International Journal of Foundations of Computer Science

Channel Synthesis for Finite Transducers*

Gilles Benattar[1], Beatrice Berard[2], Didier Lime[1],
John Mullins[3], Olivier H. Roux[1] and Mathieu Sassolas[2]

[1] *LUNAM Université, École Centrale de Nantes, IRCCyN, CNRS UMR 6597*
[2] *Université Pierre & Marie Curie, LIP6/MoVe, CNRS UMR 7606*
[3] *École Polytechnique de Montréal, Dept. of Computer & Software Eng.*

Email: gilles.benattar@clearsy.com, mathieu.sassolas@lip6.fr

Received (Day Month Year)
Accepted (Day Month Year)
Communicated by (xxxxxxxxxx)

We investigate how two agents can communicate through a noisy medium modeled as a nite non deterministic transducer. The sender and the receiver are also described by nite transducers which can respectively encode and decode binary messages. When the communication is reliable, we call the encoder/decoder pair a channel.

We study the *channel synthesis problem* which, given a transducer, asks whether or not such sender and receiver exist and builds them if the answer is positive. To that e ect we introduce the structural notion of *encoding state* in a transducer which is a necessary condition for the existence of a channel. It is not, however, a su cient condition. In fact, we prove that the problem is undecidable. Nonetheless, we obtain a synthesis procedure when the transducer is functional. We discuss these results in relation to security properties.

Keywords: Synthesis, transducers, covert communication.

1. Introduction

Given an architecture de ned by processes and communication links between them or with the environment, and a speci cation on the messages transmitted over these links, distributed synthesis aims at deciding the existence of local programs, one for each process, that together meet the speci cation, whatever the environment does. In the case of synchronous communication, the problem was proved decidable (but non-elementary) for LTL properties over pipeline architectures [14, 10], or more generally [4], when the processes are sorted in a *linear preorder* with respect to the information received from the environment. In the asynchronous setting, this problem is undecidable for total LTL speci cations [16] as soon as there are two processes.

Extended version of [2] presented at the 13th International Conference on Automata and Formal Languages (AFL'11).

In view of these negative results, we investigate another setting with only two processes (sender and receiver), modeled by nite transducers, that respectively encode and decode nite binary messages. They are organized in a pipeline architecture and communicate asynchronously through a medium that acts as noise over the link between them and also described by a xed non deterministic nite transducer. Moreover, we consider a particular basic external speci cation expressing faithful communication over nite binary words: the message received is equal to the message emitted, possibly modulo some slight modi cations or delays. The synthesis problem then asks if, given the noisy process, the encoder and decoder can be synthesized. We call such an encoder/decoder pair a *reliable channel* (or *channel* for short), and thus call this problem the *channel synthesis problem*.

We rst establish properties of such channels as well as veri cation results (Section 3). In particular, we give in Section 4 a necessary condition for the existence of a channel. We then prove (Section 5) that the channel synthesis problem is undecidable. However, we exhibit a restricted case where the problem can be decided: when the noisy process is a functional transducer (Section 6). We nally discuss the possible relations of these results with security properties in Section 7.

2. Preliminaries

Notations. The set of natural numbers is denoted by $\mathbb{N}$ and the set of *words* over a nite alphabet A is denoted by A^*, with ε for the empty word; $A^+ = A^* \setminus \{\varepsilon\}$ is the set of non-empty words over A. The length of a word u is written $|u|$ and for $1 \leq i \leq |u|$, the ith letter of u is denoted by $u[i]$. For a subset $B \subseteq A$, $|u|_B$ is the number of letters from B in u. A *language* is a subset of A^*. For two words u and v with same length n, the *distance* between u and v is the number of letters that are di erent in u and v: $d(u,v) = |\{i \mid 1 \leq i \leq n,\ u[i] \neq v[i]\}|$, where $|X|$ is the size of the nite set X.

For two words u and v, we write $v \preccurlyeq u$ when v is a *prefix* of u: there is some word w such that $u = vw$. For $k \in \mathbb{N}$, the set of *k-bounded prefixes* of u contains the pre xes v of u whose length di ers from the length of u by at most k letters: $Pref_k(u) = \{v \in A^* \mid v \preccurlyeq u \text{ and } |u| - |v| \leq k\}$.

A word w is a *factor* of u if there are some words u_1 and u_2 such that $u = u_1 w u_2$. For a word v with same length as u, the *corresponding factor* of v is the word w' such that $u = u_1 w u_2$, $v = v_1 w' v_2$, $|u_1| = |v_1|$ and $|w| = |w'|$. For two natural numbers m, p with $p \leq m \leq |u|$, the word v is *(m,p)-close* to u if $|v| = |u|$ and for any factor w of u of length $|w| = m$, the distance between w and the corresponding factor w' of v is less than or equal to p. The set of *(m,p)-substitutions* of u is de ned by: $Sub_p^m(u) = \{v \in A^* \mid v \text{ is } (m,p)\text{-close to } u\}$.

Both notations are extended in a natural way to a language $L \subseteq A^*$ as $Pref_k(L) = \bigcup_{u \in L} Pref_k(u)$ and $Sub_p^m(L) = \bigcup_{u \in L} Sub_p^m(u)$.

Codes of cardinality 2. Recall that a subset X of A^* is a code if any word in X^* admits a unique decomposition over X. We use the following properties ([12, 7]).

Proposition 1. *The three following conditions are equivalent for two words u and v over alphabet A:*
(i) the set $\{u, v\}$ is not a code,
(ii) u and v commute: $uv = vu$,
(iii) there is a word w in A^* and p, q in $\mathbb{N}$ such that $u = w^p$ and $v = w^q$.

Moreover, for a non empty word $u \in A^+$, let $Com(u) = \{v \in A^* \mid uv = vu\}$ be the set of words which commute with u. Then there exists $w \in A^*$ such that $Com(u) = w^*$.

Finite automata. A *finite automaton*, or *automaton* for short, is a tuple $A = \langle S, I, Lab, \ldots, F\rangle$, where S is a finite set of states, I is a subset of S of initial states, Lab is a finite set of labels, $\ldots \subseteq S \times Lab \times S$ is a finite transition relation and $F \subseteq S$ is a set of final states. We allow Lab to be an alphabet but also a finite subset of a monoid, although here only direct products of word monoids are considered. A *run from $s \in S$* is an alternating sequence of states and letters written as $\rho = s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} s_n$, such that $s_0 = s$ and $(s_i, a_{i+1}, s_{i+1}) \in \ldots$ for $0 \leq i \leq n-1$. The *trace* of ρ is $trace(\rho) = a_1 \cdots a_n$. We write $s \stackrel{u}{\Rightarrow} s'$ if there is a run ρ from s to s' with trace u. A run ρ as above is *accepting* if $s \in I$ and $s_n \in F$, and the *language* of A, denoted by $L(A)$, is the set of traces of accepting runs. A state $s \in S$ is *useful* if it belongs to some accepting run. Since the accepted language is the same when removing non useful states, we assume in the sequel that the set S contains only useful states. A regular language over an alphabet A is a subset of A^* accepted by a finite automaton with set of labels $Lab = A$.

Finite Transducers. A *finite transducer* (or *transducer* for short) is a finite automaton A with a finite set of labels $Lab \subseteq A^* \times B^*$ for two alphabets A and B. A label $(u, v) \in A^* \times B^*$ is often written as $u|v$ in the figures (see transducer example in Figure 2). A subset of $A^* \times B^*$ is a *rational relation* from A^* to B^* if it is the language $L(A)$ of a finite transducer A [15]. The transducer A is said to realize the relation $L(A)$.

For a rational relation M, we denote by $A_{\mathcal{M}}$ a transducer which realizes M. For a word $u \in A^*$, we write $M(u) = \{v \in B^* \mid (u, v) \in M\}$ for the image of u, $M^{-1}(v) = \{u \in A^* \mid (u, v) \in M\}$ for the inverse image of v, possibly extended to subsets of A^* or B^* respectively, $Dom(M) = \{u \in A^* \mid \exists v \in B^*, (u, v) \in M\}$ for the domain of M and $Im(M) = \{v \in B^* \mid \exists u \in A^*, (u, v) \in M\}$ for the image of M. When $M(u)$ is a singleton, it will also denote its only element, with a slight misuse of notation. If the domain of M is A^*, then M is said to be *complete*. The transducer is *functional* if it realizes a partial function: for each word $u \in A^*$, there is at most one word in $M(u)$.

For a subset P of A^*, the identity relation $\{(w, w) \mid w \in P\}$ on $A^* \times A^*$ is denoted by $Id(P)$ and $Id_k(P)$ is the relation between words and their k-bounded prefixes in P: $Id_k(P) = \{(u, v) \in P \times P \mid v \in Pref_k(u)\}$. Note that $Id_0 = Id$.

The composition of rational relations M on $A^* \times B^*$ and M' on $B^* \times C^*$,

denoted by $M \quad M'$, is a rational relation on $A^* \quad C^*$ [3]. Moreover, the image and inverse image of a regular language by a rational relation is a regular language [15].

3. Channels

We consider communication between two processes, respectively called an *encoder* and a *decoder*. The encoder E reads binary input and produces an output in A^*, while the decoder D reads words in B^* and produces a binary word. These two processes communicate through a noisy medium, modeled by a non deterministic transducer with labels in $A^* \quad B^*$ (see Figure 1). The de nition below states that a channel corresponds to reliable communication: the binary message is correctly transmitted. An example of such communication is given in Section 7, particularly in Figure 8.

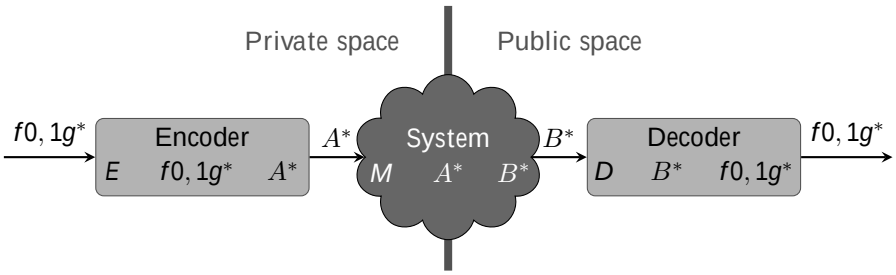


Figure 1. Implementation of a channel by transducers.

De nition 2. *Let $M \quad A^* \quad B^*$ be a rational relation. A* channel *for M is a pair $C = (E,D)$ such that E and D are rational relations in $f0,1g^* \quad A^*$ and $B^* \quad f0,1g^*$ respectively, and $E \quad M \quad D = Id(f0,1g^*)$.*

We rst prove that, given three transducers realizing rational relations M, E and D, veri cation is decidable:

Proposition 3. *Let M be a rational relation in $A^* \quad B^*$ and let E and D be two rational relations on $f0,1g^* \quad A^*$ and $B^* \quad f0,1g^*$, respectively. Given the associated transducers $A_{\mathcal{M}}$, $A_{\mathcal{E}}$, $A_{\mathcal{D}}$, it can be decided whether or not (E,D) is a channel for M.*

Proof. It can be decided whether or not a transducer is functional. Moreover, the equivalence of languages is decidable for rational relations [1]. If the transducer for relation $E \quad M \quad D$ is not functional, there is no channel for M. If the transducer is functional, it can be decided whether $E \quad M \quad D$ is equal to $Id(f0,1g^*)$, because the identity relation is the only channel for M.

□

Equality is a strong requirement regarding communication, which corresponds to an external LTL speci cation (for nite words): \the output is always equal to the input". We investigate the impact of slightly weakening this condition by admitting a bounded amount of either communication delays or substitutions. Namely, a certain number of substitutions are allowed in a sliding window of xed size.

De nition 4. *Let* M *be a rational relation in* $A^* \quad B^*$ *and let* k, m, p *be three natural numbers with* $p \quad m$*. The pair* (E, D)*, where* E *and* D *are rational relations in* $f0, 1g^* \quad A^*$ *and* $B^* \quad f0, 1g^*$ *respectively, is a channel for* M

with delay k *if* $Id(f0, 1g^*) \quad E \quad M \quad D \quad Id_k(f0, 1g^*)$*;*
with (m, p)*-substitutions if* $Id(f0, 1g^*) \quad E \quad M \quad D \quad Sub_p^m(f0, 1g^*)$.

With these de nitions, we can prove the following:

Proposition 5. *Let* M *be a rational relation.*

If there is a channel with delay k *for* M*, then there is a channel for* M*.*
If there is a channel with (m, p)*-substitutions for* M*, such that* $m > 2p$*, then there is a channel for* M*.*

Proof. For the rst point, since $Id_0 = Id$, assume there is a channel of delay $k > 0$ for M. The result is obtained by modifying the encoder so that it transmits $k + 1$ bits together, while the decoder keeps only the rst bit. Let rep_k be the morphism from $f0, 1g^*$ to $f0, 1g^*$ de ned by $rep_k(b) = b^k$ for $b \; 2 \; f0, 1g^*$. Seen as relation, rep_k is rational and it is easy to see that if $Id(f0, 1g^*) \quad E \quad M \quad D \quad Id_k(f0, 1g^*)$ then $rep_{k+1} \quad E \quad M \quad D \quad unrep_{k+1} = Id(f0, 1g^*)$. Where $unrep_k$ is de ned as follows:

$$\text{For } x \; 2 \; f0, 1g, \begin{cases} unrep_k(x^k \quad w) = x \quad unrep_k(w) \\ unrep_k(x^j) = x, \text{ for } j \quad k \end{cases}$$

For the second point, we also modify the encoder to transmit m bits together, while the decoder will choose the bit with majority. Let $maj_m : f0, 1g^m \; ! \; f0, 1g$ be the function that associates with a sequence w of length m, the bit b such that $jwj_{\{b\}} > m/2$. The relation de ned by $vote_m = f(w, maj_m(w)) \; j \; w \; 2 \; f0, 1g^m g^*$ is a rational relation. If $Id(f0, 1g^*) \quad E \quad M \quad D \quad Sub_p^m(f0, 1g^*)$ then $rep_m \quad E \quad M \quad D \quad vote_m = Id(f0, 1g^*)$. □

The two operations can be combined for $m > 2(p + k)$, treating delays as errors.

Note that the decidability result for channel veri cation cannot be easily extended to these two types of channels, since inclusion of the identity relation in a rational relation is undecidable [6].

In the rest of the paper, we address the channel synthesis problem: \given a nite transducer realizing a rational relation M, is there a pair $C = (E, D)$ of rational relations such that C is a channel for M?". Sections 5 and 6 are respectively devoted to the proofs of the following results:

Theorem 6. *The channel synthesis problem is* Π^0_1*-complete.*

Theorem 7. *The channel synthesis problem is decidable in polynomial time for a functional relation* M*. Moreover if there is a channel for* M*, the transducers realizing* E *and* D *can be built.*

The proofs of Theorems 6 and 7 partly rely on a structural necessary condition for the existence of a channel, which is established in the following section.

4. Encoding states and canonical channels

The condition is based on the notion of encoding state: such a state admits two cycles such that the respective sets of labels over alphabets A and B form codes.

Definition 8. *Let* $A = \langle S, I, A^* \times B^*, \ldots, F \rangle$ *be a transducer. An* encoding state *is a useful state* $s \in S$ *such that there exist words* $u_0, u_1 \in A^*$ *and* $v_0, v_1 \in B^*$ *such that:*

(i) $s \xrightarrow{u_0|v_0} s$ *and* $s \xrightarrow{u_1|v_1} s$*;*
(ii) The two sets $\{u_0, u_1\}$ *and* $\{v_0, v_1\}$ *are codes on* A^* *and* B^* *respectively.*

This section is devoted to the proof of the following result:

Theorem 9. *Let* M *be a rational relation in* $A^* \times B^*$*. If* M *has a channel then the corresponding transducer* $A_{\mathcal{M}}$ *has an encoding state.*

First note that the condition is not sufficient as shown by the example of system N of Figure 2. States s_3 and s_4 are encoding states, but an input u can lead to s_3, which simulates s_4, but where no word can be encoded. In this case, the non-functionality of N breaks the locality of the encoding state property.

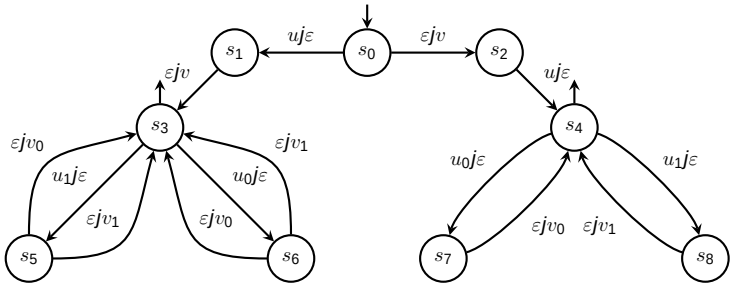


Figure 2. Transducer $\mathcal{N}$ with encoding states but no channel.

To prove Theorem 9, we first establish that the existence of a channel implies the existence of a channel under some \canonical" form, which in turn yields an

encoding state in the corresponding transducer. Both proofs rely on the following combinatorial lemma. Informally, it states that, starting from a code and an automaton, a new code can be built, related to a particular encoding node.

Lemma 10. *Let $A = \langle S, I, Lab, \ldots, F\rangle$ be a finite automaton, where Lab is a subset of a monoid with simplification (i.e., such that $xy = xy'$ implies $y = y'$ and $xy = x'y$ implies $x = x'$ for all $x, x', y, y' \in Lab$) and let $x, x_0, x_1, x' \in Lab$, such that $x \ldots \{x_0, x_1\}^* \ldots x' \ldots L(A)$ and $\{x_0, x_1\}$ is a code.*
Then, there exist states $s_0 \in I$, $s \in S$ and $s_f \in F$, and words $y \in x \ldots \{x_0, x_1\}^*$, $y_0 \in \{x_0\}^*$, $y_1 \in \{x_0, x_1\}^* \ldots x_1 \ldots \{x_0, x_1\}^*$, $y' \in \{x_0, x_1\}^* \ldots x'$ such that $\{y_0, y_1\}$ is a code and there are runs $s_0 \xRightarrow{y} s$, $s \xRightarrow{y_0} s$, $s \xRightarrow{y_1} s$ and $s \xRightarrow{y'} s_f$ in A.

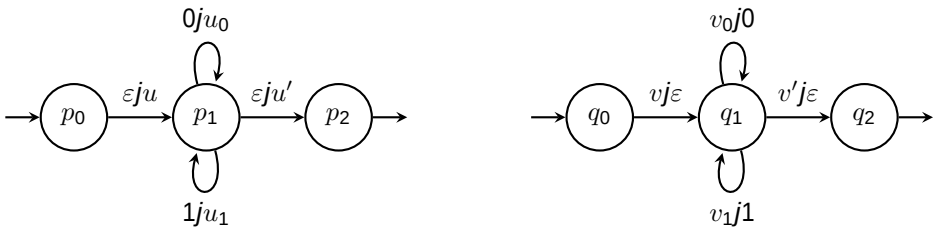
Proof. Let $p = |S|$ be the number of states in A. A simple cycle in A is a run $s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} \cdots \xrightarrow{a_n} s_1$ where all states in $\{s_1, \ldots, s_n\}$ are distinct and $n \geq 1$. There are at most $(p+1)!|I|\,|F|^p$ such simple cycles in A. For any $m > p$, any run in A labeled by x_0^m contains at least a simple cycle with trace in x_0^+, called an x_0-cycle in the sequel. Let k be the number of x_0-cycles in A, then $k \leq (p+1)!|I|\,|F|^p$.
Now we consider $u = x_0^{p+1} \ldots x_1$ and $v = x \ldots u^{k+1}x'$. Since $x \ldots \{x_0, x_1\}^* \ldots x' \ldots L(A)$, v belongs to $L(A)$. Let ρ be an accepting run in A for v, written as $s_0 \xRightarrow{x} s_0' \xRightarrow{x_0^{p+1}} s_1 \xrightarrow{x_1} s_1' \xRightarrow{x_0^{p+1}} s_{k+1} \xrightarrow{x_1} s_{k+1}' \xrightarrow{x'} s_f$, with $s_f \in F$.
For each i, $1 \leq i \leq k$, the subrun ρ_i of ρ defined by $s_i' \xRightarrow{x_0^{p+1}} s_{i+1}$ contains an x_0-cycle, hence there are $k+1$ such cycles, and two of them must be identical, say $s \xRightarrow{x_0^r} s$, with $r \geq 1$, within distinct subruns ρ_i and ρ_j. Then the run ρ can also be written as $s_0 \xRightarrow{y} s \xRightarrow{y_0} s \xRightarrow{y_1} s \xRightarrow{y_0} s \xRightarrow{y'} s_f$, with $y_0 = x_0^r$ and y_1 contains x_1 at least once. Since Lab is a subset of a monoid with simplification, the two words y_0 and y_1 do not commute, which yields the conclusions of the lemma. □

We now turn to canonical forms for channels. Let $U = (u, u_0, u_1, u')$ and $V = (v, v_0, v_1, v')$ be tuples of words in A^* and B^* respectively. The rational relations $E(U)$ and $D(V)$ on $\{0,1\}^* \times A^*$ and $B^* \times \{0,1\}^*$ respectively are defined by:

$$E(U) = (\varepsilon, u)\,\{(0, u_0), (1, u_1)\}^*\,(\varepsilon, u'), \quad D(V) = (v, \varepsilon)\,\{(v_0, 0), (v_1, 1)\}^*\,(v', \varepsilon),$$

which correspond to the transducers in Figure 3.
When $\{u_0, u_1\}$ (respectively $\{v_0, v_1\}$) is a code, we call relation $E(U)$ (respectively $D(V)$) the canonical encoder (respectively decoder) associated with U (respectively V). A channel of the form $(E(U), D(V))$ for a rational relation is denoted by $C(U, V)$ and called a *canonical channel*.

Definition 11. *Let $M \subseteq A^* \times B^*$ be a rational relation. A* canonical channel *for M is a channel of the form $C(U, V) = (E(U), D(V))$ for a pair of canonical encoder and decoder associated with tuples $U = (u, u_0, u_1, u')$ and $V = (v, v_0, v_1, v')$ in A^* and B^* respectively.*



(a) Transducer $\mathcal{E}(u, u_0, u_1, u^0)$ (b) Transducer $\mathcal{D}(v, v_0, v_1, v^0)$

Figure 3. Canonical encoder and decoder.

The next lemma shows that canonical encoders can be composed.

Lemma 12. *Let $E(X)$ and $E(U)$ be canonical encoders for tuples of words $X = (x, x_0, x_1, x')$ over $f0, 1g$ and $U = (u, u_0, u_1, u')$ over A.*
Then, there exists a tuple $W = (w, w_0, w_1, w')$ of words over A such that $E(W)$ is a canonical encoder and $E(W) = E(X)\ \ E(U)$.

Proof. We set $k = jxj$, $n = jx_0j$, $m = jx_1j$ and $\ell = jx'j$ and we identify elements of $f0, 1g$ with indices. The result is easily obtained for:
$w = u\ \ u_{x[1]} \dots u_{x[k]}$, $w_0 = u_{x_0[1]} \dots u_{x_0[n]}$, $w_1 = u_{x_1[1]} \dots u_{x_1[m]}$
and $w' = u_{x'[1]} \dots u_{x'[\ell]}\ \ u'$. □

The transformation from a channel to a canonical channel is then obtained by:

Proposition 13. *If a rational relation M in $A^*\ \ B^*$ has a channel, then there exist tuples of words $U = (u, u_0, u_1, u')$ in A^* and $V = (v, v_0, v_1, v')$ in B^*, such that $C(U, V)$ is a canonical channel for M.*

Proof. Assume the pair $C = (E, D)$ of rational relations is a channel for M. We proceed by successive transformations.

Transforming the encoding over $f0, 1g$. First, without loss of generality, we can assume that the image of E is contained in the domain of $M\ \ D$. Otherwise, the desired property is obtained by considering the relation $E' = E\ \ Id(Dom(M\ \ D))$. Let $A_{\mathcal{E}} = hS, I, Lab,\ \ , Fi$ be the transducer realizing E, where Lab is a nite subset of $f0, 1g^*\ \ A^*$. We consider the nite automaton $A^{in}_{\mathcal{E}}$ obtained by projection of $A_{\mathcal{E}}$ on its input: the structure is the same and a transition $s \overset{x|u}{\to} s'$ in $A_{\mathcal{E}}$ becomes $s \overset{x}{\to} s'$ in $A^{in}_{\mathcal{E}}$. From the hypothesis on E, the automaton $A^{in}_{\mathcal{E}}$ accepts all words over $f0, 1g$. Therefore, applying lemma 10 (with $x_0 = 0$ and $x_1 = 1$) yields states $s_0\ 2\ I$, $s\ 2\ S$ and $s_f\ 2\ F$, and a tuple of words $Y = (y, y_0, y_1, y')$ over $f0, 1g$ such that fy_0, y_1g is a code and $A^{in}_{\mathcal{E}}$ contains runs $s_0 \overset{y}{\Rightarrow} s$, $s \overset{y_0}{\Rightarrow} s$, $s \overset{y_1}{\Rightarrow} s$ and $s \overset{y'}{\Rightarrow} s_f$. In turn, this provides a tuple $W = (w, w_0, w_1, w')$ of words over A in the corresponding runs of $A_{\mathcal{E}}$: $s_0 \overset{y|w}{\Longrightarrow} s$, $s \overset{y_0|w_0}{\Longrightarrow} s$, $s \overset{y_1|w_1}{\Longrightarrow} s$ and $s \overset{y'|w'}{\Longrightarrow} s_f$. The fact

that $\{w_0, w_1\}$ is a code results from the corresponding property for $\{y_0, y_1\}$ and $E \ M \ D = Id(\{0,1\}^*)$. In addition, remark that $E(W) \ E(Y) \ E$.

Transforming encoder E into a canonical encoder. The next step consists in replacing E by an encoder using the tuple $W = (w, w_0, w_1, w')$ above (and modifying D accordingly). This is done by composition with the canonical encoder/decoder pair $C(Y, Y)$ in $\{0,1\}^* \ \{0,1\}^*$ for the tuple Y above, for which we clearly have:

$$(E(Y) \ E) \ M \ (D \ D(Y)) = Id(\{0,1\}^*).$$

Defining $E_{in} = E(W)$ and $D_{out} = D \ D(Y)$, it can be easily seen that the pair (E_{in}, D_{out}) is also a channel for M.

Transforming decoder D into a canonical decoder. We still need to transform D_{out} into a canonical decoder, by restricting its behavior to the core of the decoding relation. For this, we consider the transducer A_{enc} realizing the relation $M_{enc} = E_{in} \ M \ Id(Dom(D_{out}))$. Since $M_{enc} \ D_{out} = Id(\{0,1\}^*)$ and M_{enc} is complete over $\{0,1\}^*$, Lemma 10 can be applied again to the projection A^{in}_{enc} of A_{enc} on its input. Retrieving the corresponding runs, we obtain an initial state r_0, a state r and a final state r_f, and tuples $Z = (z, z_0, z_1, z')$ in $\{0,1\}^*$ and $V = (v, v_0, v_1, v')$ in B^* of words with runs $r_0 \xrightarrow{z|v} r$, $r \xrightarrow{z_0|v_0} r$, $r \xrightarrow{z_1|v_1} r$ and $r \xrightarrow{z'|v'} r_f$ in A_{enc}, such that $\{z_0, z_1\}$ and $\{v_0, v_1\}$ are codes.
The canonical encoder $E(Z)$ and decoder $D(V)$ satisfy:

$$E(Z) \ M_{enc} \ D(V) = Id(\{0,1\}^*) \text{ and } Id(Dom(D_{out})) \ D(V) = D(V).$$

Finally, composing the canonical encoders $E(Z)$ and $E_{in} = E(W)$ according to Lemma 12 yields a canonical encoder $E(U) = E(Z) \ E(W)$ for some tuple $U = (u, u_0, u_1, u')$ of words in A^*, and $C(U, V)$ is the required canonical channel for M, which concludes the proof. □

We end this section with the proof of Theorem 9:

Proof of Theorem 9. From Proposition 13, it suffices to prove that if a rational relation M has a canonical channel $C(U, V) = (E(U), D(V))$ for tuples of words $U = (u, u_0, u_1, u')$ in A^* and $V = (v, v_0, v_1, v')$ in B^*, then the corresponding transducer $A_{\mathcal{M}}$ has an encoding state.
The relation $E(U)$ (respectively $D(V)$) is a bijection from $\{0,1\}^*$ onto $u \ \{u_0, u_1\}^* \ u'$ (respectively from $v \ \{v_0, v_1\}^* \ v'$ onto $\{0,1\}^*$). Since $E(U) \ M \ D(V) = Id(\{0,1\}^*)$, the relation M contains $M_0 = (u, v) \ \{(u_0, v_0), (u_1, v_1)\}^* \ (u', v')$, hence all words in M_0 are accepted by $A_{\mathcal{M}}$. Moreover, the set $\{(u_0, v_0), (u_1, v_1)\}$ is a code in $A^* \ B^*$, because its components are themselves codes. Applying once again Lemma 10 to $A_{\mathcal{M}}$, we obtain $w_0 \in (u_0, v_0)^*$ and $w_1 \in \{(u_0, v_0), (u_1, v_1)\}^* \ (u_1, v_1) \ \{(u_0, v_0), (u_1, v_1)\}^*$, and a state s of $A_{\mathcal{M}}$, such that $s \xrightarrow{w_0} s$ and $s \xrightarrow{w_1} s$ are runs in $A_{\mathcal{M}}$. Since $\{w_0, w_1\}$ is a code, we can conclude that s is the required encoding state. □

5. Channel synthesis is undecidable

This section is devoted to the proof of Theorem 6. Proposition 3 states that the channel synthesis problem is in Δ_1^0. The proof of Δ_1^0-hardness is done by a reduction from Post's Correspondence Problem (PCP). Recall that an instance of PCP is a tuple $I = \langle (x_1, y_1), \ldots, (x_n, y_n) \rangle$ of pairs of words over an alphabet A. A (non trivial) solution is a non empty sequence of indices $i_1, \ldots, i_k$ such that $x_{i_1} \cdots x_{i_k} = y_{i_1} \cdots y_{i_k}$. With alphabet $N = \{1, \ldots, n\}$ of indices, an instance I can also be seen as a pair of morphisms x and y from N^* into A^* defined respectively by $x(i) = x_i$ and $y(i) = y_i$ for each $i \in N$. Hence a solution is a non empty sequence $\sigma \in N^+$ such that $x(\sigma) = y(\sigma)$. The problem of the existence of a solution is undecidable.

Starting from an instance I of PCP, we build a rational relation $M_{\mathcal{I}}$ such that I has a solution if and only if $M_{\mathcal{I}}$ has a channel.

The construction extends the undecidability proof for transducer equality [5] with an additional construction to obtain the channel property. The main idea is the following: given an input sequence $b\sigma$ such that b is a bit and σ is a solution of I, $M_{\mathcal{I}}$ will output anything except $x(\sigma) = y(\sigma)$ followed by $\overline{b}$, the complement of the input bit. Detecting this "missing word" makes the deduction of the input bit possible, and also the transmission of a message.

Construction. In addition to alphabets A and N above, we consider $B = \{>, ?\}$, which represents the bits, for the sake of readability. Hence, for $b \in B$, $\overline{b}$ is defined by $\overline{>} = ?$ and $\overline{?} = >$.

Relation $M_{\mathcal{I}}$ has input alphabet $N_B = N \cup B$ and output alphabet $A_B = A \cup B$. It is defined for $b \in B$ and $\sigma \in N^*$ by:

$$M_{\mathcal{I}}(b \otimes \sigma) = \left(A^+ \otimes b\right) \cup \left((A^+ \cap \{x(\sigma)\}) \otimes \overline{b}\right) \cup \left((A^+ \cap \{y(\sigma)\}) \otimes \overline{b}\right)$$

and extended to N_B^* by:

$$M_{\mathcal{I}}(v) = \begin{cases} \varepsilon \text{ if } v = \varepsilon, \\ M_{\mathcal{I}}(b_1 \otimes \sigma_1) \cdots M_{\mathcal{I}}(b_p \otimes \sigma_p) \text{ if } v = b_1 \otimes \sigma_1 \cdots b_p \otimes \sigma_p \\ \qquad \text{with } b_1, \ldots, b_p \in B,\ \sigma_1, \ldots, \sigma_p \in N^*, \\ \emptyset \text{ if } v \notin (B \otimes N^*)^*. \end{cases}$$

Now we can prove:

Lemma 14. *The relation $M_{\mathcal{I}}$ is a rational relation.*

Proof. The finite transducer $A_{\mathcal{I}} = \langle Q, q_0, N_B^* \times A_B^*, \Delta, \{q_0\} \rangle$ which realizes $M_{\mathcal{I}}$ is composed of two symmetrical parts, linked by the (initial and final) state q_0, that keep in memory one bit b of information (see Figure 4 for the global structure of $A_{\mathcal{I}}$). These two parts are called respectively the $>$-half and $?$-half of $A_{\mathcal{I}}$, the structure within the $>$-half being depicted in Figure 5.

In state q_*, the sequence of indices in the input is ignored and an arbitrary non empty word over A is produced, together with the input bit. Thus, this state is used to generate the language $A^+ \otimes b$ from input $b \otimes \sigma$.

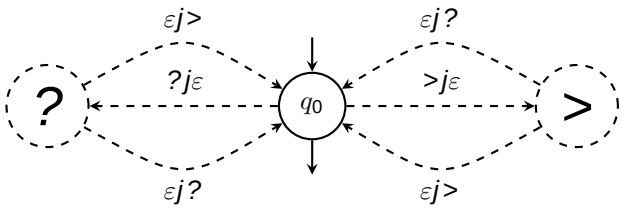


Figure 4. Symmetrical structure of $\mathcal{A}_I$.

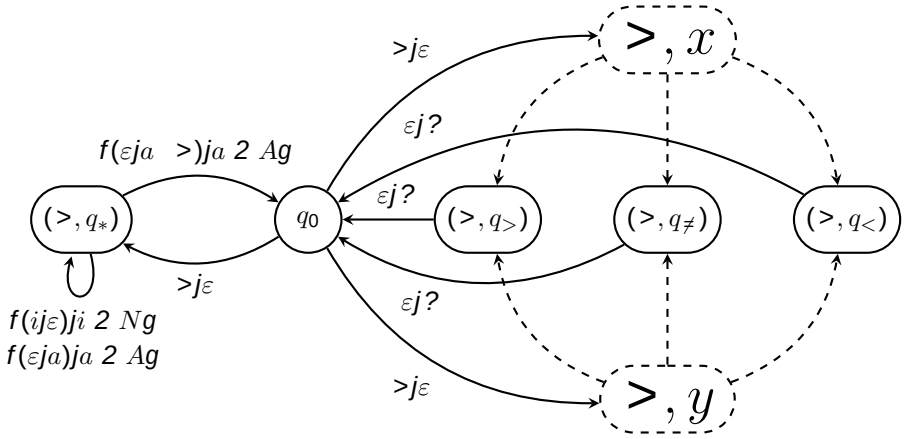


Figure 5. Structure of the $\top$-half of $\mathcal{A}_I$.

The rest of this >-half is divided into two similar parts called the $(>, x)$-quarter and the $(>, y)$-quarter respectively. The $(>, x)$-quarter produces any word which is not $x(\sigma)$, followed by $\bar{b}$, corresponding to the language $(A^+ \cap \{x(\sigma)\}) \cdot \bar{b}$ (and similarly for the $(>, y)$-quarter with language $(A^+ \cap \{y(\sigma)\}) \cdot \bar{b}$). For $z \in \{x, y\}$, avoiding $z(\sigma)$ in the $(>, z)$-quarter is achieved by:

either outputting a strict prefix of $z(\sigma)$, reaching state $q_<$,
or appending letters after $z(\sigma)$, reaching state $q_>$,
or introducing an error in $z(\sigma)$, reaching state $q_\neq$.

From these three states, arbitrary input and output can occur and return in state q_0 produces the output $\bar{b}$. Note that in these three states it is not relevant whether z represents x or y. The $(>, x)$-quarter is depicted in Figure 6. The formal description can be found in the appendix. □

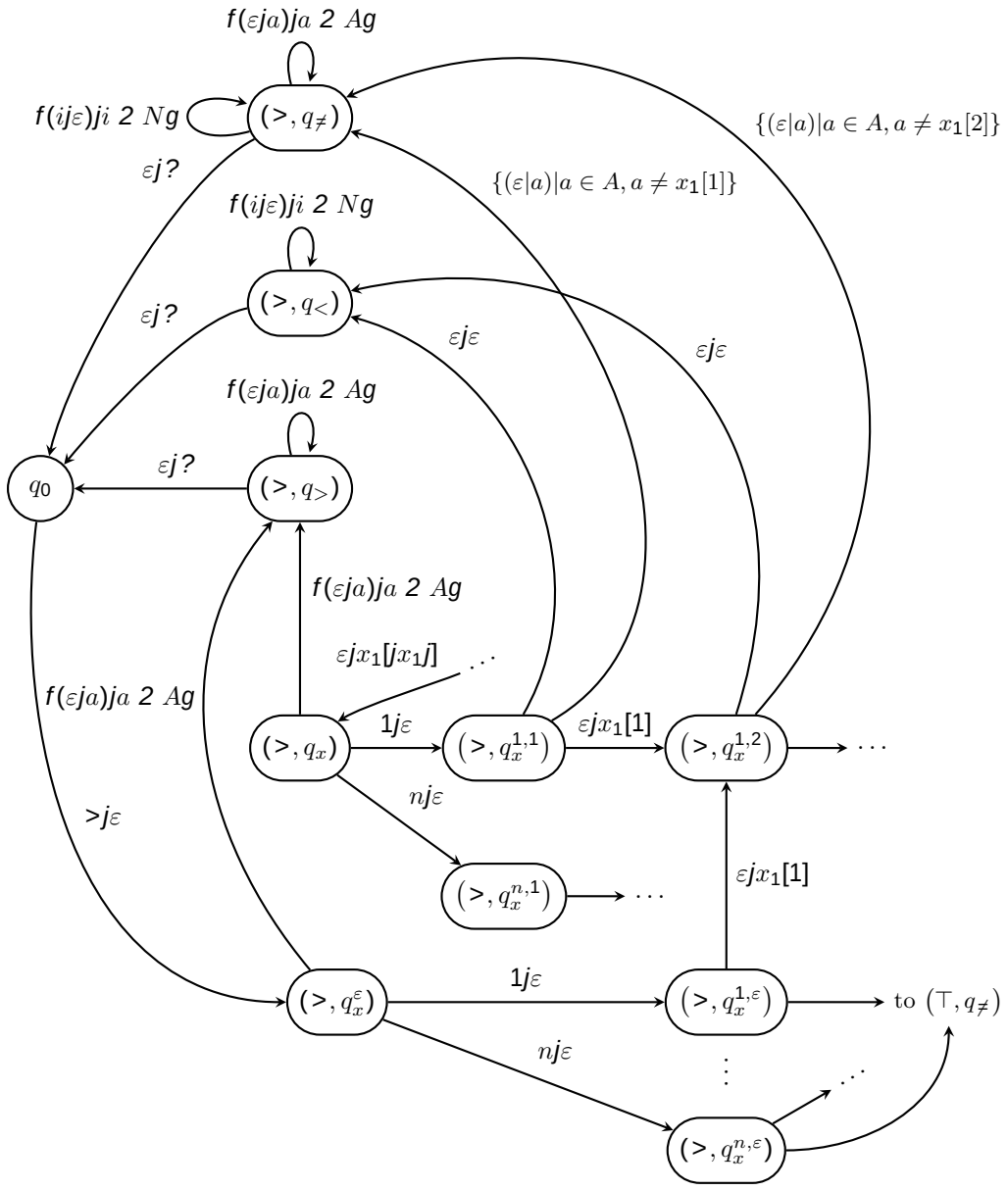


Figure 6. Structure of the $(\top, x)$-quarter of transducer $\mathcal{A}_I$: it accepts the relation $\{(\top \cdot \sigma, u \cdot \bot) \mid u \in A^+ \setminus \{x(\sigma)\}\}$.

Example 15. *Consider the following instance I_0 of PCP:*

$$\left(\begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline x & abb & b & a \\ y & a & abb & bb \end{array}\right)$$

This instance has a solution σ = 1311322 *which yields the word* w = *abbaabbabbabb. On input* $>\sigma$, *transducer* $A_{\mathcal{I}_0}$ *can output any string followed by a* $>$, *along a run looping in state* q_*. *In particular,* $w>$ *is a possible output. On the same input, some other strings followed by a* ? *may be an output, but not* $w?$.

The correspondence between a solution of the PCP instance and the channel is proved by the two following lemmata.

Lemma 16. *If instance* I *has a solution, then there is a channel for* $M_{\mathcal{I}}$.

Proof. Let σ be this solution, with $|\sigma| > 0$, and $w = x(\sigma) = y(\sigma)$. We can assume that there is no index i such that $x_i = y_i = \varepsilon$, hence $w \neq \varepsilon$. Then w is not generated by the "error" part of transducer $A_{\mathcal{I}}$. Hence, for input $b\ \sigma$, with $b \in \{>, ?\}$, $w\ b$ is an output whereas $w\ \overline{b}$ is not.

The key point in the proof is to build the channel for $M_{\mathcal{I}}$. Let $E_\sigma = \{(0|?\ \sigma), (1|>\ \sigma)\}^*$ and $D_\sigma = \{(w\ ?|0), (w\ >|1)\}^*$ realized by the transducers of Figure 7.

(a) Transducer $\mathcal{E}_\sigma$ from $\{0,1\}$ to N_B (b) Transducer $\mathcal{D}_\sigma$ from A_B to $\{0,1\}$

Figure 7. Encoder and decoders $\mathcal{E}_\sigma$ and $\mathcal{D}_\sigma$, where σ is a solution of the instance $\mathcal{I}$ of PCP and w the corresponding word.

Let $\beta \in \{0,1\}^*$ be a word of length n. The only image of β by E_σ is the word $u = \beta[1]\ \sigma\ \ \beta[n]\ \sigma$. For each factor $\beta[j]\ \sigma$, transducer $A_{\mathcal{I}}$ produces the language

$$(A^+\ \ \beta[j])\ [\ \left((A^+ \cap \{w\})\ \ \overline{\beta[j]}\right)$$

since w is produced neither by the $(\beta[j], x)$-quarter, nor by the $(\beta[j], y)$-quarter. So the image of u by $M_{\mathcal{I}}$ is the set

$$M_{\mathcal{I}}(u) = \{v_1\ \ b_1\ \ \ v_n\ \ b_n \mid \forall 1 \leq j \leq n : v_j \in A^+ \wedge b_j \in \{>, ?\} \text{ and } (v_j = w \Rightarrow b_j = \beta[j])\}.$$

In particular, the word $v = w\ \ \beta[1]\ \ \ w\ \ \beta[n]$ belongs to $M_{\mathcal{I}}(u)$, while any word of the form $w\ \ b_1\ \ \ w\ \ b_n$, with $(b_1, \ldots, b_n) \neq (\beta[1], \ldots, \beta[n])$, is not in $M_{\mathcal{I}}(u)$.

On the decoding side, a word of the form $w\ \ b_1\ \ \ w\ \ b_m$ is decoded into $b_1\ \ \ b_m$ by relation D_σ. Words that do not belong to $(w\ \ \{>, ?\})^*$ have no image by D_σ. Therefore,

$$D_\sigma(M_{\mathcal{I}}(u)) = D_\sigma(v) = \beta[1]\ \ \ \beta[n] = \beta.$$

Hence $E_\sigma \ M_{\mathcal{I}} \ D_\sigma(\beta) = \beta$ for every word β, and (E_σ, D_σ) is a channel for $M_{\mathcal{I}}$. □

Example 15 (continued) *Encoding* 0 *with* $>\sigma$ *and* 1 *with* $?\sigma$*, while decoding* 0 *with* $w>$ *and* 1 *with* $w?$*, yields a channel for* $M_{\mathcal{I}_0}$.

Lemma 17. *If* I *has no solution, then there is no channel for* $M_{\mathcal{I}}$.

Proof. This is proved by contradiction. Assume that I has no solution, then $M_{\mathcal{I}}(b\ \sigma) = A^+ \ f>, ?g$ for any $\sigma \in N^*$ and any bit $b \in f>, ?g$. Now if there is a channel for $M_{\mathcal{I}}$, using Proposition 13, we obtain tuples of words $U = (u, u_0, u_1, u')$ and $V = (v, v_0, v_1, v')$ such that fu_0, u_1g and fv_0, v_1g are codes and $(E(U), D(V))$ is a canonical channel for $M_{\mathcal{I}}$.

Consider the two words $\beta = 01$ and $\beta' = 10$. Their respective images by $E(U)$ are $n = uu_0u_1u'$ and $n' = uu_1u_0u'$. Since n and n' have images by $M_{\mathcal{I}}$, they belong to $(B \ N^*)^p$ for some $p > 0$, hence $M_{\mathcal{I}}(n) = M_{\mathcal{I}}(n') = (A^+ \ f>, ?g)^p$. We obtain $(E(U) \ M_{\mathcal{I}} \ D(V))(\beta) = (E(U) \ M_{\mathcal{I}} \ D(V))(\beta')$ which is a contradiction. □

6. Decidability for functional transducers

Theorem 7 is proved by establishing that the necessary condition from Theorem 9 is in fact su cient for a functional transducer, and building the channel. The proof relies on Lemmata 18, 19 and 20.

Lemma 18. *Let* M *be a function realized by transducer* $A_{\mathcal{M}}$*. There is a channel for* M *if and only if there exists an encoding state in* $A_{\mathcal{M}}$.

Proof. Let s be an encoding state in $A_{\mathcal{M}} = hS, I, A^* \ B^*, \ , Fi$, with codes fu_0, u_1g, fv_0, v_1g, and runs $s \xrightarrow{u_0|v_0} s$ and $s \xrightarrow{u_1|v_1} s$. Then the pair $(E(\varepsilon, u_0, u_1, \varepsilon), D(\varepsilon, v_0, v_1, \varepsilon))$ is a (canonical) channel for relation M_s realized by $A_{\mathcal{M}_s} = hS, fsg, A^* \ B^*, \ , fsgi$, which di ers from $A_{\mathcal{M}}$ only by its initial and nal states, and is also functional.

Since s is a useful state, there exist some runs $s_0 \xrightarrow{u|v} s$ and $s \xrightarrow{u'|v'} s_f$, with $s_0 \in I$, $s_f \in F$, $u, u' \in A^*$ and $v, v' \in B^*$. Since both M and M_s are functional, for any word $w \in fu_0, u_1g^*$, we have $M(u \ w \ u') = v \ M_s(w) \ v'$. Hence the pair $(E(u, u_0, u_1, u'), D(v, v_0, v_1, v'))$ is a channel for M. □

In order to nd encoding states in the transducer $A_{\mathcal{M}}$ realizing function M, we de ne for any word $u \in Dom(M)$ the set $NCI(u, M) = fu' \in A^* \mid M(u) \ M(u') \neq M(u') \ M(u)g$ of words whose image by M do not commute with the image of u. Then, we have:

Lemma 19. *For any function* M *and any word* $u \in Dom(M)$*,* $NCI(u, M)$ *is a regular subset of* A^*.

Proof. Let $v = M(u)$. Consider the language $C(v) = \{v' \in B^* \mid v \ v' = v' \ v\}$ of words commuting with v. Applying a classical result ([12] or [7]) we obtain a word $z \in B^*$ such that $C(v) = z^*$ (z is the shortest word which commutes with v), hence $C(v)$ is a regular language. Then $\overline{C}(v) = \{v' \in Im(M) \mid v \ v' \neq v' \ v\} = Im(M) \cap C(v)$ is also regular, as well as $NCI(u, M) = M^{-1}(\overline{C}(v))$. □

We now give a characterization for an encoding state:

Lemma 20. *Let $A_{\mathcal{M}} = \langle S, \{s\}, A^* \times B^*, \{s\}\rangle$ be a functional transducer realizing a function M, with $s \in S$ the only initial and final state. Then:*

> *If there exists $w \in M^{-1}(Im(M) \cap \{\varepsilon\})$ such that $NCI(w, M) \neq \emptyset$, then s is an encoding state.*
> *On the other hand, if s is an encoding state, then for any word $w \in M^{-1}(Im(M) \cap \{\varepsilon\})$, $NCI(w, M) \neq \emptyset$.*

Proof. First suppose that there is a word $w \in M^{-1}(Im(M) \cap \{\varepsilon\})$ such that $NCI(w, M) \neq \emptyset$. Then, since M is functional, $w \neq \varepsilon$ and we can conclude that s is an encoding state in M by choosing any $u_1 \in NCI(w, M)$, and setting $u_0 = w$, $v_0 = M(u_0)$, and $v_1 = M(u_1)$.

Conversely, suppose that s is an encoding state in M for some words u_0, u_1, v_0, v_1, then for $i \in \{0, 1\}$, $v_i = M(u_i)$. Consider now any $w \in M^{-1}(Im(M) \cap \{\varepsilon\})$, again with $w \neq \varepsilon$, and define $v = M(w)$. If $NCI(w, M)$ is empty, then $u_0 \notin NCI(w, M)$ and $u_1 \notin NCI(w, M)$, hence $v \ v_0 = v_0 \ v$ and $v \ v_1 = v_1 \ v$. Therefore, there exists $z \in B^*$ such that v, v_0 and v_1 all belong to z^* which is a contradiction. □

We finally give the proof of Theorem 7 with the complexity.

Proof of Theorem 7: decision procedure for a functional transducer. The decision and synthesis procedure is as follows: starting from the functional transducer $A_{\mathcal{M}} = \langle S, I, A^* \times B^*, F\rangle$ realizing function M, consider for each state $s \in S$ the transducer $A_s = \langle S, \{s\}, A^* \times B^*, \{s\}\rangle$ and the associated relation M_s.

The procedure computes a word u whose image by M_s is not ε. This can be done by looking for states $s_1, s_2 \in S$, such that $s_1 \xrightarrow{u_e|v_e} s_2$ with $u_e \in A^*$ and $v_e \in B^+$ and finding a run $\rho = s \cdots s_1 \xrightarrow{u_e|v_e} s_2 \cdots s$. If no such word can be found, then $Im(M_s) = \{\varepsilon\}$ and s is not an encoding state. Pruning S (to remove unuseful states in A_s) can be done in $O(|M|^2)$. The run ρ can be found from s_1 and s_2 in $O(|M|^2)$ too. So computing a word u whose image by M_s is not ε can be done in $O(|M|^2)$.

Let $v = M_s(u)$. The subset $Com(v)$ of B^* of words that commute with v is of the form z^* and a deterministic automaton A_z of size $O(|z|)$ accepts $\overline{z^*}$. An automaton $A_{Im(\mathcal{M}_s)}$ of size $O(|M|)$ recognizes $Im(M_s)$. Therefore the automaton B for the intersection of these languages, of size $O(|z| \times |M|)$ and with a single

initial state, recognizes $\overline{Com}(v) = Im(M_s) \cap z^*$. The emptiness problem for this automaton can be solved in linear time in the size of the product, hence in $O(|M|^2)$. If $\overline{Com}(v)$ is empty, then so is its preimage by M, and therefore $NCI(u, M) = \emptyset$ and there is no channel (by Lemma 20). Otherwise, since $\overline{Com}(v) \subseteq Im(M_s)$, we have $M_s^{-1}(\overline{Com}(v)) = NCI(u, M_s) \neq \emptyset$, and there is a channel in M_s, which can be synthesized by the construction in the proof of Lemma 20. This construction implies computing a word w in $NCI(u, M_s)$ and its image by M_s. The word w obtained as a witness by the emptiness check is thus of size $O(|M_s|^2)$, and the computation of $M_s(w)$ takes $O(|M_s| \cdot |w|)$. Hence the whole synthesis part is in $O(|M_s|^3)$.

By Lemma 18, the existence of a channel for one transducer M_s is equivalent to the existence of a channel for M, and the construction of the encoder and decoder for M from the ones for M_s can be done as in the proof of Lemma 18, in linear time with respect to $|M_s|$. Since $|z| \leq |v| \leq |M_s| \leq |M|$, the whole procedure goes in $O(|M|^4)$. □

7. Security properties for transducer systems

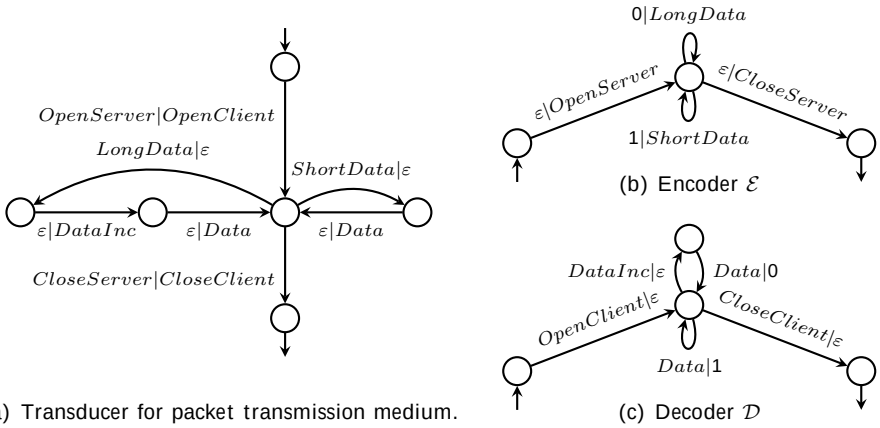
The above technique allows to discover in a system ways to transmit information. Although this transmission can be legitimate and thus of no worry, it may be the case that the channel is $covert$ [11]. This decision has to be made by the modeler, as pointed out by Millen [13]. Covert channels comprise all protocols that bypass the intended behavior of the system in order to transmit information. Practical examples have been shown in the past, such as using TCP/IP headers [17]. Some models of such channels have been devised [9, 8] although the authors define covert channels by the existence of an encoding state while we obtain this feature as a necessary condition.

The model of rational transducers offers a setting in which to study a system seen as a black-box process reading actions of users with high level of credentials (alphabet H), and outputting public or low-level actions (alphabet L). Note that any transition system over an alphabet $H \uplus L \uplus I$ (where $\uplus$ stands for the disjoint union) with a set I of internal actions can syntactically be transformed into a transducer over $H^* \times L^*$.

Typically, high-level actions are executed by a user $inside$ the system, while low-level actions are read from $outside$ it. For instance, high-level actions can be triggered by a Trojan horse in the system, trying to communicate a secret key to an external agent. The communication has to be stealthy in order not to be detected by the system, hence cannot use obvious communication channels which can be monitored. The communication also has to be reliable in order for the key to be transmitted correctly. Our model is well suited for the analysis of such threats. Future work should investigate the relation between the absence of channel and the validation of some security policies.

Let us consider the following example, inspired from [9], where a packet trans-

mission device can transmit data in two ways (see Figure 8(a)). Upon receiving a small amount of data, it can transmit it in a single (complete) packet. However, upon receiving a large amount of data, it transmits an incomplete packet followed by a complete one. An attacker can take advantage of this discrepancy in order to transmit data not inside the packets, but through the way complete and incomplete packets will be received, as shown by the encoder/decoder pair of Figure 8(b)-(c).



(a) Transducer for packet transmission medium.

(b) Encoder $\mathcal{E}$

(c) Decoder $\mathcal{D}$

Figure 8. A channel of delay 0 for the packet transmission protocol.

8. Conclusion

The model presented in this paper allows to describe reliable channels in the simple framework of transducers. Although the problem of synthesis of a channel is undecidable in general, it becomes polynomial in the case of a functional transducer. This complexity gap seems to indicate that decidability may be achieved for larger classes of transducers. We conjecture that it is the case for the class of relations defined as finite unions of functions. The case of infinite words, with on-line detection, as well as relations with security properties should also be more deeply investigated.

Acknowledgments. This work was partially supported by project ImpRo (ANR-2010-BLAN-0317), project CoChaT (Digiteo-2009-HD27) and NSERC discovery grant 13321-2010 (Government of Canada). We also would like to thank reviewers of AFL'11 and IJFCS for their insightful comments.

Bibliography

[1] Béal, M.P., Carton, O., Prieur, C., Sakarovitch, J.: Squaring transducers: An efficient procedure for deciding functionality and sequentiality of transducers. In Gonnet,

G.H., Panario, D., Viola, A., eds.: Proceedings of the 4th Latin American Symposium on Theoretical Informatics (LATIN'00). Volume 1776 of Lecture Notes in Computer Science, London, UK, Springer-Verlag (2000) 397–406.

[2] Bérard, B., Benattar, G., Lime, D., Mullins, J., Roux, O.H., Sassolas, M.: Channel synthesis for finite transducers. In Dömösi, P., Iván, S., eds.: Proceedings of the 13th International Conference on Automata and Formal Languages (AFL'11). (August 2011) 79–92.

[3] Elgot, C.C., Mezei, J.E.: On relations defined by generalized finite automata. IBM Journal Res. Develop. **9** (1965) 47–68.

[4] Finkbeiner, B., Schewe, S.: Uniform distributed synthesis. In: Proc. of LICS'05. (2005) 321–330.

[5] Gurari, E.: An introduction to the theory of computation. Computer Science Press, New York (1989).

[6] Harju, T., Hoogeboom, H., Kleijn, H.: Identities and transductions. In Karhumäki, J., Maurer, H., Rozenberg, G., eds.: Results and Trends in Theoretical Computer Science. Volume 812 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (1994) 140–144.

[7] Harrison, M.A.: Introduction to formal language theory. Addison-Wesley (1978).

[8] Hélouët, L., Roumy, A.: Covert channel detection using information theory. In Chatzikokolakis, K., Cortier, V., eds.: Proc. of the 8th Int. Workshop on Security Issues in Concurrency. (August 2010).

[9] Hélouet, L., Zeitoun, M., Degorre, A.: Scenarios and Covert channels: another game... In L. de Alfaro, ed.: Proc. of Games in Design and Verification (GDV'04). Volume 119 of ENTCS, Elsevier (2005) 93–116.

[10] Kupferman, O., Vardi, M.Y.: Synthesizing distributed systems. In Halpern, J.Y., ed.: Proc. of LICS'01, Washington, DC, USA, IEEE Computer Society (2001) 389.

[11] Lampson, B.: A note on the confinement problem. Commun. ACM **16**(10) (1973) 613–615.

[12] Lothaire, M.: Combinatorics on words. Volume 17 of Encyclopedia of Mathematics. Addison-Wesley, Reading, MA (1983).

[13] Millen, J.K.: 20 years of covert channel modeling and analysis. In: Proc. of the 1999 IEEE Symposium on Security and Privacy. (May 1999) 113 –114.

[14] Pnueli, A., Rosner, R.: Distributed reactive systems are hard to synthesize. In: Proc. of FOCS'90. Volume II, IEEE Computer Society Press (1990) 746–757.

[15] Sakarovitch, J.: Éléments de théorie des automates. Vuibert Informatique (2003).

[16] Schewe, S., Finkbeiner, B.: Synthesis of asynchronous systems. In: Proc. of LOPSTR'06. Volume 4407 of LNCS, Springer (2006) 127–142.

[17] Trabelsi, Z., El Sayed, H., Frikha, L., Rabie, T.: A novel covert channel based on the IP header record route option. Int. J. Adv. Media Commun. **1**(4) (2007) 328–350.

Appendix A. Construction of transducer $\mathcal{A}_{\mathcal{I}}$ corresponding to an instance $\mathcal{I}$ of PCP

The set of states of $A_{\mathcal{I}}$ is:

$$Q = \{q_0\} \cup (\{>, ?\} \times (\{q_*, q_x, q_y, q_x^{\varepsilon}, q_y^{\varepsilon}, q_>, q_<, q_{\neq}\} \cup Q_{\mathcal{I}} \cup Q_{\mathcal{I}}^{\varepsilon}))$$

where

$$Q_{\mathcal{I}} = \left(\bigcup_{i=1}^{n}\bigcup_{j=1}^{|x_i|}\{q_x^{i,j}\}\right) \cup \left(\bigcup_{i=1}^{n}\bigcup_{j=1}^{|y_i|}\{q_y^{i,j}\}\right)$$

and

$$Q_{\mathcal{I}}^{\varepsilon} = \left(\bigcup_{i=1}^{n}\{q_x^{i,\varepsilon}\}\right) \cup \left(\bigcup_{i=1}^{n}\{q_y^{i,\varepsilon}\}\right)$$

are sets containing respectively one state per letter of each word in I and one state per word in I. State q_0 is the unique initial and final state.

The set of transitions in $A_{\mathcal{I}}$ is built with the following rules, with $b \in \{>, ?\}$, $z \in \{x, y\}$, $i \in N$, and $a \in A$:

(R1) For $q \in \{q_*, q_x^{\varepsilon}, q_y^{\varepsilon}\}$, $q_0 \xrightarrow{b|\varepsilon} (b, q) \in$.
The transducer $A_{\mathcal{I}}$ reads input bit b and (non deterministically) chooses if it will output:
- either a (non empty) word followed by b (state q_*),
- or a (non empty) word different from $x(\sigma)$ followed by $\overline{b}$ (state q_x^{ε}),
- or a (non empty) word different from $y(\sigma)$ followed by $\overline{b}$ (state q_y^{ε}).

(R2) $(b, q_*) \xrightarrow{\varepsilon|a\cdot b} q_0 \in$.
At least one letter is produced from q_* (to avoid the empty word) followed by the input bit b.

(R3) $(b, q_*) \xrightarrow{i|\varepsilon} (b, q_*) \in$ and $(b, q_*) \xrightarrow{\varepsilon|a} (b, q_*) \in$.
With these two rules, it is possible from q_* to read (over N) and output (over A) arbitrary words.

(R4) If $|z_i| > 0$, then $(b, q_z^{\varepsilon}) \xrightarrow{i|\varepsilon} \left(b, q_z^{i,\varepsilon}\right) \in$.
If $|z_i| > 1$, then $\left(b, q_z^{i,\varepsilon}\right) \xrightarrow{\varepsilon|z_i[1]} \left(b, q_z^{i,2}\right) \in$.
If $|z_i| = 1$, then $\left(b, q_z^{i,\varepsilon}\right) \xrightarrow{\varepsilon|z_i} (b, q_z) \in$.
If $|z_i| = 0$, then $(b, q_z^{\varepsilon}) \xrightarrow{i|\varepsilon} (b, q_z^{\varepsilon}) \in$.
These transitions output the beginning of word z_i (or z_i itself) from input index i, just after reading the input bit b.

(R5) If $|z_i| > 0$, then $(b, q_z) \xrightarrow{i|\varepsilon} \left(b, q_z^{i,1}\right) \in$.
For $1 \leq j < |z_i|$, $\left(b, q_z^{i,j}\right) \xrightarrow{\varepsilon|z_i[j]} \left(b, q_z^{i,j+1}\right) \in$.
$\left(b, q_z^{i,|z_i|}\right) \xrightarrow{\varepsilon|z_i[|z_i|]} (b, q_z) \in$.

If $|z_i| = 0$, then $(b, q_z) \xrightarrow{i|\varepsilon} (b, q_z) \in \Delta$.
With these rules, $A_{\mathcal{I}}$ can read index i in state q_z and produce z_i, reaching state q_z.

(R6) For $1 \leq j < |z_i|$, $\left(b, q_z^{i,j}\right) \xrightarrow{\varepsilon|\varepsilon} (b, q_<) \in \Delta$ and $(b, q_<) \xrightarrow{i|\varepsilon} (b, q_<) \in \Delta$.
With these transitions, the production of z_i is interrupted before its end, reaching state $q_<$. There, no ouput occurs while reading the rest of the input.

(R7) $(b, q_z) \xrightarrow{\varepsilon|a} (b, q_>) \in \Delta$ and $(b, q_>) \xrightarrow{\varepsilon|a} (b, q_>) \in \Delta$.
These transitions correspond to adding a suffix without reading anything (input is complete).

(R8) Si $a \neq z_i[1]$, $\left(b, q_z^{i,\varepsilon}\right) \xrightarrow{\varepsilon|a} (b, q_{\neq}) \in \Delta$.
For $1 \leq j < |z_i|$, and if $a \neq z_i[j]$, $\left(b, q_z^{i,j}\right) \xrightarrow{\varepsilon|a} (b, q_{\neq}) \in \Delta$.
$(b, q_{\neq}) \xrightarrow{i|\varepsilon} (b, q_{\neq}) \in \Delta$ and $(b, q_{\neq}) \xrightarrow{\varepsilon|a} (b, q_{\neq}) \in \Delta$.
With these transitions, an error is inserted in z_i, reaching $q_{\neq}$ from which it is possible to read (over N) and output (over A) arbitrary words.

(R9) For $q \in \{q_<, q_>, q_{\neq}\}$, $(b, q) \xrightarrow{\varepsilon|\bar{b}} q_0 \in \Delta$.
Return from an error state produces the complement of the input bit.

Thus, transitions obtained from rules R4 and R5 build a part of $A_{\mathcal{I}}$ that produces $z(\sigma)$. However, since q_z^{ε} and q_z are not final states, the word $z(\sigma)$ itself cannot be obtained in this part. Indeed, rules R6, R7 and R8 respectively produce a strict prefix, or a word strictly containing $z(\sigma)$, or insert errors.

If nothing was produced from the beginning of the input sequence, execution stays in states q_z^{ε} et $q_z^{i,\varepsilon}$, left by reading the first letter of z_i. Note that there is no counterpart to rule R6 from $q_z^{i,\varepsilon}$, to avoid producing the empty word. Transitions corresponding to rules R2 and R9 reach state q_0, which yields the required extension of relation $M_{\mathcal{I}}$ to any input word in $(\{<, ?\} \times N^*)^*$.