Lossy Channel Games under Incomplete Information

Rayna Dimitrova
Saarland University, Germany
dimitrova@cs.uni-saarland.de

Bernd Finkbeiner
Saarland University, Germany
finkbeiner@cs.uni-saarland.de

In this paper we investigate lossy channel games under incomplete information, where two players operate on a finite set of unbounded FIFO channels and one player, representing a system component under consideration operates under incomplete information, while the other player, representing the component's environment is allowed to lose messages from the channels. We argue that these games are a suitable model for synthesis of communication protocols where processes communicate over unreliable channels. We show that in the case of finite message alphabets, games with safety and reachability winning conditions are decidable and finite-state observation-based strategies for the component can be effectively computed. Undecidability for (weak) parity objectives follows from the undecidability of (weak) parity perfect information games where only one player can lose messages.

1 Introduction

Lossy channel systems (LCSs), which are finite systems communicating via unbounded lossy FIFO channels, are used to model communication protocols such as link protocols, a canonical example of which is the Alternating Bit Protocol. The decidability of verification problems for LCSs has been well studied and a large number of works have been devoted to developing automatic analysis techniques. In the control and synthesis setting, where games are the natural computational model, this class of systems has not yet been so well investigated. In [1], Abdulla et al. establish decidability of two-player safety and reachability games where one (or both) player has downward-closed behavior (e.g., can lose messages), which subsumes games with lossy channels where one player (i.e., the environment) can lose messages. They, however, assume that the game is played under perfect information, which assumption disregards the fact that a process has no access to the local states of other processes or that it has only limited information about the contents of the channels. To the best of our knowledge, games under incomplete information where the players operate on unbounded unreliable channels have not been studied so far.

We define *lossy channel games under incomplete information* and show that in the case of finite message alphabets, games with safety and reachability winning conditions are decidable and finite-state observation-based strategies for the player who has incomplete information can be effectively computed.

Algorithms for games under incomplete information carrying out an explicit knowledge based subset construction [9] are not directly applicable to infinite-state games. Symbolic approaches [4] are effective for restricted classes of infinite-state games like discrete games on rectangular automata [5]. The symbolic algorithms that we present in this paper rely on the monotonicity of lossy channel systems w.r.t. the subword ordering, which is a well-quasi ordering (WQO). It is well known that upward and downward-closed sets of words used in the analysis of lossy channel systems can be effectively represented by finite sets of minimal elements and simple regular expressions [2], respectively. Unsurprisingly, the procedures for solving lossy channel games under incomplete information that we develop manipulate sets of sets of states. Thus, our termination arguments rely on the fact that the subword ordering is in fact a better-quasi ordering (BQO) [7, 8], a stronger notion than WQO that is preserved by the powerset operation [6].

F. Mogavero, A. Murano, and M.Y. Vardi (Eds.):
1st Workshop on Strategic Reasoning 2013 (SR'13)
EPTCS 112, 2013, pp. 43–51, doi:10.4204/EPTCS.112.9

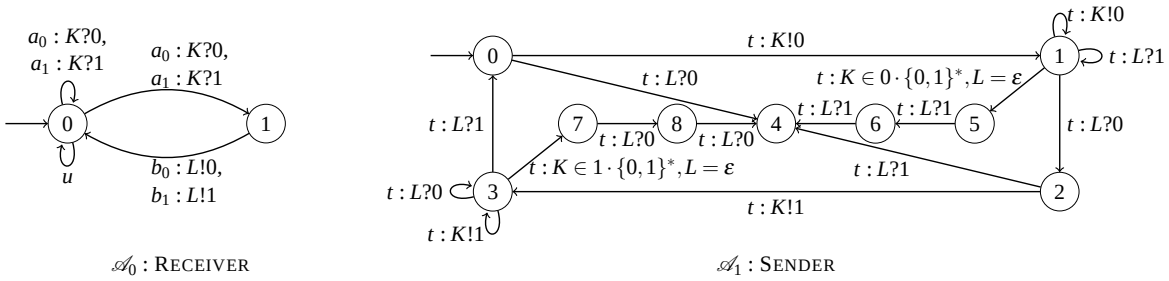

Figure 1: A communication protocol with partially specified RECEIVER process. For process RECEIVER we have $\Sigma_0 = \{a_0, a_1, b_0, b_1, u\}$ and $\Sigma_\exists = \{b_0, b_1\}$. The property that the implementation must satisfy is that location 4 in SENDER is not reachable, i.e., the receiver does not acknowledge messages that have not been sent, and once all messages and acknowledgements from previous phases have been consumed, the receiver can only send one delayed acknowledgement. Note that by using an extra channel and an extra location in process RECEIVER we can ensure that the error location is in process RECEIVER.

2 Lossy Channel Games under Incomplete Information

Lossy channel systems are asynchronous distributed systems composed of finitely many finite-state processes communicating through a finite set of unbounded FIFO channels that can nondeterministically lose messages. We consider *partially specified lossy channel systems*, where the term partially specified refers to the fact that we consider a second ("friendly") type of nondeterminism, in addition to the ("hostile") one due to the model. More specifically, this second type of nondeterminism models *unresolved implementation decisions* that can be resolved in a favorable way. We consider the case when these decisions are within a single process, and thus we can w.l.o.g. assume that the system consist of only two processes: the process under consideration and the parallel composition of the remaining processes.

Definition 1. A *partially specified lossy channel system (LCS)* is a tuple $\mathscr{L} = (\mathscr{A}_0, \mathscr{A}_1, C, M, \Sigma_0, \Sigma_1, \Sigma_\exists)$, where for each *process identifier* $p \in \{0,1\}$, $\mathscr{A}_p$ is a finite automaton describing the behavior of process p, C is a finite set of *channels*, M is a finite set of *messages*, $\Sigma = \Sigma_0 \dot\cup \Sigma_1$ is the union of the disjoint finite sets of *transition labels* for the two processes, and $\Sigma_\exists \subseteq \Sigma_0$ is a subset of the labels of the *partially specified process* $\mathscr{A}_0$. The automaton $\mathscr{A}_p = (Q_p, q_p^0, \delta_p)$ for a process p consists of a finite set Q_p of *control locations*, an *initial location* q_p^0 and a finite set δ of *transitions* of the form (q, a, Gr, Op, q'), where $q, q' \in Q_p$, $a \in \Sigma_p$, $Gr : C \to \{true, (= \varepsilon), \in (m \cdot M^*) \mid m \in M\}$ and $Op : C \to \{!m, ?m, nop \mid m \in M\}$. Intuitively, the function Gr maps each channel to a guard, which can be an emptiness test, a test of the letter at the head of the channel or the trivial guard *true*. The function Op gives the update operation for the respective channel, which is either a write, a read or *nop*, which leaves the channel unchanged.

Example. Fig.1 depicts a partially specified protocol consisting of two processes, SENDER and RECEIVER, communicating over the unreliable channels K and L. Process SENDER sends messages to RECEIVER over channel K and RECEIVER acknowledges the receipt of a message using channel L. Note that we use guards that test channels for emptiness or test the first letter of their contents.

The two processes are represented as nondeterministic finite-state automata. Process SENDER essentially runs the Alternating Bit Protocol. Process RECEIVER, however, is only *partially specified*: its alphabet of transition labels $\Sigma_0 = \{a_0, a_1, b_0, b_1, u\}$ is partitioned according to the unresolved decisions in the process specification: The subset $\Sigma_\exists = \{b_0, b_1\}$ of controllable transition labels specifies the unresolved implementation decisions, namely what bit to be sent on channel L at location 1.

The property that the protocol must satisfy is encoded as the unreachability of location 4 in process SENDER. However, the automata can easily be augmented (with an extra channel and an error location in process RECEIVER) in a way that the error location is in process RECEIVER. The property states that:

1. the receiver does not acknowledge messages that have not been sent, that is in location 2 in SENDER the language of L is 0^* and in location 0 in SENDER the language of L is 1^*,
2. once all messages and acknowledgements trailing from previous phases have been consumed (or lost), the number of delayed acknowledgements the receiver can send is bounded by one.

A *configuration* $\gamma = (q_0, q_1, w)$ of $\mathscr{L}$ is a tuple of the locations of the two processes and a function $w : C \to M^*$ that maps each channel to its contents. The *initial configuration* of $\mathscr{L}$ is $\gamma^0 = (q_0^0, q_1^0, \varepsilon)$, where $\varepsilon(c) = \varepsilon$ for each $c \in C$. The set of possible channel valuations is $W = \{w \mid w : C \to M^*\}$.

The *strong labeled transition relation* $\to \subseteq (Q_0 \times Q_1 \times W) \times \Sigma \times (Q_0 \times Q_1 \times W)$ of $\mathscr{L}$ consists of all tuples $((q_0, q_1, w), a, (q_0', q_1', w'))$ (denoted $(q_0, q_1, w) \xrightarrow{a} (q_0', q_1', w')$) such that if $a \in \Sigma_p$, then $q_{1-p}' = q_{1-p}$ and there is a transition $(q_p, a, Gr, Op, q_p') \in \delta$ such that for each $c \in C$ all of the following conditions hold: (1) if $Gr(c) = (\in m \cdot M^*)$ then $w(c) \in m \cdot M^*$, (2) if $Gr(c) = (= \varepsilon)$ then $w(c) = \varepsilon$, (3) if $Op(c) = !m$, then $w'(c) = w(c) \cdot m$, (4) if $Op(c) = ?m$, then $m \cdot w'(c) = w(c)$, and (5) if $Op(c) = nop$, then $w'(c) = w(c)$.

Let $\preceq$ denote the (not necessarily contiguous) subword relation on M^* and let us define its extension to elements of W as follows: $w_1 \preceq w_2$ for $w_1, w_2 \in W$ iff $w_1(c) \preceq w_2(c)$ for every $c \in C$.

The *weak labeled transition relation* $\Rightarrow \subseteq (Q_0 \times Q_1 \times W) \times \Sigma \times (Q_0 \times Q_1 \times W)$ for $\mathscr{L}$ is defined as follows: $(q_0, q_1, w) \stackrel{a}{\Rightarrow} (q_0', q_1', w')$ iff there exist w_1 and w_2 such that $w_1 \preceq w$ and $w' \preceq w_2$ and $(q_0, q_1, w_1) \xrightarrow{a} (q_0', q_1', w_2)$, i.e., the channels can lose messages before and after the actual transition.

Definition 2 (LC-game structure with incomplete information)**.** Let $\mathscr{L} = (\mathscr{A}_0, \mathscr{A}_1, C, M, \Sigma_0, \Sigma_1, \Sigma_\exists)$ be a partially specified LCS, and $C_{obs} \subseteq C$ be a set of *observable channels* that includes the set of all channels occurring in guards or read operations in $\mathscr{A}_0$. The *lossy channel game structure with incomplete information* for $\mathscr{L}$ and C_{obs} is $\mathscr{G}(\mathscr{L}, C_{obs}) = (S, I, \to_g, C, M, \Sigma_0, \Sigma_1, \Sigma_\exists, C_{obs})$, where:

- The set of *states* of $\mathscr{G}$ is $S = \{0,1\} \times Q_0 \times Q_1 \times W$. The first component p of a state (p, q_0, q_1, w) identifies the process to be executed and the remaining ones encode the current configuration of $\mathscr{L}$. The set of initial states of $\mathscr{G}$ is $I = \{(p, q_0, q_1, w) \mid p \in \{0,1\},\ q_0 = q_0^0,\ q_1 = q_1^0,\ w = \varepsilon\}$.
- The labeled transition relations $\to_g \subseteq S \times \Sigma \times S$ and $\Rightarrow_g \subseteq S \times \Sigma \times S$ of $\mathscr{G}$ are defined as follows: for states $s = (p, q_0, q_1, w)$ and $s' = (p', q_0', q_1', w')$ and $a \in \Sigma$ we have $s \xrightarrow{a}_g s'$ iff $a \in \Sigma_p$ and $(q_0, q_1, w) \xrightarrow{a} (q_0', q_1', w')$, and we have $s \stackrel{a}{\Rightarrow}_g s'$ iff $a \in \Sigma_p$ and $(q_0, q_1, w) \stackrel{a}{\Rightarrow} (q_0', q_1', w')$.

Remark. The first component of states in S is used to model the interleaving semantics and is updated nondeterministically in the transition relation $\to_g$ (and $\Rightarrow_g$). For simplicity, in Definition 2 we do not make any assumptions about the nondeterministic choice of which process to be executed. One natural assumption one might want to make is that the selected process must have at least one transition enabled in the current state. This and other restrictions can be easily imposed in the above model.

For the rest of the paper, let $\mathscr{G} = \mathscr{G}(\mathscr{L}, C_{obs}) = (S, I, \to_g, C, M, \Sigma_0, \Sigma_1, \Sigma_\exists, C_{obs})$ be the LC-game structure with incomplete information for a partially specified LCS $\mathscr{L}$ and observable channels C_{obs}.

Player$_\exists$ plays the game under incomplete information, observing only certain components of the current state of the game. Let $H_{obs} = C_{obs} \to (M \cup \{\varepsilon\})$ and $Obs = \{0,1\} \times Q_0 \times H_{obs}$. The *observation function* $obs : S \to Obs$ maps each state $s = (p, q_0, q_1, w)$ in $\mathscr{G}$ to the tuple $obs(s) = (p, q_0, h)$ of state components observed by *Player*$_\exists$, where for each $c \in C_{obs}$, if $p = 1$, then $h(c) = \varepsilon$ and otherwise if $w(c) = \varepsilon$, then $h(c) = \varepsilon$ and if $w(c) = m \cdot w'$ for some $m \in M$ and $w' \in M^*$, then $h(c) = m$. That is, when

$p = 0$ we have for $c \in C_{obs}$ that $h(c)$ is the letter at the head of $w(c)$, when c is not empty. For $o \in Obs$, we denote with $States(o) = \{s \in S \mid obs(s) = o\}$ the set of states whose observation is o.

Let $S_0 = \{(p, q_0, q_1, w) \in S \mid p = 0\}$ be the states where process 0 is to be executed and $S_1 = S \setminus S_0$.

The game $\mathscr{G}$ is played by $Player_\exists$ and $Player_\forall$ who build up a play $s_0 a_0^\exists a_0 s_1 a_1^\exists a_1 \ldots$, which is sequence of alternating states in S, labels in $\Sigma_\exists^\perp = \Sigma_\exists \cup \{\perp\}$ and labels in Σ, starting with a state $s_0 \in I$. Each time the current state is in S_0, $Player_\exists$ has to choose a label from the set $\Sigma_\exists \cup \{\perp\}$, that is either a label from $\Sigma_\exists$ of a transition enabled in the current state, or can be the special element $\perp$ in case no transition with label in $\Sigma_\exists$ is enabled or if there exists an enabled transition with label from $\Sigma_0 \setminus \Sigma_\exists$.

Let $\mathsf{Enabled}(s) = \{a \in \Sigma_0 \mid \exists s'.\ s \xrightarrow{a}_g s'\}$. Note that for states $s_1, s_2 \in S_0$ with $obs(s_1) = obs(s_2) = o$ it holds that $\mathsf{Enabled}(s_1) = \mathsf{Enabled}(s_2)$, and, abusing notation, we denote this set with $\mathsf{Enabled}(o)$.

For an observation $o = (0, q_0, h)$, the set $Act_\exists(o) = (\mathsf{Enabled}(o) \cap \Sigma_\exists) \cup \{\perp \mid \mathsf{Enabled}(o) \cap \Sigma_\exists = \emptyset \text{ or } \mathsf{Enabled}(o) \cap (\Sigma_0 \setminus \Sigma_\exists) \neq \emptyset\}$ consists of the transition labels that $Player_\exists$ can choose in a set $s \in S_0$ with $obs(s) = o$. For a label $a^\exists \in \Sigma_\exists^\perp$, the set $Act_\forall(o, a^\exists) = (\{a^\exists\} \cap \Sigma_\exists) \cup (\mathsf{Enabled}(o) \setminus \Sigma_\exists)$ consists of the transition labels which $Player_\forall$ can choose when the current choice of $Player_\exists$ is $a^\exists$.

The play is built by $Player_\forall$ respecting the choices of $Player_\exists$ and the transition relation $\Rightarrow_g$. When $s_i \in S_0$, then $a_i^\exists \in Act_\exists(obs(s_i))$ is the transition label chosen by $Player_\exists$ after the play prefix $s_0 a_0^\exists a_0 s_1 a_1^\exists a_1 \ldots a_{i-1}^\exists a_{i-1} s_i$ and $a_i \in Act_\forall(obs(s_i), a_i^\exists)$. After $Player_\exists$ has made his choice, $Player_\forall$ resolves the remaining nondeterminism by choosing a_i and the successor state s_{i+1} to extend the play.

A *play* in $\mathscr{G}$ is a sequence $\pi = s_0 a_0^\exists a_0 s_1 a_1^\exists a_1 s_1 \ldots \in (S \cdot (\Sigma_\exists^\perp \cdot \Sigma \cdot S)^* \cup S \cdot (\Sigma_\exists^\perp \cdot \Sigma \cdot S)^\omega)$ such that $s_0 \in I$, for every $i \geq 0$ it holds that $s_i \overset{a_i}{\Rightarrow}_g s_{i+1}$, and if $s_i \in S_1$, then $a^\exists = \perp$, and if $s_i \in S_0$ then $a_i^\exists \in Act_\exists(obs(s_i))$ and $a_i \in Act_\forall(obs(s_i), a_i^\exists)$. A play π is finite iff $\mathsf{last}(\pi)$ has no successor in $\mathscr{G}$, where $\mathsf{last}(\pi) \in S$ is the last element of π. The set $\mathsf{Prefs}(\mathscr{G}) \subseteq S \cdot (\Sigma_\exists^\perp \cdot \Sigma \cdot S)^*$ consists of the finite prefixes of plays in $\mathscr{G}$, and we denote with $\mathsf{Prefs}_\exists(\mathscr{G}) = \{\pi \in \mathsf{Prefs}(\mathscr{G}) \mid \mathsf{last}(\pi) \in S_0\}$ the set of prefixes ending in S_0.

A *strategy for* $Player_\exists$ is a total function $f_\exists : \mathsf{Prefs}_\exists(\mathscr{G}) \to \Sigma_\exists^\perp$ such that $f_\exists(\pi) \in Act_\exists(obs(\mathsf{last}(\pi)))$.

The *outcome of a strategy* $f_\exists$ is the set of plays $\mathsf{Outcome}(f_\exists)$ such that $\pi = s_0 a_0^\exists a_0 s_1 a_1^\exists a_1 \ldots \in \mathsf{Outcome}(f_\exists)$ iff for every $i \geq 0$ with $s_i \in S_0$ it holds that $a_i^\exists = f_\exists(s_0 a_0^\exists a_0 s_1 a_1^\exists a_1 \ldots s_i)$.

We define a function $obs^+ : \mathsf{Prefs}_\exists(\mathscr{G}) \to (Obs \cdot \Sigma_0)^* \cdot Obs$ that maps a prefix in $\mathsf{Prefs}_\exists(\mathscr{G})$ to the sequence of state and action observations made by $Player_\exists$: $obs^+(s_0 a_0^\exists a_0 s_1 a_1^\exists a_1 \ldots s_n) = obs'(s_0) \cdot obs'(a_0) \cdot obs'(s_1) \cdot obs'(a_1) \ldots \cdot obs'(s_n)$, where for $s \in S$, we define $obs'(s) = obs(s)$ if $s \in S_0$ and $obs'(s) = \varepsilon$ otherwise, and for $a \in \Sigma$ we define $obs'(a) = a$ if $a \in \Sigma_0$ and $obs'(a) = \varepsilon$ otherwise.

We call a strategy $f_\exists$ for $Player_\exists$ obs^+*-consistent* if for every pair of prefixes π_1 and π_2 in $\mathsf{Prefs}_\exists(\mathscr{G})$ for which $obs^+(\pi_1) = obs^+(\pi_2)$ holds, it also holds that $f_\exists(\pi_1) = f_\exists(\pi_2)$.

We are interested in *finite-state* strategies for $Player_\exists$, that is, strategies that can be implemented as finite automata. A finite state obs^+-consistent strategy for $Player_\exists$ in $\mathscr{G}$ is one that can be represented as a finite automaton $\mathscr{M}_s = (Q_s, q_s^0, (Q_0 \times H_{obs}) \times (\Sigma_\exists^\perp \times \Sigma_0), \rho)$ with alphabet $(Q_0 \times H_{obs}) \times (\Sigma_\exists^\perp \times \Sigma_0)$, whose transition relation $\rho \subseteq (Q_s \times ((Q_0 \times H_{obs}) \times (\Sigma_\exists^\perp \times \Sigma_0)) \times Q_s)$ has the following properties:

- (i) for each $q \in Q_s$, $o \in Q_0 \times H_{obs}$, $a^\exists \in \Sigma_\exists^\perp$, $a \in \Sigma_0$, and $q_1', q_2' \in Q_s$, it holds that if $(q, (o, (a^\exists, a)), q_1') \in \rho$ and $(q, (o, (a^\exists, a)), q_2') \in \rho$, then $q_1' = q_2'$ (i.e., the transition relation ρ is deterministic),
- (ii) for each $q \in Q_s$ and $o \in Q_0 \times H_{obs}$ there exist $a^\exists \in \Sigma_\exists^\perp$, $a \in \Sigma_0$, $q' \in Q_s$ with $(q, (o, (a^\exists, a)), q') \in \rho$,
- (iii) if $(q, (o, (a^\exists, a_1)), q_1') \in \rho$ and $a_2 \in Act_\forall((0, o), a^\exists)$, then $(q, (o, (a^\exists, a_2)), q_2') \in \rho$ for some $q_2' \in Q_s$,
- (iv) if $(q, (o, (a_1^\exists, a_1)), q_1') \in \rho$ and $(q, (o, (a_2^\exists, a_2)), q_2') \in \rho$, then $a_1^\exists = a_2^\exists$.

The automaton $\mathscr{M}_s$ defines an obs^+-consistent strategy $f_\exists$ for $Player_\exists$. According to the properties of $\mathscr{M}_s$, for each $\pi \in \mathsf{Prefs}_\exists(\mathscr{G})$ with $obs^+(\pi) = o_0 a_0 o_1 a_1 \ldots o_{n-1} a_{n-1} o_n$ there exists a unique sequence $a_0^\exists a_1^\exists a_{n-1}^\exists \in {\Sigma_\exists^\perp}^n$ such that there is a run of $\mathscr{M}_s$ (also unique) on the word $o_0 a_0^\exists a_0 o_1 a_1^\exists a_1 \ldots o_{n-1} a_{n-1}^\exists a_{n-1}$.

Let q be the last state of this run. We then define $f_\exists(\pi) = a^\exists$, where $a^\exists \in \Sigma_\exists^\perp$ is the unique label that exists by conditions (ii) and (iv) such that there are $a \in \Sigma_0$ and $q \in Q_s$ such that $(q,(o_n,(a^\exists,a)),q') \in \rho$.

We now turn to the definition of winning conditions in LC-games under incomplete information. We consider *safety* and *reachability* winning conditions for $Player_\exists$ defined by visible sets of states in $\mathscr{G}$. A set $T \subseteq S$ is *visible* iff for every $s \in T$ and every $s' \in S$ with $obs(s') = obs(s)$ it holds that $s' \in T$.

A *safety LC-game under incomplete information* $\mathsf{Safety}(\mathscr{G}, Err)$ is defined by a LC-game structure with incomplete information $\mathscr{G}$ and a visible set Err of error states that $Player_\exists$ must avoid. A strategy $f_\exists$ for $Player_\exists$ is *winning* in $\mathsf{Safety}(\mathscr{G}, Err)$ iff no play in $\mathsf{Outcome}(f_\exists)$ visits a state in Err.

Note that according to this definition, $Player_\exists$ wins finite plays that do not reach an error state. If we want to ensure that plays reaching a state in $\mathscr{G}$ that corresponds to a deadlock in $\mathscr{L}$ are not winning for $Player_\exists$, we can easily achieve this by appropriately instrumenting $\mathscr{L}$ and Err.

A *reachability LC-game under incomplete information* $\mathsf{Reach}(\mathscr{G}, Goal)$ is defined by a LC-game structure with incomplete information $\mathscr{G}$ and a visible set $Goal$ of goal states that $Player_\exists$ must reach. A strategy $f_\exists$ for $Player_\exists$ is *winning* in $\mathsf{Reach}(\mathscr{G}, Goal)$ iff each play in $\mathsf{Outcome}(f_\exists)$ visits a state in $Goal$.

Remark. The definition of visible sets allows that $Err \cap S_1 \neq \emptyset$ and $Goal \cap S_1 \neq \emptyset$. Thus, our definition of visible objectives does not require that for each pair of plays π_1 and π_2 with $obs^+(\pi_1) = obs^+(\pi_2)$ (where obs^+ is defined for plays analogously to prefixes) it holds that $Player_\exists$ wins π_1 iff he wins π_2. For the algorithms, which we present in the next section, for solving safety and reachability LC-games under incomplete information, the objective for $Player_\exists$ does not have to satisfy this condition.

3 Algorithms for Solving Safety and Reachability Games

Better-Quasi Orderings. The subword ordering $\preceq$ on M^* is a WQO (and so is the ordering $\preceq$ on W defined earlier). That means, it is a reflexive and transitive relation such that for every infinite sequence $w_0, w_1, \ldots$ of elements of M^* there exist indices $0 \leq i < j$ such that $w_i \preceq w_j$.

The subword ordering (as well as other WQOs commonly used in verification) is in fact also a BQO, and so is the ordering on W. Hence they are preserved by the powerset operation. Here we omit the precise definition of BQOs since it is rather technical and it is not necessary for the presentation of our results. When needed, we recall its properties relevant for our arguments.

We extend $\preceq$ to a BQO $\preceq$ on the set S of states in $\mathscr{G}$ in the following way: for $s = (p, q_0, q_1, w) \in S$ and $s' = (p', q_0', q_1', w') \in S$, we have $s \preceq s'$ iff $p = p'$, $q_0 = q_0'$, $q_1 = q_1'$, $obs(s) = obs(s')$ and $w \preceq w'$.

A set $T \subseteq S$ is *upward-closed* (respectively *downward-closed*) iff for every $s \in T$ and every $s' \in S$ with $s \preceq s'$ (respectively $s' \preceq s$) it holds that $s' \in T$. The upward-closure of a set $T \subseteq S$ is $T\uparrow = \{s' \in S \mid \exists s.\ s \in T \text{ and } s \preceq s'\}$. For each upward (respectively downward) closed set $T \subseteq S$ and $o \in Obs$, the set $T' = \{s \in T \mid obs(s) = o\}$ is also upward (respectively downward) closed. We let $\mathscr{U}_{obs}(S) = \{u \subseteq S \mid u \neq \emptyset,\ u = u\uparrow \text{ and } \exists o \in Obs. \forall s \in u.\ obs(s) = o\}$ and for $u \in \mathscr{U}_{obs}(S)$ we define $obs(u)$ in the obvious way. The set $\mathscr{D}_{obs}(S)$ and $obs : \mathscr{D}_{obs}(S) \to Obs$ are defined analogously, requiring that the elements are downward-closed instead of upward-closed. $\mathscr{D}_{obs}^{\mathsf{fin}}(S)$ is the set of finite sets in $\mathscr{D}_{obs}(S)$.

The transition relation $\Rightarrow_g$ enjoys the following property: if $s \stackrel{a}{\Rightarrow}_g s'$ and $s \preceq s''$, then $s'' \stackrel{a}{\Rightarrow}_g s'$. Thus, the set of predecessors w.r.t. some $a \in \Sigma$ of any set of states is upward-closed. For LCSs the set of successors w.r.t. some $a \in \Sigma$ of any set of states is a downward-closed set.

Let $\mathsf{Pre} : \mathscr{P}(S) \times \Sigma \to \mathscr{P}(S)$ be the function defined as $\mathsf{Pre}(T,a) = \{s \in S \mid \exists s' \in T.\ s \stackrel{a}{\Rightarrow}_g s'\}$ and let $\mathsf{Post} : \mathscr{P}(S) \times \Sigma \to \mathscr{P}(S)$ be the function defined as $\mathsf{Post}(T,a) = \{s \in S \mid \exists s' \in T.\ s' \stackrel{a}{\Rightarrow}_g s\}$. As recalled above, for each $T \subseteq S$ and each $a \in \Sigma$, $\mathsf{Pre}(T,a)$ is upward-closed and $\mathsf{Post}(T,a)$ is downward-closed.

We define the functions $\mathsf{Pre}_0 : \mathscr{U}_{obs}(S) \times \Sigma_0 \to \mathscr{P}_{\mathsf{fin}}(\mathscr{U}_{obs}(S))$ and $\mathsf{Pre}_1 : \mathscr{U}_{obs}(S) \to \mathscr{P}_{\mathsf{fin}}(\mathscr{U}_{obs}(S))$ that map a set $u \in \mathscr{U}_{obs}(S)$ to a finite set of upward-closed sets that partition the respective set of predecessors of u according to the observations $Player_\exists$ makes. Formally, $\mathsf{Pre}_0(u,a) = \{u' \in \mathscr{U}_{obs}(S) \mid \exists o \in Obs.\ u' = \mathsf{Pre}(u,a) \cap States(o)\}$ and $\mathsf{Pre}_1(u) = \{u' \in \mathscr{U}_{obs}(S) \mid \exists o \in Obs.\ u' = (\bigcup_{a \in \Sigma_1} \mathsf{Pre}(u,a)) \cap States(o)\}$. Similarly, using the function Post above, we can define the successor functions $\mathsf{Post}_0 : \mathscr{D}_{obs}(S) \times \Sigma_0 \to \mathscr{P}_{\mathsf{fin}}(\mathscr{D}_{obs}(S))$ and $\mathsf{Post}_1 : \mathscr{D}_{obs}(S) \to \mathscr{P}_{\mathsf{fin}}(\mathscr{D}_{obs}(S))$. Since the transition relation of $\mathscr{G}$ has finite branching, if $d \in \mathscr{D}^{\mathsf{fin}}_{obs}(S)$ then $d' \in \mathscr{D}^{\mathsf{fin}}_{obs}(S)$ for $d' \in \mathsf{Post}_0(d,a)$ or $d' \in \mathsf{Post}_1(d)$.

When analyzing LCSs, upward-closed sets are typically represented by their *finite sets of minimal elements*, and downward-closed sets are represented by *simple regular expressions*. These representations can be extended to obtain finite representations of elements of $\mathscr{U}_{obs}(S)$ and $\mathscr{D}_{obs}(S)$. By the definition of $\preceq$ on S, each visible set of states is upward-closed, and hence, the sets Err and $Goal$ in safety and reachability games are finitely representable. In the rest, we assume that they are represented such a way.

Our termination arguments rely on the following property: For every BQO $\preceq$ on a set X, the superset relation $\supseteq$ is a BQO on the set of upward-closed sets in $\mathscr{P}(X)$ and the subset relation $\subseteq$ is a BQO on the set of downward-closed sets. This implies that $\supseteq$ is a BQO on $\mathscr{U}_{obs}(S)$ and that $\subseteq$ is a BQO on $\mathscr{D}_{obs}(S)$.

LC-games under incomplete information with safety objectives. We describe a decision procedure for safety LC-games under incomplete information which is based on a backward fixpoint computation.

Each step in the fixpoint computation corresponds to a step in the game, which is not necessarily observable by $Player_\exists$. Thus, this construction is correct w.r.t. $Player_\exists$ strategies that are $\widetilde{obs}$-consistent, where, intuitively, the function $\widetilde{obs}$ maps a prefix to a sequence that includes also the (trivial) observations of S_1 states, and $\widetilde{obs}$-consistency is defined analogously to obs^+-consistency. To avoid this problem, our algorithm performs the fixpoint computation on a LC-game structure with incomplete information $\widetilde{\mathscr{G}}$ obtained from $\mathscr{G}$ by adding an *idle transition* for process 1. This game structure has the following property: $Player_\exists$ has an obs^+-consistent winning strategy in the game $\mathsf{Safety}(\mathscr{G}, Err)$ iff $Player_\exists$ has an $\widetilde{obs}$-consistent winning strategy in $\mathsf{Safety}(\widetilde{\mathscr{G}}, Err)$, which yields correctness of the algorithm.

Formally, the function $\widetilde{obs} : \mathsf{Prefs}_\exists(\mathscr{G}) \to (Obs^* \cdot \Sigma_0)^* \cdot Obs$ is defined as: $\widetilde{obs}(s_0 a_0^\exists a_0 \dots s_n) = obs(s_0) \cdot obs'(a_0) \cdot \ldots \cdot obs(s_n)$. The game structure $\widetilde{\mathscr{G}}$ is the tuple $\widetilde{\mathscr{G}} = (S, I, \widetilde{\to}_g, C, M, \Sigma_0, \widetilde{\Sigma_1}, \Sigma_\exists, C_{obs})$ where $\widetilde{\Sigma_1} = \Sigma_1 \cup \{idle\}$ and $idle \notin \Sigma$, and $\widetilde{\to}_g = \to_g \cup \{((1,q_0,q_1,w), idle, (p',q_0,q_1,w)) \mid p' \in \{0,1\}\}$.

We define the set $\mathscr{L}(S)$ for S as $\mathscr{L}(S) = \{l \in \mathscr{P}_{\mathsf{fin}}(\mathscr{U}_{obs}(S)) \mid l \neq \emptyset \text{ and } \exists o \in Obs. \forall u \in l.\ obs(u) = o\}$ and define $obs(l)$ for each $l \in \mathscr{L}(S)$ in the obvious way. We provide a fixpoint-based algorithm that computes a set $B \subseteq \mathscr{L}(S)$ such that each $l \in B$ has the following property: if $K \subseteq S$ is the set of states that the game can be currently in according to $Player_\exists$'s knowledge and $K \cap u \neq \emptyset$ for every $u \in l$, then $Player_\exists$ cannot win when his knowledge is K. Considering the set I of initial states, if for some $l \in B$ it holds that $I \cap u \neq \emptyset$ for all $u \in l$, then $Player_\exists$ has no obs^+-consistent winning strategy in $\mathsf{Safety}(\mathscr{G}, Err)$.

Our procedure computes a sequence $B_0 \subseteq B_1 \subseteq B_2 \dots$ of finite subsets of $\mathscr{L}(S)$. The computation starts with the set $B_0 = \{\{Err \cap States(o)\} \mid o \in Obs\}$. For $i \geq 0$, we let $B_{i+1} = B_i \cup N_{i+1}$, where the set N_{i+1} of new elements is computed based on B_i and is the smallest set that contains each $l \in \mathscr{L}(S)$ which is such that $l \subseteq \bigcup_{l' \in B_i, u' \in l'} ((\bigcup_{a \in \Sigma_0} \mathsf{Pre}_0(u',a)) \cup \mathsf{Pre}_1(u'))$ and:

- if $l \in \mathscr{P}(\mathscr{P}(S_0))$ then for every possible choice $a^\exists \in Act_\exists(obs(l))$ of $Player_\exists$, there exist an action $a \in Act_\forall(obs(l), a^\exists)$ and $l' \in B_i$ such that for every $u' \in l'$ it holds that $\mathsf{Pre}_0(u',a) \cap l \neq \emptyset$,
- if $l \in \mathscr{P}(\mathscr{P}(S_1))$ then there exists $l' \in B_i$ such that for every $u' \in l'$ it holds that $\mathsf{Pre}_1(u') \cap l \neq \emptyset$.

The ordering $\sqsubseteq$ on $\mathscr{L}(S)$ is defined such that for $l, l' \in \mathscr{L}(S)$, we have $l \sqsubseteq l'$ iff for every $u \in l$ there exists a $u' \in l'$ such that $u \supseteq u'$. The ordering $\sqsubseteq$ is a BQO, since $\supseteq$ is a BQO on $\mathscr{U}_{obs}(S)$. Intuitively, if l belongs to the set of elements of $\mathscr{L}(S)$ in which $Player_\exists$ cannot win, so does every l' with $l \sqsubseteq l'$.

We say that the sequence $B_0, B_1, B_2 \ldots$ *converges at k* if $\mathsf{Min}(B_{k+1}) \subseteq \mathsf{Min}(B_k)$, where $\mathsf{Min}(B_i)$ is the set of minimal elements of B_i w.r.t. $\sqsubseteq$. This condition can be effectively checked, since each B_i is finite. We argue that there exists a $k \geq 0$ such that the sequence computed by the procedure described above converges at k (and hence the procedure will terminate).

Let $F_0, F_1, F_2, \ldots$ be the sequence of upward-closed elements of $\mathscr{P}(\mathscr{L}(S))$ where $F_i = B_i \uparrow$ for each $i \geq 0$. As $F_0, F_1, F_2 \ldots$ is a monotonically increasing sequence of upward-closed sets of elements of $\mathscr{L}(S)$, it must eventually stabilize, i.e., there is a $k \geq 0$ such that $F_{k+1} \subseteq F_k$. Thus, since $F_{i+1} \subseteq F_i$ if and only if $\mathsf{Min}(B_{i+1}) \subseteq \mathsf{Min}(B_i)$, the sequence $B_0, B_1, B_2 \ldots$ is guaranteed to converge at some $k \geq 0$.

Proposition 1. *Let $B = B_k$, where the sequence $B_0, B_1, B_2 \ldots$ converges at k. Then, Player$_\exists$ has an $\widetilde{obs}$-consistent winning strategy in $\mathsf{Safety}(\widetilde{\mathscr{G}}, Err)$ iff for every $l \in B$ there exists $u \in l$ with $u \cap I = \emptyset$.*

If Player$_\exists$ has an $\widetilde{obs}$-consistent winning strategy in $\mathsf{Safety}(\widetilde{\mathscr{G}}, Err)$, then Player$_\exists$ has a finite-state obs^+-consistent winning strategy in the original game $\mathsf{Safety}(\mathscr{G}, Err)$.

Proof Idea. A counterexample tree for $\mathsf{Safety}(\widetilde{\mathscr{G}}, Err)$ represents a witness for the fact that *Player*$_\exists$ does not have an $\widetilde{obs}$-consistent winning strategy in $\mathsf{Safety}(\widetilde{\mathscr{G}}, Err)$. It is a finite tree with nodes labeled with elements of $\mathscr{D}_{obs}(S)$. If there is a $l \in B$ such that $u \cap I \neq \emptyset$ for every $u \in l$, a counterexample tree can be constructed in a top-down manner. For the other direction we can show by induction on the depth of the existing counterexample trees that there exists a $l \in B$ such that $u \cap I \neq \emptyset$ for every $u \in l$.

For the case when *Player*$_\exists$ wins the game $\mathsf{Safety}(\widetilde{\mathscr{G}}, Err)$ we can construct a finite-state obs^+-consistent winning strategy for *Player*$_\exists$ in the game $\mathsf{Safety}(\mathscr{G}, Err)$ by using as states for the strategy automaton functions from observations to a finite set $\mathscr{V} \subseteq \mathscr{P}_{\mathsf{fin}}(\mathscr{P}(S))$ each of whose elements V preserves the invariant that for every $l \in B$ there exists a $u \in l$ such that $u \cap \bigcup_{v \in V} v = \emptyset$. □

LC-Games under incomplete information with reachability objectives. For reachability games we give a procedure based on forward exploration of the sets of states representing the knowledge of *Player*$_\exists$ about the current state of the game. Since *Player*$_\exists$ can only observe the heads the observable channels, his knowledge at each point of the play is a finite downward-closed set, element of $\mathscr{D}^{\mathsf{fin}}_{obs}(S)$. To update this knowledge we define functions $\mathsf{Post}^{obs}_0 : \mathscr{D}^{\mathsf{fin}}_{obs}(S) \times \Sigma_0 \to \mathscr{P}_{\mathsf{fin}}(\mathscr{D}^{\mathsf{fin}}_{obs}(S))$ and $\mathsf{Post}^{obs}_1 : \mathscr{D}^{\mathsf{fin}}_{obs}(S) \to \mathscr{P}_{\mathsf{fin}}(\mathscr{D}^{\mathsf{fin}}_{obs}(S))$ that map a set $d \in \mathscr{D}^{\mathsf{fin}}_{obs}(S)$ to a finite set of elements of $\mathscr{D}^{\mathsf{fin}}_{obs}(S)$, each of which is a set of states that *Player*$_\exists$ knows, according to his current observation, the game may be in after (a transition from Σ_0 and) a sequence of transitions from Σ_1. For each $d \in \mathscr{D}^{\mathsf{fin}}_{obs}(S)$ we have $d' \in \mathsf{Post}^{obs}_0(d,a)$ (respectively $d' \in \mathsf{Post}^{obs}_1(d)$) iff there exists a sequence $d_0, d_1, \ldots, d_n \in \mathscr{D}^{\mathsf{fin}}_{obs}(S)$ such that $d_0 \in \mathsf{Post}_0(d,a)$ (respectively $d_0 = d$), for every $1 \leq i \leq n$ it holds that $d_{i-1} \subseteq S_1$ and $d_i \in \mathsf{Post}_1(d_{i-1})$, and for every $0 \leq i < j < n$ it holds that $d_i \not\subseteq d_j$ and one of the following conditions is satisfied: (1) $d' \subseteq \mathit{Goal}$, $d' = d_0$ and $n = 0$ (i.e., $d' \subseteq \mathit{Goal} \cap S_1$), or (2) there exists a $1 \leq i < n$ such that $d_i \subseteq d_n$ and $d' = d_n$ (i.e., $d' \subseteq S_1$), or (3) $d' = \{(0, q'_0, q'_1, w') \mid (1, q'_0, q'_1, w') \in \bigcup_{i=0}^{n} d_i\}$ (i.e., $d' \subseteq S_0$).

We construct a finite set of trees rooted at the different possible knowledge sets for *Player*$_\exists$ at location q^0_o. The nodes of the trees are labeled with knowledge sets, i.e., with elements of $\mathscr{D}^{\mathsf{fin}}_{obs}(S)$. The edges are labeled wit pairs of transition labels, i.e., elements of $\Sigma^{\perp}_{\exists} \times \Sigma_0$, where the first element of a pair is a possible choice of *Player*$_\exists$ and the second element is a corresponding choice of *Player*$_\forall$.

Formally, the forward exploration procedure constructs a forest $\mathscr{T}$ in which the roots are labeled with the sets $\{(0, q^0_0, q^0_1, \varepsilon)\}$ and all the sets $d \in \mathsf{Post}^{obs}_1(\{(1, q^0_0, q^0_1, \varepsilon)\} \setminus \mathit{Goal})$. At each step of the construction an open leaf node n with label d is processed in the following way:

- If $d \subseteq \mathit{Goal}$, we close the node and do not expand further from this node.
- If $d \not\subseteq \mathit{Goal}$ and either $d \subseteq S_0$ and there exists an ancestor of n that is labeled with d' and such that $d' \subseteq d$, or $d \subseteq S_1$, we close the node and do not expand further from this node.

- Otherwise, we add the set of successors of n: for each $a^{\exists} \in Act_{\exists}(obs(d))$, each $a \in Act_{\forall}(obs(d), a^{\exists})$ and each $d' \in \mathsf{Post}_0^{obs}(d,a)$ we add exactly one successor n' labeled with d' and label the edge (n,n') with $(a^{\exists},a)$. The set of successors for $(a^{\exists},a)$ is denoted with $Children(n,a^{\exists},a)$.

The finite branching of the transition relation of $\mathscr{G}$ and the fact that $\subseteq$ is a BQO on $\mathscr{D}_{obs}^{\mathsf{fin}}(S)$ imply that each of the sets $\mathsf{Post}_0^{obs}(d,a)$ and $\mathsf{Post}_1^{obs}(d)$ can be effectively computed, the set of roots and the out-degree of each node are finite, and the above procedure terminates constructing a finite forest $\mathscr{T}$.

We label each node n in $\mathscr{T}$ with a boolean value $win(n)$. For a leaf node n with $d(n) \subseteq Goal$, we define $win(n) = true$ and for any other other leaf node n we define $win(n) = false$. The value of a non-leaf node is computed based on those of its children by interpreting the choices of $Player_{\exists}$ disjunctively and the choices of $Player_{\forall}$ conjunctively. Formally, for every non-leaf node n we define $win(n) = \bigvee_{a^{\exists} \in Act_{\exists}(obs(d(n)))} \bigwedge_{a \in Act_{\forall}(obs(d(n)),a^{\exists})} \bigwedge_{n' \in Children(n,a^{\exists},a)} win(n')$, where $d(n)$ is the set of states labeling n.

Proposition 2. *$Player_{\exists}$ has an obs^{+}-consistent winning strategy in $\mathsf{Reach}(\mathscr{G}, Goal)$ iff for every root n in $\mathscr{T}$ it holds that $win(n) = true$. If $Player_{\exists}$ has an obs^{+}-consistent winning strategy in $\mathsf{Reach}(\mathscr{G}, Goal)$, then he also has a finite state obs^{+}-consistent winning strategy in $\mathsf{Reach}(\mathscr{G}, Goal)$.*

Proof Idea. If all the roots are labeled with *true* we can construct a finite-state obs^{+}-consistent strategy winning for $Player_{\exists}$ in $\mathsf{Reach}(\mathscr{G}, Goal)$, by mapping each prefix in $\mathsf{Prefs}_{\exists}(\mathscr{G})$ to a label in $\Sigma_{\exists}^{\perp}$, determined by a corresponding path in $\mathscr{T}$ and a fixed successful choice at its last node, if such path and choice exist, or given an appropriate default value otherwise. For the other direction we suppose that some root is labeled with *false* and show that for any obs^{+}-consistent strategy $f_{\exists}$ for $Player_{\exists}$, we can use the tree to construct a play $\pi \in \mathsf{Outcome}(f_{\exists})$ that never visits a state in $Goal$. □

LC-games under incomplete information with parity objectives. We now turn to more general ω-regular visible objectives for $Player_{\exists}$ where the undecidability results established in [1] for perfect information lossy channel games in which only one player can lose messages, carry on to our setting.

A *visible priority function* $pr : Obs \to \{0,1,\ldots,n\}$ for natural number $n \in \mathbb{N}$ maps each observation to a non-negative integer priority. For an infinite play $\pi = s_0 a_0^{\exists} a_0 s_1 a_1^{\exists} a_1 \ldots$ we define $pr(\pi) = \min\{pr(o) \mid o \in InfObs(\pi)\}$, where $InfObs(\pi)$ is the set of observations that occur infinitely often in π, and define $wpr(\pi) = \min\{pr(obs(s_0)), pr(obs(s_1)), \ldots\}$. A *parity* (respectively *weak parity*) LC-game under incomplete information $Parity(\mathscr{G}, pr)$ (respectively $WeakParity(\mathscr{G}, pr)$) is defined by a LC-game structure with incomplete information $\mathscr{G}$ and a visible priority function pr. A strategy $f_{\exists}$ for $Player_{\exists}$ is *winning* in the parity game $Parity(\mathscr{G}, pr)$ (weak parity game $WeakParity(\mathscr{G}, pr)$) iff for every infinite play $\pi \in \mathsf{Outcome}(f_{\exists})$ it holds that $pr(\pi)$ is even (respectively $wpr(\pi)$ is even).

Proposition 3. *The weak parity game solving problem for LC-games under incomplete information, that is, given a weak parity LC-game under incomplete information $WeakParity(\mathscr{G}, pr)$ to determine whether there exists an obs^{+}-consistent winning strategy for $Player_{\exists}$ in $WeakParity(\mathscr{G}, pr)$, is undecidable.*

Proof Idea. In [1] it was shown that in the perfect information setting the weak parity problem for B-LCS games, which are games played on a finite set of channels in which player A has a weak parity objective and only player B is allowed to lose messages, is undecidable. Their proof (given for A-LCS games but easily transferable into a proof for B-LCS games) is based on a reduction from the infinite computation problem for transition systems based on lossy channel systems, which is undecidable [3].

We argue that this reduction can be adapted for our framework, with $Player_{\exists}$ in the role of player A and $Player_{\forall}$ in the role of player B. The fact that here $Player_{\exists}$ choses only transition labels and plays under incomplete information does not affect the proof for B-LCS games, since there player A just

follows passively, while player B simulates the original system. The values of the priority function used in [1] do not depend on the contents of the channels. Thus, we can define a visible priority function. □

As a consequence, the parity game solving problem for LC-games under incomplete information is undecidable as well. As noted in [1], the construction from the proposition above can be used to show undecidability of A-LCS and B-LCS games with Büchi and co-Büchi objectives.

Summary of the results. The results of the paper are summarized in the following theorem.

Theorem 1. *For lossy channel game structures with incomplete information*

- *games with visible safety or reachability objectives for Player$_\exists$ are decidable, and when Player$_\exists$ has an observation-based winning strategy, a finite-state such strategy can be effectively computed,*
- *games with visible weak parity objectives for Player$_\exists$ are undecidable.*

4 Conclusion

We showed that the game solving problem for LC-games under incomplete information with safety or reachability objective for *Player*$_\exists$ is decidable. LC-games under incomplete information with more general winning conditions, such as weak parity (as well as Büchi and co-Büchi) condition can easily be shown to be undecidable, using a reduction similar to the one described in [1] for A-LCS games (which are perfect information games defined on LCSs in which only one player can lose channel messages). An orthogonal extension that is also clearly undecidable is decentralized control. This implies that suitable abstraction techniques are needed to address the synthesis problem within these undecidable settings.

Acknowledgements This work is partially supported by the DFG as part of SFB/TR 14 AVACS.

References

[1] Parosh Aziz Abdulla, Ahmed Bouajjani & Julien d'Orso (2008): *Monotonic and Downward Closed Games*. *J. Log. Comput.* 18(1), pp. 153–169, doi:10.1093/logcom/exm062.

[2] Parosh Aziz Abdulla, Aurore Collomb-Annichini, Ahmed Bouajjani & Bengt Jonsson (2004): *Using Forward Reachability Analysis for Verification of Lossy Channel Systems*. *FMSD* 25(1), pp. 39–65, doi:10.1023/B:FORM.0000033962.51898.1a.

[3] Parosh Aziz Abdulla & Bengt Jonsson (1996): *Undecidable Verification Problems for Programs with Unreliable Channels*. *Inf. Comput.* 130(1), pp. 71–90, doi:10.1006/inco.1996.0083.

[4] K. Chatterjee, L. Doyen, T. A. Henzinger & J.-F. Raskin (2006): *Algorithms for Omega-Regular Games with Imperfect Information*. In: *Proc. CSL'06, LNCS* 4207, Springer, pp. 287–302, doi:10.1007/11874683_19.

[5] M. De Wulf, L. Doyen & J.-F. Raskin (2006): *A Lattice Theory for Solving Games of Imperfect Information*. In: *Proc. HSCC'06, LNCS* 3927, Springer, pp. 153–168, doi:10.1007/11730637_14.

[6] Alberto Marcone (2001): *Fine Analysis of the Quasi-Orderings on the Power Set*. *Order* 18(4), pp. 339–347, doi:10.1023/A:1013952225669.

[7] E. C. Milner (1985): *Basic Wqo- and Bqo-Theory*. In: *Graphs and order*, pp. 487–502, doi:10.1007/978-94-009-5315-4_14.

[8] C. Nash-Williams (1965): *On well-quasi ordering infinite trees*. *Proceedings of the Cambridge Philosophical Society* 61, pp. 697–720, doi:10.1017/S0305004100039062.

[9] John H. Reif (1984): *The Complexity of Two-Player Games of Incomplete Information*. *J. Comput. Syst. Sci.* 29(2), pp. 274–301, doi:10.1016/0022-0000(84)90034-5.