Reliable Contracts for Unreliable Half-Duplex Communications*

Étienne Lozes[1] and Jules Villard[2]

[1] University of Kassel (DE) [2] Queen Mary, University of London (UK)

Abstract. Recent trends in formal models of web services description languages and session types focus on the asynchronicity of communications. In this paper, we study a core of these models that arose from our modelling of the Sing# programming language, and demonstrate correspondences between Sing# contracts, asynchronous session behaviors, and the subclass of communicating automata with two participants that satisfy the half-duplex property. This correspondence better explains the criteria proposed by Stengel and Bultan for Sing# contracts to be reliable, and possibly indicate useful criteria for the design of WSDL. We moreover establish a polynomial-time complexity for the analysis of communication contracts under arbitrary models of asynchronicity, and we investigate the model-checking problems against LTL formulas.

Introduction

Communication contracts are becoming commonplace in several information systems, like languages for web services (WSDL, abstract WS-BPEL, WSCL, or WSCI) and programming languages (like Sing# [12] or Axum). Theoretical foundations of communication contracts are often based on bi-partite [16,22] or multi-partite session types, with a recent trend on the asynchronicity of communications [17]. In a previous work [24], we modelled the Sing# programming language, which also features asynchronous communication contracts preventing communication errors, and proposed a formal definition of Sing# contracts that was independently discovered by Stengel and Bultan [21]. This definition is remarkably close to the one of session behaviors [3], a first-order fragment of session types.

The primary goal of session types is to ensure type-safety, *e.g.* that communications over a typed channel follow the scenario of their type at runtime. Type safety then possibly ensures other safety properties, most notably reception-safety, *i.e.* the guarantee that the received messages are always of one of the expected formats, but also buffer boundedness, absence of message orphans, etc. However, type-safe programs may go wrong under some situations. Indeed, a program may be successfully checked against a contract, and yet not be reception-safe, if the contract features either non determinism, or mixed choice. In these situations, the server and the client may follow different paths on the contract and its dual, in which case the contract might not faithfully reflect the

* The European Research Council has provided financial support under the European Community's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement no 259267.The second author acknowledges support from EPSRC.

possible head message of the incoming queues. Such bugs occurred in two contracts of the original release of Singularity [21]. The reason for this problem is that reception safety is not an intrinsic property of the syntax of the contract, but rather of the set Post^* of reachable configurations. A formal link between the properties ensured on Post^* and the properties of programs that follow $\mathcal{C}$ can be established [23], remarkably without imposing syntactic constraints on contracts or a particular communication model like FIFO. More precisely, Villard showed [23] that the boundedness of the size of the channel, the reception safety, or the absence of message orphans on Post^* directly translate into the same properties for programs (whereas deadlocks and memory leaks require extra hypotheses).

Determinism and choice uniformity are quite standard conditions on sessions, but their intrinsic relation to FIFO communications is not fully acknowledged. Consider the contract $\mathcal{C} \coloneqq ?pin; !ack; (!ok \oplus !err); \mathcal{C}$ implemented by a server S: initially, S should wait for a pin on its input buffer, send an acknowledgement on its output buffer, and then check the pin and send either an authorisation or a denial to proceed. This server can be composed with any client that follows the dual specification $\overline{\mathcal{C}}$, where sending becomes receiving and vice-versa. Assume now that the communications are not perfect FIFO, and suffer from stuttering errors. Then, even for a type-safe program, it is possible that the ack message could be received twice, thus breaking reception-safety even for a contract-abiding client.

This observation raises the question whether communication contracts can be made reliable if asynchronocity is not FIFO, say for instance out-of-order, or with stuttering errors, as it is quite common in the world-wide web network. Answering this question requires first to precise what is meant by *reliable*. The literature on session types and contracts [16,17,21,5] basically gives two kinds of answers:

- the instrumental point of view: a contract is reliable if it guarantees that the programs it types are well-behaved,
- the multiparty and message sequence chart point of view: a contract is reliable if it is realisable, or inhabited, *i.e.* if there exists a program whose conversation is exactly the one described by the contract.

These two kinds of answers actually spawned a multiplicity of notions of "reliable" contracts, obtained by adopting several notions of "well-behaved" programs [3,19,23,24,12] different notions of "conversations" [8,17,21] and considering different communication models. This multiplicity of answers suggests that there is no robust notion of reliable contracts that would be meaningful for all kinds of properties and for all kinds of asynchronicity. Moreover, each proposal justifies the determinacy and uniform choice conditions as *sufficient* conditions for contracts being reliable, and while they usually suggest that these conditions might be relaxed, they do not show how to do it *effectively*.

The issue of effectively recognising whether a contract is reliable is however crucial. Contracts are naturally modelled as communicating automata, which in the FIFO case are Turing powerful even for only one communication buffer [7]. Under other asynchronous semantics, such *dialogue* systems are often no longer Turing powerful, but still exhibit a very high complexity for reachability questions, making them difficult to analyse in practise. Channel contracts, on the other hand, can be thought of as a restricted form of such dialogues where each machine is the dual of the other. One could think

that this restriction alone makes them easier to analyse but, as we show in this paper, dualised dialogues retain their Turing power from general FIFO dialogue systems, and their high complexity from other asynchronous semantics. In fact, we show that the complexity remains the same even if contracts enjoy one (but not both) of the two syntactic restrictions mentioned before (determinism and uniform choice).

In this paper, we propose to define reliable contracts as those that are *half-duplex*, *i.e.* those where the two communication buffers are never used simultaneously, similarly to walkie-talkie conversations. Cécé and Finkel first introduced this notion in the context of FIFO communications and showed that such communications enjoy a remarkable simplicity: the set of reachable configurations is regular, and their semantics is closely related to the synchronous semantics [9]. Although this does not scale to non-FIFO communications, we show that the polynomial-time complexity of the verification problems scales to a large class of communication models, including out-of-order, lossy and stuttering communications. Adopting the half-duplex property as a definition of reliable contracts has several advantages; first, it clarifies the determinacy and uniform choice conditions on contracts: these conditions are tight with respect to the half-duplex property. In comparison, other notions of reliable contracts suggest that these properties may be relaxed but, as we show in this paper, this quickly leads to undecidability. Second, it permits to effectively ensure a flexible notion of "well-behaved" programs that may or may not take into account unspecified receptions, orphan messages, boundedness, and any regular safety property that might be of interest. Indeed, once one has shown that a particular contract is half-duplex, these questions become efficiently decidable. Third, it does not rely on each of the two parties being the "dual" of the other, and thus avoids introducing a notion of subtyping in the situations where the two parties are typed against non-dual contracts.

Our contributions are the following:

1. We show that previous foundations of contracts neither convincingly explain the determinacy and polarisation conditions, nor do they provide arguments for making the analysis of contract communications effective.
2. We show that the half-duplex property is polynomial-time and that, for half-duplex systems, boundedness, absence of unspecified receptions and message orphans can be solved in polynomial time.
3. We investigate the LTL model-checking problem over traces of either configurations or actions, and show that the former is undecidable, even for half-duplex contract communications with safe receptions, whereas the latter is decidable.

In the first section, we introduce our model of asynchronous dialogues. The second section is dedicated to defining contracts and examining previous attempts at providing foundations for them. The third section is about half-duplex communications, and we establish the polynomial-time complexity of several problems. In the last section, we consider the model-checking problem for contracts and half-duplex dialogues against linear-time temporal logic.

1 Asynchronous Dialogues

Given a finite set Σ, we write Σ^* for the set of words over alphabet Σ, ranged over by w, w'; we write $w.w'$ for the concatenation of w and w', and ϵ for the empty word. The commutative (or Parikh) image of a word w is the set of words w' equal to w up to commuting letters. The semi-linear subsets S of $\mathbb{N}^\Sigma$ are the finite unions of sets of the form $\mathbb{N}\overrightarrow{a_1} + \cdots + \mathbb{N}\overrightarrow{a_n} + \overrightarrow{b}$, and correspond to the commutative images of regular languages. When we talk about a representation of a regular (resp. semi-linear) language, we mean a non-deterministic finite automaton (resp. a base-period decomposition). The class of decision problems $\mathbf{P}$ (resp. $\mathbf{NP}$) is the one of problems that can be decided in polynomial time in the size of the input by a deterministic (resp. non-deterministic) Turing machine. A decision problem is primitive recursive if it can be decided in time $O(f(n))$ for some primitive recursive function f. We write $⁇$ to denote an element of $\{!, ?\}$. For a subset S of an ordered set $(\mathcal{S}, \succeq)$, we write $\downarrow S$ and $\uparrow S$ for respectively the downward $\{s' : \exists s \in S, s \succeq s'\}$ and upward $\{s' : \exists s \in S, s' \succeq s\}$ closures of S. We assume a fixed alphabet Σ, whose size is a parameter in all the complexity results.

Communicator A communicator is a non-deterministic finite state automaton over an alphabet of the form $\{!, ?\} \times \Sigma$.

Definition 1 (Communicator). *A* communicator *is a tuple $\mathcal{M} = (Q, \Sigma, \Delta, \dot{q}, F)$ where:*

- *Q is a finite set of states;*
- *Σ is a finite set of messages (or letters);*
- *$\Delta \subseteq Q \times (\{!, ?\} \times \Sigma) \times Q$ is a finite set of transitions;*
- *$\dot{q} \in Q$ is called the initial state;*
- *$F \subseteq Q$ is called the set of final states.*

We range over Q with $q, q', \ldots$, and over Σ with $a, b, \ldots$ Elements of the set $\mathsf{Act}_\Sigma := \{!, ?\} \times \Sigma$ are called *actions*, ranged over by $\lambda, \lambda', \ldots$, and we write $\mathsf{pol}(\lambda) \in \{!, ?\}$ for their first projection (their *polarity*). As usual, we write $q \xrightarrow{\lambda}_{\mathcal{M}} q'$ if $(q, \lambda, q') \in \Delta$ (the subscript can be omitted). Let $\mathcal{M} = (Q, \Sigma, \Delta, \dot{q}, F)$ be a fixed communicator. A state q of $\mathcal{M}$ is *terminal* if there is no $(\lambda, q') \in (\{!, ?\} \times \Sigma) \times Q$ such that $q \xrightarrow{\lambda} q'$. A (non-empty) *path* in $\mathcal{M}$ is a finite sequence of states $q_0, \ldots, q_{n+1}$ such that $q_i \xrightarrow{\lambda_i} q_{i+1}$ for all $0 \leq i \leq n$. A path is *uniform* if $\mathsf{pol}(\lambda_i) = \mathsf{pol}(\lambda_j)$ for all $i, j \in \{0, \ldots, n\}$. A communicator is *connected* if for every non initial state q, there is a path from $\dot{q}$ to q. From now on, we will implicitly consider connected communicators only. A state q is *polarised* if there are no a, b, q_1, q_2 such that $q \xrightarrow{!a} q_1$ and $q \xrightarrow{?b} q_2$. A state q is *deterministic* if for all $\lambda \in \{!, ?\} \times \Sigma$, there is at most one q' such that $q \xrightarrow{\lambda} q'$. A state q is *k-bounded* if all uniform paths (possibly cyclic) starting from q have a length less than k. A communicator is *polarised* (resp. *deterministic*) if all its states are.

Dialogues Communicators will be either communicating with themselves using a single communication buffer, and then called *monologues*, or they will be paired with another communicator using two buffers, and then called *dialogues*.

Definition 2 (Dialogue). *A dialogue is a tuple* $\mathcal{D} = (Q_0, Q_1, \Sigma, \Delta_0, \Delta_1, \dot{q}, F)$ *such that* $(Q_i, \Sigma, \Delta_i, \dot{q}, \{q_i : (q_0, q_1) \in F\})$ *is a communicator for each* i*.*

Dialogues can be used to describe the semantics of Singularity contracts, where one communicating automaton is used to describe both sides of the conversation [12,21,23]. In this case, a communicator $\mathcal{M}$ is paired with its *dual*, *i.e.* the communicator in which each transition $q \xrightarrow{!a} q'$ is replaced by $q \xrightarrow{?a} q'$, for $! \neq \bar{!}$, and the final states of the system are the pairs (q, q) of final states of $\mathcal{M}$.[1] We write $\mathcal{M} \parallel \overline{\mathcal{M}}$ for this dialogue.

We will also use the notation $\mathcal{M}_1 \parallel \mathcal{M}_2$ to denote a dialogue between $\mathcal{M}_1$ and $\mathcal{M}_2$.

Interferences The semantics of dialogue systems will interpret send and receive actions as respectively pushing messages into an outgoing buffer and popping messages from an incoming buffer. However, we will not restrict ourselves to perfect FIFO buffers, and we will rather consider that they may be subject to several kinds of interferences from the environment. We model these interferences by a preorder over words $\succeq \subseteq \Sigma^* \times \Sigma^*$ which will parametrise the semantics of dialogue systems. Intuitively, $w \succeq w'$ if w and w' are the contents of a buffer respectively before and after being submitted to interferences.

Definition 3 (Interference Model). *An interference model is a binary relation* $\succeq \subseteq \Sigma^* \times \Sigma^*$ *satisfying the following axioms:*

Reflexivity	Transitivity	Additivity	Integrity
$\dfrac{a \in \Sigma}{a \succeq a}$	$\dfrac{w \succeq w' \qquad w' \succeq w''}{w \succeq w''}$	$\dfrac{w_1 \succeq w_1' \qquad w_2 \succeq w_2'}{w_1.w_2 \succeq w_1'.w_2'}$	$\dfrac{\epsilon \succeq w}{w = \epsilon}$

Intuitively, these axioms capture the following assumptions on the interferences we consider: they may leave the communication buffers unchanged, they act locally, and they cannot fill an empty buffer. The last assumption, corresponding to the integrity axiom, will be later clarified by the half-duplex property.

The least interference model is $w \succeq w'$ if and only if $w = w'$; it models FIFO communications, *i.e.* communications without interferences. Let us review some standard forms of interferences.

Lossiness Possible leaks of messages during transmission are modelled by adding the axiom $a \succeq \epsilon$.

Corruption Possible transformation of a message a into a message b is modelled by adding the axiom $a \succeq b$.

Out-of-order Out-of-order communications are modelled by adding axioms $a.b \succeq b.a$ for all $a, b \in \Sigma$.

Stuttering Possible duplication of a message a is obtained by adding the axiom $a \succeq a.a$.

Some of these models can be put in correspondence with existing communication protocols. For instance, the FIFO model corresponds essentially to TCP, and the lossy stuttering out-of-order model to UDP. Note that the out-of-order model corresponds to

[1] This is actually a slight generalisation: in Sing# contracts, final states are also terminal, and every state leads to a final state via a special message ChannelClosed. [21]

one where buffers are just multisets, and is thus computationally equivalent to vector addition systems with states, or Petri nets. One notable exception to our definition of interferences is the model with insertion errors, where arbitrary messages can be inserted. It would be modelled by the axiom $\epsilon \succeq a$, which would not validate the integrity axiom. We may sometimes make the stronger hypothesis that the interference model is *non-expanding*, meaning that $w \succeq w'$ implies that the length of w' is smaller than the one of w. Barring the stuttering model, all the interference models we mentioned are non-expanding.

Configurations Let $\mathcal{D}$ be a fixed dialogue. A configuration of $\mathcal{D}$ is a tuple

$$\gamma = (q_0, q_1, w_0, w_1) \in \mathsf{Confs}(\mathcal{D}) \coloneqq Q_0 \times Q_1 \times \Sigma^* \times \Sigma^*.$$

The initial configuration $\dot{\gamma}$ of a dialogue $\mathcal{D}$ is $(\dot{q}, \dot{q}, \epsilon, \epsilon)$; (q_0, q_1, w_0, w_1) is final if (q_0, q_1) is final. We will view a configuration (q_0, q_1, w_0, w_1) as the word $q_0.w_0.q_1.w_1$ over the alphabet $(Q_0 \cup Q_1) \uplus \Sigma$. Similarly, a set of configurations can be considered as a language over such an alphabet. A set of configurations is regular (resp. semi-linear) if its associated language is. For instance, the set of configurations with empty buffers is both regular and semi-linear, whereas the set of configurations with buffers having as many a and b messages is semi-linear but not regular. A configuration $\gamma = (q_0, q_1, w_0, w_1)$ is *stable* if both buffers are empty: $w_0.w_1 = \epsilon$, *a message orphan* if it is not stable and $(q_0, q_1) \in F$, *an unspecified reception* if there is i such that $w_i \neq \epsilon$ and $\gamma \overset{i\lambda}{\not\longrightarrow}$ for all $\lambda \in \{!, ?\} \times \Sigma$, and *a k-out-of-bound error*, for $k \in \mathbb{N}$, if the sum of the length of w_0 and w_1 is greater than k.

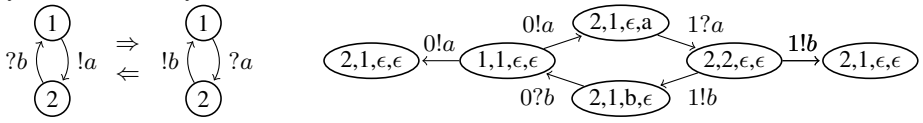
Semantics The transition system associated to a dialogue $\mathcal{D}$ for the interference model $\succeq$ is defined by a binary relation $\xrightarrow{i\lambda}_{\succeq,\mathcal{D}}$ over configurations. We write

$$(q_0, q_1, w_0, w_1) \xrightarrow{i\lambda}_{\succeq,\mathcal{D}} (q_0', q_1', w_0', w_1')$$

if and only if there is a transition $q_i \xrightarrow{\lambda}_{\mathcal{M}_i} q_i'$ such that $q_{1-i} = q_{1-i}'$, and

(send case) either $\lambda = !a$, $w_{1-i}.a \succeq w_{1-i}'$ and $w_i \succeq w_i'$;
(receive case) or $\lambda = ?a$, $w_i \succeq a.w_i'$ and $w_{1-i} \succeq w_{1-i}'$.

Example 1. The following picture represents a dialogue and its associated transition system in the lossy semantics.



We often write $\xrightarrow{i\lambda}$ instead of $\xrightarrow{i\lambda}_{\succeq,\mathcal{D}}$ when $\succeq$ and $\mathcal{D}$ are clear from the context. We write $\gamma \to \gamma'$ if $\gamma \xrightarrow{i\lambda} \gamma'$ for some i, λ, and we write Post^* for the set of reachable configurations, *i.e.* the smallest set containing $\dot{\gamma}$ and such that, for all γ, γ', if $\gamma \to \gamma'$ and $\gamma \in \mathsf{Post}^*$ then $\gamma' \in \mathsf{Post}^*$.

The semantics of a monologue is defined similarly: it behaves as if it were in a dialogue with a forwarder communicator.

A sequence of configurations $(\gamma_i)_{i\geq 0} \in \mathsf{Confs}(\mathcal{D})^{\mathbb{N}}$ is called a *trace of configurations* of a dialogue (resp. monologue) $\mathcal{D}$ if $\gamma_0 = \dot{\gamma}$, and there is some $N \in \mathbb{N} \cup \{\infty\}$ such that for all $i \leq N$, $\gamma_i \to_{\mathcal{D}} \gamma_{i+1}$, and for all $i > N$, $\gamma_i = \gamma_{i+1}$ is a final configuration. A finite sequence $\rho = \big((pid_0, \lambda_0) \dots (pid_n, \lambda_n)\big) \in (\{0,1\} \times \mathsf{Act}_\Sigma)^*$ is a *trace of actions* of the dialogue $\mathcal{D}$ if there is a trace $(\gamma_i)_{i\geq 0}$ of $\mathcal{D}$ such that $\gamma_i \xrightarrow{pid_i \lambda_i} \gamma_{i+1}$ for all $i \leq n$. A trace of send actions is a trace of actions where receive actions are skipped; a trace of receive actions is defined similarly. A finite sequence $c = \lambda_0 \dots \lambda_n \in \mathsf{Act}_\Sigma^*$ is called a *conversation* of $\mathcal{D}$ if there is a trace of send actions ρ such that c is obtained from ρ by the substitution $(0, !a) \mapsto !a$, $(1, !a) \mapsto ?a$. For instance, a conversation of $!a; ?b; !c; \mathsf{end} \parallel ?a; !b; \mathsf{end}$ is $!a.?b.!c$, obtained from the trace of send actions $(0, !a).(1, !b).(0, !c)$, which is itself a subtrace of the trace of actions $(0, !a).(1, ?a).(1, !b).(0, ?b).(0!c).(1, ?c)$. We write $\mathsf{Conv}(\mathcal{D})$ to denote the set of all conversations.

Problems of interest A safety property $P_{\mathcal{D}}$ of a dialogue $\mathcal{D}$ is a subset of $\mathsf{Confs}(\mathcal{D})$. We say that $\mathcal{D}$ satisfies $P_{\mathcal{D}}$ if $\mathsf{Post}^* \subseteq P_{\mathcal{D}}$. A safety property is polynomial-time regular if there is a polynomial-time function that associates to each dialogue system $\mathcal{D}$ a deterministic finite automaton that represents $P_{\mathcal{D}}$. Typical properties addressed by contracts are polynomial-time regular safety properties: safe receptions, *i.e.* the absence of unspecified receptions, *orphan-freedom*, *i.e.* the absence of messages in buffers at communication closure, and k-boundedness, *i.e.* absence of k-out-of-bound-errors. Boundedness, *i.e.* k-boundedness for some existentially quantified k, is also a typical property of interest, though it is not a safety property. All of these properties, as most properties depending on control state reachability, are known to be undecidable for dialogues and monologues in the FIFO semantics [7], and to be at least non-primitive recursive for the lossy semantics [18,20], and at least exponential space for the out-of-order semantics [11]. Finally, let us introduce two less standard properties. We say that a dialogue is *synchronous* if all stable reachable configurations can be reached without visiting a 2 out-of-bound configuration. In other words, a dialogue is synchronous if all stable configurations are reached in the synchronous semantics à la CCS; for instance, $!a; !b \parallel ?b; ?a$ is synchronous for the FIFO semantics (where it cannot run) but not for the out-of-order one. Following Stengel and Bultan [21], we say that a communicator $\mathcal{M}$ is *realisable* if, considered as finite state automaton, it exactly recognises $\mathsf{Conv}(\mathcal{M} \parallel \overline{\mathcal{M}})$. For instance, $!a + !b$ is realisable, whereas $!a + ?b$ is not.

2 Contracts

Contracts are communicators with certain syntactic conditions, similarly to Sing# contracts as formalised by Stengel and Bultan [21].[2]

[2] We omit the condition on final states being terminal, reached implictly in Sing# when receiving ChannelClosed messages. Moreover, Sing# requires another condition meant to ensure boundedness of the buffers, see Prop 1.5 below.

$\wedge$	$\neg$RS	$\neg$R	$\neg$S	$\neg$SC
RS	$-$	$!a + ?a$	$\mathcal{C}_1$	$!a + ?a$
R	$\mathcal{C}_2$	$-$	$\mathcal{C}_1$	$\mathcal{C}_2$
S	$!a; ?b + ?c$	$(!a; ?c^*; !b)^*$	$-$	$(!a; ?c^*; !b)^*$

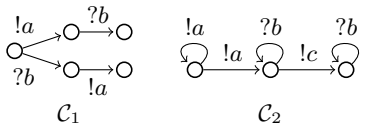


Fig. 1: Reception-safety (RS), realizability (R) and synchronism (S) are incomparable and generalise synched contracts (SC).

Definition 4 (Synched Contract). *A communicator $\mathcal{C}$ is called a* synched contract *if it is polarised, connected, and deterministic.*

Contracts are a slight generalisation of *session behaviours* [3], often presented as terms over the following grammar:

$$\mathcal{C} ::= \mathsf{end} \mid C; C \mid \oplus_{i=1,\ldots,n} !a_i; \mathcal{C}_i \mid \&_{i=1,\ldots,n} ?a_i; \mathcal{C}_i \mid \mathcal{X} \mid \mathsf{rec}\ \mathcal{X}\ \mathsf{in}\ \mathcal{C}$$

We will use this notation for concisely describing a contract, sometimes omitting trailing end. Note that session behaviours are contracts for which the final states are the terminal states. The semantics of $\mathcal{C}$ is that of the dialogue $\mathcal{C} \parallel \overline{\mathcal{C}}$. Unlike arbitrary dialogues, this semantics is very regular and thus quite easy to analyse because, at any time, at least one of the two buffers is empty, hence there is always one communicator that "follows" the other. Let us recall some facts already mentioned in the literature [15,21,24,14,5].

Proposition 1. *Let $\mathcal{C}$ be a synched contract. For $\mathcal{C} \parallel \overline{\mathcal{C}}$ in the FIFO semantics, the following properties are true:*

1. *$\mathcal{C} \parallel \overline{\mathcal{C}}$ is reception-safe;*
2. *$\mathcal{C}$ is realisable;*
3. *$\mathcal{C} \parallel \overline{\mathcal{C}}$ is synchronous;*
4. *Post^* is regular and effective;*
5. *$\mathcal{C} \parallel \overline{\mathcal{C}}$ is k-bounded if and only if all states are k-bounded;*
6. *if $\mathcal{C}$ is a session behaviour, $\mathcal{C} \parallel \overline{\mathcal{C}}$ is orphan-free.*

Properties 1, 4, 5 and 6 are of practical interest, and point out the gap between contracts and general dialogues: for the latter, they would be undecidable (see Thm. 1). Finding a good notion that generalises the contracts and justifies the determinacy and polarisation conditions first lead us to consider already discovered properties: reception-safety in the context of semantic subtyping [3], Stengel and Bultan's notion of realizability [21], and synchronism [9]. A first observation is that these properties are pairwise incomparable, and all strictly generalise the notion of synched contracts (see Fig. 1). A second observation is that they are not satisfied by synched contracts for non-FIFO communications: the synched contract $!a; !b \oplus !b; ?c$ is neither realisable, nor synchronous, nor reception-safe if communications are lossy or out-of-order.

We claim that a good candidate for being a "fundamental" property should satisfy two conditions: (1) be a decidable property, and (2) for dialogues with this property, other properties can be checked efficiently (*e.g.* boundedness can be checked efficiently for synched contracts simply by making sure that every loop in the contracts contains at

	det	pol	both
FIFO	U	U	Yes
lossy	NPR	NPR	NPR

(a) Has the contract only safe receptions?

	det	pol	both
FIFO	$U^{(1)}$	$U^{(2)}$	Yes
lossy	NPR	NPR	$NPR^{(3)}$

(b) Is the contract with safe receptions orphan-free?

Fig. 2: Complexity results for relaxed hypotheses. "det" = deterministic; "pol" = polarised; "U" = undecidable; "Yes" = always true; "NPR" = non-primitive recursive.

least one send and one receive, although synched contracts do not entail boundedness directly). Despite existing works [2,4] on related but slightly different properties, making such a call for synchronism or realizability has never been done, and it is rather unclear whether these notions satisfy requirements (1) and (2). The third candidate notion, reception-safety, is known to be undecidable for arbitrary dialogues, but Prop. 1 suggests that it could be decidable for dualised dialogues, for instance if the communicators are "almost" synched contracts. Similarly, as reception-safety plays an important role in subtyping and in the foundations of duality [3], it could be expected that properties such as boundedness or orphan-freedom are decidable for "almost synched contracts" that are reception-free. This is not the case, as the following theorem shows.

Theorem 1. *Reception-safe dialogues $\mathcal{M} \parallel \overline{\mathcal{M}}$ form neither an effective class (Fig. 2(a)), nor a class over which orphan freedom is decidable, even if $\mathcal{M}$ is assumed to be deterministic (resp. polarised), nor a tractable class if $\mathcal{M}$ is a synched contract but communications are lossy (Fig. 2(b)).*

Proof. Let us give a proof sketch for the problems marked (1), (2) and (3) in Fig. 2(b). The others are simple variants. First, let us observe that the reception-safety assumption can be lifted: a communicator C can always be extended with a sink node handling unspecified receptions without changing reachability issues (for instance, $!a; ?b; \mathsf{end}$ would be completed into $!a; (?b; \mathsf{end} \& ?a; \mathcal{C}_{\mathsf{sink}}) \oplus !b; \mathcal{C}_{\mathsf{sink}}$, where $\mathcal{C}_{\mathsf{sink}} \coloneqq (!a \oplus !b); \mathcal{C}_{\mathsf{sink}}$). Let us assume without loss of generality a synched contract $\mathcal{C}$ with a single final state and reduce the problem to the reachability of its final state in a stable configuration in the monologue semantics (which, based on standard results [7], is undecidable). Let FW denote a forwarder communicator that stops when receiving a special message $stop$, *i.e*: $\mathsf{FW} \coloneqq \big(\&_{a \in \Sigma} ?a; !a; \mathsf{FW}\big) \& ?stop; \mathsf{end}$. We then consider the following communicators, respectively deterministic for (1), polarised for (2) and both for (3), where $+$ denotes either a non-deterministic or a non-polarised choice:

$$\mathcal{C}_1 \coloneqq (!a; ?b; \mathcal{C}; !stop; ?sync; !leak) + (?b; !a; \overline{\mathsf{FW}}; ?sync)$$
$$\mathcal{C}_2 \coloneqq (!a; \mathcal{C}; !stop; ?sync; !leak) + (!a; \overline{\mathsf{FW}}; ?sync)$$
$$\mathcal{C}_3 \coloneqq (!lost; \mathcal{C}; !stop; ?sync; !leak) \oplus (!lost'; \overline{\mathsf{FW}}; ?sync)$$

Then $\mathcal{C}_1$ and $\mathcal{C}_2$ (resp. $\mathcal{C}_3$) leak the message $leak$ in the dualised FIFO (resp. lossy) semantics if and only if $\mathcal{C}$ reaches a stable final state in the FIFO monologue semantics. □

3 Half-Duplex Dialogues

Half-duplex dialogues were introduced by Cécé and Finkel in the context of FIFO communications [9]; this notion captures an idea we informally mentioned for contract communications, namely that two communication buffers are never used at the same time. In other words, a dialogue is half-duplex if, at every moment, at most one of the communicators is allowed to send messages, like in a walkie-talkie conversation.

Definition 5 (Half-duplex property).

- *A configuration* (q_0, q_1, w_0, w_1) *is* half-duplex *if either* $w_0 = \epsilon$ *or* $w_1 = \epsilon$.
- *A dialogue* $\mathcal{D}$ *is* half-duplex *if all its reachable configurations are half-duplex.*
- *A communicator* $\mathcal{C}$ *is a* half-duplex contract *if* $\mathcal{C} \parallel \overline{\mathcal{C}}$ *is half-duplex.*

Example 2. The synched contract $\mathcal{C} \coloneqq !a; ?b; \mathcal{C}$ presented in Ex. 1 is half-duplex for the FIFO, lossy, out-of-order and corruption interferences, and any combination thereof. It is not half-duplex if a or b have duplication errors. Similarly, $\mathcal{U} \coloneqq (!a; !b; !c) \oplus (!b; ?c)$ is a synched contract, but it is not half-duplex in the lossy semantics as $\mathcal{U} \parallel \overline{\mathcal{U}}$ can reduce to $!c \parallel !c$.

The half-duplex property may be imposed by the communication medium (for instance, a bus or radio communications), or it may be a design choice for optimising the implementation of a communication channel. It applies a priori to any communication model, but it should be stressed that it only makes sense for those satisfying the integrity condition of interferences: if $\succeq$ does not satisfy the integrity axiom, half-duplex dialogues are communication-free dialogues. Similarly, the class of half-duplex dialogues is the one of unidirectional communications for the stuttering model if none of the letters is ensured to be duplication-free.

Any synched contract is a half-duplex contract for the FIFO semantics. However, as observed in Ex. 2, this is not true for unreliable communications. Moreover, even for FIFO communications, the converse is false: half-duplex contracts are not necessarily synched contracts. Despite these differences, half-duplex contracts are not very different in nature from synched contracts. For a communicator $\mathcal{C}$, we write $\det(\mathcal{C})$ for the communicator obtained by determinization of $\mathcal{C}$ as a finite state automaton.

Proposition 2. *A connected communicator* $\mathcal{C}$ *is a half-duplex contract in the FIFO semantics if and only if* $\det(\mathcal{C})$ *is a synched contract.*

If we move now to unreliable communications, the connection with the synched contracts becomes a bit looser, but still exhibit some remarkable similarities:

Proposition 3. *Let* $\mathcal{C}$ *be a half-duplex contract. Then the following holds:*

1. $\det(\mathcal{C})$ *is polarised;*
2. *if* $\succeq$ *is non-expanding, then* $\mathcal{C} \parallel \overline{\mathcal{C}}$ *is* k*-bounded if and only if all states are* k*-bounded.*

These results might explain and justify some of the aspects of contracts. First, the condition of polarisation seems a rather fundamental one, and is intimately related to the half-duplex condition. Second, the computation of the bound on the buffer's size is always extremely simple under the half-duplex hypothesis. Third, the synched hypothesis simplifies the check on polarisation, but it does not suffice to guarantee the half-duplex property for unreliable communications. To make a case for the half-duplex property rather than synched contracts, we need to address two issues:

- being half-duplex is a semantic notion hence might be complex to check, particularly in the light of the previous complexity results, whereas the syntactic synch property can be checked linearly in the number of transitions in the contract;
- it does not prevent unspecified receptions or orphan messages, as already observed in the introduction.

Before addressing these concerns, let us first identify which models of interferences support efficient decision procedures. We propose two notions of "reasonably simple" interference models: an interference model $\succeq$ is *regular* if it is axiomatized by the axioms of Def. 3 plus any subset of the lossy, corruption, and stuttering axiom. It is *semi-linear* if it is axiomatized by such a set of axioms and the out-of-order axiom. The crucial property of these interference models is that deciding the emptiness of $\uparrow L \cap \downarrow L'$ for regular languages L is in $\mathbf{P}$ for regular interferences, and in $\mathbf{NP}$ for semi-linear ones.

Theorem 2. *Let $\succeq$ be any fixed regular (resp. semi-linear) interference model. Then the following decision problem is in $\mathbf{P}$ (resp. $\mathbf{NP}$):*

Input *A dialogue $\mathcal{D}$.*
Problem *Is $\mathcal{D}$ half-duplex?*

Despite their semantic definition, half-duplex contracts are thus an effective subclass of communicators.

Theorem 3. *Let $\mathcal{D}$ be a half-duplex dialogue, and $\succeq$ a regular (resp. semi-linear) interference model. Then Post^* is regular (resp. semi-linear), and a representation of it is computable in polynomial-time (resp. in non-deterministic polynomial time).*

This result addresses the second concern: if $\succeq$ is a regular (resp. semi-linear) interference model then the problem of deciding whether a half-duplex dialogue satisfies a given regular safety property (*e.g.* safe receptions, or orphan-freedom) is in $\mathbf{P}$ (resp. $\mathbf{NP}$). Boundedness is moreover in $\mathbf{P}$ if either the communication model is non-expanding (by Prop. 3), or regular (by Thm. 3), and in $\mathbf{NP}$ for expanding, semi-linear interferences.

We only sketch the proof of these two results. First observe that the proof argument of Cécé and Finkel [9] does not scale to unreliable communications: it is not the case that all reachable stable configurations can be computed by considering synchronous executions. For instance, in the out-of-order semantics,

$$!a; !b; end \parallel ?b; ?a; end \rightarrow^* end \parallel end$$

but such a reduction is not possible in the synchronous semantics. The proofs of Thm. 2 and 3 are indeed based on a different observation: the set $\mathsf{Post}^*_{\mathsf{HD}}(\{\dot{\gamma}\})$ of configurations reached from $\dot{\gamma}$ by visiting half-duplex configurations corresponds to the disjunct

$$\bigcup_{\gamma \in \mathsf{Post}^*_{\mathsf{HD}}(\{\dot{\gamma}\}), \gamma \text{ stable}} \mathsf{Post}^*_{\mathsf{uni}}(\{\gamma\})$$

where $\mathsf{Post}^*_{\mathsf{uni}}(\{\gamma\})$ denotes the set of configurations that are reachable from γ by forbidding communications over one of the two queues. It can then be observed that the set of reachable stable configurations γ appearing in the disjunct is finite (since there are only finitely many stable configurations), but moreover computable in polynomial-time for order-preserving unreliable communications, and in non-deterministic polynomial-time for out-of-order unreliable communications. Finally, $\mathsf{Post}^*_{\mathsf{uni}}(\{\gamma\})$ is regular for order-preserving unreliable communications, and semi-linear for out-of-order unreliable communications.

4 LTL Model-Checking

It is sometimes desirable to express temporal properties about communications[3]; for instance, some works on subtyping introduce a liveness condition that is not automatically satisfied by contracts [19]. In this section, we introduce two notions of LTL model-checking against half-duplex dialogues, one based on traces of configurations, and the other on traces of actions. We show that the former is undecidable, even for contracts, whereas the later is decidable if only one kind of actions (either send or receive) is taken into consideration.

4.1 Traces of configurations

We consider formulas ϕ of LTL as those given by the following grammar:

$$P ::= \langle q, q' \rangle \mid \langle \rightarrow \rangle \mid \langle \leftarrow \rangle \qquad \phi ::= P \mid \phi \wedge \phi \mid \neg\phi \mid \mathsf{X}\,\phi \mid \phi \,\mathsf{U}\, \phi$$

where $q, q' \in Q$. LTL formulas express special properties on the runs of a dialogue: $\langle q, q' \rangle$ asserts that the first configuration of a trace is in control states (q, q'); $\langle \rightarrow \rangle$ (resp. $\langle \leftarrow \rangle$) asserts that the first (resp. the second) queue is empty, $\mathsf{X}\,\phi$ asserts that ϕ is true at the next step of the trace; $\phi' \,\mathsf{U}\, \phi$ that ϕ is true after some time, and meanwhile ϕ' holds. We say that $\mathcal{D}$ satisfies ϕ if for all traces $(\gamma_i)_{i \geq 0}$ of $\mathcal{D}$, $(\gamma_i)_{i \geq 0}$ satisfies ϕ. For instance, $\langle \leftarrow \rangle \,\mathsf{U}\, (\langle \rightarrow \rangle \wedge X(\mathsf{end}, \mathsf{end}))$ asserts that the first party always closes the conversation after replying once to the messages of the second party.

Theorem 4. *The decision problem:*

Input *A synched, half-duplex contract $\mathcal{C}$ with safe receptions.*

[3] Note however that temporal properties of contracts do not necessarily translate into the same properties for the programs they type, as the set of (projections of) runs of a program is in general a subset of the runs of its contract.

Input *A formula ϕ of LTL.*
Problem *Does $\mathcal{C} \parallel \overline{\mathcal{C}}$ satisfy ϕ?*

is undecidable if $\succeq$ is regular (resp. semi-linear).

The proof is by reduction of the model-checking problem for monologues, which is undecidable by standard results (for the lossy and lossy out-of-order semantics, by undecidability of visiting a control state infinitely often in lossy FIFO and lossy counter machines [20], and for out-of-order semantics by undecidability of reachability in Minsky machines –note that $\langle\leftarrow\rangle \wedge \langle\rightarrow\rangle$ encodes zero tests). The reduction of an instance $(\mathcal{M}, \phi)$ of the monologue model-checking problem to an instance $(\mathcal{C}, \phi')$ of the contract model-checking problem uses a very simple contract $\mathcal{C}$ that allows to send any message at any time, whereas the formula ϕ' is a conjunct $\phi_{\mathsf{sched}} \wedge \phi_0$ with ϕ_{sched} forcing a scheduling between the sender and the receiver that simulates the monologue $\mathcal{M}$, and ϕ_0 replicates ϕ up to this simulation. Note that atomic predicates in ϕ_{sched} do not talk about queue contents, thus the model-checking problem remains undecidable in the FIFO and lossy case if predicates P are restricted to control state observations $\langle q, q'\rangle$.

4.2 Traces of Actions

We now consider LTL formulas where the atomic predicates P in the previous grammar range over $\langle pid, \lambda\rangle \in \{0, 1\} \times \mathsf{Act}_\Sigma$ and interpret formulas over traces of actions. We say that a dialogue $\mathcal{D}$ satisfies a formula ϕ in the send (resp. receive, resp. send/receive) semantics if all traces of send actions (resp. receive actions, resp. all actions) satisfy ϕ. For instance, $\langle 0, !a\rangle \wedge \mathsf{X}\langle 0, !b\rangle$ asserts that in all executions, only 0 sends messages, which are one a followed by one b; in the send/receive semantics, it moreover asserts that 1 is not receiving these messages.

Theorem 5. *Let $\succeq$ be any fixed regular (resp. semi-linear) interference model. The decision problem:*

Input *A half-duplex dialogue $\mathcal{D}$, a formula ϕ.*
Problem *Does $\mathcal{D}$ satisfy ϕ in the send (resp. receive) semantics?*

are decidable. However, this problem is undecidable under the FIFO or lossy send/receive semantics, even for synched half-duplex contract dialogues with safe receptions.

The undecidability results come from a slight adaptation of the proof of Thm. 4. The decidability result for the send semantics is based on the following property:

Proposition 4. *Let $\mathcal{D}$ be a half-duplex dialogue. Then the set of traces of send actions is regular. It is moreover effective for a regular (resp. semi-linear) interference model.*

On the other hand, the set of traces of receive actions is not always regular. For instance, the set of traces of receive actions of $(!a; !b)^* \parallel ?a^*; ?b^*$ for an out-of-order semantics is $\{(1, ?a)^n.(1, ?b)^n : n \geq 0\}$. For the communication models of Thm. 5, the set of traces of receive actions is however recognisable by Parikh automata, which keeps the model-checking problem decidable.

Conclusion

Related Work Channel contracts have been popularised by web services programming languages, and influenced *e.g.* the Sing# programming language. The first work formalising the semantics of Sing# contracts as communicating finite state machines, and the deterministic and polarised (aka autonomous) conditions they should satisfy was by Stengel and Bultan [21]. They focus on realizability of which they show that contracts are a special case in the FIFO semantics. Stengel and Bultan designed the TUNE model-checker for contracts conversations in Sing# (hence with bounded buffers) against LTL formulas, using the SPIN model-checker as a back-end.

Some of the properties we formalised in Prop. 1 are fairly old, and to the best of our knowledge can be traced back to the work of Gouda, Manning and Yu [15]. Cécé and Finkel first proved the theorems of Sec. 2 in the restricted case of FIFO communications; their proofs rely on the observation that all stable reachable configurations are reached in the synchronous semantics as well. As we have shown, the same argument does not hold for any interference model. Cécé and Finkel also showed the undecidability of LTL model-checking over traces of configurations for the FIFO semantics. Their proof technique is significantly more specialised than ours, and could not be used in our setting (the communicators they consider are not contracts, and the proof strongly relies on the FIFO semantics), and they do not consider traces of actions.

Some of the kinds of unreliable communications we consider, like lossiness, stuttering, or message corruption, have been introduced and studied by many authors (*e.g.* Abdullah and Johnson [1], Purushothaman et al. [10], Mayr [18]). In these works, no axiomatization of the interference model is proposed, as the decidability of the problems they consider depend on the particular choice of the communication model; in our case, on the contrary, we are able to provide a generic axiomatization that makes the proofs rather uniform.

It is worth noting that our framework is orthogonal to that of well-structured transition systems [13]: some of the preorders $\succeq$ that we consider are not well-quasi-orders, and our proofs do not rely on these techniques. Our method is rather based on the idea of regular model-checking [6,25].

Perspectives Our main goal was to recast the theory of contracts in the half-duplex framework, a novel contribution missed by the literature, and to emphasise the problems of effectivity while proposing a new notion of reliable contracts. We have argued that contract communications should be seen as exactly the dualised half-duplex communications; we observed that duality does not drastically reduce the complexity of the communications, which propose an interesting alternative to subtyping. A more refined analysis of the complexity of fixed properties over fixed communication models could however show some advantages of contracts over non-dual half-duplex communications.

Beside complexity issues, the practical relevance of the half-duplex hypothesis would probably merit a more experimental study. On the one hand, scaling this hypothesis to multipartite sessions, if possible, is not obvious, and our results do not cover interesting and useful parts of MSCs and multipartite session types. On the other hand, the half-duplex property could be a quite frequent one in message-passing programming, first of

all in MPI, where non half-duplex communications are often avoided as paving the way to head-to-head deadlocks.

References

1. P. A. Abdulla and B. Jonsson. Verifying programs with unreliable channels. *Inf. Comput.*, 1996.
2. R. Alur, K. Etessami, and M. Yannakakis. Inference of message sequence charts. In *Software Concepts and Tools*, 2003.
3. F. Barbanera and U. de'Liguoro. Two notions of sub-behaviour for session-based client/server systems. In *PPDP*, 2010.
4. S. Basu and T. Bultan. Choreography conformance via synchronizability. In *WWW*, 2011.
5. V. Bono, C. Messa, and L. Padovani. Typing Copyless Message Passing. In *ESOP*, vol. LNCS 6602, 2011.
6. A. Bouajjani, B. Jonsson, M. Nilsson, and T. Touili. Regular model checking. In *CAV*, 2000.
7. D. Brand and P. Zafiropulo. On communicating finite-state machines. *J. ACM*, 1983.
8. M. Bravetti and G. Zavattaro. Contract compliance and choreography conformance in the presence of message queues. In *WS-FM*, 2008.
9. G. Cécé and A. Finkel. Verification of programs with half-duplex communication. *Inf. Comput.*, 2005.
10. G. Cécé, A. Finkel, and S. Purushothaman Iyer. Unreliable channels are easier to verify than perfect channels. *Inf. Comput.*, Jan. 1996.
11. J. Esparza. Decidability and complexity of Petri net problems - an introduction. In *Petri Nets*, 1996.
12. M. Fähndrich, M. Aiken, C. Hawblitzel, O. Hodson, G. C. Hunt, J. R. Larus, and S. Levi. Language support for fast and reliable message-based communication in Singularity OS. In *EuroSys*. 2006.
13. A. Finkel and Ph. Schnoebelen. Well-structured transition systems everywhere! *Theor. Comput. Sci.*, Apr. 2001.
14. S. J. Gay and V. T. Vasconcelos. Linear type theory for asynchronous session types. *J. Funct. Program.*, 2010.
15. M. G. Gouda, E. G. Manning, and Y.-T. Yu. On the progress of communications between two finite state machines. *Information and Control*, 1984.
16. K. Honda, V. T. Vasconcelos, and M. Kubo. Language primitives and type discipline for structured communication-based programming. In *ESOP*, vol. 1381 of *lncs*. 1998.
17. K. Honda, N. Yoshida, and M. Carbone. Multiparty asynchronous session types. In *POPL*. 2008.
18. R. Mayr. Undecidable problems in unreliable computations. *Theor. Comput. Sci.*, 2003.
19. L. Padovani. Fair subtyping for multiparty session types. In *COORDINATION*, vol. 6721 of *LNCS*, 2011.
20. P. Schnoebelen. Lossy counter machines decidability cheat sheet. In *RP*, vol. 6227 of *LNCS*. 2010.
21. Z. Stengel and T. Bultan. Analyzing singularity channel contracts. In *ISSTA*, 2009.
22. K. Takeuchi, K. Honda, and M. Kubo. An interaction-based language and its typing system. In *PARLE*, vol. 817 of *lncs*. 1994.
23. J. Villard. *Heaps and Hops*. PhD Thesis, LSV, ENS Cachan, Feb. 2011.
24. J. Villard, É. Lozes, and C. Calcagno. Proving copyless message passing. In *APLAS*, vol. 5904 of *lncs*. 2009.
25. P. Wolper and B. Boigelot. Verifying systems with infinite but regular state spaces. In *CAV*, 1998.