Les systèmes à canaux non-fiables vus comme des transducteurs

Jérémie Dimino

Laboratoire Spécification et Vérification
ENS de Cachan

Résumé Ce document constitue le rapport de mon stage de M1 que j'ai effectué au Laboratoire Spécification et Vérification à l'ENS de Cachan, sous la direction de Philippe Schnoebelen.

Durant ce stage nous nous sommes intéressés à un encodage des transducteurs dans les systèmes à canaux, qui nous a permis d'obtenir plusieurs résultats sur les systèmes à canaux non-fiables. Certains de ceux-ci sont des résultats existants mais redémontrés de manière plus élégante, d'autres sont nouveaux.

Table des matières

1 Introduction

Les systèmes à canaux sont des automates finis communiquant de manière asynchrone par l'intermédiaire de canaux fifo non-bornés. Ils sont un modèle naturel pour les protocoles de communication asynchrone, et sont en pratique utilisés comme sémantique de base pour des langages de spécification de protocole tels que SDL et Estelle.

Il est assez facile de voir que les systèmes à canaux sont un modèle de calcul puissant. Ils ont en fait la puissance de calcul des machines de Turing ; un canal pouvant simuler une bande de travail d'une machine de Turing. Il suffit ensuite d'ajouter un marqueur pour noter l'emplacement de la tête de lecture.

Partant de là, on ne dispose pas de méthode de vérification systématique pour les systèmes à canaux. On doit se restreindre à des méthodes algorithmiques répondant partiellement au problème posé.

Le paradoxe des canaux non fiables Les systèmes à canaux non fiables ont été introduits il y a quelques années par Finkel [Fin94a], et de manière indépendante par Abdulla et Jonsson [AJ96b]. Dans de tels systèmes les messages peuvent être perdus à tout moment sans notification. Cela peut servir par exemple à modéliser des protocoles de communication dans lesquels la communication n'est pas supposée être fiable.

De manière surprenante, certaines propriétés deviennent décidables lorsque l'on passe d'une sémantique parfaite à une sémantique avec perte. Parmi elles

on trouve plusieurs problèmes utilisés en vérification tels que la terminaison ou l'accessibilité.

On pourrait penser que le fait d'autoriser les pertes de messages rend les systèmes à canaux triviaux. Il n'en est rien. Beaucoup de problèmes sont indécidables même avec une sémantique non fiable. En fait la frontière entre les problèmes décidables et les problèmes indécidables pour les systèmes à canaux non fiables n'est pas encore bien explorée.

En particulier les réductions utilisées dans les preuves d'indécidabilité sont difficiles à réutiliser, et se basent souvent sur des résultats d'indécidabilité admis par analogie. N'étant pas satisfait de cet état des choses, Philippe Schnoebelen à cherché des preuves plus simples (cf. notes du cours 2-9 du MPRI) ; elles sont basées sur des systèmes à canaux bornés. Cependant les preuves reposent sur des problèmes sur les systèmes à canaux bornés pour lesquels l'indécidabilité n'a pas été prouvée.

Mayr [May00] propose un autre modèle des systèmes non fiables basé sur les machines à compteurs. Les systèmes qu'il étudie se comportent comme des machines de Minsky à l'exception que les compteurs peuvent diminuer à tout moment. Les réductions qu'il utilise partent des machines de Minsky et sont donc plus simples. Cependant son travail n'a pas été poursuivi à notre connaissance. De plus les machines à compteurs non fiables peuvent être vues comme un cas particulier de systèmes à canaux non fiables, où un compteur est représenté par un canal où le nombre est écrit en unaire et le début est marqué par une lettre spéciale. En conséquence tous les résultats obtenus sur les machines à compteurs ne se transportent pas sur les systèmes à canaux non fiables, seuls les résultats négatifs restent valables. En effet les machines à compteurs non fiables constituent un modèle qui n'a pas la puissance des systèmes à canaux non fiables [CS08].

Contribution apportée Dans ce rapport nous allons proposer une nouvelle approche pour les systèmes à canaux non fiables. Pour cela nous allons exploiter l'analogie qui existe entre un transducteur[1] et un système à un canal. En effet, on peut voir un système à un canal comme un transducteur ; tout deux sont dirigés par un ensemble fini d'états de contrôle, changent d'état et écrivent en fonction de ce qu'ils lisent. Les différences sont les conditions aux bornes, i.e. pour un transducteur, il est bien défini quand celui-ci commence (état initial) et termine un calcul en acceptant le mot d'entré (états finaux) alors que ce n'est pas le cas pour les systèmes à canaux, ceux-ci peuvent lire ce qu'ils ont précédemment écrit et ainsi boucler.

Nous donnerons un codage des transducteurs dans les systèmes à canaux qui évite d'introduire des boucles, et nous donnerons plusieurs lemmes pour relier les sémantiques opérationnelles des deux modèles. En particulier on montrera comment encoder le calcul des itérés d'un transducteur dans les systèmes à canaux.

[1] L'idée d'utiliser des transducteurs est due à Pierre Chambart.

Partant du fait que les transducteurs sont un modèle ancien et richement étudié, nous partirons de problèmes indécidables pour les transducteurs et obtiendrons des réductions simples et réutilisables pour démontrer l'indécidabilité de problèmes de terminaisons et d'équité sur les systèmes à canaux non fiables.

2 Définitions et notations

Définition 2.1 (Système à canaux). *Un système à un canal est un triplet* $S = (Q, \Sigma, \Delta)$ *où :*

- $Q = \{r, s, \ldots\}$ *est un ensemble fini d'états de contrôle,*
- $\Sigma = \{a, b, \ldots\}$ *est un ensemble fini de messages,*
- $\Delta \subseteq Q \times \Sigma^* \times Q \times \Sigma^*$ *est un ensemble fini de règles de transitions.*

Une règle $\delta \in \Delta$ de la forme (s, w_l, r, w_e) s'écrit $s \xrightarrow{?w_l\ !w_e} r$ et signifie que S peut passer de l'état de contrôle s à l'état de contrôle r en lisant w_l et en écrivant w_e. Lire w_l n'est possible que si le canal commence par le mot w_l.

Remarque 2.2. La définition 2.1 suppose que le système comporte uniquement un canal. En effet l'analogie transducteurs/systèmes à canaux se fait en n'utilisant que des systèmes à un canal. De plus comme tous les résultats que nous allons montrer sont des résultats de dureté, le fait de n'utiliser qu'un seul canal n'est pas une perte de généralité.

Remarque 2.3. La définition 2.1 suppose qu'il y a une seule partie automate dans un système à canaux, i.e. un seul ensemble d'états de contrôle. Ceci n'est pas une perte de généralité puisque plusieurs parties peuvent être combinées en une seule via un produit asynchrone classique d'automates.

2.1 Sémantique opérationnelle fiable

Étant donné un système à canaux S, on lui associe le système de transition $\mathcal{T}_{\text{fbl}}(S) = \langle Conf_S, \rightarrow_{\text{fbl}} \rangle$ où $Conf_S = Q \times \Sigma^*$ est l'ensemble des configurations du système, et $\rightarrow_{\text{fbl}} \subseteq Conf_S \times Conf_S$ est la relation de transition fiable non étiquetée.

Une *configuration* de S est une paire $\sigma = \langle r, w \rangle$ où $r \in Q$ est un état de contrôle et $w \in \Sigma^*$ représente le contenu du canal.

Les transitions possibles entre les configurations sont données par les règles de transitions de S. Formellement, pour $\sigma, \sigma' \in Conf_S$, on a $\sigma \rightarrow_{\text{fbl}} \sigma'$ ssi σ est de la forme $\langle s, w_l.w \rangle$, il existe une règle de transition $s \xrightarrow{?w_l\ !w_e} r$ dans Δ et $\sigma' = \langle r, w.w_e \rangle$.

2.2 Sémantique opérationnelle non fiable

La sémantique des systèmes à canaux non fiables diffère de celle des système à canaux fiables : les messages peuvent être perdus durant la communication. Pour formaliser la notion de perte, nous allons introduire un ordre sur les mots et les configurations.

Définition 2.4. *On définit la relation $\sqsubseteq$ sur les mots de la manière suivante : étant donnés deux mots $u = a_1 \dots a_n$ et $v = b_1 \dots b_m$, $u \sqsubseteq v$ ssi il existe une suite croissante $1 \leq i_1 < i_2 < \dots < i_n \leq m$ telle que pour tout $1 \leq j \leq n$, $a_j = b_{i_j}$.*

$u \sqsubseteq v$ se lit "u est un sous-mot de v". Cette relation est bien un ordre sur Σ^* et même un beau préordre (un "well-quasi-order" en anglais) d'après le lemme de Higman.

On étend cet ordre sur les mots en un ordre sur les configurations de la manière suivante :

$$\langle s, w\rangle \sqsubseteq \langle s', w'\rangle \stackrel{\text{def}}{\equiv} s = s' \wedge w \sqsubseteq w'.$$

Cet ordre est aussi un beau préordre.

On peut maintenant définir la sémantique opérationnelle des système à canaux non fiables de la manière suivante : étant donné un système à canaux S, on lui associe le système de transition $\mathcal{T}_{\neg\text{fbl}}(S) = \langle Conf_S, \rightarrow_{\neg\text{fbl}}\rangle$ où :

- $Conf_S = Q \times \Sigma^*$ est l'ensemble des configurations du système, qui est le même que pour la sémantique opérationnelle fiable,
- $\rightarrow_{\neg\text{fbl}} \subseteq Conf_S \times Conf_S$ est la relation de transition non fiable non étiquetée définie par :

$$\sigma \rightarrow_{\neg\text{fbl}} \sigma' \stackrel{\text{def}}{\Leftrightarrow} \exists \theta, \theta' \in Conf, \sigma \sqsupseteq \theta \rightarrow_{\text{fbl}} \theta' \sqsupseteq \sigma'.$$

Avec cette définition, on peut perdre des messages avant ou après avoir franchi une règle de transition. C'est la notion de perte la plus générale, qui a été introduite par Abdulla et Jonsson [AJ96a].

Dans la suite nous utiliserons la notation $\rightarrow$ pour $\rightarrow_{\neg\text{fbl}}$.

2.3 Transducteurs

Un transducteur est un automate fini qui de plus produit un mot en sortie. Il "transforme" le mot d'entrée. Formellement, un transducteur se définit de la manière suivante :

Définition 2.5 (Transducteur). *Un transducteur est un quintuple $T = (Q, \Sigma, q_0, F, \Delta)$ où :*

- *$Q = \{r, s, \dots\}$ est un ensemble fini d'états,*
- *$\Sigma = \{a, b, \dots\}$ est un ensemble fini de lettres,*
- *$q_0 \in Q$ est l'état initial,*
- *$F \subseteq Q$ est un ensemble fini d'états finaux,*
- *$\Delta \subseteq Q \times \Sigma \times Q \times \Sigma^*$ est un ensemble fini de règles de transitions.*

Une règle $(q, a, q', w) \in \Delta$ de T s'écrit $q \xrightarrow{?a\ !w} q'$ et signifie que T peut passer de l'état q à l'état q' en lisant a dans le mot d'entré et écrivant w sur le mot de sortie.

Dans ce rapport nous nous intéresserons en particulier aux transducteurs temps réel, qui écrivent exactement une lettre pour chaque lettre lue :

Définition 2.6 (Transducteur temps réel). *Un transducteur temps réel est un transducteur $T = (Q, \Sigma, q_0, F, \Delta)$ tel que :*

$$\Delta \subseteq Q \times \Sigma \times Q \times \Sigma.$$

2.4 Sémantique opérationnelle des transducteurs

Étant donné un transducteur T on lui associe le système de transition $\mathcal{T}(T) = \langle Conf, \rightarrow \rangle$ où $Conf = Q \times \Sigma^* \times \Sigma^*$ est l'ensemble des configurations du transducteur et $\rightarrow \subseteq Conf \times Conf$ est la relation de transition. [2]

Une configuration de T est un triplet $\langle q, w_e, w_s \rangle$ où q est l'état de contrôle, w_e est le mot d'entré et w_s est le mot de sortie.

Les transitions sont dirigées par les règles de transitions de T. Étant données deux configuration σ et σ' de $Conf$ on a $\sigma \rightarrow \sigma'$ ssi :

- σ s'écrit $\langle q, a.w_e, w_s \rangle$,
- il existe une règle de la forme $q \xrightarrow{?a\ !w} q'$ dans Δ,
- $\sigma' = \langle q', w_e, w_s.w \rangle$.

La transduction R_T se définit maintenant comme suit à partir de la fermeture réflexive transitive $\xrightarrow{*}$ de $\rightarrow$:

Définition 2.7. *On définit la relation $R_T \subseteq \Sigma^* \times \Sigma^*$ de la manière suivante : pour tous mots w et w' de Σ^* :*

$$w R_T w' \overset{\text{def}}{\Leftrightarrow} \exists q_f \in F, \langle q_0, w, \varepsilon \rangle \xrightarrow{*} \langle q_f, \varepsilon, w' \rangle.$$

3 Les systèmes à canaux non-fiables vus comme des transducteurs

3.1 Définitions

Dans cette partie nous allons décrire une manière de construire un système à canaux calculant les itérés d'un transducteur temps réel (excepté pour le mot vide).

Le système à canaux que nous construisons doit refléter le calcul du transducteur et donc ne pas introduire de boucles dans le calcul d'une transformation par le transducteur, même s'il y a perte de message.

Pour cela nous allons utiliser une technique consistant à travailler avec deux copies du même alphabet. Étant donné un alphabet $\Sigma = \{a, b, c, \ldots\}$ on en construit une copie $\overline{\Sigma} = \{\overline{a}, \overline{b}, \overline{c}, \ldots\}$. $\overline{\Sigma}$ est tel que $\Sigma \cap \overline{\Sigma} = \emptyset$.

Étant donnée une lettre $a \in \Sigma$, on dira que $\overline{a}$ est une lettre marquée. Pour simplifier les définitions, on introduit une notation pour l'alphabet de toutes les lettres, marquées ou non :

$$\ddot{\Sigma} \overset{\text{def}}{=} \Sigma \uplus \overline{\Sigma}.$$

[2] La sémantique opérationnelle des transducteurs est souvent donnée avec moins de granularité, mais celle que nous donnons est équivalente.

Enfin on définit la notion de mot marqué de la façon suivante : étant donné un mot $w = a_1 \dots a_n$ de Σ^*, on notera $\overline{w}$ le mot de $\overline{\Sigma}^*$ défini par :

$$\overline{w} \stackrel{\text{def}}{=} \overline{a_1} \dots \overline{a_n}.$$

Nous pouvons maintenant définir un système à canaux qui simule le calcul d'un transducteur :

Définition 3.1. *Étant donné un transducteur temps réel $T = (Q, \Sigma, q_0, F, \Delta)$, on lui associe le système à canaux $S_{T,marq}$ défini par :*

$$S_{T,marq} \stackrel{\text{def}}{=} (Q, \ddot{\Sigma}, \Delta_{marq})$$

où :

$$\Delta_{marq} \stackrel{\text{def}}{=} \{q \xrightarrow{?a\ !\overline{b}} q' \mid (q, a, q', b) \in \Delta\}.$$

Pour $w \in \Sigma^+$, un calcul fiable de $S_{T,marq}$ partant de $\langle q_0, w\rangle$ correspond précisement à un calcul du transducteur T sur w (cf. lemme 3.7). Nous verrons par la suite comment gérer le cas où le mot de départ contient aussi des lettres marquées.

Enfin, comme nous voulons calculer les itérés du transducteur, il nous reste à le faire boucler, en démarquant les lettres pour que le calcul puisse continuer :

Définition 3.2. *Étant donné un transducteur temps réel $T = (Q, \Sigma, q_0, F, \sigma)$, on lui associe le système à canaux S_T qui calcule les itérés de T, défini de la manière suivante :*

$$S_T \stackrel{\text{def}}{=} (Q \uplus \{q_{net}\}, \ddot{\Sigma}, \Delta_{marq} \cup \Delta_{net}),$$

où :

$$\begin{aligned}\Delta_{net} \stackrel{\text{def}}{=} &\{q \xrightarrow{?\overline{x}\ !x} q_{net} \mid q \in F, x \in \Sigma\}\\ &\cup \{q_{net} \xrightarrow{?\overline{x}\ !x} q_{net} \mid x \in \Sigma\}\\ &\cup \{q_{net} \xrightarrow{?\epsilon\ !\epsilon} q_0\}.\end{aligned}$$

Remarque 3.3. S_T est une extension de $S_{T,marq}$.

Cette construction est illustrée dans Fig. 1. L'indice *net* dénote "nettoyage".

On remarquera qu'un calcul de S_T ne peut pas boucler dans $S_{T,marq}$ puisque le nombre de lettres non marquées du canal diminue strictement à chaque transition, et qu'il ne peut pas boucler indéfiniment en q_{net} car le nombre de lettres marquées du canal diminue strictement à chaque transition. Le système obtenu n'introduit donc pas de "nouvelle boucle" par rapport au transducteur de départ.

Remarque 3.4. On observe que toutes les règles de transitions de S_T sont de la forme $q \xrightarrow{?a\ !b} q'$ ou $q \xrightarrow{?\varepsilon\ !\varepsilon} q'$. Le système à canaux obtenu conserve donc les longueurs, i.e. pour tous mots u et v de $\ddot{\Sigma}^*$ et tous états q et q' de $Q \uplus \{q_{net}\}$, on a :

$$\langle q, u\rangle \xrightarrow{+}_{\text{fbl}} \langle q', v\rangle \Rightarrow |u| = |v|.$$

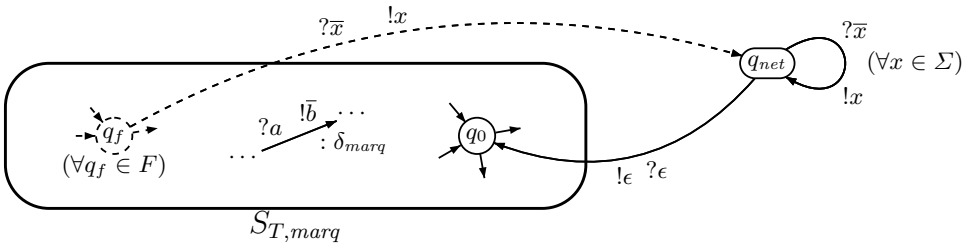

Fig. 1. S_T : un système à canaux calculant les itérés du transducteur T

Nous allons maintenant introduire une notion qui nous permet d'exprimer le fait que le calcul dans S_T fasse exactement "un tour", ce qui signifie un calcul d'une image par le transducteur qui reste donc dans $S_{T,marq}$ puis une phase de démarquage :

Définition 3.5. *Soient T un transducteur temps réel, S_T son système à canaux associé et u et v deux mots de $\ddot{\Sigma}^*$. On définit la relation $\triangleright$ de $\ddot{\Sigma}^* \times \ddot{\Sigma}^*$ de la manière suivante : $u \triangleright v$ ssi les deux propriétés suivantes sont vérifiées :*

–
$$u, v \in \Sigma\ddot{\Sigma}^* \cap \ddot{\Sigma}^*\Sigma, \tag{1}$$

– *il existe un calcul dans S_T de la forme :*

$$\langle r_0, u_0\rangle \rightarrow_{\text{fbl}} \langle r_1, u_1\rangle \rightarrow_{\text{fbl}} \dots \langle r_n, u_n\rangle$$

vérifiant $u_0 = u$, $u_n = v$ et $r_0 \dots r_n \in q_0 Q^* {q_{net}}^+ q_0$.

Expliquons la condition (1) : le fait d'imposer que u commence et se termine par une lettre non marquée permet d'éviter les calculs qui ne conserveraient par la même alternance de paquets de lettres marquées/non marquées. Le fait d'imposer que v commence et se termine également par une lettre non marquée permet d'éviter les calculs qui nettoieraient moins que le maximum possible en laissant des lettres marquées à la fin du canal, et donc ne conserveraient pas non plus la même alternance de paquets de lettres marquées/non marquées.

On notera que $u \triangleright v$ peut être vu à la fois comme le fait que u et v sont en relation et à la fois comme un run de $\langle q_0, u\rangle$ à $\langle q_0, v\rangle$. Nous utiliserons la même notation pour dénoter les deux notions.

Remarque 3.6. En particulier, si $u \in \Sigma^+$, on a le chemin suivant :

$$\underbrace{\langle q_0, u\rangle \xrightarrow{+}_{\text{fbl}} \langle q_f, \overline{v}\rangle}_{\text{dans } S_{T,marq}} \xrightarrow{+}_{\text{fbl}} \langle q_0, v\rangle$$

Dans la suite on notera $\triangleright^k$ la composée k-fois de $\triangleright$, $\triangleright^+$ la fermeture transitive de $\triangleright$ et $\triangleright^*$ la fermeture transitive réflexive de $\triangleright$.

3.2 Relation entre les transducteurs et les systèmes à canaux non fiables

Dans cette partie nous allons expliciter et prouver le lien que nous avons construit entre les systèmes à canaux et les transducteurs. On prend T un transducteur temps réel et $S_{T,marq}$, S_T et $\triangleright$ (lié à S_T) sont définis comme précédemment.

Tout d'abord, la sémantique opérationnelle de $S_{T,marq}$ et R_T sont liées par la relation suivante :

Lemme 3.7. *Pour tous mots u et v dans Σ^*,*

$$uR_Tv \text{ ssi } \exists q_f \in F, \langle q_0, u\rangle \xrightarrow{+}_{\text{fbl}} \langle q_f, \overline{v}\rangle \text{ dans } S_{T,marq}.$$

Démonstration. On notera tout d'abord que $|u| = |v|$ si uR_Tv (T étant temps réel), ainsi que si $\langle q_0, u\rangle \xrightarrow{+}_{\text{fbl}} \langle q_f, \overline{v}\rangle$ (Rem. 3.4). On peut donc poser $n = |u|$, $u = a_1 \dots a_n$ et $v = b_1 \dots b_n$.
($\Rightarrow$) : Si uR_Tv alors on a un run dans T de la forme :

$$\langle q_0, u, \varepsilon\rangle \rightarrow \dots \langle q_i, a_{i+1} \dots a_n, b_1 \dots b_i\rangle \rightarrow \dots \langle q_n, \varepsilon, v\rangle$$

avec $q_n \in F$. Par définition de Δ_{marq} on en déduit le run suivant de $S_{T,marq}$:

$$\langle q_0, u\rangle \rightarrow_{\text{fbl}} \dots \langle q_i, a_{i+1} \dots a_n\overline{b_1} \dots \overline{b_i}\rangle \rightarrow_{\text{fbl}} \dots \langle q_n, \overline{v}\rangle$$

d'où le résultat.
($\Leftarrow$) : Supposons que $S_{T,marq}$ admette un run de la forme $\langle q_0, u\rangle \xrightarrow{+}_{\text{fbl}} \langle q_f, \overline{v}\rangle$ avec q_f dans F. Comme chaque transition de $S_{T,marq}$ lit une lettre non marquée et écrit une lettre marquée, le calcul est de longueur n et est de la forme :

$$\langle q_0, u\rangle \rightarrow_{\text{fbl}} \dots \langle q_i, a_{i+1} \dots a_n\overline{b_1} \dots \overline{b_i}\rangle \rightarrow_{\text{fbl}} \dots \langle q_n, \overline{v}\rangle$$

avec $q_n = q_f$. On en déduit que les règles suivantes sont dans Δ_{marq}, pour $1 \leq i \leq n$: $q_{i-1} \xrightarrow{?a_i\ !\overline{b_i}} q_i$. D'où l'on déduit par définition de Δ_{marq} que T à des règles de transitions de la forme $q_{i-1} \xrightarrow{?a_i\ !b_i} q_i$ et donc que T admet le run suivant :

$$\langle q_0, u, \varepsilon\rangle \rightarrow \dots \langle q_i, a_{i+1} \dots a_n, b_1 \dots b_i\rangle \rightarrow \dots \langle q_n, \varepsilon, v\rangle$$

donc uR_Tv.

□

Maintenant nous montrons le lien qui existe entre R_T et la sémantique opérationnelle de S_T :

Lemme 3.8. *Soient u et v deux mots de Σ^+, on a :*

$$uR_Tv \text{ ssi } u \triangleright v.$$

Démonstration. ($\Rightarrow$) : Si uR_Tv, d'après Lem. 3.7 il existe $q_f \in F$ tel que $\langle q_0, u\rangle \xrightarrow{+}_{\text{fbl}} \langle q_f, \overline{v}\rangle$ dans $S_{T,marq}$. On en déduit donc le run suivant dans S_T, en posant $v = b_1 \dots b_n$:

$$\underbrace{\langle q_0, u\rangle \xrightarrow{+}_{\text{fbl}} \langle q_f, \overline{v}\rangle}_{\text{dans } S_{T,marq}} \rightarrow_{\text{fbl}} \underbrace{\langle q_{net}, \overline{b_2 \dots b_n} b_1\rangle \rightarrow_{\text{fbl}} \dots \langle q_{net}, v\rangle}_{\text{nettoyage}} \rightarrow_{\text{fbl}} \langle q_0, v\rangle$$

où les états visités sont dans $q_0 Q^* {q_{net}}^+ q_0$. On a donc $u \triangleright v$.
($\Leftarrow$) : Si $u \triangleright v$, alors on a un run dans S_T de la forme :

$$\underbrace{\langle q_0, u\rangle \xrightarrow{+}_{\text{fbl}} \langle r, u'\overline{w}\rangle}_{\text{dans } S_{T,marq}} \rightarrow_{\text{fbl}} \langle q_{net}, x\overline{y}z\rangle \xrightarrow{+}_{\text{fbl}} \langle q_{net}, v\rangle \rightarrow_{\text{fbl}} \langle q_0, v\rangle$$

où u' est un suffixe de u. Or les seules règles de transition de S_T qui vont en q_{net} depuis un état de Q partent d'un état final de T et lisent une lettre marquée. On en déduit donc que r appartient à F et que $u' = \varepsilon$.

De plus, puisque la partie nettoyage écrit v, elle doit lire $\overline{v}$, on en déduit donc que $\overline{w} = \overline{v}$.

Ce run contient donc le run suivant qui est dans $S_{T,marq}$:

$$\langle q_0, u\rangle \xrightarrow{+}_{\text{fbl}} \langle r, \overline{v}\rangle$$

avec $r \in F$. D'après Lem. 3.7 on en déduit uR_Tv.

□

Le lemme 3.8 ne s'applique qu'aux mots de Σ^+, or dans le cas général, le contenu du canal de S_T est dans $\ddot{\Sigma}^+$. Nous allons maintenant voir comment simplifier $\triangleright^*$ pour toujours se ramener au cas où les mots u et v sont dans Σ^+.

Lemme 3.9 (Forme des images par $\triangleright$). *Soient u et v deux mots de Σ, w un mot de $\Sigma\ddot{\Sigma}^*$ et z un mot de $\ddot{\Sigma}^*$. Si $u\overline{v}w \triangleright z$, alors il existe un mot u' de Σ^+ tel que $z = w\overline{u'}v$.*

Démonstration. Par définition de $\triangleright$ on a un calcul dans S_T de la forme :

$$\langle r_0, u_0\rangle \rightarrow_{\text{fbl}} \langle r_1, u_1\rangle \rightarrow_{\text{fbl}} \dots \langle r_n, u_n\rangle$$

vérifiant $u_0 = u\overline{v}w$, $u_n = z$ et $r_0 \dots r_n \in q_0 Q^* {q_{net}}^+ q_0$.

Avant de passer en q_{net} le calcul doit lire le maximum de lettres non marquées. En effet la seule manière d'aller en q_{net} depuis Q est de lire une lettre marquée. De plus comme z commence par une lettre non marquée (par définition de $\triangleright$) le calcul doit lire le plus de lettres marquées possible. Comme w commence par une lettre non marquée on a donc $z = w\overline{u'}v$ pour un certain mot u' de Σ^+.

□

Lemme 3.10 (Forme des images par $\triangleright^+$). *Soit u un mot de $\Sigma\ddot{\Sigma}^* \cap \ddot{\Sigma}^*\Sigma$ s'écrivant sous la forme :*

$$u = u_1\overline{u_2}u_3\overline{u_4}\dots\overline{u_{2n}}u_{2n+1} = \Big[\prod_{i=1}^{n}(u_{2i-1}\overline{u_{2i}})\Big]u_{2n+1}$$

où $u_1, \dots u_{2n+1}$ sont dans Σ^+. Alors tous les mots v tels que $u \triangleright^{2n+1} v$ sont de la forme :

$$v = v_1\overline{v_2}v_3\overline{v_4}\dots\overline{v_{2n}}v_{2n+1} = \Big[\prod_{i=1}^{n}(v_{2i-1}\overline{v_{2i}})\Big]v_{2n+1}$$

où $v_1, \dots v_{2n+1}$ sont dans Σ^+ et vérifient :

$$\forall 1 \leq i \leq 2n+1, u_i R_T v_i.$$

Démonstration. Pour bien comprendre ce qui se passe, nous faisons la preuve pour $n = 2$. Celle-ci s'adapte facilement dans le cas général.

D'après le lemme 3.9 on a :

$$\begin{aligned} u_1\overline{u_2}u_3\overline{u_4}u_5 &\triangleright u_3\overline{u_4}u_5\overline{v_1}u_2 \\ &\triangleright u_5\overline{v_1}u_2\overline{v_3}u_4 \\ &\triangleright u_2\overline{v_3}u_4\overline{v_5}v_1 \\ &\triangleright u_4\overline{v_5}v_1\overline{v_2}v_3 \\ &\triangleright v_1\overline{v_2}v_3\overline{v_4}v_5 \end{aligned}$$

Et d'après le lemme 3.7, on sait que pour tout i entre 1 et 5, on $u_i R_T v_i$. □

Lemme 3.11 (Forme des images par $\triangleright^+$ (2)). *Soit u un mot de $\Sigma\ddot{\Sigma}^* \cap \ddot{\Sigma}^*\Sigma$ s'écrivant sous la forme :*

$$u = u_1\overline{u_2}u_3\overline{u_4}\dots\overline{u_{2n}}u_{2n+1} = \Big[\prod_{i=1}^{n}(u_{2i-1}\overline{u_{2i}})\Big]u_{2n+1}$$

où $u_1, \dots u_{2n+1}$ sont dans Σ^+. Alors tous les mots v tels que $u \triangleright^{k(2n+1)} v$, pour k un entier naturel, sont de la forme :

$$v = v_1\overline{v_2}v_3\overline{v_4}\dots\overline{v_{2n}}v_{2n+1} = \Big[\prod_{i=1}^{n}(v_{2i-1}\overline{v_{2i}})\Big]v_{2n+1}$$

où $v_1, \dots v_{2n+1}$ sont dans Σ^+ et vérifient :

$$\forall 1 \leq i \leq 2n+1, u_i R_T^k v_i.$$

Démonstration. On montre le résultat par récurrence sur k.

<u>*cas $k = 0$*</u> Si $u \triangleright^0 v$, alors $v = u$ et pour tout $1 \leq i \leq 2n+1$ on a $u_i R_T^0 u_i$.

étape d'induction Supposons la propriété vraie pour un certain $k \in \mathbb{N}$.

Si $u \triangleright^{(k+1)(2n+1)} v$ alors on a pour un certain mot w : $u \triangleright^{k(2n+1)} w \triangleright^{2n+1} v$. Par supposition w s'écrit de la forme :

$$w = w_1\overline{w_2}w_3\overline{w_4}\dots\overline{w_{2n}}w_{2n+1} = \Big[\prod_{i=1}^{n}(w_{2i-1}\overline{w_{2i}})\Big]w_{2n+1}$$

où $w_1, \dots w_{2n+1}$ sont dans Σ^+ et vérifient :

$$\forall 1 \leq i \leq 2n+1, u_i R_T^k w_i.$$

En appliquant le lemme 3.10 à w et v on obtient que v est de la forme :

$$v = v_1\overline{v_2}v_3\overline{v_4}\dots\overline{v_{2n}}v_{2n+1} = \Big[\prod_{i=1}^{n}(v_{2i-1}\overline{v_{2i}})\Big]v_{2n+1}$$

où $v_1, \dots v_{2n+1}$ sont dans Σ^+ et vérifient :

$$\forall 1 \leq i \leq 2n+1, w_i R_T v_i.$$

On a donc :

$$\forall 1 \leq i \leq 2n+1, u_i R_T^{(k+1)} v_i.$$

D'où le résultat. □

4 Applications

4.1 Problèmes de terminaison

Dans cette partie nous allons nous intéresser à des problèmes de terminaison pour les systèmes à canaux non fiables.

Le principal résultat connu concernant la terminaison dans les systèmes à canaux non fiables est celui de la décidabilité de la terminaison simple [Fin94b], [MS02] :

Terminaison simple
Données : un système à canaux non fiable S et une configuration de départ σ_0,
Question : est-ce que tous les calculs non fiables dans S partant de σ_0 terminent ?

Dans le cas de la terminaison simple la configuration de départ est fixée. Nous allons nous intéresser à deux problèmes de terminaison uniforme :

Terminaison uniforme (UNI-TERM) :
Données : un système à canaux S et un état de contrôle q,
Question : est-ce que pour tous mots w, tous les calculs partant de $\langle q, w\rangle$ terminent ?

Terminaison uniforme 2 (UNI-TERM2) :
Donnée : un système à canaux non fiable S,
Question : est-ce que pour tout état q et tous mots w, tous les calculs partant de $\langle q, w\rangle$ terminent ?

Paradoxalement ces deux problème sont indécidables :

Théorème 4.1. UNI-TERM *et* UNI-TERM2 *sont indécidables.*

Pour montrer ce résultat nous allons partir d'un problème sur les itérés de transducteurs :

Point fixe d'un transducteur (TRANS-FIX) :
Donnée : un transducteur T,
Question : existe-t-il un mot non vide w et un entier $k > 0$ tel que $w(R_T)^k w$?

Ce problème est indécidable, même si l'on se restreint aux transducteurs temps réel déterministes. Ce résultat est une conséquence directe de l'indécidabilité du problème de pavage périodique du plan par un ensemble de tuiles de Wang [AD97], même si l'on se restreint aux ensembles "déterministes" de tuiles qui se comportent comme des transducteurs déterministes [MR99].

Lemme 4.2. *Soient T un transducteur et S_T son système à canaux non fiables associé. S_T est une instance positive de* UNI-TERM2 *ssi T est une instance négative de* TRANS-FIX.

Démonstration. ($\Rightarrow$) : On notera tout d'abord que comme S_T ne peut pas augmenter la taille du canal, si un run est infini, il devient parfait au bout d'un certain temps.

De plus, la seule manière d'avoir un run infini dans S_T est d'alterner infiniment entre q_{net} et Q. En effet, dans $S_{T,mark}$ le nombre de lettres non marquées du canal diminue strictement à chaque transition, et en restant dans l'état q_{net} le nombre de lettres marquées diminue strictement à chaque transition.

Le run admet donc un suffixe qui se décompose en une succession de runs de la forme :

$$\overbrace{\langle q_0, u\rangle \xrightarrow{+}_{\text{fbl}} \langle q_f, v\rangle}^{\text{dans } S_{T,mark}} \rightarrow_{\text{fbl}} \langle q_{net}, w\rangle \xrightarrow{+}_{\text{fbl}} \langle q_0, x\rangle.$$

u peut s'écrire sous la forme $u = \prod_{i=1}^{n}(u_{2i-1}\overline{u_{2i}})u_{2n+1}$, où u_2, ... u_{2n} sont des mots de Σ^+ et u_1 et u_{2n+1} sont des mots de Σ^*. On notera cependant que x admet la même décomposition mais avec toutes ses parties dans Σ^+, i.e. :

$$\begin{cases} x = \prod_{i=1}^{n}(x_{2i-1}\overline{x_{2i}})x_{2n+1} \\ \forall 1 \leq i \leq 2n+1, x_i \in \Sigma^+. \end{cases}$$

Le calcul admet donc un suffixe qui peut s'écrire :

$$u_0 \triangleright u_1 \triangleright u_2 \ldots$$

Considérons la suite $(u_{2kn})_{k\in\mathbb{N}}$. C'est une suite de mot de même longueur donc deux au moins sont égaux. Il existe donc deux entiers p et q tels que $u_{2pn} = u_{2qn}$, avec $p < q$.

Or $u_{2pn} \triangleright^{2(q-p)n} u_{2qn}$. D'après le lemme 3.11 on en déduit donc qu'il existe un préfixe v dans Σ^+ de u_{2pn} tel que $v \triangleright^{(q-p)} v$. D'après Lem. 3.8, on en déduit $vR_T^{(q-p)}v$.

($\Leftarrow$) : Si T est une instance positive de `Trans-Fix`, alors il existe un mot u de Σ^+ et un entier naturel $k > 0$ tel que $uR_T^k u$.

D'après le lemme 3.8 on en déduit le run infini suivant dans S_T :

$$u \triangleright^k u$$

qui se trouve d'ailleurs être un run fiable.

□

Ce qui nous montre que `Uni-Term2` est indécidable. La même démonstration vaut également pour `Uni-Term` en prenant comme état de contrôle $s = q_{init}$.

4.2 Problèmes de vivacité

Dans cette partie nous allons nous intéresser à l'indécidabilité de problèmes de vivacité dans les systèmes à canaux non fiables.

On considère les deux problèmes suivants :

Acceptance de Büchi (`Büchi`) :
Donnée : un système à canaux non fiables S, une configuration σ_0 et un état de contrôle s dans Q,
Question : existe-t-il un calcul partant de σ_0 qui visite s infiniment souvent ?

Boucle sur état de contrôle (`CSL`) :
Donnée : un système à canaux non fiables S, une configuration σ_0 et un état de contrôle s dans Q,
Question : y a-t-il un run non fiable de la forme :

$$\sigma_0 \xrightarrow{*} \langle s, u\rangle \xrightarrow{+} \langle s, u\rangle$$

pour un certain mot u dans S ?

`Büchi` et `CSL` coïncident : à partir d'un run visitant infiniment souvent s, on peut construire un run bouclant sur s.

Théorème 4.3. `CSL` *et* `Büchi` *sont indécidables pour les systèmes à canaux non fiables.*

Ce théorème est dû à Abdulla et Jonsson [AJ96a], mais la réduction utilisée dans la preuve est assez complexe. Nous en donnons ici une plus simple :

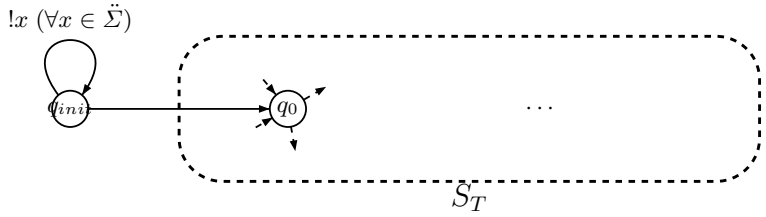

Fig. 2. représentation de S

Démonstration. Par réduction de TRANS-FIX à BÜCHI.

Soit T un transducteur, et S_T son système à canaux associé. On ajoute à S_T un état q_{init} qui met n'importe quel mot sur le canal puis saute en q_0. On note S le système obtenu, celui-ci est illustré en Fig. 2. On prend $\sigma_0 = \langle q_{init}, \varepsilon \rangle$ et $s = q_0$.

Si on a un run qui visite infiniment souvent q_0 en partant de σ_0, alors on a un run infini dans S_T, et d'après Lem. 4.2, T est une instance positive de TRANS-FIX.

Si il existe u dans Σ^+ et k dans $\mathbb{N} \setminus \{0\}$ tel que $uR_T^k u$, alors on en déduit le run infini suivant qui est un témoin pour BÜCHI :

$$\langle q_{init}, \varepsilon \rangle \xrightarrow{+}_{\text{fbl}} \langle q_{init}, u \rangle \rightarrow_{\text{fbl}} u \triangleright u \triangleright u \ldots$$

D'où le résultat.

□

5 Conclusion

Dans ce rapport nous avons exposé et exploité une analogie entre les transducteurs et les systèmes à canaux. Nous avons construit un système à canaux capable de calculer les itérés d'un transducteur (Fig. 1), et nous avons donné un lemme pour exhiber la relation entre le système construit et le transducteur (Lem. 3.11).

Nous avons ensuite exploité cette construction pour démontrer l'indécidabilité de plusieurs problèmes de terminaison et de vivacité sur les systèmes à canaux non fiables.

On conjecture que d'autres problèmes peuvent être prouvés indécidables en utilisant le même genre de réduction, tel que le suivant (déjà connu pour être indécidable) :

Infinitude

Données : un système à canaux non fiables S et une configuration de départ σ_0,

Question : existe-t-il un calcul partant de σ_0 qui visite une infinité de configurations différentes ?

Nous nous sommes également intéressé à de nouveaux problèmes d'accessibilité et avons prouvé que le problème suivant est indécidable :

Accessibilité universelle en un état
Données : un système à canaux non fiable S et une configuration de départ σ_0 et un état de contrôle s,
Question : est ce que pour tout mot w la configuration $\langle s, w\rangle$ est accessible en partant de σ_0 ?

Par contre nous n'avons pas encore de réponse pour le problème suivant :

Accessibilité universelle
Données : un système à canaux non fiable S et une configuration de départ σ_0
Question : est ce que pour tout état q et tout mot w la configuration $\langle q, w\rangle$ est accessible en partant de σ_0 ?

Références

AD97. C. Allauzen and B. Durand. Tiling problems. In E. Börger, E. Grädel, and Y. Gurevich, editors, *The Classical Decision Problem*, Perspectives in Mathematical Logic, pages 407–420. Springer, 1997.

AJ96a. P. A. Abdulla and B. Jonsson. Undecidable verification problems for programs with unreliable channels. *Information and Computation*, 130(1) :71–90, 1996.

AJ96b. P. A. Abdulla and B. Jonsson. Verifying programs with unreliable channels. *Information and Computation*, 127(2) :91–101, 1996.

CS08. Pierre Chambart and Ph. Schnoebelen. The ordinal recursive complexity of lossy channel systems. In *LICS*, pages 205–216. IEEE Computer Society, 2008.

Fin94a. A. Finkel. Decidability of the termination problem for completely specificied protocols. *Distributed Computing*, 7(3) :129–135, 1994.

Fin94b. Alain Finkel. Decidability of the termination problem for completely specified protocols. *Distributed Computing*, 7(3) :129–135, 1994.

May00. Richard Mayr. Undecidable problems in unreliable computations. In *Theoretical Computer Science*, pages 377–386. Springer, 2000.

MR99. J. Mazoyer and I. Rapaport. Global fixed point attractors of circular cellular automata and periodic tilings of the plane : Undecidability results. *Discrete Mathematics*, 199(1–3) :103–122, 1999.

MS02. Benoît Masson and Ph. Schnoebelen. On verifying fair lossy channel systems. In Krzysztof Diks and Wojciech Rytter, editors, *MFCS*, volume 2420 of *Lecture Notes in Computer Science*, pages 543–555. Springer, 2002.