On Computing Fixpoints in Well-Structured Regular Model Checking, With Applications to Lossy Channel Systems*

C. Baier[1], N. Bertrand[2], and Ph. Schnoebelen[2]

[1] Universität Bonn, Institut für Informatik I, Germany
[2] LSV, ENS de Cachan & CNRS, France

Abstract. We prove a general finite convergence theorem for "upward-guarded" fixpoint expressions over a well-quasi-ordered set. This has immediate applications in regular model checking of well-structured systems, where a main issue is the eventual convergence of fixpoint computations. In particular, we are able to directly obtain several new decidability results on lossy channel systems.

1 Introduction

Regular model checking [19, 11] is a popular paradigm for the symbolic verification of models with infinite state space. It has been applied to varied families of systems ranging from distributed algorithms and channel systems to hybrid systems and programs handling dynamic data structures.

In regular model checking, one works with regular sets of states and handles them via finite descriptions, e.g., finite-state automata or regular expressions. Models amenable to regular model checking are such that, when $S \subseteq Conf$ is regular, then $Post(S)$ (or $Pre(S)$), the set of 1-step successors (resp., predecessors), is again a regular set that can be computed effectively from S. Since regular sets are closed under Boolean operations, one can[3] try to compute the reachability set $Post^*(Init)$, as the limit of the sequence

$$S_0 := Init; \quad S_1 := S_0 \cup Post(S_0); \quad \dots \quad S_{n+1} := S_n \cup Post(S_n); \quad \dots \qquad (*)$$

Since equality of regular sets is decidable, the computation of (*) can contain a test that detects if the limit is reached in finite time, i.e., if $S_{n+1} = S_n$ for some $n \in \mathbb{N}$,

With infinite-state models, the main difficulty is *convergence*. It is very rare that a fixpoint computation like (*) converges in finite time [10].

Well-structured transition systems (WSTS) are a generic family of models for which the co-reachability set $Pre^*(Final)$ can be computed symbolically with a backward-chaining version of (*) [3, 16]. For WSTS's, convergence of the fixpoint computation is ensured by WQO theory: one handles upward-closed sets, and increasing sequences

* The first author is supported by the DFG-NWO project VOSS II and the DFG-project PROBPOR. The last two authors were supported by the ACI Sécurité & Informatique project Persée.

[3] Actually, such symbolic computations are possible with any class of representation closed under, and providing algorithms for, *Pre* or *Post*, Boolean operations, vacuity [19, 18].

of upward-closed sets always converge in finite time when the underlying ordering is a well-quasi-ordering (a WQO), as is the case with WSTS's.

Computing $Pre^*(Final)$ for reachability analysis is just a special case of fixpoint computation. When dealing with richer temporal properties, one is interested in more complex fixpoints. E.g., the set of states satisfying the CTL formula $\exists[Cond \mathsf{U} Goal]$ is definable via a least-fixpoint expression: $\mu X.Goal \cup (Cond \cap Pre(X))$. For game-theoretic properties, similar fixpoints are involved. E.g., the states from which one can enforce reaching a goal in a turn-based game is given by $\mu X.Goal \cup Pre(\overline{Pre(\overline{X})})$.

Our contribution. In this paper, we define a notion of μ-expressions where recursion is guarded by upward-closure operators, and give a general finite convergence theorem for all such expressions. The consequence is that these fixpoint expressions can be evaluated symbolically by an iterative procedure. The guarded fragment we isolate is very relevant for the verification of well-structured transition systems as we demonstrate by providing several new decidability results on channel systems.

Related work. Henzinger *et al.* give general conditions for the convergence of fixpoints computations for temporal [18] or game-theoretic [14] properties, but the underlying framework (finite quotients) is different and has different applications (timed and hybrid systems). Our applications to well-structured transition systems generalize results from [2, 27, 28, 21] that rely on more ad-hoc finite convergence lemmas.

2 A guarded mu-calculus

We assume basic understanding of μ-calculi techniques (otherwise see [7]) and of well-quasi-ordering (WQO) theory (otherwise see [24, 20], or simply [16, sect. 2.1]).

Let $(W, \sqsubseteq)$ be a well-quasi-ordered set. A subset V of W is *upward-closed* if $w \in V$ whenever $v \sqsubseteq w$ for some $v \in V$. From WQO theory, we mostly need the following:

Fact 2.1 (Finite convergence) *If $V_0 \subseteq V_1 \subseteq V_2 \subseteq \cdots$ is an infinite increasing sequence of upward-closed subsets of W, then for some index $k \in \mathbb{N}$, $\bigcup_{i \in \mathbb{N}} V_i = V_k$.*

The *upward-closure* of $V \subseteq W$, denoted $C_\uparrow(V)$, is the smallest upward-closed set that contains V. The *upward-kernel* of V, denoted $K_\uparrow(V)$, is the largest upward-closed set included in V. There are symmetric notions of *downward-closed* subset of W, of *downward-closure*, $C_\downarrow(V)$, and of *downward-kernel*, $K_\downarrow(V)$, of V. The complement of an upward-closed subset is downward-closed. Observe that $C_\uparrow(V) = V = K_\uparrow(V)$ iff V is upward-closed, and that $C_\uparrow$ and $K_\downarrow$ (resp., $C_\downarrow$ and $K_\uparrow$) are dual:

$$W \smallsetminus K_\uparrow(V) = C_\downarrow(W \smallsetminus V), \qquad W \smallsetminus K_\downarrow(V) = C_\uparrow(W \smallsetminus V). \tag{1}$$

Monotonic region algebra. In symbolic model-checking, a *region algebra* is a family of sets of states (subsets of W) that is closed under Boolean and other relevant operators like *Pre* and *Post* [18].

Here we consider regions generated by a family $O = \{o_1, o_2, \ldots\}$ of (monotonic) operators. By a k-ary *operator*, we mean a monotonic mapping $o : (2^W)^k \to 2^W$ that associates a subset $o(V_1, \ldots, V_k) \subseteq W$ with any k subsets $V_1, \ldots, V_k$. Monotonicity means that $o(V_1,\ldots,V_k) \subseteq o(V'_1,\ldots,V'_k)$ when $V_i \subseteq V'_i$ for $i = 1,\ldots,k$. We allow nullary operators, i.e., fixed subsets of W. Finally, we require that O contains at least four special unary operators: $C_\uparrow$, $C_\downarrow$, $K_\uparrow$, $K_\downarrow$, and two special nullary operators: $\emptyset$ and W.

The *region algebra generated by* O, denoted with $\mathcal{R}_O$, or simply $\mathcal{R}$, is the set of all the subsets of W, called *regions*, that can be obtained by applying operators from O on already constructed regions, starting with nullary operators. Equivalently, $\mathcal{R}$ is the least subset of 2^W that is closed under O.

We say the region algebra generated by O is *effective* if there are algorithms implementing the operators in O and an effective membership algorithm saying whether $w \in R$ for some $w \in W$ and some region $R \in \mathcal{R}_O$. Such effectiveness assumptions presuppose a finitary encoding of regions and elements of W: if there are several possible encodings for a same region, we assume an effective equality test.

Extending the region algebra with fixpoints. Let $\chi = \{X_1, X_2, \cdots\}$ be a countable set of variables. $L_\mu(W,\sqsubseteq,O)$, or shortly L_μ when $(W,\sqsubseteq)$ and O are understood, is the set of *O-terms with least and greatest fixpoints* given by the following abstract syntax:

$$L_\mu \ni \varphi,\psi \ ::= o(\varphi_1,\ldots,\varphi_k) \bigm| X \bigm| \mu X.\varphi \bigm| \nu X.\varphi \bigm| C_\uparrow(\varphi) \bigm| C_\downarrow(\varphi) \bigm| K_\uparrow(\varphi) \bigm| K_\downarrow(\varphi)$$

where X runs over variables from χ, and o over operators from O. $\mu X.\varphi$ and $\nu X.\varphi$ are fixpoint expressions. Free and bound occurrences of variables are defined as usual. We assume that no variable has both bound and free occurrences in some φ, and that no two fixpoint subterms bind the same variable: this can always be ensured by renaming bound variables. (The abstract syntax for L_μ could be shorter but we wanted to stress that $C_\uparrow$, $C_\downarrow$, $K_\uparrow$, and $K_\downarrow$ are required to be present in O.)

The meaning of L_μ terms is as expected: an *environment* is a mapping $env : \chi \to 2^W$ that interprets each variable $X \in \chi$ as a subset of W. Given env, a term $\varphi \in L_\mu$ denotes a subset of W, written $[\![\varphi]\!]_{env}$ and defined by induction on the structure of φ:

$$\begin{aligned}
[\![X]\!]_{env} &\stackrel{\text{def}}{=} env(X) & [\![o(\varphi_1,\ldots,\varphi_k)]\!]_{env} &\stackrel{\text{def}}{=} o([\![\varphi_1]\!]_{env},\ldots,[\![\varphi_k]\!]_{env})\\
[\![C_\uparrow(\varphi)]\!]_{env} &\stackrel{\text{def}}{=} C_\uparrow\big([\![\varphi]\!]_{env}\big) & [\![C_\downarrow(\varphi)]\!]_{env} &\stackrel{\text{def}}{=} C_\downarrow([\![\varphi]\!]_{env})\\
[\![K_\uparrow(\varphi)]\!]_{env} &\stackrel{\text{def}}{=} K_\uparrow([\![\varphi]\!]_{env}) & [\![K_\downarrow(\varphi)]\!]_{env} &\stackrel{\text{def}}{=} K_\downarrow\big([\![\varphi]\!]_{env}\big)\\
[\![\mu X.\varphi]\!]_{env} &\stackrel{\text{def}}{=} \mathrm{lfp}\big(\Omega[\varphi,X,env]\big) & [\![\nu X.\varphi]\!]_{env} &\stackrel{\text{def}}{=} \mathrm{gfp}\big(\Omega[\varphi,X,env]\big)
\end{aligned}$$

where, for all φ, X, and env, $\Omega[\varphi,X,env] : 2^W \to 2^W$ is a unary operator defined by $\Omega[\varphi,X,env](V) \stackrel{\text{def}}{=} [\![\varphi]\!]_{env[X:=V]}$, using the standard variant notation "$env[X := V]$" for the environment that agrees with env everywhere except on X where it returns V. As usual, $[\![\varphi]\!]_{env}$ does not depend on $env(X)$ if X is not free in φ, so that we may shortly write $[\![\varphi]\!]$ when φ is a closed term, i.e., a term with no free variables.

We recall that the semantics of the fixpoint terms is well-defined since, for every φ, X and env, $\Omega[\varphi,X,env]$ is monotonic (and since $(2^W,\subseteq)$ is a complete lattice). Moreover, if env and env' are such that $env(X) \subseteq env'(X)$ for all $X \in \chi$, shortly written $env \subseteq env'$, then $\mathrm{lfp}(\Omega[\varphi,X,env]) \subseteq \mathrm{lfp}(\Omega[\varphi,X,env'])$ and $\mathrm{gfp}(\Omega[\varphi,X,env]) \subseteq \mathrm{gfp}(\Omega[\varphi,X,env'])$.

Definition 2.2 (Upward- and downward-guardedness).

1. *A variable X is* upward-guarded *in φ if all free occurrences of X in φ are in the scope of either a $C_\uparrow$ or a $K_\uparrow$ operator, i.e., appear in a subterm of the form $C_\uparrow(\psi)$ or $K_\uparrow(\psi)$.*
2. *Dually, X is* downward-guarded *in φ if all its free occurrences are in the scope of a $C_\downarrow$ or a $K_\downarrow$ operator.*
3. *A term φ is* guarded *if all its least-fixpoint subterms $\mu X.\psi$ have X upward-guarded in ψ, and all its greatest-fixpoint subterms $\nu X.\psi$ have X downward-guarded in ψ.*

Given some φ, X and *env*, the approximants of $\text{lfp}(\Omega[\varphi, X, env])$ are given by the sequence $(M_i)_{i\in\mathbb{N}}$ of subsets of W defined inductively with $M_0 = \emptyset$ and $M_{i+1} = [\![\varphi]\!]_{env[X:=M_i]}$. Monotonicity yields

$$M_0 \subseteq M_1 \subseteq M_2 \subseteq \cdots \subseteq \text{lfp}(\Omega[\varphi, X, env]). \tag{2}$$

Similarly we define $(N_i)_{i\in\mathbb{N}}$ by $N_0 = W$ and $N_{i+1} = [\![\varphi]\!]_{env[X:=N_i]}$, so that

$$N_0 \supseteq N_1 \supseteq N_2 \supseteq \cdots \supseteq \text{gfp}(\Omega[\varphi, X, env]). \tag{3}$$

Lemma 2.3 (Finite convergence of approximants). *If X is upward-guarded in φ, then there exists an index $k \in \mathbb{N}$ such that*

$$[\![\mu X.\varphi]\!]_{env} = M_k = M_{k+1} = M_{k+2} = \ldots \tag{4}$$

Dually, if X is downward-guarded in φ, then there exists a $k' \in \mathbb{N}$ such that

$$[\![\nu X.\varphi]\!]_{env} = N_{k'} = N_{k'+1} = N_{k'+2} = \ldots \tag{5}$$

Proof. We only prove the first half since the other half is dual. Let $\psi_1, \ldots, \psi_m$ be the maximal subterms of φ that are immediately under the scope of a $C_\uparrow$ or a $K_\uparrow$ operator. Then φ can be decomposed under the form

$$\varphi \equiv \Phi(\Uparrow \psi_1, \ldots, \Uparrow \psi_m)$$

where the context $\Phi(Y_1, \ldots, Y_m)$ uses fresh variables $Y_1, \ldots, Y_m$ to be substituted in, and where $\Uparrow \psi_i$ is either $C_\uparrow(\psi_i)$ or $K_\uparrow(\psi_i)$, depending on how ψ_i appears in φ. In either case, and for any environment env', the set $[\![\Uparrow \psi_i]\!]_{env'}$ is upward-closed.

For $V_1, \ldots, V_m \subseteq W$ we shortly write $[\![\Phi]\!](V_1, \ldots, V_m)$ for $[\![\Phi]\!]_{env[Y_1:=V_1, \ldots, Y_m:=V_m]}$. Since X is upward-guarded in φ, it has no occurrence in Φ, only in the ψ_i's, so that

$$\begin{aligned} M_{i+1} = [\![\varphi]\!]_{env[X:=M_i]} &= [\![\Phi]\!]([\![\Uparrow \psi_1]\!]_{env[X:=M_i]}, \ldots, [\![\Uparrow \psi_m]\!]_{env[X:=M_i]}) \\ &= [\![\Phi]\!](L_{i,1}, \ldots, L_{i,m}) \end{aligned}$$

writing $L_{i,j}$ for $[\![\Uparrow \psi_j]\!]_{env[X:=M_i]}$. From $M_0 \subseteq M_1 \subseteq M_2 \subseteq \cdots$, we deduce $L_{0,j} \subseteq L_{1,j} \subseteq L_{2,j} \subseteq \cdots$ Since $K_\uparrow$ and $C_\uparrow$ return upward-closed sets, the $L_{i,j}$'s are upward-closed subsets of W. For all $j = 1, \ldots, m$, Fact 2.1 implies that there is an index k_j such that $L_{i,j} = L_{k_j,j}$ for all $i \geq k_j$. Picking $K = \max(k_1, \ldots, k_j)$ gives for any $i \geq K$

$$M_{i+1} = [\![\Phi]\!](L_{i,1}, \ldots, L_{i,m}) = [\![\Phi]\!](L_{k_1,1}, \ldots, L_{k_m,m}) = [\![\Phi]\!](L_{K,1}, \ldots, L_{K,m}) = M_{K+1}.$$

Thus, $\bigcup_{i\in\mathbb{N}} M_i = M_{K+1} = M_{K+2}$ and M_{K+1} is a fixpoint of $\Omega[\varphi, X, env]$, hence the least one thanks to (2). Picking $k = K + 1$ satisfies (4). $\square$

Regions with guarded fixpoints. We can now prove our main result: subsets defined by L_μ terms are regions (and can be computed effectively if the underlying region algebra is effective).

By a *region-environment* we mean an environment $env : \chi \to \mathcal{R}$ that associates regions with variables. If env is a region-environment, and φ has only free variables, i.e., has no fixpoint subterms, then $[\![\varphi]\!]_{env}$ is a region.

Theorem 2.4. *If $\varphi \in L_\mu$ is guarded and env is a region-environment then $[\![\varphi]\!]_{env}$ is a region. Furthermore, if the region algebra is effective, then $[\![\varphi]\!]_{env}$ can be computed effectively from φ and env.*

Proof. By structural induction on the structure of φ. If $\varphi = o()$ is a nullary operator, the result holds by definition of the region algebra. If $\varphi = o(\varphi_1, \cdots, \varphi_k)$, the $[\![\varphi_i]\!]_{env}$'s are (effectively) regions by induction hypothesis, so that $[\![\varphi]\!]_{env}$ is an (effective) region too by definition. In particular, this argument applies when o is a nullary operator, or is one of the unary operators we singled out: $C_\uparrow$, $C_\downarrow$, $K_\uparrow$, and $K_\downarrow$.

If $\varphi = \mu X.\psi$, we can apply Lemma 2.3 after we have proved that each one of the approximants $M_0, M_1, M_2, \ldots$, of $[\![\varphi]\!]_{env}$ are regions. In particular, $M_0 = \emptyset$ is a region, and if M_i is a region, then $M_{i+1} = [\![\psi]\!]_{env[X:=M_i]}$ is one too, since $env' = env[X := M_i]$ is a region-environment, and since by induction hypothesis $[\![\psi]\!]_{env'}$ is a region when env' is a region-environment. When $\mathcal{R}_O$ is effective, the M_i can be computed effectively, and one can detect when $M_k = M_{k+1}$ since region equality is decidable by definition. Then $[\![\varphi]\!]_{env} = M_k$ can be computed effectively. Finally, the case where $\varphi = \nu X.\psi$ is dual. ☐

Corollary 2.5 (Decidability for guarded $\mathcal{L}_\mu$ properties). *The following problems are decidable for effective monotonic region algebras:*

Model-checking: *"Does $w \in [\![\varphi]\!]$?" for a $w \in W$ and a closed and guarded $\varphi \in L_\mu$.*
Satisfiability: *"Is $[\![\varphi]\!]$ non-empty?" for a closed and guarded $\varphi \in L_\mu$.*
Universality: *"Does $[\![\varphi]\!] = W$?" for a closed and guarded $\varphi \in L_\mu$.*

A region algebra of regular languages. Consider $W = \Sigma^*$, the set of finite words over some finite alphabet Σ. The *subword ordering*, defined by "$u \sqsubseteq v$ iff u can be obtained by erasing some letters from v", is a WQO (Higman's Lemma). Regular languages over Σ are a natural choice for regions: observe that the closure operators $C_\uparrow$ and $C_\downarrow$ preserve regularity and have effective implementations.[4] Natural operators to be considered in O are $\cup$ (union) and $\cap$ (intersection). However, any operation on languages that is monotonic, preserve regularity, and has an effective implementation on regular languages can be added. This includes concatenation (denoted $R.R'$), star-closure (denote R^*), left- and right-residuals ($R^{-1}R' \stackrel{\text{def}}{=} \{v \mid \exists u \in R, uv \in R'\}$), shuffle product (denoted $R \parallel R'$), reverse (denoted $\overleftarrow{R}$), conjugacy ($\widetilde{R} \stackrel{\text{def}}{=} \{vu \mid uv \in R\}$), homomorphic and inverse-homomorphic images, and many more [26]. Complementation is not allowed in O (it is not monotonic)

[4] From a FSA for R, one obtains a FSA for $C_\uparrow(R)$ simply by adding loops $q \xrightarrow{a} q$ on all states q of the FSA and for all letters $a \in \Sigma$. A FSA for $C_\downarrow(R)$ is obtained by adding ε-transitions $q \xrightarrow{\varepsilon} q'$ whenever there is a $q \xrightarrow{a} q'$. From this, $K_\uparrow$ and $K_\downarrow$ can be implemented using (1).

but the duals of all above-mentioned operators can be included in O (without compromising effectiveness) so that, for all practical purposes, complement can be used with the restriction that bound variables in L_μ terms are under an even number of complementations.

An application of Theorem 2.4 is that, if R_1 and R_2 are regular languages, then the language defined as $\mu X.\nu Y.\big(K_\uparrow\big[R_1 \parallel (X^* \cap C_\downarrow(Y^{-1}\overleftarrow{\overline{X}} \cap X^{-1}R_2))\big]\big)$ is regular and a finite representation for it (e.g., a regular expression or a minimal DFA) can be constructed from R_1 and R_2.

3 Verification of lossy channel systems

Theorem 2.4 has several applications for regular model checking of lossy channel systems [5] (LCS) and other families of well-structured systems [3, 16]. In the rest of this paper we concentrate on LCS's.

3.1 Channel systems, perfect and lossy

A channel system is a tuple $\mathcal{L} = (Q, \mathsf{C}, \mathsf{M}, \Delta)$ consisting of a finite set $Q = \{p, q, \ldots\}$ of *locations*, a finite set $\mathsf{C} = \{c, \ldots\}$ of *channels*, a finite *message alphabet* $\mathsf{M} = \{m, \ldots\}$ and a finite set $\Delta = \{\delta, \ldots\}$ of *transition rules*. Each transition rule has the form $q \xrightarrow{op} p$ where op is an *operation*: $c!m$ (sending message $m \in \mathsf{M}$ along channel $c \in \mathsf{C}$), $c?m$ (receiving message m from channel c), or $\surd$ (an internal action to some process, no I/O-operation).

Operational semantics. Let $\mathcal{L} = (Q, \mathsf{C}, \mathsf{M}, \Delta)$ be a channel system. A *configuration* (also, a *state*) is a pair $\sigma = (q, w)$ where $q \in Q$ is a location and $w : \mathsf{C} \to \mathsf{M}^*$ is a channel valuation that associates with any channel its content (a sequence of messages). The set $Q \times \mathsf{M}^{*\mathsf{C}}$ of all configurations is denoted by $\mathit{Conf} = \{\sigma, \rho, \ldots\}$. For a subset V of Conf, we let $\overline{V} \stackrel{\text{def}}{=} \mathit{Conf} \smallsetminus V$.

Steps between configurations are as expected. Formally, $\sigma = (q, w)$ leads to $\sigma' = (q', w')$ by firing $\delta = p \xrightarrow{op} r$, denoted $\sigma \xrightarrow{\delta}_{\text{perf}} \sigma'$, if and only if $q = p$, $q' = r$ and w' is obtained from w by the effect of op (the "perf" subscripts emphasizes that the step is perfect: without losses). Precisely

$$w'(c) = \begin{cases} w(c)m & \text{if } op = c!m, \\ m^{-1}w(c) & \text{if } op = c?m, \end{cases}$$

where the notation "$w(c)m$" (for concatenation) and "$m^{-1}w(c)$" (for left-residuals) are as in section 2. Furthermore, $w'(c) = w(c)$ for all channels c that are not touched upon by op.

Thus, when $op = c?m$, w' is only defined if $w(c)$ starts with m and indeed this is the intended condition for firing δ. Whenever $\sigma \xrightarrow{\delta} \rho$ for some ρ, we say that δ is *enabled* in σ, written $\delta \in \Delta(\sigma)$.

Below we restrict our attention to LCS's where from each $q \in Q$ there is at least one rule $q \xrightarrow{op} p$ in Δ where op is not a receiving action: this ensures that the LCS has no deadlock states and simplifies many technical details without losing any generality.

Lossy systems. In *lossy* channel systems, losing messages is formalized via the subword ordering, extended from M^* to $Conf$: $(q,w) \sqsubseteq (q',w')$ if $q = q'$ and $w(c) \sqsubseteq w'(c)$ for all channels $c \in \mathsf{C}$.

A (possibly lossy) step in the LCS is made of a perfect step followed by arbitrary losses:[5] formally, we write $\sigma \xrightarrow{\delta} \rho$ whenever there is a perfect step $\sigma \xrightarrow{\delta}_{\text{perf}} \sigma'$ such that $\rho \sqsubseteq \sigma'$. This gives rise to a labeled transition system $LTS_{\mathcal{L}} \stackrel{\text{def}}{=} (Conf, \Delta, \rightarrow)$.

Remark 3.1. Our choice of operational semantics has the consequence that $LTS_{\mathcal{L}}$ is *not* turned into a WSTS by $\sqsubseteq$ because message losses only occur after a step. However, the WSTS structure is recovered with the following relation: $\sigma \preceq \rho \stackrel{\text{def}}{\Leftrightarrow} \sigma \sqsubseteq \rho \wedge \Delta(\sigma) = \Delta(\rho)$. Both $\sqsubseteq$ and $\preceq$ turns $Conf$ into a WQO. From now on we assume, for the sake of simplicity, that $(Conf, \sqsubseteq)$ is the WQO on which L_μ is defined. □

Following standard notations for transition systems $(Conf, \Delta, \rightarrow)$ labeled over some Δ, we write $Pre[\delta](\sigma) \stackrel{\text{def}}{=} \{\rho \in Conf \mid \rho \xrightarrow{\delta} \sigma\}$ for the set of predecessors via δ of σ in $\mathcal{L}$. Then $Pre(\sigma) \stackrel{\text{def}}{=} \bigcup_{\delta \in \Delta} Pre[\delta](\sigma)$ has all 1-step predecessors of σ, and $Pre(V) = \bigcup_{\sigma \in V} Pre(\sigma)$ has all 1-step predecessors of states in V. The dual $\widetilde{Pre}$ of Pre is defined by $\widetilde{Pre}(V) = \overline{Pre(\overline{V})}$. Thus $\sigma \in \widetilde{Pre}(V)$ iff all 1-step successors of σ are in V (this includes the case where σ is a deadlock state).

Seen as unary operators on 2^{Conf}, both Pre and $\widetilde{Pre}$ are monotonic and even continuous for all transition systems [30]. For LCS's, the following lemma states that Pre is compatible with the WQO on states, which will play a crucial role later when we want to show that some L_μ term is guarded.

Lemma 3.2. *Let $V \subseteq Conf$ in the transition system $LTS_{\mathcal{L}}$ associated with a LCS $\mathcal{L}$. Then $Pre(V) = Pre(C_\uparrow(V))$ and $\widetilde{Pre}(V) = \widetilde{Pre}(K_\downarrow(V))$.*

Proof. $V \subseteq C_\uparrow(V)$ implies $Pre(V) \subseteq Pre(C_\uparrow(V))$. Now $\sigma \in Pre(C_\uparrow(V))$ implies that $\sigma \rightarrow \rho \sqsupseteq \rho'$ for some $\rho' \in V$. But then $\sigma \rightarrow \rho'$ by definition of lossy steps and $\sigma \in Pre(V)$. The second equality is dual. □

An effective region algebra for LCS's. We are now ready to apply the framework of section 2 to regular model checking of lossy channel systems. Assume $\mathcal{L} = (Q, \mathsf{C}, \mathsf{M}, \Delta)$ is a given LCS. A region $R \in \mathcal{R}$ is any "regular" subset of $Conf$. More formally, it is any set $R \subseteq Conf$ that can be written under the form

$$R = \sum_{i \in I} (q_i, R_i^1, \ldots, R_i^{|\mathsf{C}|})$$

[5] Note that, with this definition, message losses only occur *after* steps (thus, not in the initial configuration). The usual definition allows arbitrary losses before and after a step [5]. There is no essential semantical difference between these two ways of grouping atomic events into single "steps", except for the first step. The definition from [5] is technically smoother when LCS's are viewed as nondeterministic systems, but becomes unnatural in situations where several adversarial processes compete, e.g., in probabilistic LCS's [9] or the game-theoretical settings we explore in sections 4 and 5.

where I is a *finite* index set, the q_i's are locations from Q, and each R_i^j is a regular language on alphabet M. The notation has obvious interpretation, with summation denoting set union (the empty sum is denoted $\emptyset$). We are not more precise on how such regions could be effectively represented (see [6]), but they could be handled as, e.g., regular expressions or FSAs over the extended alphabet $\mathsf{M} \cup Q \cup \{'(', ')', ','\}$.

The set O of operators includes union, intersection, $C_\uparrow$, $C_\downarrow$, $K_\uparrow$, $K_\downarrow$: these are monotonic, regularity-preserving, and effective operators as explained in our example at the end of section 2. Operators specific to regular model-checking are Pre and $\widetilde{Pre}$. That they are regularity-preserving and effective is better seen by first looking at the special case of perfect steps. We use

$$Pre_{\mathrm{perf}}[p \xrightarrow{c_i?m} q](q, R_p^1, \cdots, R_p^{|\mathsf{C}|}) = (p, R_p^1, \ldots, R_p^{i-1}, mR_p^i, R_p^{i+1}, \ldots, R_p^{|\mathsf{C}|}),$$
$$Pre_{\mathrm{perf}}[p \xrightarrow{c_i!m} q](q, R_p^1, \cdots, R_p^{|\mathsf{C}|}) = (p, R_p^1, \ldots, R_p^{i-1}, R_p^i m^{-1}, R_p^{i+1}, \ldots, R_p^{|\mathsf{C}|})$$

completed with the obvious

$$Pre_{\mathrm{perf}}[p \xrightarrow{op} q](r, R_p^1, \cdots, R_p^{|\mathsf{C}|}) = \emptyset \text{ when } r \neq q,$$
$$Pre_{\mathrm{perf}}\Big(\sum_{i \in I}(q_i, R_i^1, \ldots, R_i^{|\mathsf{C}|})\Big) = \sum_{i \in I}\sum_{\delta \in \Delta} Pre_{\mathrm{perf}}[\delta](q_i, R_i^1, \ldots, R_i^{|\mathsf{C}|}).$$

Then lossy steps are handled with $Pre(R) = Pre_{\mathrm{perf}}(C_\uparrow(R))$.

Clearly, both Pre_{perf} and Pre are effective operators on regions.

3.2 Regular model-checking for lossy channel systems

Surprising decidability results for lossy channel systems is what launched the study of this model [15, 5, 12]. We reformulate several of these results as a direct consequence of Theorem 2.4, before moving to new problems and new decidability results in the next sections. Note that our technique is applied here to a slightly different operational semantics (cf. footnote 5) but it would clearly apply as directly to the simpler semantics.

Reachability analysis. Thanks to Lemma 3.2, the co-reachability set can be expressed as a guarded L_μ term:

$$Pre^*(V) = \mu X.V \cup Pre(X) = \mu X.V \cup Pre(C_\uparrow(X)). \tag{6}$$

Corollary 3.3. *For regular $V \subseteq Conf$, $Pre^*(V)$ is regular and effectively computable.*

Safety properties. More generally, safety properties can be handled. In CTL, they can be written $\forall(V_1 \mathsf{R} V_2)$. Recall that R, the Release modality, is dual to Until: a state σ satisfies $\forall(V_1 \mathsf{R} V_2)$ if and only if along all paths issuing from σ, V_2 always holds until maybe V_1 is visited. Using Lemma 3.2, $[\![\forall(V_1 \mathsf{R} V_2)]\!]$, the set of states where the safety property holds, can be defined as a guarded L_μ term:

$$[\![\forall(V_1 \mathsf{R} V_2)]\!] = \nu X.\big(V_2 \cap (\widetilde{Pre}(X) \cup V_1)\big) = \nu X.\big(V_2 \cap (\widetilde{Pre}(K_\downarrow(X)) \cup V_1)\big). \tag{7}$$

Corollary 3.4. *For regular $V_1, V_2 \subseteq Conf$, $[\![\forall(V_1 \mathsf{R} V_2)]\!]$ is regular and effectively computable.*

Another formulation is based on the duality between the "$\forall$R" and the "$\exists$U" modalities.

Theorem 3.5. *[21, sect. 5] If f is a temporal formula in the $\mathsf{TL}(\exists\mathsf{U}, \exists\mathsf{X}, \wedge, \neg)$ fragment of CTL (using regions for atomic propositions), then $[\![f]\!]$ is regular and effectively computable.*

Proof. By induction on the structure of f, using $[\![\exists\mathsf{X} f]\!] \stackrel{\text{def}}{=} Pre([\![f]\!])$, and the fact that regions are (effectively) closed under complementation. □

Beyond safety. Inevitability properties, and recurrent reachability can be stated in L_μ. With temporal logic notation, this yields

$$[\![\forall\Diamond V]\!] = \mu X.\big(V \cup (Pre(Conf) \cap \widetilde{Pre}(X))\big),$$
$$[\![\exists\Box\Diamond V]\!] = \nu X.\big(\mu Y.((V \cup Pre(Y)) \cap Pre(X))\big).$$

These two terms are not guarded and Lemma 3.2 is of no help here. However this is not surprising: firstly, $\sigma \models \exists\Box\Diamond V$ is undecidable [4]; secondly, and while $\sigma \models [\![\forall\Diamond V]\!]$ is decidable, the set $[\![\forall\Diamond V]\!]$ cannot be computed effectively [23].

3.3 Generalized lossy channel systems

Transition rules in LCS's do not carry guards, aka preconditions, beyond the implicit condition that a reading action $c?m$ is only enabled when $w(c)$ starts with m. This barebone definition is for simplification purpose, but actual protocols sometimes use guards that probe the contents of the channel before taking this or that transition. The simplest such guards are emptiness tests, like "$p \xrightarrow{c=\varepsilon?} q$" that only allows a transition from p to q if $w(c)$ is empty.

We now introduce *LCS's with regular guards* (GLCS's), an extension of the barebone model where any regular set of channel contents can be used to guard a transition rule. This generalizes emptiness tests, occurrence tests (as in [25]), etc., and allows expressing priority between rules since whether given rules are enabled is a regular condition.

Formally, we assume rules in Δ now have the form $p \xrightarrow{G:op} q$ with p, q, op as before, and where G, the guard, can be any regular region. The operational semantics is a expected: when $\delta = p \xrightarrow{G:op} q$, there is a perfect step $\sigma \xrightarrow{\delta}_{\text{perf}} \theta$ iff $\sigma \in G$ and θ is obtained from σ by the rule $p \xrightarrow{G:op} q$ (without any guard). Then, general steps $\sigma \xrightarrow{\delta} \rho$ are obtained from perfect steps $\sigma \xrightarrow{\delta}_{\text{perf}} \sigma'$ by message losses $\rho \sqsubseteq \sigma'$.

Verification of GLCS's. For GLCS's, *Pre* and *Post* are effective monotonic regularity-preserving operators as in the LCS case since

$$Pre[p \xrightarrow{G:op} q](R) = G \cap Pre[p \xrightarrow{op} q](R),$$
$$Post[p \xrightarrow{G:op} q](R) = Post[p \xrightarrow{op} q](G \cap R).$$

Observe that Lemma 3.2 holds for GLCS's as well, so that Equations (6) and (7) entail a generalized version of Theorem 3.5:

Theorem 3.6. *For all GLCS's $\mathcal{L}$ and formulae f in the $\mathsf{TL}(\exists\mathsf{U},\exists\mathsf{X},\wedge,\neg)$ fragment, $[\![f]\!]$ is regular and effectively computable.*

4 Solving games on lossy channel systems

In this section, we consider turn-based games on GLCS's where two players, A and B, alternate their moves. Games play a growing role in verification where they address situations in which different agents have different, competing goals. We assume a basic understanding of the associated concepts: arena, play, strategy, etc. (otherwise see [17]).

Games on well-structured systems have already been investigated in [2, 27, 28]. The positive results in these three papers rely on ad-hoc finite convergence lemmas that are special cases of our Theorem 2.4.

4.1 Symmetric LCS-games with controllable message losses

We start with the simplest kind of games on a GLCS: A and B play in turn, choosing the next configuration, i.e., picking what rule $\delta \in \Delta$ is fired, and what messages are lost.

Formally, a *symmetric LCS-game* is a GLCS $\mathcal{L} = (Q_A, Q_B, \mathsf{C}, \mathsf{M}, \Delta)$ where the set of locations $Q = Q_A \cup Q_B$ is partitioned into two sets, one for each player, and where the rules ensure strict alternation: for all $p \xrightarrow{G:op} q \in \Delta$, $p \in Q_A$ iff $q \in Q_B$. Below, we shortly write $Conf_A$ for $Q_A \times \mathsf{M}^{*|\mathsf{C}|}$, the regular region where it is A's turn to play. $Conf_B$ is defined similarly. Strict alternation means that the arena, $LTS_{\mathcal{L}}$, is a bipartite graph partitioned in $Conf_A$ and $Conf_B$.

Reachability games. Reachability and invariance are among the simplest objectives for games. In a reachability game, A tries to reach a state in some set V, no matter how B behaves. This goal is denoted $\Diamond V$. It is known that such games are determined and that memoryless strategies are sufficient [17]. The set of winning configurations for A is denoted with $\langle\!\langle A\rangle\!\rangle\Diamond V$, and can be defined in L_μ:

$$\langle\!\langle A\rangle\!\rangle\Diamond V = \mu X.\Big[V \cup \big[Conf_A \cap Pre(X)\big] \cup \big[Conf_B \cap \widetilde{Pre}(X)\big]\Big]. \tag{8}$$

The first occurrence of X can be made upward-guarded by replacing $Pre(X)$ with $Pre(C_\uparrow(X))$ (Lemma 3.2). For the second occurrence, we can unfold the term, relying on the fixpoint equation $[\![\mu X.\varphi(X)]\!] = [\![\mu X.\varphi(\varphi(X))]\!]$. This will replace $Conf_B \cap \widetilde{Pre}(X)$ in (8) with

$$Conf_B \cap \widetilde{Pre}\Big(V \cup \big[Conf_A \cap Pre(X)\big] \cup \big[Conf_B \cap \widetilde{Pre}(X)\big]\Big). \tag{+}$$

Now, the strict alternation between $Conf_A$ and $Conf_B$ lets us simplify (+) into

$$Conf_B \cap \widetilde{Pre}\Big(V \cup Pre(X)\Big). \tag{9}$$

Hence (8) can be rewritten into

$$\langle\!\langle A\rangle\!\rangle\Diamond V = \mu X.\Big[V \cup \big[Conf_A \cap Pre(C_\uparrow(X))\big] \cup \big[Conf_B \cap \widetilde{Pre}(V \cup Pre(C_\uparrow(X)))\big]\Big]. \tag{8'}$$

Invariance games. In invariance games, A's goal is to never leave some set $V \subseteq \mathit{Conf}$, no matter how B behaves. Invariance games are dual to reachability games, and the set of winning configurations $\langle\!\langle A\rangle\!\rangle\Box V$ is exactly $\overline{\langle\!\langle B\rangle\!\rangle\Diamond\overline{V}}$.

Repeated reachability games. Here A's goal is to visit V infinitely many times, no matter how B behaves. The set of winning configurations is given by the following L_μ term:

$$\langle\!\langle A\rangle\!\rangle\Box\Diamond V = \nu Y.\langle\!\langle A\rangle\!\rangle\Diamond\Big[V \cap (\varphi_A(Y) \cup \varphi_B(Y))\Big], \tag{10}$$

where

$$\varphi_A(Y) \stackrel{\text{def}}{=} \mathit{Conf}_A \cap \mathit{Pre}\big(C_\uparrow(\widetilde{\mathit{Pre}}(K_\downarrow(Y)))\big),$$
$$\varphi_B(Y) \stackrel{\text{def}}{=} \mathit{Conf}_B \cap \widetilde{\mathit{Pre}}(K_\downarrow(Y)).$$

and where we reuse (8') for $\langle\!\langle A\rangle\!\rangle\Diamond[\ldots]$.

Persistence games. In a persistence game, A aims at remaining inside V from some moment on, no matter how B behaves. Dually, this can be seen as a repeated reachability game for B. Note that $\langle\!\langle A\rangle\!\rangle\Diamond\Box V \neq \langle\!\langle A\rangle\!\rangle\Diamond(\langle\!\langle A\rangle\!\rangle\Box V)$.

Theorem 4.1 (Decidability of symmetric LCS-games). *For symmetric LCS-games $\mathcal{L}$ and regular regions V, the four sets $\langle\!\langle A\rangle\!\rangle\Diamond V$, $\langle\!\langle A\rangle\!\rangle\Box V$, $\langle\!\langle A\rangle\!\rangle\Diamond\Box V$, and $\langle\!\langle A\rangle\!\rangle\Box\Diamond V$, are (effective) regions. Hence reachability, invariance, repeated reachability, and persistence symmetric games are decidable on GLCS's.*

Proof (Sketch). The winning sets can be defined by guarded L_μ terms.

Remark 4.2. There is no contradiction between the undecidability of $\exists\Box\Diamond V$ and the decidability of $\langle\!\langle A\rangle\!\rangle\Box\Diamond V$. In the latter case, B does not cooperate with A, making the goal harder to reach for A (and the property easier to decide for us). □

4.2 Asymmetric LCS-games with 1-sided controlled loss of messages

Here we adopt the setting considered in [2]. It varies from the symmetric setting of section 4.1 in that only player B can lose messages (and can control what is lost), while player A can only make perfect steps. Note that this generalizes games where A plays moves in the channel system, and B is an adversarial environment responsible for message losses. We use the same syntax as for symmetric LCS-games.

Reachability and invariance games. Let us first consider games where one player tries to reach a regular region V (goal $\Diamond V$), no matter how the other player behaves.

The configurations where B can win a reachability game are given by:

$$\begin{aligned}\langle\!\langle B\rangle\!\rangle\Diamond V &= \mu X.V \cup \Big(\mathit{Conf}_B \cap \mathit{Pre}(X)\Big) \cup \Big(\mathit{Conf}_A \cap \widetilde{\mathit{Pre}}_{\text{perf}}(X)\Big)\\ &= \mu X.V \cup \Big(\mathit{Conf}_B \cap \mathit{Pre}(C_\uparrow(X))\Big) \cup \Big(\mathit{Conf}_A \cap \widetilde{\mathit{Pre}}_{\text{perf}}(V \cup \mathit{Pre}(C_\uparrow(X)))\Big)\end{aligned}$$

where guardedness is obtained via Lemma 3.2 and unfolding.

When we consider a reachability game for A, the situation is not so clear:

$$\langle\langle A\rangle\rangle\Diamond V = \mu X.V \cup \Big(Conf_A \cap Pre_{\mathrm{perf}}(X)\Big) \cup \Big(Conf_B \cap \widetilde{Pre}(X)\Big).$$

Neither Lemma 3.2 nor unfolding techniques can turn this into a guarded term. This should be expected since the set $\langle\langle A\rangle\rangle\Diamond V$ cannot be computed effectively [2].

Theorem 4.3 (Decidability of asymmetric LCS-games [2]). *For asymmetric LCS-games $\mathcal{L}$ and regular regions V, the sets $\langle\langle B\rangle\rangle\Diamond V$ and $\langle\langle A\rangle\rangle\Box V$ are (effective) regions. Hence reachability games for B, and invariance games for A are decidable on GLCS's.*

Proof (Sketch). Invariance games are dual to reachability games, and the winning set $\langle\langle B\rangle\rangle\Diamond V$ is defined by a guarded L_μ term.

5 Channel systems with probabilistic losses

LCS's where messages losses follow probabilistic rules have been investigated as a less pessimistic model of protocols with unreliable channels (see [29, 1, 9] and the references therein).

In [9], we present decidability results for LCS's seen as combining *nondeterministic* choice of transition rules with *probabilistic* message losses. The semantics is in term of Markovian decision processes, or $1\frac{1}{2}$-player games, whose solutions can be defined in L_μ. Indeed, we found the inspiration for L_μ and our Theorem 2.4 while extending our results in the MDP approach to richer sets of regions.

In this section, rather than rephrasing our results on $1\frac{1}{2}$-player games on LCS's, we show how to deal with $2\frac{1}{2}$-player games [13] on LCS's, i.e., games opposing players A and B (as in section 4) but where message losses are probabilistic. This relies on new characterizations, like equations (11) or (12) below, for which the proof will be found in the full version of this paper.

Formally, a *symmetric probabilistic LCS-game* $\mathcal{L} = (Q_A, Q_B, \mathsf{C}, \mathsf{M}, \Delta)$ is exactly like a symmetric LCS-game but with an altered semantics: in state $\sigma \in Conf_A$, player A selects a fireable rule $\delta \in \Delta$ (B picks the rule if $\sigma \in Conf_B$) and the system moves to a successor state ρ where $\sigma \xrightarrow{\delta}_{\mathrm{perf}} \sigma' \sqsupseteq \rho$ and ρ is chosen probabilistically in $C_\downarrow(\{\sigma'\})$. The definition of the probability distribution $\mathbf{P}(\sigma,\delta,\rho)$ can be found in [29, 9] where it is called *the local-fault model*. It satisfies $\mathbf{P}(\sigma,\delta,\rho) > 0$ iff $\rho \sqsubseteq \sigma'$ (assuming $\sigma \xrightarrow{\delta}_{\mathrm{perf}} \sigma'$). Additionally it guarantees a *finite-attractor property*: the set of states where all channels are empty will be visited infinitely many times almost surely [1, 8].

Reachability games. Assume A tries to reach region V (goal $\Diamond V$) *with probability 1* no matter how B behaves. The set $\langle\langle A\rangle\rangle[\Diamond V]_{=1}$ of states in which A has an almost-sure winning strategy is given by

$$\langle\langle A\rangle\rangle[\Diamond V]_{=1} = \nu Y.\mu X.\begin{pmatrix} V \cup \Big[Conf_A \cap Pre_{\mathrm{perf}}(C_\uparrow(X) \cap K_\downarrow(Y))\Big] \\ \cup \Big[Conf_B \cap \widetilde{Pre}_{\mathrm{perf}}(C_\uparrow(X) \cap K_\downarrow(Y))\Big]\end{pmatrix}. \qquad (11)$$

Remark 5.1. Justifying (11) is outside the scope of this paper, but we can try to give an intuition of why it works: the inner fixpoint "$\mu X.V \cup \cdots$" define the largest set from which A has a strategy to reach V no matter what B does *if the message losses are favorable*. However, whatever messages are lost, A's strategy also guarantees that the system will remain in Y, from which it will be possible to retry the strategy for $\Diamond V$ as many times as necessary. This will eventually succeed almost surely thanks to the finite-attractor property. □

Invariance games. Assume now A tries to stay in V almost surely (goal $[\Box V]_{=1}$), no matter how B behaves. Then A must ensure $\Box V$ surely and we are considering a 2-player game where message losses are adversarial and could as well be controlled by B. This leads to

$$\begin{aligned}\langle\!\langle A\rangle\!\rangle[\Box V]_{=1} &= \nu X.V \cap \Big(\big[Conf_A \cap Pre_{\text{perf}}(K_\downarrow(X))\big] \cup \big[Conf_B \cap \widetilde{Pre}(X)\big]\Big)\\ &= \nu X.V \cap \Big(\big[Conf_A \cap Pre_{\text{perf}}(K_\downarrow(X))\big] \cup \big[Conf_B \cap \widetilde{Pre}(K_\downarrow(X))\big]\Big).\end{aligned} \tag{12}$$

In (12), the subterm $Pre_{\text{perf}}(K_\downarrow(X))$ accounts for states in which A can choose a perfect move that will end in $K_\downarrow(X)$, i.e., that can be followed by any adversarial message losses and still remain in X. The subterm $\widetilde{Pre}(X)$ accounts for states in which B cannot avoid going to X, even with message losses under his control. $\widetilde{Pre}(X)$ can be rewritten into $\widetilde{Pre}(K_\downarrow(X))$ thanks to Lemma 3.2, so that we end up with a guarded term.

Goals to be satisfied with positive probability. In $2\frac{1}{2}$-player games, it may happen that a given goal can only be attained with some non-zero probability [13]. Observe that, since the games we consider are determined [22], the goals $[\Diamond V]_{>0}$ or $[\Box V]_{>0}$ are the opposite of goals asking for probability 1:

$$\langle\!\langle A\rangle\!\rangle[\Diamond V]_{>0} = \overline{\langle\!\langle B\rangle\!\rangle[\Box \overline{V}]_{=1}}, \qquad \langle\!\langle A\rangle\!\rangle[\Box V]_{>0} = \overline{\langle\!\langle B\rangle\!\rangle[\Diamond \overline{V}]_{=1}}.$$

Theorem 5.2 (Decidability of qualitative symmetric probabilistic LCS-games). *For symmetric probabilistic LCS-games $\mathcal{L}$ and regular regions V, the sets $\langle\!\langle A\rangle\!\rangle[\Diamond V]_{=1}$, $\langle\!\langle A\rangle\!\rangle[\Diamond V]_{>0}$, $\langle\!\langle A\rangle\!\rangle[\Box V]_{=1}$, and $\langle\!\langle A\rangle\!\rangle[\Box V]_{>0}$ are (effective) regions. Hence qualitative reachability and invariance games are decidable on GLCS's.*

Proof (Sketch). These sets can be defined by guarded L_μ terms. □

6 Conclusion

We defined a notion of upward/downward-guarded fixpoint expressions that define subsets of a well-quasi-ordered set. For these guarded fixpoint expressions, a finite convergence theorem is proved, that shows how the fixpoints can be evaluated with a finite number of operations. This has a number of applications, in particular in the symbolic verification of well-structured systems, our original motivation. We illustrate this in the second part of the paper, with lossy channel systems as a target. For these systems, we derive in an easy and uniform way, a number of decidability theorems that extend or generalize the main existing results in the verification of temporal properties or game-theoretical properties.

These techniques can be applied to other well-structured systems, with a region algebra built on, e.g., upward-closed sets. Such regions are not closed by complementation, hence fewer properties can be written in L_μ. Admittedly, many examples of well-structured systems do not enjoy closure properties as nice as our Lemma 3.2 for LCS's, which will make it more difficult to express interesting properties in the guarded fragment of L_μ. But this can still be done, as witnessed by [27, 28] where the authors introduced a concept of B-games and BB-games that captures some essential closure assumptions allowing the kind of rewritings and unfoldings we have justified with Lemma 3.2.

References

1. P. A. Abdulla, N. Bertrand, A. Rabinovich, and Ph Schnoebelen. Verification of probabilistic systems with faulty communication. *Information and Computation*, 202(2):141–165, 2005.
2. P. A. Abdulla, A. Bouajjani, and J. d'Orso. Deciding monotonic games. In *Proc. 17th Int. Workshop Computer Science Logic (CSL 2003) and 8th Kurt Gödel Coll. (KGL 2003), Vienna, Austria, Aug. 2003*, volume 2803 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 2003.
3. P. A. Abdulla, K. Čerāns, B. Jonsson, and Yih-Kuen Tsay. Algorithmic analysis of programs with well quasi-ordered domains. *Information and Computation*, 160(1/2):109–127, 2000.
4. P. A. Abdulla and B. Jonsson. Undecidable verification problems for programs with unreliable channels. *Information and Computation*, 130(1):71–90, 1996.
5. P. A. Abdulla and B. Jonsson. Verifying programs with unreliable channels. *Information and Computation*, 127(2):91–101, 1996.
6. P. A. Abdulla and B. Jonsson. Channel representation in protocol verification. In *Proc. 12th Int. Conf. Concurrency Theory (CONCUR 2001), Aalborg, Denmark, Aug. 2001*, volume 2154 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2001.
7. A. Arnold and D. Niwiński. *Rudiments of μ-Calculus*, volume 146 of *Studies in Logic and the Foundations of Mathematics*. Elsevier Science, 2001.
8. C. Baier, N. Bertrand, and Ph. Schnoebelen. A note on the attractor-property of infinite-state Markov chains. *Information Processing Letters*, 97(2):58–63, 2006.
9. C. Baier, N. Bertrand, and Ph. Schnoebelen. Verifying nondeterministic probabilistic channel systems against ω-regular linear-time properties. *ACM Transactions on Computational Logic*, 2006. To appear, available at http://arxiv.org/abs/cs.LO/0511023.
10. S. Bardin, A. Finkel, J. Leroux, and Ph. Schnoebelen. Flat acceleration in symbolic model checking. In *Proc. 3rd Int. Symp. Automated Technology for Verification and Analysis (ATVA 2005), Taipei, Taiwan, Oct. 2005*, volume 3707 of *Lecture Notes in Computer Science*, pages 474–488. Springer, 2005.
11. A. Bouajjani, B. Jonsson, M. Nilsson, and T. Touili. Regular model checking. In *Proc. 12th Int. Conf. Computer Aided Verification (CAV 2000), Chicago, IL, USA, July 2000*, volume 1855 of *Lecture Notes in Computer Science*, pages 403–418. Springer, 2000.
12. G. Cécé, A. Finkel, and S. Purushothaman Iyer. Unreliable channels are easier to verify than perfect channels. *Information and Computation*, 124(1):20–31, 1996.
13. K. Chatterjee, L. de Alfaro, and T. A. Henzinger. The complexity of stochastic Rabin and Streett games. In *Proc. 32nd Int. Coll. Automata, Languages, and Programming (ICALP 2005), Lisbon, Portugal, July 2005*, volume 3580 of *Lecture Notes in Computer Science*, pages 878–890. Springer, 2005.

14. L. de Alfaro, T. A. Henzinger, and R. Majumdar. Symbolic algorithms for infinite-state games. In *Proc. 12th Int. Conf. Concurrency Theory (CONCUR 2001), Aalborg, Denmark, Aug. 2001*, volume 2154 of *Lecture Notes in Computer Science*, pages 536–550. Springer, 2001.
15. A. Finkel. Decidability of the termination problem for completely specificied protocols. *Distributed Computing*, 7(3):129–135, 1994.
16. A. Finkel and Ph. Schnoebelen. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1–2):63–92, 2001.
17. E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002.
18. T. A. Henzinger, R. Majumdar, and J.-F. Raskin. A classification of symbolic transition systems. *ACM Trans. Computational Logic*, 6(1):1–32, 2005.
19. Y. Kesten, O. Maler, M. Marcus, A. Pnueli, and E. Shahar. Symbolic model checking with rich assertional languages. *Theoretical Computer Science*, 256(1–2):93–112, 2001.
20. J. B. Kruskal. The theory of well-quasi-ordering: A frequently discovered concept. *Journal of Combinatorial Theory, Series A*, 13(3):297–305, 1972.
21. A. Kučera and Ph. Schnoebelen. A general approach to comparing infinite-state systems with their finite-state specifications. *Theoretical Computer Science*, 358(2-3):315–333, 2006.
22. D. A. Martin. The determinacy of Blackwell games. *The Journal of Symbolic Logic*, 63(4):1565–1581, 1998.
23. R. Mayr. Undecidable problems in unreliable computations. *Theoretical Computer Science*, 297(1–3):337–354, 2003.
24. E. C. Milner. Basic WQO- and BQO-theory. In I. Rival, editor, *Graphs and Order. The Role of Graphs in the Theory of Ordered Sets and Its Applications*, pages 487–502. D. Reidel Publishing, 1985.
25. J. Ouaknine and J. Worrell. On metric temporal logic and faulty Turing machines. In *Proc. 9th Int. Conf. Foundations of Software Science and Computation Structures (FOSSACS 2006), Vienna, Austria, Mar. 2006*, volume 3921 of *Lecture Notes in Computer Science*, pages 217–230. Springer, 2006.
26. D. Perrin. Finite automata. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 1, pages 1–57. Elsevier Science, 1990.
27. J.-F. Raskin, M. Samuelides, and L. Van Begin. Petri games are monotonic but difficult to decide. Tech. Report 2003.21, Centre Fédéré en Vérification, 2003. Available at http://www.ulb.ac.be/di/ssd/cfv/TechReps.
28. J.-F. Raskin, M. Samuelides, and L. Van Begin. Games for counting abstractions. In *Proc. 4th Int. Workshop on Automated Verification of Critical Systems (AVoCS 2004), London, UK, Sep. 2004*, volume 128(6) of *Electronic Notes in Theor. Comp. Sci.*, pages 69–85. Elsevier Science, 2005.
29. Ph. Schnoebelen. The verification of probabilistic lossy channel systems. In *Validation of Stochastic Systems – A Guide to Current Research*, volume 2925 of *Lecture Notes in Computer Science*, pages 445–465. Springer, 2004.
30. J. Sifakis. A unified approach for studying the properties of transitions systems. *Theoretical Computer Science*, 18:227–258, 1982.