Channel Synthesis for Finite Transducers*

Gilles Benattar[1] Béatrice Bérard[2] Didier Lime[1]
John Mullins[3] Olivier H. Roux[1] Mathieu Sassolas[2]

[1]École Centrale de Nantes, IRCCyN, CNRS UMR 6597
[2]Université Pierre & Marie Curie, LIP6/MoVe, CNRS UMR 7606
[3]École Polytechnique de Montréal

Email: Gilles.Benattar@irccyn.ec-nantes.fr,
mathieu.sassolas@lip6.fr

Abstract

We investigate how two agents can communicate through a noisy medium modeled as a finite transducer. The sender and the receiver are also described by finite transducers which can respectively encode and decode binary messages. When the communication is reliable, modulo some transmission delay, we call the encoder/decoder pair a channel. We study the channel synthesis problem, which asks whether, given a system, such sender and receiver exist and builds them if the answer is positive.

We prove that the problem is undecidable. However, we obtain a synthesis procedure when the transducer is a finite union of functions. We discuss these results in relation to security properties.

1 Introduction

Given an architecture defined by processes and communication links between them or with the environment, and a specification on the messages transmitted over these links, distributed synthesis aims at deciding existence of local programs, one for each process, that together meet the specification, whatever the environment does. In the case of synchronous

*Work partially supported by project DOTS (ANR-06-SETI-003), project CoChaT (Digiteo-2009-HD27), NSERC discovery grant 13321-2007 (Government of Canada).

communication, the problem was proved decidable (but non-elementary) for LTL properties over pipeline architectures [14, 10], or more generally [5], when the processes are sorted in a *linear preorder* related to the information received from the environment. In the asynchronous setting, the problem of synthesis of at least two processes is undecidable for general LTL specifications [17].

Here, we consider a simpler setting with only two processes (sender and receiver), respectively encoding and decoding binary messages, and a particular basic specification expressing faithful communication: the message received is equal to the message emitted, possibly modulo some transmission delay. However, these processes communicate asynchronously through a (non-deterministic) medium, acting as noise over the link between them. The three processes (encoder, decoder and noise) are modeled by finite transducers. The synthesis problem then asks if, given the noisy process, the encoder and decoder can be synthesized. We call *reliable channel* (or *channel* for short) such an encoder/decoder pair, and thus call this problem the *channel synthesis problem.*

We prove that, even in this most simple setting, the channel synthesis problem is undecidable (Section 4), except in a restricted case where the noisy process is a finite union of functional transducers (Section 5). Then, we discuss in Section 6 the possible relations of these results with security properties.

2 Preliminaries

Notations. The set of *words* over a finite alphabet A is denoted by A^*, with ε for the empty word. The length of a word u is written $|u|$ and for $1 \leq i \leq |u|$, the ith letter of u is denoted by $u[i]$. A *language* is a subset of A^*.

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 $u = vw$. The set of *k-bounded prefixes* of u is defined by $\mathit{Pref}_k(u) = \{v \in A^* | v \preccurlyeq u \wedge |u| - |v| \leq k\}$, it contains the prefixes v of u whose length differ from the length of u by at most k letters.

Finite automata. A finite automaton, or automaton for short, is a tuple $\mathcal{M} = \langle S, s_0, \mathit{Lab}, \Delta, F\rangle$, where S is a set of states, s_0 is the initial state, Lab is a set of labels, $\Delta \subseteq S \times \mathit{Lab} \times S$ is the transition relation and $F \subseteq S$ is a subset of final states. Note that Lab can be an alphabet but also a (part of a) monoid. A *run from* $s \in S$ is a path starting from s in the graph, written as $\rho = s \xrightarrow{a_1} s_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} s_n$, with $a_i \in \mathit{Lab}$ and $s_i \in S$, for $1 \leq i \leq n$. The *trace* of ρ is $\mathit{trace}(\rho) = a_1 \cdots a_n$. We write $s \overset{u}{\Rightarrow} s'$ if there is a run ρ from s to s' with trace u. A run ρ as above is *accepting* if $s = s_0$ and $s_n \in F$ and the language of $\mathcal{M}$ 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 $\mathit{Lab} = A$.

Finite Transducers. A finite transducer (or transducer for short) is a finite automaton $\mathcal{M}$ with set of labels $\mathit{Lab} \subseteq A^* \times B^*$ for two alphabets A and B. A label $(u, v) \in A^* \times B^*$ is often written as $u|v$ in the figures (see examples in Figure 1). The subset of $A^* \times B^*$ containing the traces of accepting runs of $\mathcal{M}$ is a *rational relation* [16] from A^* to B^*. The transducer $\mathcal{M}$ is said to realize the corresponding relation which is also denoted by $\mathcal{M}$. With this slight abuse of notation, for a word $u \in A^*$, we write $\mathcal{M}(u) = \{v \in B^* \mid (u, v) \in \mathcal{M}\}$ for the image of u, $\mathcal{M}^{-1}(v) = \{u \in A^* \mid (u, v) \in \mathcal{M}\}$ for the inverse image of v, possibly extended to subsets of A^* or B^* respectively, $\mathit{Dom}(\mathcal{M}) = \{u \in A^* \mid \exists v \in B^*, (u, v) \in \mathcal{M}\}$ for the domain of $\mathcal{M}$ and $\mathit{Im}(\mathcal{M}) = \{v \in B^* \mid \exists u \in A^*, (u, v) \in \mathcal{M}\}$ for the image of $\mathcal{M}$. When $\mathcal{M}(w)$ is a singleton, it will also denote its only element, again with a slight abuse of notation. If the domain of $\mathcal{M}$ is A^*, then $\mathcal{M}$ is said to be *complete*. The transducer is *functional* if it realizes a partial function: for each word $w \in A^*$, there is at most one word in $\mathcal{M}(w)$.

For a subset P of A^*, the identity relation $\{(w, w) \mid w \in P\}$ on $A^* \times A^*$ is denoted by $\mathit{Id}(P)$ and $\mathit{Id}_k(P)$ is the relation between words and their k-bounded prefixes in P: $\mathit{Id}_k(P) = \{(u, v) \in P \times P \mid v \in \mathit{Pref}_k(u)\}$. Note that $\mathit{Id}_0 = \mathit{Id}$.

The composition of rational relations $\mathcal{M}$ on $A^* \times B^*$ and $\mathcal{M}'$ on $B^* \times C^*$, denoted by $\mathcal{M} \cdot \mathcal{M}'$, is a rational relation on $A^* \times C^*$ ([4]). Moreover, the image and inverse image of a regular language by a rational relation is a regular language [16].

3 Channels

We consider communication between two processes, respectively called an *encoder* and a *decoder*. The encoder $\mathcal{E}$ reads binary input and produces an output in A^*, while the decoder $\mathcal{D}$ reads words over B and produces a binary word. The intermediate noisy agent is modeled by a transducer over $A^* \times B^*$. The definition below states that a channel corresponds to reliable communication: the binary message is correctly transmitted, modulo some delay to take into account transmission time.

Definition 1. *A* channel *with delay k for a transducer $\mathcal{M} = \langle S, s_0, A^* \times B^*, \Delta, F\rangle$ is a pair $(\mathcal{E}, \mathcal{D})$ such that $\mathcal{E}$ is a transducer on $\{0,1\}^* \times A^*$, $\mathcal{D}$ is a transducer on $B^* \times \{0,1\}^*$ and $Id(\{0,1\}^*) \subseteq \mathcal{E} \cdot \mathcal{M} \cdot \mathcal{D} \subseteq Id_k(\{0,1\}^*)$.*

The first inclusion means that any binary words can be encoded, hence it implies that encoder $\mathcal{E}$ is complete. This inclusion could be weakened but it must ensure that the channel permits communication.

First recall some properties of channels from [2]. The first one shows that verification is possible when the pair $(\mathcal{E}, \mathcal{D})$ is given, for channels with delay 0.

Proposition 2 ([2])**.** *Let $\mathcal{M}$ be a transducer on $A^* \times B^*$ and let $\mathcal{E}$ and $\mathcal{D}$ be two transducers on $\{0,1\}^* \times A^*$ and $B^* \times \{0,1\}^*$, respectively. It can be decided whether $(\mathcal{E}, \mathcal{D})$ is a channel with delay* 0 *for $\mathcal{M}$.*

Also, by compacting together sequences of letters, it can be shown that looking only for channels with no delay is sufficient.

Proposition 3 ([2])**.** *If there is a channel with delay k for transducer $\mathcal{M}$, then there is also a channel with no delay for $\mathcal{M}$.*

We consider now the channel synthesis problem: "given a transducer $\mathcal{M}$, is there a channel for $\mathcal{M}$?" and prove the two following results in the sections 4 and 5, respectively:

Theorem 4. *The channel synthesis problem is Σ_1^0-complete.*

Theorem 5. *The channel synthesis problem for a functional transducer $\mathcal{M}$ is decidable in polynomial time. Moreover if there is a channel for $\mathcal{M}$, it can be computed.*

Both proofs partly rely on a structural necessary condition for the existence of a channel, based on the following notion of encoding state.

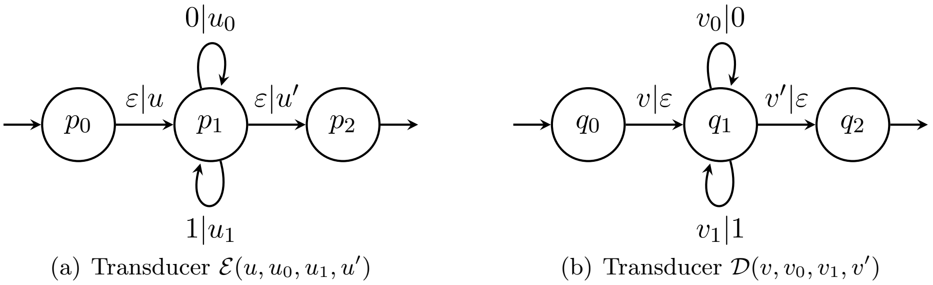
Definition 6. *Let $\mathcal{M} = \langle S, s_0, A^* \times B^*, \Delta, F\rangle$ be a transducer. An* encoding state *is a state $s \in S$ such that there exist four words $u_0, u_1 \in A^*$ and $v_0, v_1 \in B^*$ such that (i) $s \xRightarrow{u_0|v_0} s$ and $s \xRightarrow{u_1|v_1} s$ and (ii) $u_1 \cdot u_0 \neq u_0 \cdot u_1$ and $v_1 \cdot v_0 \neq v_0 \cdot v_1$.*

Note that condition (i) means that there are two cycles on state s while condition (ii) expresses the fact that the pair $\{u_0, u_1\}$ (resp. $\{v_0, v_1\}$) is a code ([12, 7]). For tuples $U = (u, u_0, u_1, u')$ and $V = (v, v_0, v_1, v')$ of words in A^* and B^* respectively, we define the relations

$$\begin{array}{rcl} \mathcal{E}(U) & = & (\varepsilon, u) \cdot \{(0, u_0), (1, u_1)\}^* \cdot (\varepsilon, u') \\ \text{and} \quad \mathcal{D}(V) & = & (v, \varepsilon) \cdot \{(v_0, 0), (v_1, 1)\}^* \cdot (v', \varepsilon), \end{array}$$

which correspond to the transducers in Figure 1. Combinatorial arguments then yield the following result:

Theorem 7 ([2]). *Let $\mathcal{M} = \langle S, s_0, A^* \times B^*, \Delta, F\rangle$ be a transducer. If there is a channel for $\mathcal{M}$ then:*
1. There exist tuple of words $U = (u, u_0, u_1, u')$ in A^* and $V = (v, v_0, v_1, v')$ in B^*, such that $u_0 \cdot u_1 \neq u_1 \cdot u_0$ and $v_0 \cdot v_1 \neq v_1 \cdot v_0$, and $(\mathcal{E}(U), \mathcal{D}(V))$ is a channel for $\mathcal{M}$.
2. there is an encoding state.



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

Figure 1: General form of encoder and decoder.

The condition is not sufficient as shown by the example of the (non functional) system $\mathcal{N}$ of Figure 2. State s_4 is an encoding state, but an u can also lead to s_3, which simulates s_4, but where no word can be encoded: indeed, after u_0 or u_1, both v_0 and v_1 can be produced. In this case, the non-functionality of $\mathcal{N}$ breaks the locality of the encoding state property.

4 Channel synthesis is undecidable

This section is devoted to a sketch of the proof of Theorem 4. Proposition 2 states that the channel synthesis problem is in Σ_1^0. The proof of Σ_1^0-hardness is done by a reduction from Post's Correspondence Problem (PCP) [15].

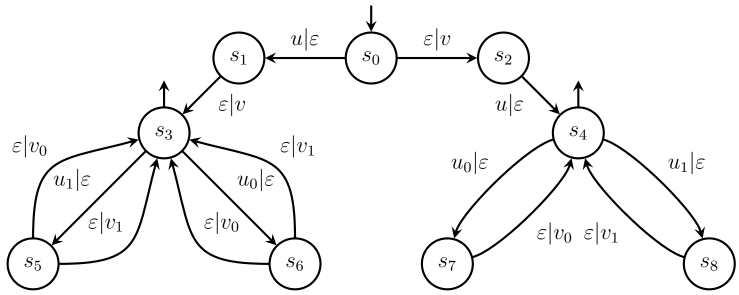


Figure 2: Transducer $\mathcal{N}$ with an encoding state s_4 but no channel

Recall that an instance of PCP is a tuple $\mathcal{I} = \langle (x_1, y_1), \ldots, (x_n, y_n)\rangle$ of pairs of words over an alphabet A. A (non trivial) solution is a (non empty) sequence of indexes $i_1, \ldots, i_k$ such that $x_{i_1} \cdots x_{i_k} = y_{i_1} \cdots y_{i_k}$. The problem of existence of a solution is undecidable.

Starting from an instance $\mathcal{I} = \langle (x_1, y_1) \ldots, (x_n, y_n)\rangle$ of PCP, we build a transducer $\mathcal{M}_\mathcal{I}$ such that:

$$\mathcal{I} \text{ has a solution if and only if there is a channel for } \mathcal{M}_\mathcal{I}.$$

The construction extends the undecidability proof for transducer equality [6] with an additional construction to obtain the channel property. The main idea is the following: the transducer $\mathcal{M}_\mathcal{I}$ reads a bit followed by a sequence of indexes. If the sequence is a non-trivial solution of the instance $\mathcal{I}$, then the bit is transmitted. Otherwise, $\mathcal{M}_\mathcal{I}$ may flip the input bit. More precisely, given an input sequence $bi_1, \ldots, i_k$ such that $i_1, \ldots, i_k$ is a solution of $\mathcal{I}$ and b a bit, $\mathcal{M}_\mathcal{I}$ will output anything except $x_{i_1} \cdots x_{i_k} = y_{i_1} \cdots y_{i_k}$ followed by $\overline{b}$ the complement of the input bit. Detecting this "missing word" will allow to deduce the input bit, and hence to transmit a message.

We give an intuition of the construction of transducer $\mathcal{M}_\mathcal{I}$; the detailed proof can be found in [3].

Construction. We consider the alphabets $B = \{\top, \bot\}$, $N = \{1, \ldots, n\}$, $A_B = A \cup B$, and $N_B = N \cup B$. For $b \in B$, we define $\overline{b}$ by $\overline{\top} = \bot$ and $\overline{\bot} = \top$. Recall that an instance $\mathcal{I}$ can also be seen as a pair of morphisms x and y, with $x(\sigma) = x_{i_1} \cdots x_{i_k}$ and $y(\sigma) = y_{i_1} \cdots y_{i_k}$ for any word $\sigma = i_1 \cdots i_k \in N^*$. Hence PCP can be reformulated as the existence of a sequence σ, with $|\sigma| > 0$, such that $x(\sigma) = y(\sigma)$.

Transducer $\mathcal{M}_\mathcal{I} = \langle Q, q_0, N_B^* \times A_B^*, \Delta, \{q_0\}\rangle$ realizes a relation in $N_B^* \times$

A_B^* such that for $b \in B$ and $\sigma \in N^+$:

$$\begin{array}{rcl} \mathcal{M}_{\mathcal{I}}(b \cdot \sigma) & = & (A^* \cdot b) \cup \big((A^* \setminus \{x(\sigma)\}) \cdot \overline{b}\big) \cup \big((A^* \setminus \{y(\sigma)\}) \cdot \overline{b}\big) \\ \text{and} \quad \mathcal{M}_{\mathcal{I}}(b) & = & A^+ \cdot \{b, \overline{b}\} \end{array}$$

This means that this transducer taking as input a bit and a sequence of indexes, outputs:

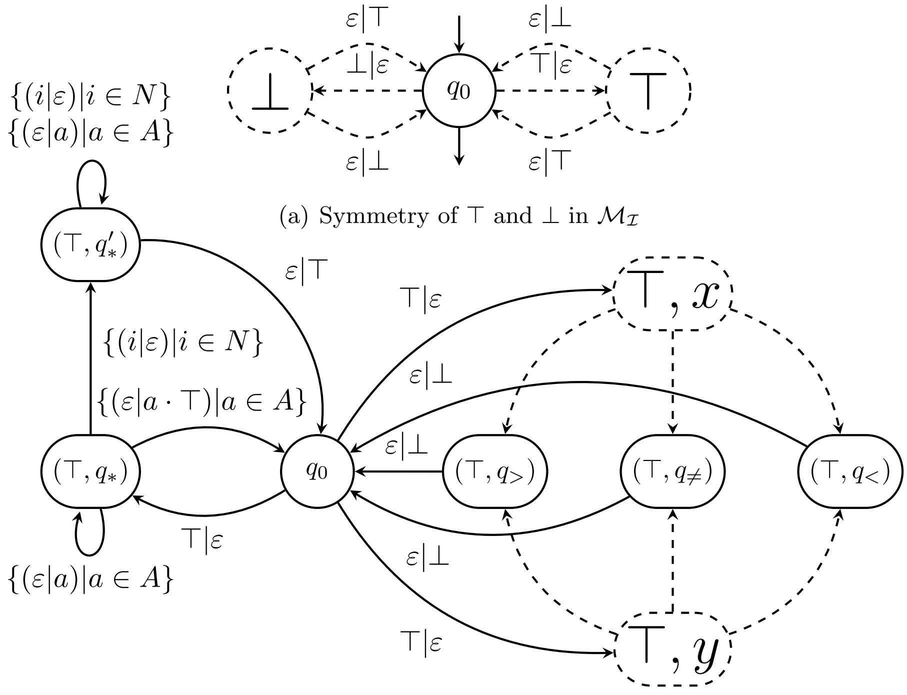
- either any word followed by the same bit,
- or a word which is not the image of the sequence by x followed by the opposite of the input bit,
- or a word which is not the image of the sequence by y followed by the opposite of the input bit.

The case of an empty sequence of indexes is treated separately: the word followed by the input bit has to be non-empty. This relation is extended to N_B^* by $\mathcal{M}_{\mathcal{I}}(\varepsilon) = \{\varepsilon\}$ and for $b_1, \ldots, b_p \in B$ and $\sigma_1, \ldots, \sigma_p \in N^*$: $\mathcal{M}_{\mathcal{I}}(b_1 \cdot \sigma_1 \cdots b_p \cdot \sigma_p) = \mathcal{M}_{\mathcal{I}}(b_1 \cdot \sigma_1) \cdots \mathcal{M}_{\mathcal{I}}(b_p \cdot \sigma_p)$ while $\mathcal{M}_{\mathcal{I}}(v) = \emptyset$ if $v \notin (B \cdot N^*)^*$.

Transducer $\mathcal{M}_{\mathcal{I}}$ is composed of two symmetrical parts that keep in memory one bit b of information (see Figure 3(a)). The part of $\mathcal{M}_{\mathcal{I}}$ consisting of state q_* and q'_* does not look at its input and generates any word of A^* (or a non empty word if no input was read), appending b after it. On the other hand, on input $b \cdot \sigma$, the other states (which will be called the diff-part in the sequel) generate either a word which is not $x(\sigma)$, or a word which is not $y(\sigma)$, appending $\overline{b}$ after it (see Figure 3(b)). This is achieved by introducing errors in $x(\sigma)$ (or $y(\sigma)$, without loss of generality), by either outputting a strict prefix of $x(\sigma)$ (reaching $q_<$), or appending letters after $x(\sigma)$ (reaching $q_>$), or by introducing an error in $x(\sigma)$ (reaching $q_{\neq}$). This part is depicted in Figure 5 in the case of an example.

Sketch of correctness proof. If, for an input $b \cdot \sigma$ with $|\sigma| > 0$, the sequence σ is a solution of $\mathcal{I}$, then $w = x(\sigma) = y(\sigma)$ will not be generated by the diff-part of $\mathcal{M}_{\mathcal{I}}$, hence $w \cdot b$ will be an output whereas $w \cdot \overline{b}$ will not[1]. Conversely, if the sequence σ is not a solution of $\mathcal{I}$, then $w = x(\sigma) \neq y(\sigma)$ will be generated by the diff-part of $\mathcal{M}_{\mathcal{I}}$ (in this case in the "y part" of the transducer), hence both $w \cdot b$ and $w \cdot \overline{b}$ will be outputs. Note that in both cases, there will be other outputs: all $u \cdot b$ and $u \cdot \overline{b}$ for $u \in A^* \setminus \{w\}$. When $|\sigma| = 0$, which is always a trivial solution of $\mathcal{I}$, the empty word ε cannot

[1] We can assume that there is no index i such that $x_i = y_i = \varepsilon$, hence $w \neq \varepsilon$.



(a) Symmetry of $\top$ and $\bot$ in $\mathcal{M}_\mathcal{I}$

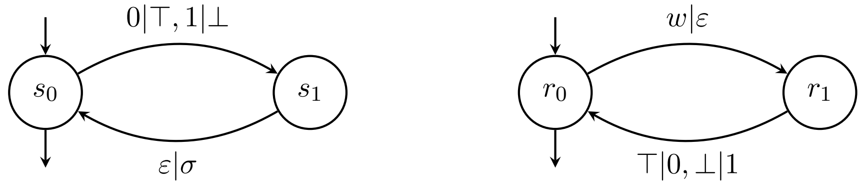
(b) Structure of the $\top$ part of $\mathcal{M}_\mathcal{I}$

Figure 3: Global structure of $\mathcal{M}_\mathcal{I}$.

be produced in the q_* part of $\mathcal{M}$. Hence neither b nor $\overline{b}$ will be produced (alone). The case above, with an input in $B \cdot N^*$, can be generalized to an input in $(B \cdot N^*)^*$. Indeed, q_0 is the only initial and final state and the structure of $\mathcal{M}_\mathcal{I}$ ensures that q_0 is left reading a letter of B, reached producing a (possibly different) letter of B, and that no other transition either reads or outputs a letter of B.

The key point in the proof is to build the channel for $\mathcal{M}_\mathcal{I}$ when $\mathcal{I}$ has a solution. Let $\sigma = i_1 \cdots i_k$ be this solution, with $k > 0$, and $w = x(\sigma) = y(\sigma)$. Then the pair $(\mathcal{M}_\sigma, \mathcal{M}_w)$ of transducers depicted in Figure 4 are a channel for $\mathcal{M}_\mathcal{I}$.

Example 1. *Consider the instance $\mathcal{I}_0 = \langle (abb, a), (b, abb), (a, bb) \rangle$ of PCP. The corresponding transducer $\mathcal{M}_{\mathcal{I}_0}$ is partly depicted (only the $\top, x$ part) in Figure 5.This instance has a solution $\sigma =$ 1311322 which yields the word $w = abbaabbabbabb$. On input $\top$1311322, $\mathcal{M}_{\mathcal{I}_0}$ can output any string followed by a $\top$, along a run looping through states q_* and q'_*. In particular, $w\top$ is a possible output. On the same input, some other strings followed*



(a) Transducer $\mathcal{M}_\sigma$ from $\{0,1\}^*$ to N_B^*

(b) Transducer $\mathcal{M}_w$ from A_B^* to $\{0,1\}^*$

Figure 4: Encoder and decoders $\mathcal{M}_\sigma$ and $\mathcal{M}_w$, where σ is a solution of the instance $\mathcal{I}$ of PCP and w the corresponding word.

by a $\bot$ may be an output, e.g. *$abbaabbabaa\bot$, produced by a run*

$$q_0 \xrightarrow{\top|\varepsilon} q_x \xrightarrow{1|\varepsilon} q_x^{1,1} \xrightarrow{\varepsilon|a} q_x^{1,2} \xrightarrow{\varepsilon|b} q_x^{1,3} \xrightarrow{\varepsilon|b} q_x \xrightarrow{3|\varepsilon} q_x^{3,1} \xrightarrow{\varepsilon|a} q_x \xrightarrow{1|\varepsilon} q_x^{1,1} \xrightarrow{\varepsilon|a} \cdots$$

$$\cdots \xrightarrow{\varepsilon|a} q_x^{1,2} \xrightarrow{\varepsilon|b} q_x^{1,3} \xrightarrow{\varepsilon|b} q_x \xrightarrow{1|\varepsilon} q_x^{1,1} \xrightarrow{\varepsilon|a} q_x^{1,2} \xrightarrow{\varepsilon|b} q_x^{1,3} \xrightarrow{\varepsilon|a} q_{\neq} \xrightarrow{\varepsilon|a} q_{\neq} \xrightarrow{\varepsilon|\bot} q_0.$$

However, $w\bot$ is not an output, since after reading $\top 1311322$ *and producing w, the run ends in state q_x (or q_y) which is not accepting and cannot reach q_0 without reading more input. Hence encoding* 0 *with* $\top 1311322$ *and* 1 *with* $\bot 1311322$*, while decoding* 0 *with* $w\top$ *and* 1 *with* $w\bot$ *yields a channel for* $\mathcal{M}_{\mathcal{I}_0}$.

The converse is proved by contradiction: if $\mathcal{I}$ has no solution, and if there is a channel for $\mathcal{M}_\mathcal{I}$, then, using theorem 7, we can find two words u and u' such that $(\mathcal{E} \cdot \mathcal{M}_\mathcal{I} \cdot \mathcal{D})(u) = (\mathcal{E} \cdot \mathcal{M}_\mathcal{I} \cdot \mathcal{D})(u')$ while $u \neq u'$, thus a contradiction.

5 Decidability for functional transducers

Theorem 5 is proved by establishing that the necessary condition from Theorem 7 is in fact sufficient for a functional transducer, and building the channel. The proof of Theorem 5 is based on lemmata 8, 9 and 10.

Lemma 8. *Let $\mathcal{M} = \langle S, s_0, A^* \times B^*, \Delta, F\rangle$ be a functional transducer. There is a channel for $\mathcal{M}$ if and only if there exists an encoding state.*

Proof. Let s be an encoding state in $\mathcal{M}$, with $s \xRightarrow{u_0|v_0} s$ and $s \xRightarrow{u_1|v_1} s$, $u_1{\cdot}u_0 \neq u_0{\cdot}u_1$ and $v_1{\cdot}v_0 \neq v_0{\cdot}v_1$. We denote by $\mathcal{E}_0(u_0,u_1)$ and $\mathcal{D}_0(v_0,v_1)$ respectively the transducers in Figure 6. Then the pair $(\mathcal{E}_0(u_0,u_1), \mathcal{D}_0(v_0,v_1))$ is a channel for $\mathcal{M}_s = \langle S, s, A^* \times B^*, \Delta, \{s\}\rangle$, which differs from $\mathcal{M}$ only

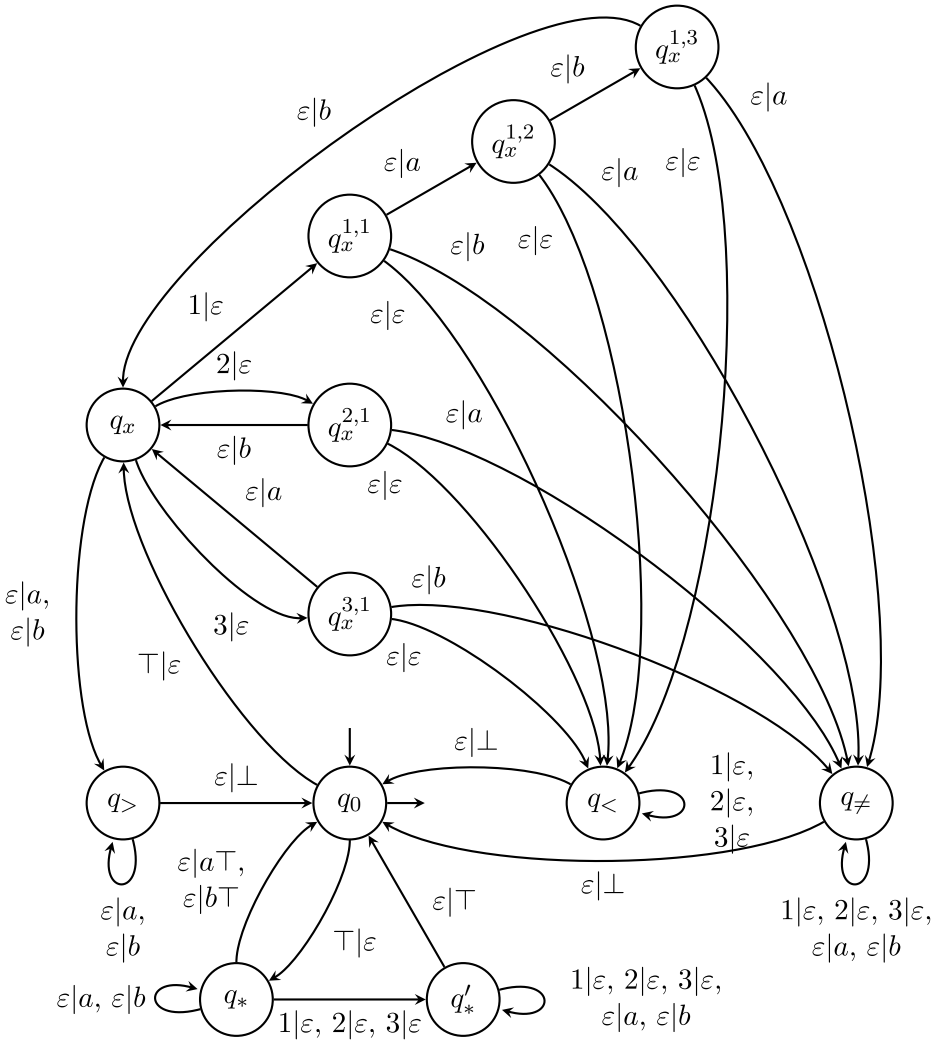
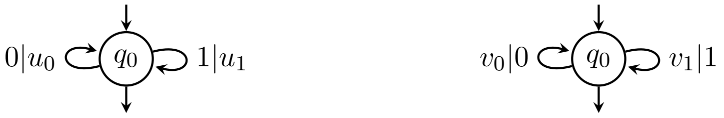

$q_x^{1,3}$
$\varepsilon|b$
$\varepsilon|a$
$\varepsilon|b$
$q_x^{1,2}$
$\varepsilon|a$
$\varepsilon|a$
$\varepsilon|\varepsilon$
$q_x^{1,1}$
$\varepsilon|b$
$\varepsilon|\varepsilon$
$1|\varepsilon$
$\varepsilon|\varepsilon$
$2|\varepsilon$
q_x
$q_x^{2,1}$
$\varepsilon|a$
$\varepsilon|b$
$\varepsilon|\varepsilon$
$\varepsilon|a$
$\varepsilon|a$, $\varepsilon|b$
$\varepsilon|b$
$q_x^{3,1}$
$3|\varepsilon$
$\varepsilon|\varepsilon$
$\top|\varepsilon$
$\varepsilon|\bot$
$\varepsilon|\bot$
$1|\varepsilon$, $2|\varepsilon$, $3|\varepsilon$
$q_>$
q_0
$q_<$
$q_{\neq}$
$\varepsilon|\bot$
$\varepsilon|a\top$, $\varepsilon|b\top$
$\varepsilon|\top$
$\varepsilon|a$, $\varepsilon|b$
$1|\varepsilon$, $2|\varepsilon$, $3|\varepsilon$, $\varepsilon|a$, $\varepsilon|b$
$\top|\varepsilon$
$\varepsilon|a$, $\varepsilon|b$
q_*
q'_*
$1|\varepsilon$, $2|\varepsilon$, $3|\varepsilon$
$1|\varepsilon$, $2|\varepsilon$, $3|\varepsilon$, $\varepsilon|a$, $\varepsilon|b$

Figure 5: Part of transducer $\mathcal{M}_{\mathcal{I}_0}$ encoding PCP instance $\mathcal{I}_0$. Only the $\top$-x quarter has been represented.

by its initial and final states, and is also functional. Since s is a useful state, there exist some runs $s_0 \overset{(u|v)}{\Longrightarrow} s$ and $s \overset{(u'|v')}{\Longrightarrow} s_f$, with $s_f \in F$, $u, u' \in A^*$ and $v, v' \in B^*$. Since both $\mathcal{M}$ and $\mathcal{M}_s$ are functional, for any word $w \in \{u_0, u_1\}^*$, we have $\mathcal{M}(u \cdot w \cdot u') = v \cdot \mathcal{M}_s(w) \cdot v'$. Hence the pair $(\mathcal{E}(u, u_0, u_1, u'), \mathcal{D}(v, v_0, v_1, v'))$ is a channel for $\mathcal{M}$. □



(a) Transducer $\mathcal{E}_0(u_0, u_1)$

(b) Transducer $\mathcal{D}_0(v_0, v_1)$

Figure 6: Encoder and decoder for $\mathcal{M}_s$.

In order to find encoding states in a transducer $\mathcal{M}$, for any word $u \in \mathit{Dom}(\mathcal{M})$, we define the set $\mathit{NCI}(u, \mathcal{M}) = \{u' \in A^* \mid \mathcal{M}(u) \cdot \mathcal{M}(u') \neq \mathcal{M}(u') \cdot \mathcal{M}(u)\}$ of words whose image by $\mathcal{M}$ do not commute with the image of u. Then, we have:

Lemma 9. *Given a functional transducer $\mathcal{M}$ and a word $u \in \mathit{Dom}(\mathcal{M})$, $\mathit{NCI}(u, \mathcal{M})$ is a regular subset of A^*.*

Proof. Let $v = \mathcal{M}(u)$. Consider the language $C(v) = \{v' \in B^* | v \cdot v' = v' \cdot v\}$ 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 commute with v), hence $C(v)$ is a regular language. Then $\overline{C}(v) = \{v' \in \mathit{Im}(\mathcal{M}) \mid v \cdot v' \neq v' \cdot v\} = \mathit{Im}(\mathcal{M}) \setminus C(v)$ is also regular, as well as $\mathit{NCI}(u, \mathcal{M}) = \mathcal{M}^{-1}(\overline{C}(v))$. □

We now prove that it can be decided whether a state is encoding.

Lemma 10. *Let $\mathcal{M} = \langle S, s, A^* \times B^*, \Delta, \{s\}\rangle$ be a functional transducer, with $s \in S$ the initial and only finite state. Then:*

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

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

Conversely, suppose that s is an encoding state in $\mathcal{M}$ for some words u_0, u_1, v_0, v_1, then for $i \in \{0, 1\}$, $v_i = \mathcal{M}(u_i)$. Consider now any $w \in \mathcal{M}^{-1}(\mathit{Im}(\mathcal{M}) \backslash \{\varepsilon\})$, again with $w \neq \varepsilon$, and define $v = \mathcal{M}(w)$. If $\mathit{NCI}(w, \mathcal{M})$ is empty, then $u_0 \notin \mathit{NCI}(w, \mathcal{M})$ and $u_1 \notin \mathit{NCI}(w, \mathcal{M})$, hence $v \cdot v_0 = v_0 \cdot v$

and $v \cdot v_1 = v_1 \cdot v$. Therefore, there exists $z \in B^*$ such that v, v_0 and v_1 all belong to z^* which is a contradiction. □

Proof of Theorem 5. The decision and synthesis procedure is as follows: for each state $s \in S$, consider transducer $\mathcal{M}_s$ (in which all states are assumed to be useful). Then compute a word u whose image by $\mathcal{M}_s$ is not ε. This can be done by looking if there is $s_1, s_2 \in S$, s.t. $s_1 \xrightarrow{u_e|v_e} s_2$ with $u_e \in A^*$ and $v_e \in B^+$ and finding a run $\rho = s \Rightarrow s_1 \xrightarrow{u_e|v_e} s_2 \Rightarrow s$. If no such word can be found, then $\mathit{Im}(\mathcal{M}_s) = \{\varepsilon\}$ and it is clear that there is no channel for $\mathcal{M}_s$. Pruning S (to remove unuseful states) can be done in $O(|\mathcal{M}|^2)$. The run ρ can be found from s_1 and s_2 in $O(|\mathcal{M}|^2)$ too. So computing u whose image by $\mathcal{M}_s$ is not ε can be done in $O(|\mathcal{M}|^2)$.

Let $v = \mathcal{M}_s(u)$. The subset $C(v)$ of B^* of words that commute with v is of the form z^* and a deterministic automaton $\mathcal{A}_z$ of size $O(|z|)$ accepts $\overline{z^*}$. An automaton $\mathcal{A}_{\mathit{Im}(\mathcal{M}_s)}$ of size $O(|\mathcal{M}|)$ recognizes $\mathit{Im}(\mathcal{M}_s)$. Therefore the automaton $\mathcal{B}$ for the intersection of these languages, of size $O(|z| \times |\mathcal{M}|)$ and with a single initial state, recognizes $\overline{C}(v) = \mathit{Im}(\mathcal{M}_s) \setminus z^*$. The emptiness problem for this automaton can be solved in linear time in the size of the product, hence in $O(|\mathcal{M}|^2)$. If $\overline{C}(v)$ is empty, then so is its preimage by $\mathcal{M}$, and therefore $\mathit{NCI}(u, \mathcal{M}) = \emptyset$ and there is no channel (by Lemma 10). Otherwise, since $\overline{C}(v) \subseteq \mathit{Im}(\mathcal{M}_s)$, $\mathcal{M}_s^{-1}(\overline{C}(v)) = \mathit{NCI}(u, \mathcal{M}_s) \neq \emptyset$, and there is a channel in $\mathcal{M}_s$, which can be synthesized by the construction in the proof of Lemma 10. This construction implies computing a word w in $\mathit{NCI}(u, \mathcal{M}_s)$ and its image by $\mathcal{M}_s$. The word w obtained as a witness of the emptiness check and thus of size $O(|\mathcal{M}_s|^2)$, and the computation of $\mathcal{M}_s(w)$ takes $O(|\mathcal{M}_s| \times |w|)$. Hence the whole synthesis part is in $O(|\mathcal{M}_s|^3)$.

By Lemma 8, the existence of a channel for one transducer $\mathcal{M}_s$ is equivalent to the existence of a channel for $\mathcal{M}$, and the construction of the encoder and decoder for $\mathcal{M}$ from the ones for $\mathcal{M}_s$ can be done as in the proof of Lemma 8, in linear time with respect to $|\mathcal{M}_s|$.

Since $|z| \leq |v| \leq |\mathcal{M}_s| \leq |\mathcal{M}|$, the whole procedure goes in $O(|\mathcal{M}|) \times O(|\mathcal{M}|^2 + |\mathcal{M}|^2 + |\mathcal{M}|^3) = O(|\mathcal{M}|^4)$. □

This result can be extended to the case of transducers built as a union of n functional transducers:

$$\mathcal{M} = \mathcal{F}_1 \cup \cdots \cup \mathcal{F}_n \text{ where } \forall 1 \leq i \leq n,\ \mathcal{F}_i \text{ is functional.}$$

By induction, the problem boils down to the fact that if a transducer $\mathcal{M}$ has a covert channel, then so does $\mathcal{M}' = \mathcal{M} \cup \mathcal{F}$. Intuitively, either the encoder can *force* $\mathcal{M}'$ to be in the component $\mathcal{M}$, or $\mathcal{F}$ itself contains a channel. The full proof can be found in [1].

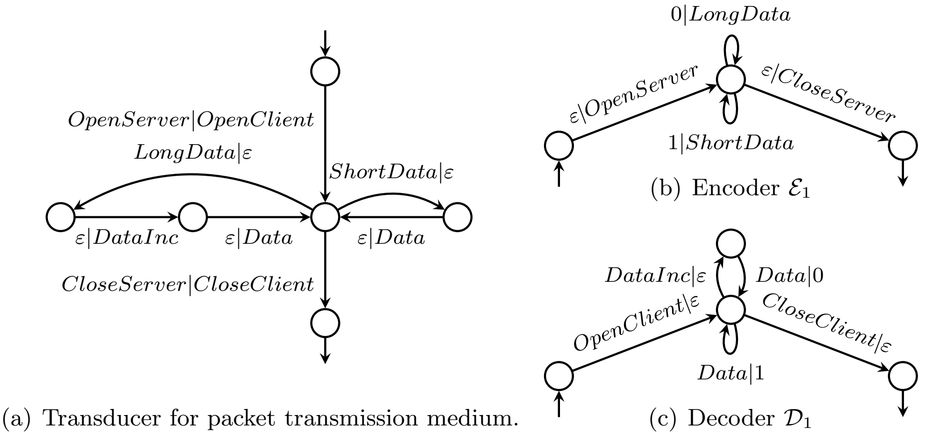
6 Security properties for transducer systems

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 [18]. 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 the exterior. 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 a covert channel and the validation of some security policies.

Let us consider the following example, inspired from [9], where a packet transmission device can transmit data in two ways (see Figure 7(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 7(b)-(c).



(a) Transducer for packet transmission medium.

(b) Encoder $\mathcal{E}_1$

(c) Decoder $\mathcal{D}_1$

Figure 7: A channel of delay 0 for the packet transmission protocol.

7 Conclusion

The model presented in this paper can be used to specify reliable channels in the simple framework of transducers. Although the problem of existence 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.

References

[1] Benattar, G.: Synthèse de systèmes informatiques temporisés non interférents. PhD thesis, Université de Nantes (2011).

[2] Benattar, G., Bérard, B., Lime, D., Mullins, J., Roux, O.H., Sassolas, M.: Covert channels with sequential transducers. In: Workshop on Foundations of Computer Security. (August 2009).

[3] Benattar, G., Bérard, B., Lime, D., Mullins, J., Roux, O.H., Sassolas, M.: Covert channels synthesis for transducers. Technical report, IRCCyN - LIP6 - École Polytechnique de Montréal (March 2010).

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

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

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

[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] Post, E.L.: A variant of a recursively unsolvable problem. Bulletin of the American Mathematical Society **52**(4) (April 1946) 264–268.

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

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

[18] 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.