Canaux fiables et non-fiables : frontières de la décidabilité

P. Chambart, encadré par Ph. Schnoebelen

LSV, ENS Cachan, CNRS
61, av. Pdt. Wilson, F-94230 Cachan, France

1 Introduction

Les systèmes a canaux sont des systèmes de transition semblables aux automates à piles, où les piles sont remplacées par des files. Ce modèle peut servir par exemple à représenter des réseaux, pour vérifier des protocoles. Mais comme ces systèmes sont Turing-complets, il est impossible de faire de la vérification automatique dessus.

Par contre si l'on autorise les pertes dans les canaux, le modèle devient étonnamment plus simple et certain problèmes deviennent décidable, comme l'accessibilité d'un état de contrôle [6,2], ce qui nous intéresse ici.

Le modèle peut ensuite être étendu pour permettre d'avoir, dans un même système, des canaux fiables et non fiables. Il y a des cas où l'accessibilité reste décidable, par exemple dans un réseau en boucle où il y a au moins un canal non fiable [4].

Le but de ce stage était de caractériser les formes de réseau où l'accessibilité est décidable. Nous avons répondu à cette question, mais il se trouve que ces configurations ne sont pas particulièrement intéressantes pour modéliser des protocoles.

Par contre cette étude a permit d'obtenir des résultats de complexité intéressants. En effet l'accessibilité dans les systèmes a canaux non fiables est un problème qui est connu comme non primitif récursif [13] depuis quelques années et qui a déjà servit à démontrer la difficulté d'autres problèmes [3,9,11,8,1,12,5,7,10]. Pour résoudre le problème, nous avons été amené a en définir un autre, semblable au problème de correspondance de Post, équivalent à l'accessibilité dans les systèmes à canaux non fiables. Nous espérons donc que ce problème sera plus facile à utiliser pour faire des réductions.

2 systèmes à canaux

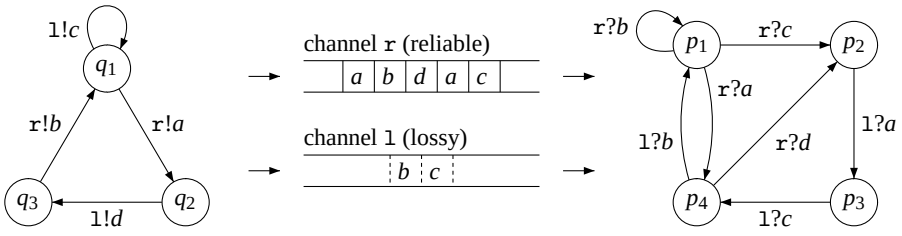

FIG. 1. Un exemple de système avec un canal fiable et un non fiable

Un système à canaux est un tuple $S = (P, \mathtt{M}, \mathtt{C})$. $\mathtt{M} = \{a_1, a_2, \ldots\}$ est l'alphabet fini des messages, $\mathtt{C} = \{\mathtt{c}_1, \mathtt{c}_2, \ldots\}$ est l'ensemble des canaux, $P = \{p_1, p_2, \ldots\}$ est un ensemble de participants, Ce sont les noeuds du réseau. Chaque participant est un tuple $p_i = (\mathtt{Q}_i, S_i, R_i, \Delta_i)$ où $\mathtt{Q}_i = \{q_1, q_2, \ldots\}$ est l'ensemble fini des états, $S_i \subset \mathtt{C}$ est l'ensemble des canaux dans lesquels p_i peut écrire, $R_i \subset \mathtt{C}$ est l'ensemble des canaux dans lesquels p_i peut lire, $\Delta_i \subseteq \mathtt{Q}_i \times S_i \times \mathtt{M} \times \mathtt{Q}_i \cup \mathtt{Q}_i \times R_i \times \mathtt{M} \times \mathtt{Q}_i$ est l'ensemble des règles de transition. Ces règles sont usuellement notée sous la forme $q \xrightarrow{\mathtt{c}!a} q'$ pour les écritures et $q \xrightarrow{\mathtt{c}?a} q'$ pour les lectures.

De plus, il n'existe pas $p_i, p_j \in P$ tels que $R_i \cap R_j$ et $S_i \cap S_j$ ne soient pas vide, c'est à dire que pour tout canal, il y a au plus un émetteur et un récepteur.

Soit $S = (P, \mathtt{M}, \mathtt{C})$ un système muni de n participants, $p_1, \ldots, p_n$, et m canaux, $c_1, \ldots, c_m$. Une configuration est un tuple $C = (q_1, \ldots, q_n, w_1, \ldots, w_m)$ où $q_1, \ldots, q_n$ sont les états et $w_1, \ldots, w_m$ sont les mots contenus dans les canaux.

Les règles des Δ_i engendrent un système de transitions sur les configurations. Si il y a une réglé d'écriture $q \xrightarrow{\mathtt{c}!a} q'$ dans Δ_i, alors il existe dans ce système une transition $(\ldots, q_i, \ldots, m_c, \ldots) \xrightarrow{\mathtt{c}!a} (\ldots, q'_i, \ldots, m_c.$ et pour une lecture $q \xrightarrow{\mathtt{c}?a} q'$, il y a $(\ldots, q_i, \ldots, a.m_c, \ldots) \xrightarrow{\mathtt{c}?a} (\ldots, q'_i, \ldots, m_c, \ldots)$.

De manière plus intuitive, les participants d'un système sont des automates qui ont la possibilité d'envoyer des messages à d'autres participants sur des canaux. Par contre ces autres participants ne lisent pas forcement tout de suite ces messages. Les canaux peuvent servir ainsi d'espace de stockage. C'est ce qui rend ce modèle Turing complet.

L'ensemble des message envoyés dans le canal mais pas encore lu forme un mot sur l'alphabet des messages. C'est ce qu'on appellera le mot contenu dans un canal.

Un système à canaux non fiables (LCS) est définit de manière similaire, mais le système de transition engendré comprends des règles supplémentaires, représentant les pertes possibles. Ainsi lors d'une écriture $q \xrightarrow{\mathtt{c}!a} q'$, la transition $(\ldots, q_i, \ldots) \xrightarrow{\mathtt{c}!a} (\ldots, q'_i, \ldots)$ est aussi possible.

Il existe d'autres définitions représentant les pertes à la lecture, ou même directement dans le canal, mais elles sont au final équivalente. Celle-ci est la plus pratique dans notre cas.

Nous nous intéresserons désormais à des systèmes contenant ces deux types de canaux $S = (P, \mathtt{M}, \mathtt{C})$ avec $\mathtt{C} = F \cup NF$ où F est l'ensemble des canaux fiables, et NF les non fiables.

2.1 accessibilité d'états et de configurations

Il est parfois plus commode de regarder la question de l'accessibilité pour des configuration du système, parfois sur des états (quand on parle d'état sur un système, cela correspond à un tuple d'états des participants). Notons que ce sont au final les même problèmes.

Pour un système S, on peut en construire un autre S', tel que l'accessibilité d'un état q dans S soit équivalent à l'accessibilité d'une configuration C dans S'. Pour cela, il faut ajouter un nouvel état q' dans S' accessible uniquement depuis q tel qu'il soit possible de vider les canaux si q' est atteint, quel que soient leur contenus initiaux. donc q est accessible ssi q' est accessible avec les canaux vide.

L'on peut dans l'autre sens construire un système vérifiant le contenu des canaux après avoir atteint un état donné, puis si ce contenu est celui attendu, le système va dans un nouvel état. Ainsi ce nouvel états n'est accessible que si une configuration particulière pouvait être atteinte.

3 formes

Nous définissons la notion de formes pour pouvoir parler de classes de systèmes. Une forme est un graphe oriente avec différents types d'arêtes (fiables, non fiable) et ayant la possibilité d'avoir plusieurs arêtes de même type entre deux noeuds. Les noeuds du graphe correspondent à des participants et les arêtes à des canaux. Chaque forme correspond a un problème : l'accessibilité dans les systèmes de cette forme.

3.1 définition

Plus formellement une forme F est un tuple (P, A_f, A_n, f) où P est l'ensemble fini des noeuds. A_f est l'ensemble fini des arêtes fiables, A_n est l'ensemble fini des arêtes non fiables. $f : A_f \cup A_n \longrightarrow P \times P$ est la fonction qui définit les origines et arrivés de chaque arête.

Pour une arête a telle que $f(a) = (n_e, n_r)$, n_e est appelé le noeud émetteur et n_r le récepteur.

Un système S est de forme F si chaque noeud de F correspond a un participant de S, et que chaque arête de F correspond a un canal de S.

Par exemple le système décrit dans la figure 1 est de la forme décrite par la figure 4

Formellement, un système $S = (P_s, \mathtt{M}, F \cup NF)$ est de forme $F = (N, A_f, A_n, f)$, si il existe des bijection b_p entre les noeuds de F et les participants de S, b_f entre les arêtes fiables de F et les canaux fiables de S et b_n entre les arêtes non fiables de F et les canaux non fiables de S, telles que $b_p(n_1) = (\mathtt{Q}_1, S_1, R_1, \Delta_1), b_p(n_2) = (\mathtt{Q}_2, S_2, R_2, \Delta_2), b_f(a) = c$ et $f(c) = (n_1, n_2)$ ssi $c \in S_1, c \in R_2$ et $c \in F$

Une forme est dite décidable si le problème de l'accessibilité dans les systèmes ayant cette forme est décidable.

3.2 propriétés

Une sous forme est définie de manière similaire à un sous graphe. Soient les formes $F = (P, A_f, A_n, f)$ et $F' = (P', A'_f, A'_n, f')$, si $P' \subset P, A'_f \subset A_f, A'_n \subset A_n$ et f' est la restriction de f à $A'_f \cup A'_n$, alors F' est sous forme de F.

Proposition 3.1 (Sous forme). *Soit F une forme, F' une sous forme de F. si F' est indécidable alors F l'est aussi.*

Démonstration. Pour tout système de forme F' il est possible d'ajouter des participants ne faisant rien pour obtenir un système de forme F. Résoudre l'accessibilité dans les systèmes de forme F la résout pour ceux de forme F'.

On dit que $F' = (P, A'_f, A'_n, f')$ est une simplification de $F = (P, A_f, A_n, f)$ si on peut obtenir F' à partir de F en redirigeant des chemins passant par un noeud. Plus précisément, si p est un noeud. $c_1, \ldots, c_n$ sont des arêtes sortantes de p vers les noeuds $p_1, \ldots, p_n$ et c_e est une arête allant de p_e vers p. les arêtes $c_1, \ldots, c_n$ sont remplacées dans A'_f et A'_n par des arêtes allant de p_e vers $p_1, \ldots, p_n$. Si c_e et c_i sont des arêtes fiables, alors l'arête qui remplace c_i est fiable, sinon elle est non fiable. c_e n'est pas dans F'. On peut faire de même en choisissant un canal sortant de p et plusieurs entrant. le premier cas est une simplification en émission, le deuxième en réception.

par exemple une simplification pourrait se faire ainsi :

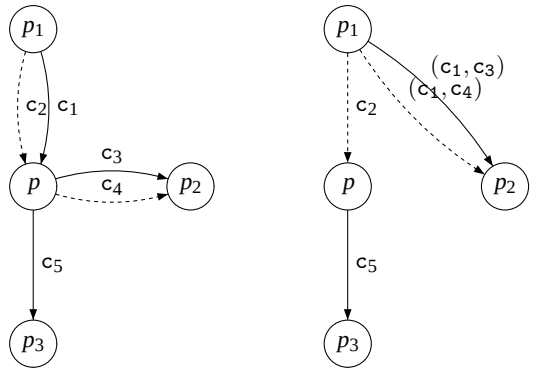

FIG. 2.

Proposition 3.2 (Simplification). *Soit F une forme, F' une simplification de F. si F' est indécidable alors F l'est aussi.*

Démonstration. Supposons que ce soit une simplification en émission. appelons p le participant autour duquel s'est faite la simplification et p_e le participant qui est à l'origine du canal c supprimé. L'idée, pour construire un système de forme $S\,F$ à partir d'un système S' de forme F', est de se servir du canal c pour transmettre les messages de plusieurs canaux. p ensuite retransmet ces messages dans les canaux correspondant. Pour cela, l'on étend l'alphabet des message en ajoutant le nom du canal sur lequel le message doit être retransmit. p est modifié pour que dans tout état il puisse lire sur c et ré-émettre sur le canal sortant correspondant au message.

pour tout run de S' il existe donc un run correspondant, dans S, mais où p lit les messages dés qu'ils arrivent et les retransmet, ce qui mime le run de S'. Les comportement que S a en plus ne sont que les comportements où p ne renvoie pas les messages. S' peut faire ces comportements en ne lisant pas sur les canaux qui ne sont pas fourni en message dans S.

Si les deux canaux du chemin remplacé sont fiables, il ne peut pas y avoir de perte de cette manière. Si l'un des deux est non fiable, il peut y avoir des pertes comme dans un canal non fiable.

On peut faire de la même manière si c'est une simplification en réception.

Soit F une forme contenant un canal c reliant les noeuds p_1 et p_2. On dit que la forme F' est la *fusion suivant un canal c* de F si on peut obtenir F' à partir de F en supprimant p_2 et c en reliants tous les canaux, reliés à p_2 dans F, à p_1 dans F'.

par exemple une fusion pourrait se faire ainsi :

Proposition 3.3 (Fusion fiable). *Soit F une forme, F' la fusion suivant le canal c de F. si c est fiable et F' est indécidable alors F l'est aussi.*

Démonstration. Pour cela, nous allons encore construire un système de forme F simulant un système de forme F'. Si le canal c relie les participants p_e et p_r dans F, qui sont remplacés par p dans F', alors p_r est p mais au lieu de lire et écrire sur les canaux qui arrivent et partent de p_e, p_r lit sur c un message correspondant à l'action. S'il ne le fait pas, il passe dans un état bloquant.

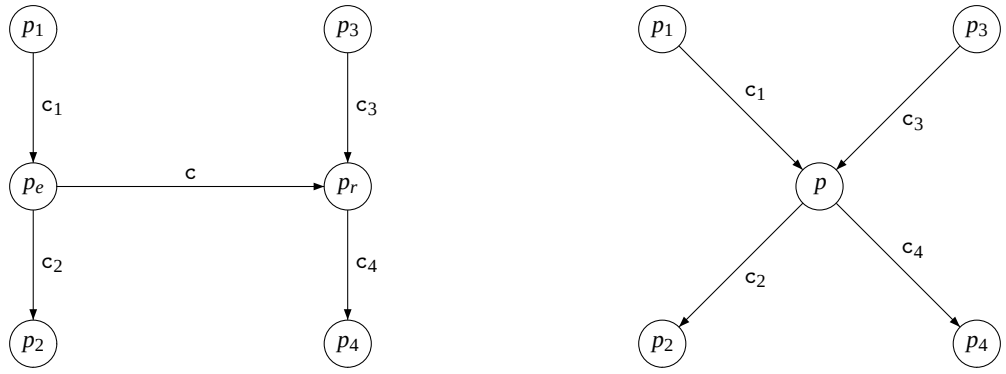

FIG. 3.

Ainsi si le run ne s'arrête pas dans un état bloquant, c'est que p_e a fait les actions correspondantes à ces message, dans le même ordre que p l'aurais fait, et l'a signalé sur le canal c.

Soit $r = \delta_1, \ldots, \delta_i, \delta_{i+1}, \ldots, \delta_l$ un run d'un système S. Si δ_i et δ_{i+1} sont des transitions de participants différents et que δ_i n'écrit pas une lettre lue par δ_{i+1}, alors $r' = \delta_1, \ldots, \delta_{i+1}, \delta_i, \ldots, \delta_l$ (en inversant δ_i et δ_{i+1}) est un run de S et est dit égal à réordonnement près à r.

Lemma 3.4. *Soit un système $S = (P, \mathtt{M}, \mathtt{C})$ de forme F, $c \in \mathtt{C}$ un canal et C_1, C_2 deux configurations de S où le canal c contient au plus un message.*

Si pour tout run r de S, il existe un run r' égal à réordonnement près tel que le canal $c \in \mathtt{C}$ ne contiennent jamais plus d'un message dans le canal, alors on peut construire un système S' de forme F', fusion de F suivant le canal c, et des configuration C_1', C_2' de S' telles que C_2 soit accessible depuis C_1 dans S ssi C_2' est accessible depuis C_1' dans S'.

L'idée est que si on peut faire que le canal ne contienne jamais plus d'un message, alors la donnée qu'il mémorise est finie et donc peut être stockée dans les états d'un participant. On peut ainsi construire une sorte d'automate produit des participants qui communiquent par ce canal.

Démonstration. Soient $p_1 = (\mathtt{Q}_1, S_1, R_1, \Delta_1)$ le participant émetteur sur le canal c et $p_2 = (\mathtt{Q}_2, S_2, R_i 2\Delta_2)$ le récepteur. Construisont un participant $p' = (\mathtt{Q}', S', R', \Delta')$ avec $\mathtt{Q}' = \mathtt{Q}_1 \times \{\mathtt{M} \cup \varepsilon\} \times \mathtt{Q}_2$ (ce sont les états de l'automate produit, dans lesquels on ajoute de la mémoire pour 1 message : $\{\mathtt{M} \cup \varepsilon\}$), $S' = (S_1/c) \cup S_2$, $R' = R_1 \cup (R_2/c)$, et

- si $q_1 \xrightarrow{c'!a} q_1' \in \Delta_1, c_i \neq c$ alors pour tout $m \in \mathtt{M}, q_2 \in \mathtt{Q}_2$, $q_1, m, q_2, \xrightarrow{c_i!a} q_1', m, q_2, \in \Delta'$, et de la même manière pour les écritures et les transitions de Δ_2
- si $q_1 \xrightarrow{c!a} q_1' \in \Delta_1$ alors pour tout $q_2 \in \mathtt{Q}_2$, $q_1, \varepsilon, q_2, \xrightarrow{c_i!a} q_1', a, q_2, \in \Delta'$
- si c est non fiable et si $q_1 \xrightarrow{c!a} q_1' \in \Delta_1$ alors pour tout $m \in \mathtt{M}, q_2 \in \mathtt{Q}_2$, $q_1, m, q_2, \xrightarrow{c_i!a} q_1', m, q_2, \in \Delta'$
- si $q_2 \xrightarrow{c?a} q_2' \in \Delta_2$ alors pour tout $q_1 \in \mathtt{Q}_1$, $q_1, a, q_2, \xrightarrow{c_i?a} q_1, \varepsilon, q_2', \in \Delta'$

Le système S' est $(P', \mathtt{M}, \mathtt{C}')$ avec $P' = (P/\{p_1, p_2\}) \cup \{p'\}$ et $\mathtt{C}' = \mathtt{C}/\{c\}$. À une configuration $C = (q_1, q_2, q_3 \ldots, w_c, w_{c_1}, \ldots)$ de S où $|w_c| \leq 1$ on fait correspondre $C' = ((q_1, w_c, q_2), q_3, \ldots, w_{c_1}, \ldots)$ dans S'.

Pour tout run d'une configuration C_1 à une configuration C_2 dans S, on a un run équivalent n'ayant jamais plus d'un message dans c, ainsi par construction, on obtient un run correspondant de C'_1 à C'_2 dans S'.

Pour pouvoir utiliser cette propriété sur l'exécution, il faut avoir une propriété structurelle disant quand ces conditions sont réunies.

On peut par exemple remarquer que cela s'applique dans le cas où un participant n'a qu'un canal en écriture. Comme la seule contrainte pour le réordonnement des transition de ce participant ne viennent que de ce canal, on peut toujours retarder ses écritures juste avant les lectures par le receveur. Ainsi il n'y a jamais plus d'un message dans le canal.

Ceci peut être généralisé par :

Proposition 3.5 (Réordonnement). *Soit une forme F, c un canal de F. p_e le participant écrivant sur c, p_r celui lisant sur c. Si il n'y a pas d'autre chemin dans F de p_e à p_r, alors pour tout run de tout système de forme F, ne commençant ni ne finissant dans une configuration où c contient plus d'un message, il existe un run égal a reordonnement près, dans lequel c ne contient jamais plus d'un message.*

Démonstration. Soit un système S de forme F et un run r de S allant de la configuration C_1 à C_2, c ne contenant pas plus d'un message dans C_1 et C_2. Si δ_e une écriture sur c et δ_r la lecture correspondante. On peut réordonner les règles de telles manière qu'il n'y ait pas de transition de p_e entre ces deux règles. En effet, comme il n'existe pas d'autre chemin entre p_e et p_r que c, aucun des participants qui attendent un message de p_e pour pouvoir continuer leur exécution ne peut écrire de message à p_r (ou a un participant pouvant écrire un message à p_r,...). Ainsi toutes les transitions qui doivent forcement être exécutées avant δ_r peuvent être faites avant les transitions de p_e.

Donc δ_r peut être faite avant la deuxième écriture sur c.

On dit qu'un système est réductible par fusion si il existe un canal reliant le participant p_e à p_r tel qu'il n'y ait pas d'autre chemin de p_e à p_r.

Ici, par fermeture vers le haut, on entends ici, suivant l'ordre $\sqsubseteq$ (l'ordre sous mot, $m \sqsubseteq m'$ si l'on peut obtenir m en effaçant des lettres de m'). On notera $[R_1,\ldots,R_n]$ l'ensemble $\{(w_1,\ldots,w_n | w_1 \in R_1,\ldots,w_n \in R_n\}$, et $\uparrow L$ la fermeture vers le haut de L. On peut étendre l'ordre sous mot aux tuples de mots par $(w_1,\ldots,w_n) \sqsubseteq (w'_1,\ldots,w'_n) si \forall i, 1 \leq i \leq n, w_i \sqsubseteq w'_i$.

Proposition 3.6 (Découpage). *Soit une forme F constituée des deux formes F_1 et F_2, et telle qu'il n'y ait entre F_1 et F_2 que des canaux non fiables, orientés de F_1 vers F_2. F est décidable ssi F_1 et F_2 sont décidables.*

Démonstration. Supposons F_1 et F_2 décidables.

Soit $S = (P, \mathtt{M}, \mathtt{C})$ un système de forme F. $c_1,\ldots,c_n$ sont les canaux reliant les participants de F_1 à ceux de F_2. $S_1 = (P_1, \mathtt{M}, \mathtt{C}_1)$ et $S_2 = (P_2, \mathtt{M}, \mathtt{C}_2)$ sont les parties de S correspondantes à F_1 et F_2 (en gardant les canaux $c_1,\ldots,c_n$ dans $\mathtt{C}_1$ et $\mathtt{C}_2$).

Soient C_i, C_f des configurations de S où tous les canaux sont vide. Si $w = (w_1,\ldots,w_n)$, $C_{(i,w)}, C_{(f,w)}$ sont les configurations C_i, C_f, avec les mots $w_1,\ldots,w_n$ dans les canaux $c_1,\ldots,c_n$.

$C_{[i_1,w)}, C_{[f_1,w)}$ sont les parties de $C_{(i,w)}, C_{(f,w)}$ correspondant à S_1 et $C_{[i_2,w)}, C_{[f_2,w)}$ celles correspondant à S_2.

On peut tester si C_{f_2} est accessible depuis $C_{[i_2,(w_1,\ldots,w_n))}$. En effet comme F_2 est décidable, ajouter n participant pour écrire dans les canaux $c_1,\ldots,c_n$ donne une forme, qui peut se réduire par fusion à la forme F_2, c'est donc une forme décidable. Donc si ces participants écrivent les mots $(w_1,\ldots,w_n)$ dans $c_1,\ldots,c_n$ (on peut ici choisir que ces canaux sont fiables), on peut tester l'accessibilité d'une configuration donnée depuis $C_{[i_2,(w_1,\ldots,w_n))}$.

Soient $L \subset \mathtt{M}^n$ le langage des tuples de mots $(w_1,\ldots,w_n)$ tels que C_{f_2} est accessible depuis $C_{[i_2,w)}$. C'est à dire que C_f est accessible depuis C_i ssi $C_{[f_1,w)}$ est accessible depuis C_{i_1}, avec $w \in L$.

Comme les canaux $c_1,\ldots,c_n$ sont non fiables, ce que S_1 peut écrire dans les canaux est en fait le langage $\uparrow L$. Comme $\sqsubseteq$ est un bon ordre (lemme de Higman), et un ensemble fermé vers le haut par un bon ordre a un nombre fini d'éléments minimaux, $\uparrow L$ a donc un nombre fini d'éléments minimaux.

Grâce à cela on a : Soit W un ensemble fini de tuples de mots. Il est aussi possible de tester que $\uparrow L / \uparrow W$ est vide. Pour cela, il faut tester si il existe un tuple de mot w dans $(\Sigma^\star)^n / \uparrow W$ tel que C_{f_2} est accessible depuis $C_{[i_2,(w_1,\ldots,w_n))}$, c'est à dire si $((\Sigma^\star)^n / \uparrow W) \cap \uparrow L$ est vide. Pour cela, on se réfère au lemme suivant. qui nous dit que $(\mathtt{M}^\star)^n / \uparrow W$ est une une union finie d'ensembles de la forme $[R_1,\ldots,R_n]$ avec $R_1,\ldots,R_n$ des langages rationnels. On peut donc construire pour chaque ensemble $[R_1,\ldots,R_n]$ des participants $p_1,\ldots,p_n$ écrivant ces langages sur les canaux $c_1,\ldots,c_n$. On teste ainsi si $[R_1,\ldots,R_n] \cap \uparrow L$ est vide. On peut donc savoir si $\uparrow L / \uparrow W$ est vide en testant l'accessibilité dans un nombre fini de systèmes de forme F_2.

Avec cela, on est capable de construire l'ensemble $\uparrow L$, il suffit de tester tous les tuples de mots pour savoir si ils sont dans $\uparrow L$ et de s'arrêter quand il n'existe plus de tuples dans $\uparrow L$ qui ne soit pas supérieur a un tuple déjà trouvé.

Il faut ensuite tester si S_1 peut écrire ces mots dans les canaux. Pour cela on utilises des participants supplémentaires, de la même manière que précédemment, mais qui lisent sur les canaux au lieu d'écrire.

Et comme $\uparrow L$ est vide ssi L est vide. Et L est vide ssi il n'existe pas w tel que $C_{[f_1,w)}$ est accessible depuis C_{i_1} dans S_1 et C_{f_2} est accessible depuis $C_{[i_2,w)}$, c'est à dire ssi C_f n'est pas accessible depuis C_i. On est donc capable de calculer l'accessibilité, F est donc décidable.

L'autre sens du ssi découle de la propriété de sous forme.

Lemma 3.7. *Soit Σ un alphabet fini. Soit $W \in \Sigma^{\star n}$ un ensemble fini de tuples de mots de $\Sigma^s tar$. $(\Sigma^\star)^n / \uparrow W$ peut être décrit comme une union finie de langages de la forme $[R_1,\ldots,R_n]$ avec $R_1,\ldots,R_n$ des langages rationnels.*

Démonstration. Soient $R_1,\ldots,R_n$ des langages rationnels sur $\Sigma^s tar$. Soient $w_1,\ldots,w_n$ des mots de $\Sigma^s tar$. On peut déjà remarquer que $[R_1,\ldots,R_n] / \uparrow (w_1,\ldots,w_n)$ est l'ensemble tel que pour tout tuple de mots $(m_1,\ldots,m_n)$ il y a au moins un mot m_i tel que $w_i \not\sqsubseteq m_i$. Donc $[R_1,\ldots,R_n] / \uparrow (w_1,\ldots,w_n) = \bigcup_{1\leq i' n} [R_1,\ldots,R_i / \uparrow w_i,\ldots,R_n]$. Comme $\uparrow w_i$ est rationnel, $R_i / \uparrow w_i$ est rationnel.

De même, si L est une union finie d'ensembles de la forme $[R_1,\ldots,R_n]$ avec $R_1,\ldots,R_n$ rationnels, $L / \uparrow (w_1,\ldots,w_n)$ est de la même forme.

Pour $W = \{(w_1^1, \ldots, w_n^1), \ldots, (w_1^k, \ldots, w_n^k)\}$ un ensemble de k tuples de mots. $L/\uparrow W = L/\uparrow (w_1^1, \ldots, w_n^1) \ldots /\uparrow (w_1^k, \ldots, w_n^k)$. Ainsi par récurrence $(\Sigma^\star)^n/\uparrow W$ est une union finie d'ensembles de la forme, $[R_1, \ldots, R_n]$ avec $R_1, \ldots, R_n$ rationnels.

On dit qu'une forme F est découpable si elle est constituée des deux formes F_1 et F_2, et telle qu'il n'y ait entre F_1 et F_2 que des canaux non fiables, orientés de F_1 vers F_2.

Avec les propriétés de réduction par fusion et de découpage, nous pouvons démontrer la décidabilité d'un système complexe à partir de systèmes simples. Avec les propriétés de sous forme, de simplification, de fusion fiable et de découpage, c'est l'indécidabilité que nous pouvons prouver.

4 Cas indécoupables et irréductibles par fusion

Nous allons d'abord voir ces cas pour les formes à deux noeuds, qui nous servirons ensuite a généraliser. Dans les figures, les arêtes en pointillé représente les canaux non fiables, les autres les canaux fiables.

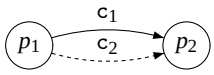

FIG. 4. La seule forme indécoupable, irréductible à deux noeuds, contenant des canaux fiables et décidable.

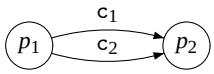

FIG. 5.

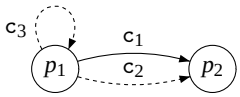

FIG. 6.

Proposition 4.1. *la forme décrite par la figure 5 est indécidable.*

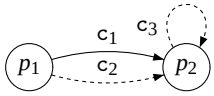

FIG. 7.

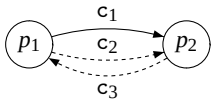

FIG. 8.

Démonstration. Soit une instance $I = \{(u_1, v_1, \ldots, u_n, v_n\}$ de PCP. Pour $\sigma = \sigma_1 \ldots \sigma_k$ une série d'indices de $1 \ldots n$ on note u_σ le mot $u_{\sigma_1} \ldots u_{\sigma_n}$. On peut construire un participant p_e qui peut,pour tout σ, écrire u_σ sur le canal c_1 et v_σ sur le canal c_2. On peut aussi construire un second participant p_2 qui vérifie si les deux canaux contiennent les même message. Ainsi il est possible de construire un système S et deux configurations C_i, C_f tels que l'accessibilité de C_f depuis C_i dans S soit équivalent à l'existence d'une solution à I.

Comme PCP est indécidable, cette forme est indécidable.

Proposition 4.2. *les formes décrites par les figures 6 7 8 et 9 sont indécidable.*

Démonstration. On peut se restreindre à une variante de PCP dans laquelle si σ est solution de l'instance $I = \{(u_1, v_1, \ldots, u_n, v_n\}$, alors pour tout préfixé σ' de σ, $u_{\sigma'}$ est préfixés de $v_{\sigma'}$. Ce problème est aussi indécidable. En fait la preuve classique de l'indécidabilité de PCP prouve aussi cela.

Soit une instance $I = \{(u_1, v_1, \ldots, u_n, v_n\}$ de PCP. Dans les 4 cas, p_1 écrira les mots u_i dans c_1, le canal fiable, et les mots v_i dans c_2, et p_2 vérifiera que les contenu des deux canaux sont égaux. Il faut ensuite s'assurer qu'il n'y a pas de pertes de message dans le canal non fiable. Pour cela, on s'assure que ce qui est écrit (ou lu suivant le cas) dans le canal non fiable est toujours plus court que ce qui y est ecrit dans le fiable. Ainsi, si p_2 lit le même contenu dans les deux canaux, c'est qu'il n'y a pas eu de pertes. Pour s'assurer que le contenu écrit (ou lu) dans c_2 est plus court, on compte la différence de taille entre les mots de c_1 et c_2 dans c_3. Comme ce compteur est non fiable, sont contenu peut changer. Comme le problème choisi nous autorise à ne compter que des valeurs positives, les pertes ne font que diminuer cette valeur. Ainsi les pertes dans le

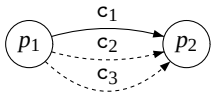

FIG. 9.

compteur ne permettent pas au mot de c_2 d'être plus long que celui de c_1. La différence entre ces 4 cas se trouve dans le participant qui incrémente et décrémente le compteur.

- Dans le cas 6, p_1 incrémente et décrémente. a chaque fois que p_1 écrit un mot plus long dans c_1 que dans c_2, il ajoute la différence de longueur au compteur. à chaque fois qu'il ecrit un mot plus long dans c_2, il retranche la différence de longueur au compteur. Si il atteint une valeur négative, il bloque.
- pour le cas 7, p_2 incrémente et décrémente. Pour que p_2 puisse compter la différence de taille, p_1 doit envoyer ces informations. Il les envoie en plus des mots sur le canal c_1. Comme il est fiable, c'est possible d'envoyer des valeurs positives ou négatives.
- pour le cas 8, p_2 incrémente, p_1 décrémente. C'est fait de la même manière que le cas précédent, mais p_1 n'envoie que les informations pour incrémenter.
- pour le cas 9 c'est identique, mais les rôles sont inversés.

Ainsi on peut tester qu'il n'y a pas eu de pertes. Si le contenu des deux canaux est identique et qu'il n'y a pas eu de pertes, c'est que le système a résolu une instance de cette variante de PCP. Ces formes sont donc aussi indécidables.

Proposition 4.3. *Si une forme est indécoupable et irréductible par fusion, qu'elle a au moins 3 noeuds, et qu'elle contient au moins une arête fiable, alors elle est indécidable.*

Démonstration. Soient p_e et p_r des noeuds émetteur et récepteur sur une arête fiable. Comme la forme est irréductible par fusion, il existe un autre chemin de p_e à p_r.

- Si ce chemin est direct de p_e à p_r. Comme il existe un troisième noeud, et que la forme est indécoupable, alors il y a une arête depuis p_e ou p_r vers ce noeud (p_3). (si il y a plus de 3 noeuds, il y en a au moins un qui vérifie cela).
  - si cette arête est non fiable, alors , comme la forme est indécoupable, il existe un chemin depuis p_3 vers p_e ou p_r. On se ramène donc par sous forme et simplifications à une des 4 formes décrites par les figures 6 7 8 et 9
  - si l'arête est fiable, alors comme la forme est irréductible, il existe un autre chemin qui mène à p_3. Si on fusionne l'arête fiable allant à p_3 on se retrouve avec les cas 6 ou 7.
- Si ce chemin n'est pas direct, alors il passe par un noeud p_3. Comme la forme est irréductible, il existe un deuxième chemin de p_e vers p_3 et de p_3 vers p_r.
  - si le deuxième chemin de p_e vers p_3 passe par p_r alors il y a une boucle de p_r vers p_3 puis de p_3 vers p_r. On est dans le cas 7 par sous forme et simplifications.
  - si il ne passe pas par p_r
    - si le deuxième chemin de p_3 vers p_r passe par p_e, il y a une boucle sur p_e ,c'est le cas 6
    - sinon il y a 3 chemins distincts de p_e vers p_r, c'est le cas 9.

Nous pouvons maintenant lister tous les cas indécoupables et irréductibles par fusion :

- si il n'y a pas de canaux fiables, les systèmes de cette forme sont des systèmes à canaux non fiables, et l'accessibilité y est décidable.
- si il y a au moins une arête fiable :
  - si il y a un seul noeud, il a une boucle fiable. C'est une machine de Turing

- si il y a deux noeuds, c'est soit le cas 4, qui est montré décidable dans l'article en annexe, soit cela contient l'un des 5 autres cas 5 6 7 8 et 9 et c'est indécidable.
- si il y a plus de 3 noeuds, c'est indécidable.

Dit plus simplement, une forme est indécoupables et irréductibles par fusion est décidable ssi elle ne contiens pas de canaux fiables ou que c'est la forme 4. Nous appellerons ces cas, les cas minimaux décidables.

Proposition 4.4. *Si une forme F est indécoupable, mais réductible par fusion, et si F ne peut pas se réduire dans les formes minimales décidables, alors F est indécidable.*

Démonstration. Si F n'est pas réductible en des formes minimales décidables, alors F contient des arête fiables et toutes ses réductions en contiennent aussi.

- Soit F contient un chemin qui forme une boucle fiable, alors c'est une forme indécidable.
- Soit F n'en contient pas. Soit F' est une forme réduite suivant des arêtes fiables à partir de F et qu'il n'est plus possible de réduire ainsi (il peut y avoir d'autres réductions mais seulement avec des arêtes non fiables). Il existe donc une arête fiable c, reliant deux noeuds p_e à p_r doublé par un autre chemin dans F'.
  - si cet autre chemin est fiable alors F' est indécidable, donc F aussi.
  - si ce chemin non fiable contient une boucle, alors c'est simplifiable dans le cas des figures 6 et 7.
  - Il y a deux noeuds p_1, p_2 dans le chemin tels qu'il y ait deux chemins de p_1 à p_2, alors c'est simplifiable dans le cas de la figure 9.
  - Sinon, comme la forme ne peut pas se réduire dans une forme minimale décidable, elle contient donc un noeud p hors de ce chemin. Comme la forme est indécoupable, il y a une arête depuis ou vers p
    - soit il y a une arête fiable vers un noeud hors de p_e, p_r et ceux constituant le chemin de p_e à p_r. comme toutes les arêtes fiables sont irréductibles, elle est donc doublé. En fusionnant suivant cette arête fiable, on obtient les cas des figures 6 et 7
    - si il n'y a pas d'arête fiable, alors comme la forme est indécoupable, il existe un chemin non fiable dans l'autre sens. Ceci permet donc de simplifier vers l'un des 4 cas indécidables à 2 noeuds évoqués précédemment.

5 résultat complet

Theorem 5.1. *Une forme est décidable ssi elle est découpable en formes qui peuvent se réduire par fusions aux cas minimaux décidables.*

Démonstration. Remarquons d'abord que la fusion ne retire pas de chemins, donc si une forme n'est pas découpable, toute fusion depuis cette forme reste indécoupable. On peut donc bien faire tous les découpages avant les réductions par fusions. Comme le nombre de découpages et de fusions possibles est fini, il est possible de tous les essayer. Si il existe une manière de découper en des formes qui se réduisent dans les cas minimaux décidables, alors on peut la trouver.

Découper n'ajoute pas de chemins, donc ne retire pas de découpes possibles. Quel que soit l'ordre de découpe d'une forme, on se retrouve avec les même formes non découpables. On peut alors caractériser les formes décidables plus directement.

Une forme est décidable ssi c'est un graphe acyclique de formes réductibles dans les cas minimaux, reliés par des canaux non fiables.

6 Conclusion

Avec cette caractérisation, on voit qu'il est possible d'utiliser des canaux fiables seulement dans une boucle avec au moins un canal non fiable. Les autres cas décidables sont des cas ou la communication est unidirectionnelle, ce qui n'a pas vraiment de sens pour modéliser des protocoles.

Par contre, comme précisé précédemment, l'étude de ce problème, en particulier le cas décrit par la figure 4, nous a permis d'étendre le champ des problèmes équivalents à l'accessibilité dans les systèmes à canaux non fiable.

Nous nous intéressons désormais à caractériser plus finement cette classe de problèmes, et en particulier, nous avons obtenu une borne inférieur de complexité plus élevée que non primitive récursive.

Références

1. P. A. Abdulla, J. Deneux, J. Ouaknine, and J. Worrell. Decidability and complexity results for timed automata via channel machines. In *Proc. ICALP 2005*, volume 3580 of *Lecture Notes in Computer Science*, pages 1089–1101. Springer, 2005.
2. P. A. Abdulla and B. Jonsson. Verifying programs with unreliable channels. *Information and Computation*, 127(2) :91–101, 1996.
3. R. Amadio and Ch. Meyssonnier. On decidability of the control reachability problem in the asynchronous π-calculus. *Nordic Journal of Computing*, 9(2) :70–101, 2002.
4. G. Cécé. Ring networks are easier to verify. Unpublished manuscript, ? ?
5. S. Demri and R. Lazić. LTL with the freeze quantifier and register automata. In *Proc. LICS 2006*, pages 17–26. IEEE Comp. Soc. Press, 2006.
6. A. Finkel. Decidability of the termination problem for completely specificied protocols. *Distributed Computing*, 7(3) :129–135, 1994.
7. D. Gabelaia, A. Kurucz, F. Wolter, and M. Zakharyaschev. Non-primitive recursive decidability of products of modal logics with expanding domains. *Annals of Pure and Applied Logic*, 142(1–3) :245–268, 2006.
8. B. Konev, F. Wolter, and M Zakharyaschev. Temporal logics over transitive states. In *Proc. CADE 2005*, volume 3632 of *Lecture Notes in Computer Science*, pages 182–203. Springer, 2005.
9. S. Lasota and I. Walukiewicz. Alternating timed automata. In *Proc. FOSSACS 2005*, volume 3441 of *Lecture Notes in Computer Science*, pages 250–265. Springer, 2005.
10. R. Lazić, T. Newcomb, J. Ouaknine, A. W. Roscoe, and J. Worrell. Nets with tokens which carry data. In *Proc. ICATPN 2007*, volume 4546 of *Lecture Notes in Computer Science*, pages 301–320. Springer, 2007.
11. J. Ouaknine and J. Worrell. On the decidability of metric temporal logic. In *Proc. LICS 2005*, pages 188–197. IEEE Comp. Soc. Press, 2005.
12. J. Ouaknine and J. Worrell. On metric temporal logic and faulty Turing machines. In *Proc. FOSSACS 2006*, volume 3921 of *Lecture Notes in Computer Science*, pages 217–230. Springer, 2006.
13. Ph. Schnoebelen. Verifying lossy channel systems has nonprimitive recursive complexity. *Information Processing Letters*, 83(5) :251–261, 2002.