wedge (x, y, y' \in \mathbb{N})\}) & \\ \quad = \{\langle \textit{PX'}, x, y \rangle | x, y \in \mathbb{N}\} & \textit{si } \sigma = stop\_prod' \\ \emptyset & \textit{sinon} \end{cases}$$

Donc, le contrôleur $\mathcal{C}_1$ interdit toujours $stop\_prod'$, car il n'observe pas les variables x' et y'. De manière analogue, $\mathcal{C}_2$ interdit toujours $stop\_prod$ et il inhibe $stop\_prod'$ lorsque $x' \leq 12$. Le système contrôlé est obtenu en restreignant la garde de δ_4, qui ne peut plus être tirée lorsque $x \leq 12$, et la garde de δ_8, qui ne peut plus être tirée lorsque $x' \leq 12$.

4.2. *Le contrôle décentralisé d'interdiction d'états non-bloquant*

Calcul de $I(Bad)$. Cet ensemble d'états (et plus généralement la fonction $I(.)$) est donné par la fonction $\mathrm{Coreach}_{uc}^{bl} : 2^{\mathcal{D}_V} \mapsto 2^{\mathcal{D}_V}$ définie ci-dessous. Cet ensemble correspond à l'ensemble des états, qui sont bloquants dans le système contrôlé ou qui mènent à un état interdit en ne tirant que des transitions incontrôlables. Pour calculer $\mathrm{Coreach}_{uc}^{bl}(Bad)$, nous calculons d'abord $\mathrm{Coreach}_{uc}(Bad)$ (défini par [3]). Ensuite, si nous rendons inaccessible les états interdits en coupant toutes les transitions contrôlables qui mènent à un mauvais état, le système contrôlé correspondant pourrait avoir de nouveaux états bloquants. Nous devons alors rajouter ces états bloquants à l'ensemble des états à interdire. La fonction $\mathrm{Pre}_{bl}(B)$ calcule, pour un ensemble $B \subseteq \mathcal{D}_V$ d'états à interdire, l'ensemble des états, qui seraient bloquants dans le système contrôlé, si les états de B n'étaient plus accessibles. Le calcul des états bloquants est basé sur les fonctions $\mathcal{F}_i$ $(\forall i \in [1,n])$ définies en [4]. Pour assurer la convergence dans le calcul de $\mathrm{Coreach}_{uc}^{bl}(Bad)$, Pre_{bl}, et par conséquent $\mathcal{F}_i$, doivent être monotones. Nous utilisons donc les fonctions monotones $\widehat{\mathcal{F}_i}$ au lieu de $\mathcal{F}_i$ dans le calcul du contrôleur pour le cas non-bloquant :

$$\widehat{\mathcal{F}_i}(\sigma, B) = \begin{cases} M_i(\mathrm{Pre}_\sigma(B)) & \text{si } \sigma \in \Sigma_{i,c} \\ \emptyset & \text{sinon} \end{cases}$$

Nous expliquons maintenant comment calculer les états bloquants dans le système contrôlé $\mathcal{T}_{/\langle \mathcal{C}_1,\ldots,\mathcal{C}_n\rangle}$. Un état $\vec{\nu} \in \mathcal{D}_V$ est bloquant dans $\mathcal{T}_{/\langle \mathcal{C}_1,\ldots,\mathcal{C}_n\rangle}$, si les deux conditions suivantes sont satisfaites dans le système $\mathcal{T}$:

1) l'état $\vec{\nu}$ n'a pas de transitions incontrôlables sortantes.

2) pour chaque transition contrôlable δ, cette transition ne peut pas être tirée à partir de $\vec{\nu}$ (c'est-à-dire que $\vec{\nu} \notin G_\delta$) ou l'action σ_δ est interdite par le contrôle global pour les observations $\langle M_1(\vec{\nu}), \ldots, M_n(\vec{\nu})\rangle$ (c'est-à-dire que $\sigma_\delta \in \mathcal{R}^{\mathcal{S}}(\mathcal{S}_1(M_1(\vec{\nu})), \ldots, \mathcal{S}_n(M_n(\vec{\nu})))$). Cette seconde condition est équivalente à $\forall i \in \mathrm{In}(\sigma_\delta) : M_i(\vec{\nu}) \in \widehat{\mathcal{F}_i}(\sigma_\delta, B)$, parce que la règle de fusion donnant les actions à interdire globalement correspond à l'ensemble des actions σ interdites par chaque contrôleur pouvant inhiber cette action.

Définition 8 *Formellement, pour un ensemble d'états $B \subseteq \mathcal{D}_V$ à interdire, un état $\vec{\nu}$ est bloquant si :* 1) $\forall \delta \in \Delta_{uc} : \vec{\nu} \notin G_\delta$, *et* 2) $\forall \delta \in \Delta_c : (\vec{\nu} \notin G_\delta) \vee (\forall i \in \mathrm{In}(\sigma_\delta) : M_i(\vec{\nu}) \in \widehat{\mathcal{F}_i}(\sigma_\delta, B))$.

Puisque $\widehat{\mathcal{F}_i}(\sigma, B) = \emptyset$ ($\forall \sigma \in \Sigma_{uc}$), la fonction Pre_{bl}, qui calcule les états qui seraient bloquants dans le système contrôlé, peut être définie comme suit : $\mathrm{Pre}_{bl}(B) = B \cup [\bigcap_{\delta\in\Delta}(\overline{G_\delta} \cup \bigcap_{i\in \mathrm{In}(\sigma_\delta)}(M_i^{-1}(\widehat{\mathcal{F}_i}(\sigma_\delta, B))))]$

En ajoutant les états bloquants aux états à interdire, de nouveaux états peuvent mener de manière incontrôlable à un mauvais état. En conséquence, l'ensemble $\mathrm{Coreach}_{uc}^{bl}(Bad)$ est calculé par l'équation de point fixe suivante :

$$\mathrm{Coreach}_{uc}^{bl}(Bad) = \mathrm{lfp}(\lambda B.Bad \cup \mathrm{Pre}_{bl}(\mathrm{Coreach}_{uc}(B))) \qquad [6]$$

Calcul du contrôleur local $\mathcal{C}_i$ ($\forall i \in [1, n]$) et du système contrôlé $\mathcal{T}_{/\langle \mathcal{C}_1,\ldots,\mathcal{C}_n\rangle}$. Les contrôleurs $\mathcal{C}_i$ et le système contrôlé sont calculés de manière analogue à ce qui est fait dans la section 4.1.

Proposition 2 *Les n-uples de contrôleurs $\mathcal{C}_i = \langle \mathcal{S}_i, E_i\rangle$, calculés par l'algorithme ci-dessus, résolvent CDIENb (Kalyon* et al.*, 2009b).*

5. Calcul effectif des contrôleurs au moyen de l'interprétation abstraite

Le calcul actuel des contrôleurs, basé sur une équation de point fixe pour calculer $I(Bad)$, n'est généralement pas possible pour des raisons d'indécidabilité ou de complexité. Pour surmonter ce problème, nous utilisons des techniques *d'interprétation abstraite* (Cousot *et al.*, 1977), qui permettent de calculer une sur-approximation du

point fixe $I(Bad)$. Cette sur-approximation assure que Bad n'est pas accessible dans le système contrôlé, qui interdira potentiellement plus de transitions que nécessaire. Nous obtenons donc un contrôleur correct, mais qui est plus restrictif.

L'idée est de ne pas effectuer les calculs directement avec des ensembles d'états, mais de remplacer le treillis des ensembles d'états $(2^{\mathcal{D}_V}, \subseteq)$ par un treillis abstrait $(\Lambda, \sqsubseteq)$ plus facile à manipuler. Une fonction d'abstraction $\alpha : 2^{\mathcal{D}_V} \mapsto \Lambda$ et une fonction de concrétisation $\gamma : \Lambda \mapsto 2^{\mathcal{D}_V}$ permettent de passer de l'un à l'autre. En pratique, ce treillis abstrait sera celui des polyèdres convexes (Cousot *et al.*, 1978). Les fonctions et opérateurs employés dans les équations définies en section 4 seront remplacés par les opérateurs abstraits correspondant. Ces opérateurs sont définis de telle sorte que le résultat obtenu dans le treillis abstrait sera une sur-approximation du point fixe concret. Ainsi, la fonction correspondant à $\mathrm{Pre}_{uc} : 2^{\mathcal{D}_V} \mapsto 2^{\mathcal{D}_V}$ est notée $\mathrm{Pre}^\sharp_{uc} : \Lambda \mapsto \Lambda$, et est définie pour tout $l \in \Lambda$ comme suit : $\mathrm{Pre}^\sharp_{uc}(l) = \bigsqcup_{\delta \in \Delta_{uc}} \mathrm{Pre}^\sharp_\delta(l)$, où $\mathrm{Pre}^\sharp_\delta(l) = \alpha(G_\delta \cap A_\delta^{-1}(\gamma(l)))$. $\mathrm{Coreach}^\sharp_{uc}(Bad)$ est le plus petit point fixe de la fonction $\lambda l.\alpha(Bad) \sqcup \mathrm{Pre}^\sharp_{uc}(l)$ et nous calculons l_∞, défini comme la limite de la séquence donnée par $l_1 = \alpha(Bad)$ et $l_{i+1} = l_i \nabla \mathrm{Pre}^\sharp_{uc}(l_i)$ où ∇ est l'opérateur d'élargissement utilisé. La théorie de l'interprétation abstraite assure que cette séquence se stabilise après un nombre fini d'étapes, et que sa concrétisation $\gamma(l_\infty)$ est une sur-approximation de $I(Bad)$. Ainsi, nous obtenons $I'(Bad) = \gamma(l_\infty)$ et nous définissons les contrôleurs comme dans la section 4, en utilisant $I'(Bad)$ au lieu de $I(Bad)$.

Qualité des approximations. La méthode présentée ici calcule toujours un contrôleur correct, mais sans la garantie que ce soit un maximal. Moins nous faisons d'approximations durant les calculs, plus l'approximation de $I(Bad)$ que nous obtenons est précise. On peut améliorer la qualité des approximations en utilisant un treillis abstrait adapté au type de données du système à contrôler et en utilisant des stratégies de calcul de point-fixe plus efficaces (Bourdoncle, 1992). Il existe cependant peu de résultats théoriques sur la qualité de l'abstraction. Nous pouvons seulement montrer sur des exemples que nos abstractions permettent le calcul d'un contrôleur utile.

Expérimentations. Nous avons implémenté les algorithmes des sections 4 et 5 dans notre outil SMACS (Synthèse MAsquée de ContrôleurS), écrit en Objective CAML (OCa, 2009), qui utilise la librairie APRON (APR, 2009) et un solveur de point fixe générique (Fix, 2009). Nous avons expérimenté cet outil sur plusieurs exemples, dont celui qui a été détaillé sur la figure 1. Dans cet exemple qui modélise un système de gestion de stocks, chaque contrôleur observe une des deux unités qui peuvent produire des pièces ou en consommer. SMACS obtient rapidement le système contrôlé (<20 ms) qui restreint les gardes comme il a été détaillé dans la section 4.1. Un autre exemple figure dans (Kalyon *et al.*, 2009b).

6. Comparaison entre le contrôle centralisé et le contrôle décentralisé

Dans cette section, nous comparons la permissivité des n contrôleurs décentralisés $\mathcal{C}_i$ ($\forall i \in [1, n]$) calculés dans la section 4 avec un contrôleur centralisé $\mathcal{C}$ également calculé par les algorithmes de la section 4 (on pose alors le nombre de contrôleurs à 1). Ce contrôleur peut agir sur les actions dans Σ_c (c'est-à-dire qu'il peut interdire une action si elle peut être interdite par au moins un des contrôleurs $\mathcal{C}_i$) et observe le système à travers le masque $M = \prod_{i\in[1,n]} M_i$ (c'est-à-dire que $\mathcal{C}$ ne peut pas distinguer deux états $\vec{\nu}$ et $\vec{\nu}'$, si $\forall i \in [1, n], M_i(\vec{\nu}) = M_i(\vec{\nu}')$). En conséquence, $\mathcal{C}$ ne peut pas distinguer un état $\vec{\nu} \in \mathcal{D}_V$ des états appartenant à $\bigcap_{i=1}^{n} M_i^{-1}(M_i(\vec{\nu}))$. La propriété suivante montre que $\mathcal{C}$ donne une solution plus permissive pour CDIEB.

Proposition 3 *Le contrôleur centralisé $\mathcal{C}$ calculé par l'algorithme de la section 4.1 est plus permissif que les n contrôleurs décentralisés calculés par l'algorithme de la section 4.1 (Kalyon* et al.*, 2009b).*

Nous pouvons également montrer qu'un contrôleur non-bloquant centralisé est plus permissif que les contrôleurs non-bloquants décentralisés.

La proposition suivante donne une condition suffisante (basée sur la condition de M-contrôlabilité dans (Takai *et al.*, 1994)) pour avoir une équivalence entre le contrôleur centralisé et les contrôleurs décentralisés.

Proposition 4 *Le contrôleur centralisé et les contrôleurs décentralisés, tous deux calculés par l'algorithme de la section 4.1, ont la même permissivité si* $\forall \vec{\nu} \notin \mathrm{Coreach}_{uc}(Bad), \forall \sigma \in \Sigma_c : [\mathrm{Post}_\sigma(M^{-1}(M(\vec{\nu})) \setminus \mathrm{Coreach}_{uc}(Bad)) \cap \mathrm{Coreach}_{uc}(Bad) = \emptyset] \Rightarrow [\exists i \in \mathrm{In}(\sigma) : \mathrm{Post}_\sigma(M_i^{-1}(M_i(\vec{\nu})) \setminus \mathrm{Coreach}_{uc}(Bad)) \cap \mathrm{Coreach}_{uc}(Bad) = \emptyset]$ *(Kalyon* et al.*, 2009b).*

Intuitivement, cette condition stipule que si une action contrôlable n'est pas inhibée par le contrôleur centralisé, alors un des contrôleurs décentralisés qui peut la contrôler ne l'interdira pas non plus. Une condition semblable peut être donnée pour l'algorithme de la section 4.2 en remplaçant $\mathrm{Coreach}_{uc}(Bad)$ par les ensembles obtenus lors des itérations du point fixe.

7. Travaux futurs

Cet article présente un moyen de faire du contrôle décentralisé de systèmes symboliques. Nous le voyons comme une étape vers la résolution du problème du contrôle de systèmes symboliques distribués. Autrement dit, nous comptons nous passer de l'hypothèse de communication synchrone entre les différents contrôleurs. Dans (Tripakis, 2002), il a été établi que ce problème est indécidable. Mais en utilisant des techniques d'interprétation abstraite, nous pouvons espérer obtenir des résultats de

manière similaire à celle exposée dans le présent article. Nous comptons également améliorer notre outil SMACS pour pouvoir nous attaquer à des études de cas plus conséquentes que les exemples mentionnés ici.

8. Bibliographie

APR, « The APRON library », 2009. http ://apron.cri.ensmp.fr/.

Bourdoncle F., Sémantiques des Langages Impératifs d'Ordre Supérieur et Interprétation Abstraite, PhD thesis, Ecole Polytechnique, 1992.

Cassandras C., Lafortune S., *Introduction to Discrete Event Systems*, Springer, 2008.

Cousot P., Cousot R., « Abstract interpretation : a unified lattice model for static analysis of programs by construction or approximation of fixpoints », *POPL'77*, p. 238-252, 1977.

Cousot P., Halbwachs N., « Automatic discovery of linear restraints among variables of a program », *POPL '78*, p. 84-96, 1978.

Fix, « Fixpoint : an OCaml library implementing a generic fix-point engine », 2009. http ://pop-art.inrialpes.fr/people/bjeannet/bjeannet-forge/fixpoint/.

Henzinger T., Majumdar R., Raskin J.-F., « A classification of symbolic transition systems », *ACM Trans. Comput. Logic*, vol. 6, n° 1, p. 1-32, 2005.

Jeannet B., « Dynamic Partitioning In Linear Relation Analysis. Application To The Verification Of Reactive Systems », *Formal Meth. in Syst. Design*, vol. 23, n° 1, p. 5-37, 2003.

Jeannet B., Jéron T., Rusu V., Zinovieva E., « Symbolic Test Selection based on Approximate Analysis », *TACAS'05, Volume 3440 of LNCS*, Edinburgh, p. 349-364, April, 2005.

Kalyon G., Le Gall T., Marchand H., Massart T., « Control of Infinite Symbolic Transition Systems under Partial Observation », *European Control Conference*, Hungary, August, 2009a.

Kalyon G., Le Gall T., Marchand H., Massart T., Contrôle décentralisé de systèmes symboliques infinis sous observation partielle, Technical report of the verification group n° 121, Université Libre de Bruxelles, September, 2009b.

Kumar R., Garg V., « On Computation of State Avoidance Control for Infinite State Systems in Assignment Program Model », *IEEE Trans. on Autom. Science and Engineering*, vol. 2, n° 2, p. 87-91, 2005.

Kumar R., Garg V., Marcus S., « Predicates and Predicate Transformers for Supervisory Control of Discrete Event Dynamical Systems », *IEEE Trans. Autom. Control*, vol. 38, n° 2, p. 232-247, 1993.

Le Gall T., Jeannet B., Marchand H., « Supervisory Control of Infinite Symbolic Systems using Abstract Interpretation », *CDC/ECC'05*, December, 2005.

OCa, « The programming language Objective CAML », 2009. http ://caml.inria.fr/.

Ramadge P., Wonham W., « The Control of Discrete Event Systems », *Proceedings of the IEEE ; Special issue on Dynamics of Discrete Event Systems*, vol. 77, n° 1, p. 81-98, 1989.

Takai S., Kodama S., « Characterization of all M-controllable subpredicates of a given predicate », *International Journal of Control*, vol. 70, p. 541-549(9), 10 July 1998.

Takai S., Kodama S., Ushio T., « Decentralized state feedback control of discrete event systems », *Syst. Control Lett.*, vol. 22, n° 5, p. 369-375, 1994.

Tarski A., « A Lattice-theoretical Fixpoint Theorem and its applications », *Pacific Journal of Mathematics*, vol. 5, p. 285-309, 1955.

Tripakis S., « Decentralized Control of Discrete Event Systems with Bounded or Unbounded Delay Communication », *WODES '02*, IEEE Computer Society, 2002.