From Quantified CTL to QBF

Akash Hossain
IRIF, Univ. Paris Diderot, France
akashoss42@gmail.com

François Laroussinie
IRIF, Univ. Paris Diderot, France
https://www.irif.fr/~francoisl/
francoisl@irif.fr

—— **Abstract** ——————————————————————————————

QCTL extends the temporal logic CTL with quantifications over atomic propositions. This extension
is known to be very expressive: QCTL allows us to express complex properties over Kripke structures
(it is as expressive as MSO). Several semantics exist for the quantifications: here, we work with the
structure semantics, where the extra propositions label the Kripke structure (and not its execution
tree), and the model-checking problem is known to be PSPACE-complete in this framework. We
propose a model-checking algorithm for QCTL based on a reduction to QBF. We consider several
reduction strategies, and we compare them with a prototype (based on the SMT-solver Z3) on
several examples.

1 Introduction

Temporal logics have been introduced in computer science in the late 1970's [14]; they provide
a powerful formalism for specifying correctness properties of evolving systems. Various kinds
of temporal logics have been defined, with different expressiveness and algorithmic properties.
For instance, the *Computation Tree Logic* (CTL) expresses properties of the computation
tree of the system under study (time is branching: a state may have several successors), and
the *Linear-time Temporal Logic* (LTL) expresses properties of one execution at a time (a
system is viewed as a set of executions).

Temporal logics allow *model checking*, i.e. the automatic verification that a finite state
system satisfies its expected behavioral specifications [15, 3]. It is well known that CTL model-
checking is PTIME-complete and LTL model-checking (based on automata techniques) is
PSPACE-complete. Verification tools exist for both logics and model-checking is now com-
monly used in the design of critical reactive systems. The main limitation to this approach
is the state-explosion problem: symbolic techniques (for example with BDD), SAT-based
approaches, or partial order reductions have been developed and they are impressively
successful. The SAT-based model-checking consists in using SAT-solvers in the decision
procedures. It was first developed for bounded model-checking (to search for executions
whose length is bounded by some integer, satisfying some temporal property) which can be
reduced to some satisfiability problem and then can be solved by a SAT-solver [2]. SAT
approaches have also been extended to unbounded verification and combined with other
techniques [12]. Many studies have been done in this area, and it is widely considered as
an important approach in practice, which complements other symbolic techniques like BDD
(see [1] for a survey).

In terms of expressiveness, CTL (or LTL) still has some limitations: in particular, it lacks the ability of *counting*. For instance, it cannot express that an event occurs (at least) at every even position along a path, or that a state has two successors. In order to cope with this, temporal logics have been extended with *propositional quantifiers* [16]: those quantifiers allow for adding fresh atomic propositions in the model before evaluating the truth value of a temporal-logic formula. That a state has at least two successors can then be expressed (in *quantified* CTL, hereafter written QCTL) by saying that it is possible to label the model with atomic proposition p in such a way that there is a successor that is labelled with p and one that is not.

Different semantics for QCTL have been studied in the literature depending on the definition of the labelling: either it refers to the finite-state model – it is the *structure semantics* – or it refers to the execution tree – it is the *tree semantics*. Both semantics are interesting and have been extensively studied [9, 7, 13, 8, 4, 10]. While the tree semantics allow us to use the tree automata techniques to get decision procedures (model-checking and satisfiability are TOWER-complete [10]), the situation is quite different for the structure semantics: in this framework, model-checking is PSPACE-complete and satisfiability is undecidable [7].

In this paper, we focus on the structure semantics, and we propose a model-checking algorithm based on a reduction to QBF (propositional logic augmented with quantifiers): given a Kripke structure $\mathcal{K}$ and a QCTL formula Φ, we show how to build a QBF formula $\widehat{\Phi}^{\mathcal{K}}$ which is valid iff $\mathcal{K} \models \Phi$. It is natural to use QBF quantifiers to deal with propositional quantifiers of QCTL. Of course, QBF-solvers are not as efficient as SAT-solvers, but still much progress has been made (and QBF-solvers have already been considered for model-checking, as in [6]). We propose several reductions depending on the way of dealing with nested temporal modalities, and we have implemented a prototype (based on Z3 SMT-solver [5]) to compare these reductions over several examples. As far as we know, it is the first implementation of a model-checker for QCTL.

Here, our first objective is to use the QBF-solver as a tool to check complex properties over limited size models, and this is therefore different from the classical use of SAT-based techniques which are precisely applied to solve verification problems for very large systems.

The outline of the paper is as follows: we begin with setting up the necessary formalism in order to define QCTL. We then devote Section 3 to the different reductions to QBF. Finally, Section 4 contains several practical results and examples.

2 Definitions

2.1 Kripke structures

Let AP be a set of atomic propositions.

▶ **Definition 1.** *A* Kripke structure *is a tuple $\mathcal{K} = \langle V, E, \ell \rangle$, where V is a finite set of vertices (or states), $E \subseteq V \times V$ is a set of edges (we assume that for any $x \in V$, there exists $x' \in V$ s.t. $(x, x') \in E$), and $\ell\colon V \to 2^{\mathsf{AP}}$ is a labelling function.*

An infinite path (also called an execution) in a Kripke structure is an infinite sequence $\rho = x_0 x_1 x_2 \ldots$ such that for any i we have $x_i \in V$ and $(x_i, x_{i+1}) \in E$. We write $\mathsf{Path}_{\mathcal{K}}^{\omega}$ for the set of infinite paths of $\mathcal{K}$ and $\mathsf{Path}_{\mathcal{K}}^{\omega}(x)$ for the set of infinite paths issued from $x \in V$. Given such a path ρ, we use $\rho_{\leq i}$ to denote the i-th prefix $x_0 \ldots x_i$, $\rho_{\geq i}$ for the i-th suffix $x_i x_{i+1} \ldots$, and $\rho(i)$ for the vertex x_i. The size of $\mathcal{K}$ is $|V| + |E|$.

Given a set $P \subseteq \mathsf{AP}$, two Kripke structures $\mathcal{K} = (V, E, \ell)$ and $\mathcal{K}' = (V', E', \ell')$ are said P-equivalent (denoted by $\mathcal{K} \equiv_P \mathcal{K}'$) if $V = V'$, $E = E'$, and for every $x \in V$ we have: $\ell(x) \cap P = \ell'(x) \cap P$.

2.2 QCTL

This section is devoted to the definition of the logic QCTL, which extends the classical branching-time temporal logic CTL with quantifications over atomic propositions.

▶ **Definition 2.** *The syntax of* QCTL *is defined by the following grammar:*

$$\mathsf{QCTL} \ni \varphi, \psi ::= q \mid \neg\varphi \mid \varphi \lor \psi \mid \mathbf{EX}\varphi \mid \mathbf{E}\varphi\mathbf{U}\psi \mid \mathbf{A}\varphi\mathbf{U}\psi \mid \exists p.\ \varphi$$

where q and p range over AP.

QCTL formulas are evaluated over states of Kripke structures:

▶ **Definition 3.** *Let $\mathcal{K} = \langle V, E, \ell \rangle$ be a Kripke structure, and $x \in V$. The semantics of* QCTL *formulas is defined inductively as follows:*

$\mathcal{K}, x \models p$ *iff* $p \in \ell(x)$

$\mathcal{K}, x \models \neg\varphi$ *iff* $\mathcal{K}, x \not\models \varphi$

$\mathcal{K}, x \models \varphi \lor \psi$ *iff* $\mathcal{K}, x \models \varphi$ *or* $\mathcal{K}, x \models \psi$

$\mathcal{K}, x \models \mathbf{EX}\varphi$ *iff* $\exists(x, x') \in E$ *s.t.* $\mathcal{K}, x' \models \varphi$

$\mathcal{K}, x \models \mathbf{E}\varphi\mathbf{U}\psi$ *iff* $\exists\rho \in \mathsf{Path}^\omega_\mathcal{K}(x), \exists i \geq 0$ *s.t.* $\mathcal{K}, \rho(i) \models \psi$ *and*

$\qquad\qquad\qquad$ *for any* $0 \leq j < i$, *we have* $\mathcal{K}, \rho(j) \models \varphi$

$\mathcal{K}, x \models \mathbf{A}\varphi\mathbf{U}\psi$ *iff* $\forall\rho \in \mathsf{Path}^\omega_\mathcal{K}(x), \exists i \geq 0$ *s.t.* $\mathcal{K}, \rho(i) \models \psi$ *and*

$\qquad\qquad\qquad$ *for any* $0 \leq j < i$, *we have* $\mathcal{K}, \rho(j) \models \varphi$

$\mathcal{K}, x \models \exists p.\ \varphi$ *iff* $\exists\mathcal{K}' \equiv_{\mathsf{AP}\setminus\{p\}} \mathcal{K}$ *s.t.* $\mathcal{K}', x \models \varphi$

In the sequel, we use standard abbreviations such as $\top, \bot, \land, \Rightarrow$ and $\Leftrightarrow$. We also use the additional (classical) temporal modalities of CTL: $\mathbf{AX}\varphi = \neg\mathbf{EX}\neg\varphi$, $\mathbf{EF}\varphi = \mathbf{E}\top\mathbf{U}\varphi$, $\mathbf{AF}\varphi = \mathbf{A}\top\mathbf{U}\varphi$, $\mathbf{EG}\varphi = \neg\mathbf{AF}\neg\varphi$, $\mathbf{AG}\varphi = \neg\mathbf{EF}\neg\varphi$, $\mathbf{E}\varphi\mathbf{W}\psi = \neg\mathbf{A}\neg\psi\mathbf{U}(\neg\psi \land \neg\varphi)$ and $\mathbf{A}\varphi\mathbf{W}\psi = \neg\mathbf{E}\neg\psi\mathbf{U}(\neg\psi \land \neg\varphi)$.

Moreover, we use the following abbreviations related to quantifiers over atomic propositions: $\forall p.\ \varphi = \neg\exists p.\ \neg\varphi$, and for a set $P = \{p_1, \ldots, p_k\} \subseteq \mathsf{AP}$, we write $\exists P.\varphi$ for $\exists p_1.\ldots.\exists p_k.\varphi$ and $\forall P.\varphi$ for $\forall p_1.\ldots.\forall p_k.\varphi$.

The size of a formula $\varphi \in \mathsf{QCTL}$, denoted $|\varphi|$, is defined inductively by: $|q| = 1$, $|\neg\varphi| = |\exists p.\varphi| = |\mathbf{EX}\varphi| = 1 + |\varphi|$, $|\varphi \lor \psi| = |\mathbf{E}\varphi\mathbf{U}\psi| = |\mathbf{A}\varphi\mathbf{U}\psi| = 1 + |\varphi| + |\psi|$. Moreover we use $\mathsf{ht}(\varphi)$ to denote the *temporal height* of φ, that is the maximal number of nested temporal modalities in φ. And given a subformula ψ in Φ, the *temporal depth* of ψ in Φ (denoted $\mathsf{td}_\Phi(\psi)$) is the number of temporal modalities having ψ in their scope.

In the following, we denote by $\mathsf{SubF}(\Phi)$ (resp. $\mathsf{SubTF}(\Phi)$) the set of subformulas of Φ (resp. the set of subformulas starting with a temporal modality).

Two QCTL formulas φ and ψ are said to be *equivalent* (written $\varphi \equiv \psi$) iff for any structure $\mathcal{K}$, any state x, we have $\mathcal{K}, x \models \varphi$ iff $\mathcal{K}, x \models \psi$. This equivalence is substitutive.

2.2.1 Discussion on the semantics

The semantics we defined is classically called the *structure semantics*: a formula $\exists p.\varphi$ holds true in a Kripke structure $\mathcal{K}$ iff there exists a p-labelling of the structure $\mathcal{K}$ such that φ is satisfied. Another well-known semantics coexists in the literature for propositional quantifiers,

the *tree semantics*: $\exists p.\ \varphi$ holds true when there exists a p-labelling of the *execution tree* (the infinite unfolding) of the Kripke structure under which φ holds. If, for CTL, interpreting formulas over the structure or the execution tree is equivalent, this is not the case for QCTL. Moreover, these two semantics do not have the same algorithmic properties: if QCTL model-checking and satisfiability are TOWER-complete for the tree semantics (the algorithms are based on tree automata techniques), QCTL model-checking is PSPACE-complete for the structure semantics but satisfiability is undecidable. (see [10] for a survey). Nevertheless, in both semantics, QCTL and QCTL* (the extension of CTL* with quantifications) are equally expressive, and are as expressive [1] as the Monadic Second-Order Logic over the finite structure or the infinite trees (depending on the semantics). Note also that any QCTL formula is equivalent to a formula in Prenex normal form (we will use this result in next sections).

Finally, there is also the *amorphous semantics* [7], where $\exists p.\ \varphi$ holds true at a state s in some Kripke structure $\mathcal{K}$ if, and only if, there exists some Kripke structure $\mathcal{K}'$ with a state s' such that s and s' are bisimilar, and for which there exists a p-labelling making φ hold true at s'. With these semantics, the logic is insensitive to unwinding, and more generally it is bisimulation-invariant (contrary to the two previous semantics, see below).

2.3   Examples of QCTL formulas

QCTL allows us to express complex properties over Kripke structures: for example, we can build a characteristic formula (up to isomorphism) of a structure, we can reduce model-checking problems for multi-player games to QCTL model-checking [11]…Below, we give several examples of counting properties, to illustrate the expressive power of propositional quantifiers.

The first one of the formulas below expresses that there exists a unique reachable state satisfying φ, and the second one states that there exists a unique immediate successor satisfying φ:

$$\mathbf{E}_{=1}\mathbf{F}\varphi \ = \ \mathbf{EF}\varphi \wedge \forall p.\big(\mathbf{EF}(p \wedge \varphi) \Rightarrow \mathbf{AG}(\varphi \Rightarrow p)\big) \tag{1}$$

$$\mathbf{E}_{=1}\mathbf{X}\varphi \ = \ \mathbf{EX}\varphi \wedge \forall p.\ \big(\mathbf{AX}(\varphi \Rightarrow p) \vee \mathbf{AX}(\varphi \Rightarrow \neg p)\big) \tag{2}$$

where we assume that p does not appear in φ. Consider the formula 1: if there were two reachable states satisfying φ, then labelling only one of them with p would falsify the $\mathbf{AG}$ subformula. For 2, the argument is similar.

The existence of at least k successors satisfying a given property can be expressed with:

$$\mathbf{E}_{\geq k}\mathbf{X}\varphi = \exists P.\ \Big(\bigwedge_{1 \leq i \leq k} \mathbf{EX}\big(p_i \wedge \bigwedge_{i' \neq i} \neg p_{i'}\big) \wedge \mathbf{AX}\Big(\big(\bigvee_{1 \leq i \leq k} p_i\big) \Rightarrow \varphi\Big)\Big) \tag{3}$$

And we can define $\mathbf{E}_{=k}\mathbf{X}\varphi$ as $\mathbf{E}_{\geq k}\mathbf{X}\varphi \ \wedge \ \neg\mathbf{E}_{\geq k+1}\mathbf{X}\varphi$. Note that these examples show why QCTL formulas are not bisimulation-invariant.

When using QCTL to specify properties, one often needs to quantify (existentially or universally) over *one* reachable state we want to mark with a given atomic proposition. To this aim, we add the following abbreviations:

$$\exists^1 p.\varphi = \exists p.\big((\mathbf{E}_{=1}\mathbf{F}\ p) \wedge \varphi\big) \qquad \forall^1 p.\varphi = \forall p.\big((\mathbf{E}_{=1}\mathbf{F}\ p) \Rightarrow \varphi\big)$$

---

[1] This requires adequate definitions, since a temporal logic formula may only deal with the reachable part of the model, while MSO has a more *global* point of view.

3 Model-checking QCTL

Model-checking QCTL is a PSPACE-complete problem (for detailed results about program complexity and formula complexity, see [10]), and it is NP-complete for the restricted set of formulas of the form $\exists P.\varphi$, with $P \subseteq \mathsf{AP}$ and $\varphi \in \mathsf{CTL}$ [9]. In this section, we give a reduction from the QCTL model-checking problem to QBF.

In the following, we assume a Kripke structure $\mathcal{K} = \langle V, E, \ell \rangle$, an initial state $x_0 \in V$ and a QCTL formula Φ to be fixed. Let V be $\{x_0, \ldots, x_n\}$. We also assume w.l.o.g. that every quantifier $\exists$ and $\forall$ in Φ introduces a fresh atomic proposition, and distinct from the propositions used in $\mathcal{K}$. We use AP_Q^Φ to denote the set of quantified atomic propositions in Φ.

These assumptions allow us to use an alternative notation for the semantics of Φ-subformulas: the truth value of φ will be defined for a state x in $\mathcal{K}$ within an *environment* $\varepsilon : \mathsf{AP}_Q^\Phi \mapsto 2^V$, that is a partial mapping associating a subset of vertices to a proposition in AP_Q^Φ. We use $\mathcal{K}, x \models_\varepsilon \varphi$ to denote that φ holds at x in $\mathcal{K}$ within ε. This ensures that the $\mathcal{K}$'s labelling ℓ is not modified when a subformula is evaluated, only ε is extended with labellings for new quantified propositions. Formally the main changes of the semantics are as follows:

$$\mathcal{K}, x \models_\varepsilon p \text{ iff } \Big((p \in \mathsf{AP}_Q^\Phi \text{ and } x \in \varepsilon(p)) \text{ or } (p \notin \mathsf{AP}_Q^\Phi \text{ and } p \in \ell(x)) \Big)$$

$$\mathcal{K}, x \models_\varepsilon \exists p. \varphi \text{ iff } \exists V' \subseteq V \text{ s.t. } \mathcal{K}, x \models_{\varepsilon[p \mapsto V']} \varphi$$

where $\varepsilon[p \mapsto V']$ denotes the mapping which coincides with ε for every proposition in $AP_Q^\Phi \setminus \{p\}$ and associates V' to p.

We use this new notation in order to better distinguish initial $\mathcal{K}$'s propositions and quantified propositions to make proofs simpler. Of course, there is no semantic difference: $\mathcal{K}, x \models \Phi$ iff $\mathcal{K}, x \models_\emptyset \Phi$.

In next sections, we consider general *quantified propositional formulas* (QBF) of the form:

$$\mathsf{QBF} \ni \alpha, \beta ::= q \mid \alpha \wedge \beta \mid \alpha \vee \beta \mid \neg\alpha \mid \alpha \Leftrightarrow \beta \mid \alpha \Rightarrow \beta \mid \exists q.\alpha \mid \forall q.\alpha$$

The formal semantics of a formula α is defined over a Boolean valuation for free variables in α (*i.e.* propositions which are not bound by a quantifier [2]), and it is defined as usual. A formula is said to be closed when it does not contain free variables. In the following, we use the standard notion of validity for closed QBF formulas.

Our aim is then to build a (closed) QBF formula $\widehat{\Phi}^{x_0}$ such that $\widehat{\Phi}^{x_0}$ is valid iff Φ holds true at x_0 in $\mathcal{K}$.

3.1 Overview

We present several reductions of QCTL model-checking problem to QBF validity problem. Given a Φ-subformula φ, a vertex $x \in V$, and a subset $P \subseteq AP_Q^\Phi$, we define a QBF formula $\widehat{\varphi}^{x,P}$ whose variables belong to $\mathsf{AP}_Q^\Phi \times V$ (in the following, we use the notation p^x for $p \in \mathsf{AP}_Q^\Phi$ and $x \in V$). The first two reductions are based on different encodings of temporal modalities, but share a common part given in Figure 1.

---

[2] We assume w.l.o.g. that every quantifier $\exists$ and $\forall$ introduces a new proposition.

$$\widehat{\neg\varphi}^{x,P} = \neg\widehat{\varphi}^{x,P} \qquad \widehat{\varphi \vee \psi}^{x,P} = \widehat{\varphi}^{x,P} \vee \widehat{\psi}^{x,P} \qquad \widehat{\mathbf{EX}\varphi}^{x,P} = \bigvee_{(x,x')\in E} \widehat{\varphi}^{x',P}$$

$$\widehat{\exists p.\varphi}^{x,P} = \exists p^{x_0}\dots p^{x_n}.\widehat{\varphi}^{x,P\cup\{p\}} \qquad \widehat{p}^{x,P} = \begin{cases} p^x & \text{if } p \in P \\ \top & \text{if } p \notin P \text{ and } p \in \ell(x) \\ \bot & \text{otherwise} \end{cases}$$

Figure 1 Reduction for basic modalities for reductions UU and FP.

$$\widehat{\mathbf{E}\varphi\mathbf{U}\psi}^{x,P} = \overline{\mathbf{E}\varphi\mathbf{U}\psi}^{x,P,\{x\}} \qquad \text{with:}$$
$$\overline{\mathbf{E}\varphi\mathbf{U}\psi}^{x,P,X} = \widehat{\psi}^{x,P} \vee \left(\widehat{\varphi}^{x,P} \wedge \bigvee_{(x,x')\in E\,s.t.\,x'\notin X} \overline{\mathbf{E}\varphi\mathbf{U}\psi}^{x',P,X\cup\{x'\}} \right)$$
$$\widehat{\mathbf{A}\varphi\mathbf{U}\psi}^{x,P} = \overline{\mathbf{A}\varphi\mathbf{U}\psi}^{x,P,\{x\}} \qquad \text{with:}$$
$$\overline{\mathbf{A}\varphi\mathbf{U}\psi}^{x,P,X} = \begin{cases} \widehat{\psi}^{x,P} & \text{if } \exists (x,x')\in E,\ x'\in X \\ \widehat{\psi}^{x,P} \vee \left(\widehat{\varphi}^{x,P} \wedge \bigwedge_{(x,x')\in E} \overline{\mathbf{A}\varphi\mathbf{U}\psi}^{x',P,X\cup\{x'\}} \right) & \text{otherwise} \end{cases}$$

Figure 2 Reduction for temporal modalities **EU** and **AU** – variant UU.

3.1.1 Unfolding characterization of the until operators

First, we can complete previous construction rules of Figure 1 with those of Figure 2 to get the first method (called UU). This is a naive approach consisting in encoding the temporal modalities as unfoldings of the transition relation. The rules of Figure 2 can be seen as a depth-first way to look for a path satisfying an Until modality (for **EU**) or its negation (for **AU**) among all the simple paths issued from x.

Before stating the correctness of the construction, we need to associate a Boolean valuation v_ε for variables in $\mathsf{AP}_Q^\Phi \times V$ to an environment ε for AP_Q^Φ. We define v_ε as follows: for any $p \in \mathsf{AP}_Q^\Phi$ and $x \in V$, $v_\varepsilon(p^x) = \top$ iff $x \in \varepsilon(p)$.

Now we have the following theorem whose proof is in Appendix A:

▶ **Theorem 4.** *Given a* QCTL *formula* Φ, *a Kripke structure* $\mathcal{K} = \langle V, E, \ell \rangle$, *a state* $x \in V$, *an environment* $\varepsilon : \mathsf{AP}_Q^\Phi \mapsto 2^V$ *and a* Φ-*subformula* φ, *if* $\widehat{\varphi}^{x,\mathsf{dom}(\varepsilon)}$ *is defined inductively w.r.t. the rules of Figures 1 and 2, we have:* $\mathcal{K}, x \models_\varepsilon \varphi \quad$ *iff* $\quad v_\varepsilon \models \widehat{\varphi}^{x,\mathsf{dom}(\varepsilon)}$

It remains to define $\widehat{\Phi}^x$ as $\widehat{\Phi}^{x,\emptyset}$ and we get the reduction: $\mathcal{K}, x_0 \models \Phi$ iff $\widehat{\Phi}^{x_0}$ is valid.

The main drawback of this reduction is the size of the QBF formula: indeed any Until modality may induce a formula whose size is in $O(|V|!)$, and the size of the resulting formula $\widehat{\Phi}^{x_0}$ is then in $O((|\Phi| \cdot |V|!)^{\mathsf{ht}(\Phi)})$. Nevertheless, one can notice that the reduction does not use new quantified propositons to encode the temporal modalities, contrary to other methods we will see later.

3.2 Fixed point characterization of the until operators

Here we present the fixed point method (called FP) for dealing with the modalities **AU** and **EU**. Let φ and ψ be two QCTL-formulas. The idea of the method is to build a QCTL formula that is equivalent to $\mathbf{E}\varphi\mathbf{U}\psi$ (or $\mathbf{A}\varphi\mathbf{U}\psi$), using only the modalities **EX**, **AX** and **AG**.

We first have the following lemma:

▶ **Lemma 5.** *For any* QCTL *formula* $\mathbf{E}\varphi\mathbf{U}\psi$*, we have:*

$$\mathbf{E}\varphi\mathbf{U}\psi \quad \equiv \quad \forall z.\Big(\mathbf{AG}\big(z \Leftrightarrow (\psi \vee (\varphi \wedge \mathbf{EX}\, z))\big) \;\Rightarrow\; z\Big)$$

Proof. Let x be a state in a Kripke structure $\mathcal{K}$. Let θ be the formula $\big(\mathbf{AG}\big(z \Leftrightarrow (\psi \vee (\varphi \wedge \mathbf{EX}\, z))\big) \;\Rightarrow\; z\big)$.

Assume $\mathcal{K}, x \models \mathbf{E}\varphi\mathbf{U}\psi$. We can use the standard characterization of **EU** as fixed point: x belongs to the least fixed point of the equation $Z = \psi \vee (\varphi \wedge \mathbf{EX}\, Z)$ where ψ (resp. φ) is here interpreted as the set of states satisfying ψ (resp. φ). Therefore any z-labelling of reachable states from x [3] corresponding to a fixed point will have the state x labelled. This is precisely what is specified by the QCTL formula.

Now if $\mathcal{K}, x \models \theta$ for every z-labelling corresponding to a fixed point of the previous equation, this is the case for the z-labelling of the states reachable from x and satisfying $\mathbf{E}\varphi\mathbf{U}\psi$, and we deduce $\mathcal{K}, x \models \mathbf{E}\varphi\mathbf{U}\psi$. ◀

And we have the same result for **AU**:

▶ **Lemma 6.** *For any* QCTL *formula* $\mathbf{A}\varphi\mathbf{U}\psi$*, we have:*

$$\mathbf{A}\varphi\mathbf{U}\psi \quad \equiv \quad \forall z.\Big(\mathbf{AG}\big(z \Leftrightarrow (\psi \vee (\varphi \wedge \mathbf{AX}\, z))\big) \;\Rightarrow\; z\Big)$$

Proof. Similar to the proof of Lemma 5. ◀

As a direct consequence, we get the following result:

▶ **Proposition 7.** *For any* QCTL *formula* Φ*, we can build an equivalent* QCTL *formula* $\mathsf{fpc}(\Phi)$ *such that: (1)* $\mathsf{fpc}(\Phi)$ *is built up from atomic propositions, Boolean operators, propositional quantifiers and modalities* **EX** *and* **AG***, and (2) the size of* $\mathsf{fpc}(\Phi)$ *is linear in* $|\Phi|$*.*

Note that the size of $\mathsf{fpc}(\Phi)$ comes from the fact that there is no duplication of subformulas when applying the transformation rules based on equivalences of Lemmas 5 and 6. Moreover, the temporal height of $\mathsf{fpc}(\Phi)$ is smaller than $\mathsf{ht}(\Phi) + 1$[4].

And to translate $\mathsf{fpc}(\Phi)$ into QBF, it remains to add a single rule [5] to the definitions of Figure 1 to deal with **AG**:

$$\widehat{\mathbf{AG}\varphi}^{x,P} = \bigwedge_{(x,y)\in E^*} \widehat{\varphi}^{\,y,P} \tag{4}$$

Where E^* is the reflexive and transitive closure of E. Then we have:

---

[3] Labelling other states does not matter.
[4] The temporal height will be increased by 1 if Φ has an until operator whose members are Boolean combinations of atomic propositions.
[5] For **AX** we can either change the rule for **EX** with a disjunction, or express it with **EX** and $\neg$.

▶ **Theorem 8.** *Given a* QCTL *formula* Φ*, a Kripke structure* $\mathcal{K} = \langle V, E, \ell \rangle$*, a state* $x \in V$ *and an environment* $\varepsilon : \mathsf{AP}_Q^\Phi \mapsto 2^V$*, for any* Φ*-subformula* φ*, if* $\widehat{\mathsf{fpc}(\varphi)}^{x, \mathsf{dom}(\varepsilon)}$ *is defined inductively w.r.t. the rules of Figure 1 and the rule 4, we have:*

$$\mathcal{K}, x \models_\varepsilon \varphi \quad \textit{iff} \quad v_\varepsilon \models \widehat{\mathsf{fpc}(\varphi)}^{x, \mathsf{dom}(\varepsilon)}$$

Proof. The proof is a direct consequence of Proposition 7. ◀

With this approach, the size of $\widehat{\mathsf{fpc}(\Phi)}^x$ is in $O((|\Phi| \cdot |\mathcal{K}|)^{\mathsf{ht}(\Phi)})$. Indeed, an **EU** (or **AU**) modality gives rise to a QBF formula of size $(|V| + |E|)$. The exponential size comes from the potential nesting of temporal modalities: to avoid it, one could consider the DAG-size of formulas. In the next section, we consider another solution. Note also that the number of propositional variables in the QBF formula is bounded by $|\mathsf{AP}_Q^\Phi| \cdot |V|$.

3.3   Reduction via flat formulas

To avoid the size explosion of $\widehat{\Phi}^x$, one can use an alternative approach for Prenex QCTL formulas. Remember that any QCTL formula can be translated into an equivalent QCTL formula in Prenex normal form whose size is linear in the size of the original formula [10].

In the following, a CTL formula is said to be *basic* when it is of the form $\mathbf{EX}\alpha$, $\mathbf{E}\alpha\mathbf{U}\beta$ or $\mathbf{A}\alpha\mathbf{U}\beta$ where α and β are Boolean combinations of atomic propositions. It is easy to observe that any CTL formula can be translated into a QCTL formula with a temporal height less or equal to 2:

▶ **Proposition 9.** *Any* CTL *formula* Φ *is equivalent to some* QCTL *formula* Ψ *of the form:*

$$\exists\{\kappa_1 \dots \kappa_m\}.\Big(\Phi_0 \wedge \bigwedge_{i=1\dots m} \mathbf{AG}(\kappa_i \Leftrightarrow \theta_i)\Big)$$

where Φ_0 *is a Boolean combination of basic* CTL *formulas and every* θ_i *is a basic* CTL *formula (for any* $1 \leq i \leq m$*). Moreover,* $|\Psi|$ *is in* $O(|\Phi|)$*.*

Proof. Let S_Φ be the set of *temporal* subformulas occurring in Φ at a temporal depth greater or equal to 1. We will prove the proposition by induction over the size of S_Φ. If $|S_\Phi| = 0$, the formula satisfies the property. Now consider a formula with $|S_\Phi| > 0$. Φ must have at least one basic (strict) subformula θ_1. Then Φ is equivalent to the formula $\exists\kappa_1.(\Phi[\theta_1 \leftarrow \kappa_1] \wedge \mathbf{AG}(\kappa_1 \Leftrightarrow \theta_1))$, where κ_1 is a fresh atomic proposition, and $\varphi[\alpha \leftarrow \beta]$ is φ where every occurrence of α is replaced by β. Indeed, any state reachable from the current state x will be labelled by κ_1 iff θ_1 holds true at that state (NB: the states that are not reachable from x do not matter for the truth value of Φ), and this enforces the equivalence. We have $|S_{\Phi[\theta_1 \leftarrow \kappa_1]}| < |S_\Phi|$, thus we can apply induction hypothesis to get:

$$\Phi[\theta_1 \leftarrow \kappa_1] \equiv \exists\{\kappa_2 \dots \kappa_m\}.\Big(\Phi_0 \wedge \bigwedge_{i=2\dots m} \mathbf{AG}(\kappa_i \Leftrightarrow \theta_i)\Big) = \Psi'$$

Where $\kappa_2 \dots \kappa_m$ are fresh atomic propositions, Φ_0 is a Boolean combination of basic CTL formulas and every θ_i is a basic CTL formula. We can conclude that:

$$\Phi \equiv \exists\kappa_1.\Big[\Big(\exists\{\kappa_2 \dots \kappa_m\}.\Big(\Phi_0 \wedge \bigwedge_{i=2\dots m} \mathbf{AG}(\kappa_i \Leftrightarrow \theta_i)\Big)\Big) \wedge \mathbf{AG}(\kappa_1 \Leftrightarrow \theta_1)\Big]$$

$$\equiv \exists\{\kappa_1 \dots \kappa_m\}.\Big(\Phi_0 \wedge \bigwedge_{i=1\dots m} \mathbf{AG}(\kappa_i \Leftrightarrow \theta_i)\Big) = \Psi$$

Note that the last equivalence comes from the fact that no κ_i with $i > 1$ occurs in θ_1. By i.h. the size of Ψ' is linear in $|\Phi[\theta_1 \leftarrow \kappa_1]|$, and the size of $\Phi[\theta_1 \leftarrow \kappa_1]$ is smaller than that of Φ, therefore the size of Ψ is linear in $|\Phi|$. ◀

It is important to note that S_Φ can be described as $\{\varphi_1 \dots \varphi_m\}$, where $\varphi_1 = \theta_1$, and $\varphi_{i+1} = \theta_{i+1}[\kappa_1 \leftarrow \varphi_1, \dots \kappa_i \leftarrow \varphi_i]$. The correspondance $\theta_i - \varphi_i$ will be crucial later on.

In the following, a QCTL formula of the form $\mathcal{Q}.\big(\Phi_0 \wedge \bigwedge_{i=1\dots m} \mathbf{AG}(\kappa_i \Leftrightarrow \theta_i)\big)$, where $\mathcal{Q}$ is a sequence of quantifications, Φ_0 is a Boolean combination of basic CTL formulas, every κ_i is an atomic proposition, and every θ_i (for $i = 1, \dots, m$) is a basic CTL formula, is said to be a *flat* formula.

As a corollary of previous results, we have:

▶ **Proposition 10.** *Any* QCTL *formula Φ is equivalent to some flat formula whose size is linear in $|\Phi|$.*

Given a QCTL formula Φ, we use $\mathsf{flat}(\Phi)$ to denote the flat formula equivalent to Φ, obtained by first translating Φ into Prenex normal form, and then transform the CTL subformula as described in Proposition 9.

Applying method FP to some flat formula provides a QBF formula of polynomial size since a flat formula has a temporal height less or equal to 2:

▶ **Corollary 11.** *Given a* QCTL *formula Φ, a Kripke structure $\mathcal{K} = \langle V, E, \ell \rangle$ and a state x, the* QBF *formula $\widehat{\mathsf{fpc}(\mathsf{flat}(\Phi))}^x$ obtained by applying the rules of Figure 1 and the rule 4 is valid iff $\mathcal{K}, x \models \Phi$. And the size of $\widehat{\mathsf{fpc}(\mathsf{flat}(\Phi))}^x$ is in $O(|V| \cdot (|V| + |E|) \cdot |\Phi|)$.*

Therefore this reduction (called FFP) provides a PSPACE algorithm for QCTL model-checking. But there are two disadvantages to this approach. First, putting the formula into Prenex normal form may increase the number of quantified atomic propositions and the number of alternations (which is *in fine* linear in the number of quantifiers in the original formula) [10]. For example, when extracting a quantifier $\forall$ from some $\mathbf{EX}$ modality, we need to introduce two propositions, this can be seen for the formula $\mathbf{EX}(\forall p.(\mathbf{AX}p \vee \mathbf{AX}\neg p))$ which is translated as:

$$\exists z.\forall p.\forall z'.\Big((\mathbf{EX}(z \wedge z') \Rightarrow \mathbf{AX}(z \Rightarrow z')) \wedge \mathbf{EX}(z \wedge (\mathbf{AX}p \vee \mathbf{AX}\neg p))\Big)$$

where the proposition z is used to mark a state, and z' is used to enforce that only at most one successor is labelled by z. Of course, these two remarks may have a strong impact on the complexity of the decision procedure. Finally, note also that the resulting QBF formula is not in Prenex normal form.

3.4 Variant of FFP

In the previous reduction, the modalities $\mathbf{EU}$ and $\mathbf{AU}$ may introduce an alternation of quantifiers: an atomic proposition κ is introduced by an existential quantifier, and then a universal quantifier introduces a variable z to encode the fixed point characterisation of $\mathbf{U}$. We propose another reduction in order to avoid this alternation: for this, we will use bit vectors (instead of single Boolean values) associated with every state to encode the distance from the current state to a state satisfying the right-hand side of the Until modality. This reduction produces a prenex QBF formula.

$$\widehat{\varphi \wedge \psi}^{x,P} = \widehat{\varphi}^{x,P} \wedge \widehat{\psi}^{x,P} \qquad \widehat{\varphi \vee \psi}^{x,P} = \widehat{\varphi}^{x,P} \vee \widehat{\psi}^{x,P} \qquad \widehat{\neg p}^{x,P} = \neg \widehat{p}^{x,P}$$

$$\widehat{\mathbf{EX}\varphi}^{x,P} = \bigvee_{(x,y) \in E} \widehat{\varphi}^{y,P} \qquad \widehat{\mathbf{AX}\varphi}^{x,P} = \bigwedge_{(x,y) \in E} \widehat{\varphi}^{y,P} \qquad \widehat{\mathbf{AG}\varphi}^{x,P} = \bigwedge_{(x,y) \in E^*} \widehat{\varphi}^{y,P}$$

$$\text{For } Q \in \{\exists, \forall\} \quad \widehat{Qp.\varphi}^{x,P} = Q p^{x_0} \dots p^{x_n} \widehat{\varphi}^{x,P \cup \{p\}}$$

$$\text{For } Q \in \{\exists, \forall\} \quad \widehat{Q\dot{\kappa}_i.\varphi}^{x,P} = \begin{cases} Q\overline{\kappa_i}^{x_0} \dots \overline{\kappa_i}^{x_n}.\widehat{\varphi}^{x,P \cup \{\overline{\kappa_i}\}} & \text{if } \dot{\kappa}_i \text{ encodes} \\ \text{a least fixed point.} \\ Q\kappa_i^{x_0} \dots \kappa_i^{x_n}.\widehat{\varphi}^{x,P \cup \{\kappa_i\}} & \text{otherwise} \end{cases}$$

$$\widehat{p}^{x,P} = \begin{cases} p^x & \text{if } p \in P \text{ and } p \text{ is a Boolean var.} \\ [p^x < |V|] & \text{if } p \in P \text{ and } p \text{ is a Bit vect.} \\ \top & \text{if } p \notin P \wedge p \in \ell(x) \\ \bot & \text{otherwise} \end{cases}$$

Figure 3 Transformation rules for method FBV.

First, we consider a QCTL formula Φ under negation normal form (NNF), where the negation is only applied to atomic propositions. This transformation makes that Φ is built from temporal modalities in $S_{tmod} = \{\mathbf{EX}, \mathbf{AX}, \mathbf{EU}, \mathbf{AU}, \mathbf{EW}, \mathbf{AW}\}$

We can then reformulate Proposition 10 as the following proposition, whose proof is in Appendix B:

▶ **Proposition 12.** *Any* QCTL *formula Φ is equivalent to some* QCTL *formula Ψ in NNF of the form:* $\Psi = \mathcal{Q} \exists \{\kappa_1 \dots \kappa_m\}.\left(\Phi_0 \wedge \bigwedge_{i=1\dots m} \mathbf{AG}(\kappa_i \Rightarrow \theta_i)\right)$ *where $\mathcal{Q}$ is a sequence of quantifications, Φ_0 is a* CTL *formula containing only the temporal modalities* $\mathbf{EX}$*,* $\mathbf{AX}$ *or* $\mathbf{AG}$ *and whose temporal height is less or equal to 1, and every θ_i is a basic* CTL *formula (with $1 \leq i \leq m$). Moreover, $|\Psi|$ is in $O(|\Phi|)$.*

From the previous proposition, we derive a new reduction to QBF (called FBV). For modalities $\mathbf{EW}$ and $\mathbf{AW}$, we use the same encoding as for method FP, except that we use the corresponding Boolean proposition κ_i directly for the greatest fixed point. And for Until-based modalities, corresponding to least fixed points, we will use bit vectors instead of atomic propositions to encode the truth value of $\mathbf{EU}$ or $\mathbf{AU}$: for a formula $\theta_i = \mathbf{E}\varphi\mathbf{U}\psi$, we will consider a bit vector $\overline{\kappa_i}$ of length $\lceil \log(|V| + 1) \rceil$ for every state instead of a single Boolean value κ_i. The idea is that in a state x, the value $\overline{\kappa_i}$ encodes in binary the distance (in terms of number of transitions) from x to a state satisfying ψ along a path satisfying φ. And for $\theta_i = \mathbf{A}\varphi\mathbf{U}\psi$, the value $\overline{\kappa_i}$ encodes the maximal distance before a state satisfying ψ (along a path where φ is true). In the following, such a $\overline{\kappa_i}$ associated to a θ_i based on an Until is called an Until-κ. Note that given a bit vector $\overline{\kappa_i}$ and an integer value d encoded in binary, we will use $[\overline{\kappa_i}^y = d]$ and $[\overline{\kappa_i}^y < d]$ to denote the corresponding propositional formulas over $\overline{\kappa_i}^y$.

The new reduction is based on the rewriting rules of Figure 3 for several operators, and we define the reduction for $\mathbf{AG}(\kappa_i \Rightarrow \theta_i)$ for $\theta_i = \mathbf{E}\varphi\mathbf{W}\psi$ or $\theta_i = \mathbf{A}\varphi\mathbf{W}\psi$ as follows:

$$\overline{\mathbf{AG}(\kappa_i \Rightarrow \mathbf{E}\varphi\mathbf{W}\psi)}^{\,x,P} = \bigwedge_{(x,y)\in E^*} \left(\kappa_i^y \Rightarrow \left(\widehat{\psi}^{y,P} \vee \left(\widehat{\varphi}^{y,P} \wedge \bigvee_{(y,y')\in E} \kappa_i^{y'} \right) \right) \right) \tag{5}$$

$$\overline{\mathbf{AG}(\kappa_i \Rightarrow \mathbf{A}\varphi\mathbf{W}\psi)}^{\,x,P} = \bigwedge_{(x,y)\in E^*} \left(\kappa_i^y \Rightarrow \left(\widehat{\psi}^{y,P} \vee \left(\widehat{\varphi}^{y,P} \wedge \bigwedge_{(y,y')\in E} \kappa_i^{y'} \right) \right) \right) \tag{6}$$

And for $\mathbf{AG}(\overline{\kappa_i} \Rightarrow \theta_i)$ with $\theta_i = \mathbf{E}\varphi\mathbf{U}\psi$ or $\theta_i = \mathbf{A}\varphi\mathbf{U}\psi$, we have:

$$\overline{\mathbf{AG}(\overline{\kappa_i} \Rightarrow \mathbf{E}\varphi\mathbf{U}\psi)}^{\,x,P} = \bigwedge_{(x,y)\in E^*} \left[\left([\overline{\kappa_i}^y = 0] \Rightarrow \widehat{\psi}^{y,P} \right) \right.$$
$$\left. \wedge \bigwedge_{1\le d < |V|} \left([\overline{\kappa_i}^y = d] \Rightarrow \left(\widehat{\varphi}^{y,P} \wedge \bigvee_{(y,y')\in E} [\overline{\kappa_i}^{y'} = d-1] \right) \right) \right] \tag{7}$$

where $[\overline{\kappa_i}^y = d]$ and $[\overline{\kappa_i}^y < d]$ with a given value d correspond to propositional formulas over $\overline{\kappa_i}^y$. And for $\mathbf{AG}(\overline{\kappa_i} \Rightarrow \theta_i)$ with $\theta_i = \mathbf{A}\varphi\mathbf{U}\psi$, we have:

$$\overline{\mathbf{AG}(\overline{\kappa_i} \Rightarrow \mathbf{A}\varphi\mathbf{U}\psi)}^{\,x,P} = \bigwedge_{(x,y)\in E^*} \left[\left([\overline{\kappa_i}^y = 0] \Rightarrow \widehat{\psi}^{y,P} \right) \right.$$
$$\left. \wedge \bigwedge_{1\le d < |V|} \left([\overline{\kappa_i}^y = d] \Rightarrow \left(\widehat{\varphi}^{y,P} \wedge \bigwedge_{(y,y')\in E} [\overline{\kappa_i}^{y'} < d] \right) \right) \right] \tag{8}$$

Note that in the equivalences 7 and 8, we cannot replace the implication by an equivalence: for a state y s.t. $\widehat{\varphi}^{y,P}$, the existence of some successor y' s.t. $[\overline{\kappa_i}^{y'} < d]$ is not sufficient to imply $[\overline{\kappa_i}^y = d]$. This is why we consider only implications here.

Method FBV is defined from previous rules and we get a Prenex QBF formula:

▶ **Corollary 13.** *Given a* QCTL *formula* Φ, *a Kripke structure* $\mathcal{K} = \langle V, E, \ell \rangle$, *and a state* x, *the Prenex* QBF *formula* $\widehat{\mathsf{flat}(\Phi)}^x$ *obtained by applying the rules of Figure 3 and rules 5, 6, 7, and 8 above, is valid iff* $\mathcal{K}, x \models \Phi$, *and* $|\widehat{\mathsf{flat}(\Phi)}^x|$ *is in* $O(|V| \cdot (|V| + |E|) \cdot |\Phi|)$.

This approach allows us to easily adapt the algorithm to bounded model-checking: instead of considering values from 1 to $|V|$ for d in the definition of Until modalities, one can restrict the range to a smaller interval, to get a smaller QBF formula to check. Note also that obtaining a Prenex QBF formula is interesting in practice, because many QBF-solvers require Prenex formulas as inputs.

We can also observe that this reduction proceeds a bit like the classical model-checking algorithm for CTL where for deciding $x \models \Phi$, the truth value of every Φ-subformula is computed in every state of the model. In some case, this may induce additional work compared to methods like FP or UU (for example to decide whether the initial state satisfies a **EU** formula).

3.4.1 Using BitVectors for $\exists^1$ and $\forall^1$

The quantifiers $\exists^1$ and $\forall^1$ are very useful in many specifications. It can be interesting to develop ad-hoc algorithms in order to improve the generated QBF formulas. For the first methods we described, they are translated into propositional formulas (instead of introducing extra quantified atomic propositions as they are formally defined). Another method consists in using bit vectors as in the treatment of Until modalities described above: in this case, these quantifiers introduce a unique bit vector of size $\lceil \log(|V|+1) \rceil$ to store the number of the state

selected by the quantifier (that is the state which will be labelled by the proposition). This method is interesting, since it reduces the number of quantified propositions, it is integrated in the method FBV.

4   Experimental results

In this section, we consider three examples to illustrate the QBF-based model-checking approach for QCTL. The problems we will consider can be solved more efficiently without a QCTL model-checker, but they provide valuable insights on the performance of the reduction strategies, and the properties to be checked cannot be expressed with classical temporal logics. The first example shows that the performances can change significally when we choose different formulas to check, even if they are equivalent. The second problem illustrates how QCTL can naturally express properties related to game theory, and it is a situation where we can observe how bounded model-checking can improve the performance of FBV. In the last example, we check formulas with a large temporal height, thus we can see how much space is gained by flattening the formulas. We wanted to manipulate graphs that could easily be scaled up by tweaking a few parameters, so we chose to use grids.

We have implemented a prototype to try the different reduction strategies [6] . Our tool is available online [7]: given a Kripke structure $\mathcal{K}$ with a state x and a formula Φ, it produces a specification file (corresponding to $\widehat{\Phi}^x$) for the SMT-solver Z3 [5]. The choice of Z3 was motivated by the fact that the generated QBF formulas are not always Prenex, which many QBF-solvers require, unlike Z3.

4.1 k-connectivity

Here, we consider an undirected graph, and we want to check whether there exist (at least) k internally disjoint paths [8] from a vertex x to some vertex y. A classical result in graph theory due to Menger ensures that, given two vertices x and y in a graph G, the minimum number of vertices whose deletion makes that there is no more paths between x and y is equal to the maximum number of internally disjoint paths between these two vertices.

We can encode these two ideas by the following QCTL formulas (interpreted over x):

$$\Phi_k \;=\; \exists p_1 \dots \exists p_{k-1} \Big(\bigwedge_{1 \le i < k} \mathbf{EX}\big(\mathbf{E}(p_i \wedge \bigwedge_{j \ne i} \neg p_j)\,\mathbf{U}\,y\big) \;\wedge\; \mathbf{EX}\,\mathbf{E}(\bigwedge_{1 \le i < k} \neg p_i)\,\mathbf{U}\,y \Big) \qquad (9)$$

$$\Psi_k \;=\; \forall^1 p_1 \dots \forall^1 p_{k-1}\, \mathbf{EX}\,\Big(\mathbf{E}(\bigwedge_{1 \le i < k} \neg p_i)\,\mathbf{U}\,y \Big) \qquad (10)$$

Φ_k uses the labelling by the p_i's to mark the internal vertices of k paths between the current position and the vertex y. The modality $\mathbf{EX}$ is used to consider only the intermediate states (and not the starting state). The formula Ψ_k proceeds differently: the idea is to mark exactly $k-1$ states with $p_1, \dots, p_{k-1}$ and to verify that there still exists at least one path leading to y without going through the states labelled by some p_i.

By Menger's Theorem, we know that these formulas are equivalent over undirected graphs.

---

[6] NB: For reductions FFP and FBV the formula has to be given in Prenex normal form.

[7] `https://www.irif.fr/~francoisl/qctlmc.html`

[8] Two paths $src \leftrightarrow r_1 \leftrightarrow \dots \leftrightarrow r_k \leftrightarrow dest$ and $src \leftrightarrow r'_1 \leftrightarrow \dots \leftrightarrow r'_{k'} \leftrightarrow dest$ are internally disjoint iff $r_i \ne r'_j$ for any $1 \le i \le k$ and $1 \le j \le k'$. And note that with this def., if there is an edge (x, y), there exist k internally disjoint paths from x to y for any k.

We interpret these formulas over Kripke structures $\mathcal{S}_{n,m}$ with $n \geq m$ (see Figure 4) which correspond to two kinds of grids $n \times n$ connected by m edges (these edges are of the form $(q_{i,n}, r_{1,i})$ or $(q_{n,i}, r_{i,1})$. The initial state is $q_{1,1}$ and when evaluating Φ_k or Ψ_k we assume the state $r_{n,n}$ to be labelled by y. In this context, we clearly have that Φ_k and Ψ_k hold for true at $q_{1,1}$ iff $k \leq m$.
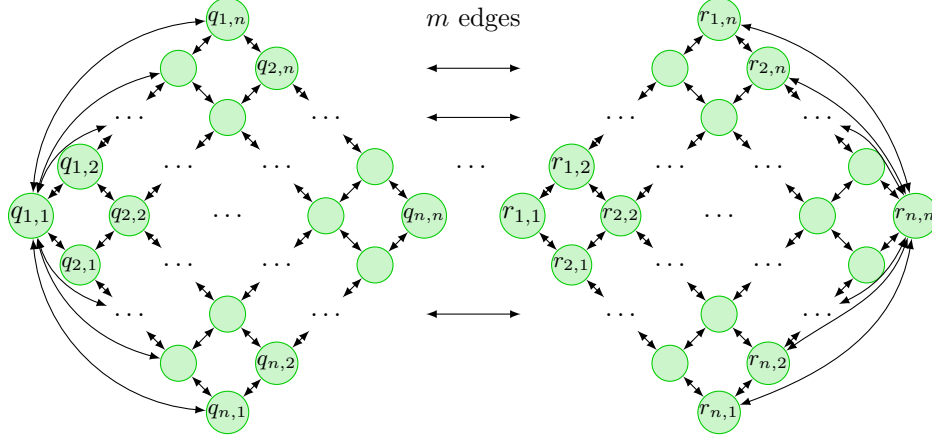
Figure 4 Structure $\mathcal{S}_{n,m}$ for the k-connectivity problem.

Detailed results are presented in Appendix C. The main lessons we can see are:

- Formula Φ_k is much more difficult to verify: the number of temporal modalities is probably one explanation. Another one for methods FP and FFP(based on the fixed point characterization of $\mathbf{U}/\mathbf{W}$ modalities) could be an alternation of quantifiers: in Φ_k, there is an existential quantification over the p_is and the $\mathbf{EU}$s introduce a universal quantification, but it is not the case for Ψ_k, where there are only universal quantifications for these reductions.
- The reduction FP is the most efficient: it can be used to verify models with more than two thousands states when m is small. The method FFP is also rather efficient, but the flattening seems to be too costly for such a simple formula.
- The reduction UU produces very large QBF formulas, but rather simple to check.

We can generalise the problem by verifying that there exist at least k internally disjoint paths between any pair of reachable vertices x and y in a given structure. These previous formulas can be modified as follows:

$$\Phi_k^g = \forall^1 y \exists p_1 \dots \exists p_{k-1} \, \mathbf{AG}\Big(\bigwedge_{1 \leq i < k} \mathbf{EX}\big(\mathbf{E}(p_i \wedge \bigwedge_{j \neq i} \neg p_j) \, \mathbf{U} \, y\big) \, \wedge \, \mathbf{EX}\, \mathbf{E}(\bigwedge_{1 \leq i < k} \neg p_i) \, \mathbf{U} \, y\Big) \tag{11}$$

$$\Psi_k^g = \forall^1 y \forall^1 p_1 \dots \forall^1 p_{k-1} \mathbf{AG}\Big[\mathbf{EX} \Big(\mathbf{E}(\bigwedge_{1 \leq i < k} \neg p_i) \, \mathbf{U} \, y\Big)\Big] \tag{12}$$

In that case, Φ_k^g is useless: too complex to be verified. And the method FP is still the most efficient.

4.2 Nim game

Nim game is a turn-based two-player game. A configuration is a set of heaps of objects and a boolean value indicating whose turn it is. At each turn, a player has to choose one non-empty heap and remove at least one object from it. The aim of each player is to remove

the last object. Given a configuration c and a Player-J with $J \in \{1,2\}$, we can build a finite Kripke structure $\mathcal{S}_J$, where x_c is a state corresponding to the configuration c ; and use a QCTL formula Φ_{win}^J such that $\mathcal{S}, x_c \models \Phi_{\mathsf{win}}^J$ iff Player-J has a wining strategy from c. Note that there is a simple and well-known criterion over the numbers of objects in each heap to decide who has a winning strategy, but we consider this problem just because it is interesting to illustrate what kind of problem we can solve with QCTL.

Each configuration corresponds to a state in $\mathcal{S}_J$. Every move for Player-$\overline{J}$ from a configuration c to a configuration c' provides a transition $(x_c, x_{c'})$ in $\mathcal{S}_J$. However, a move of Player-J from c to c' is encoded as two transitions $x_c \rightarrow x_{c,c'} \rightarrow x_{c'}$ where $x_{c,c'}$ is then an intermediary state we use to encode a strategy for Player-J (marking $x_{c,c'}$ by an atomic proposition will correspond to Player-J choosing c' from c). We assume that every state x_c is labelled by t_1 if it's Player-1's turn to play at c, and by t_2 otherwise. Every intermediary state $x_{c,c'}$ is labelled by int. We also label empty configurations by w_1 or w_2, depending on which player played the last move.

Clearly, the size of $\mathcal{S}$ will depend on the number of objects in each set in the initial configuration. The formula Φ_{win}^J depends only on J:

$$\Phi_{\mathsf{win}}^J \;=\; \exists m. \Big(\mathbf{AG}\big(t_J \;\Rightarrow\; \mathbf{EX}m\big) \;\wedge\; \mathbf{AF}\big(w_J \vee (\mathrm{int} \wedge \neg m)\big)\Big)$$

This formula holds true in a state corresponding to some configuration c iff there exists a labelling by m such that every reachable configuration where it's Player-J's turn, has a successor labelled by m (thus a possible choice to do) and every execution from the current state leads to either a winning state for Player-J or a non-selected intermediary state, therefore all outcomes induced by the underlying strategy have to verify $\mathbf{F}w_J$. Note that in this example, the Kripke structure is acyclic (except the self-loops on the ending states).
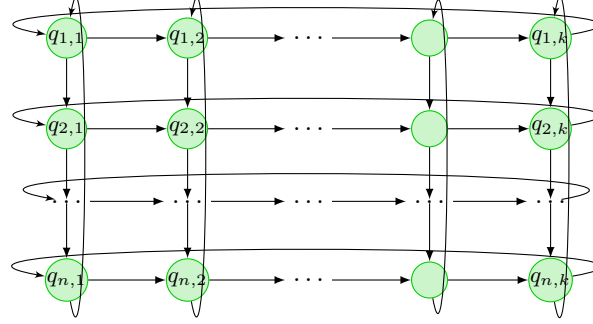
From detailed results in appendix, we can see:

- One can consider structures with more than 10 thousand states. Note that the number of heaps is important for the size of the model, but the maximal length of a game depends on the number of objects (and is rather small in our examples).
- The most efficient method is FFP (with FP).
- Method FBV is more efficient when we consider bounded model-checking. Note that in this case, the verification may not be complete: if the QBF formula is valid, the property is satisfied by the structure, otherwise, no conclusion can be done, except if we can prove that the chosen bound was big enough to be sure that there is no solution. In our case, we can easily compute the maximal bound: at each turn, a player has to pick at least one object, such a move may give rise to one transition in the model (for the opponent), or two transitions (for the player for whom we look for a strategy). Thus, if there are n objects in the initial configuration, we can choose $\frac{3n}{2}$ for the bound.

4.3   Resources distribution

The last example is as follows: given a Kripke structure $\mathcal{S}$ and two integers k and d, we aim at choosing at most k states (called targets in the following) such that every reachable state (from the initial one) can reach a target in less than d transitions. This problem can be encoded with the following QCTL formula where d modalities $\mathbf{EX}$ are nested:

$$\Phi_{\mathsf{res}} \;=\; \exists^1 c_1 \dots \exists^1 c_k \, \mathbf{AG}\Big((\bigvee_{1 \le i \le k} c_i) \vee \mathbf{EX}\Big((\bigvee_{1 \le i \le k} c_i) \vee \Big(\dots \vee \mathbf{EX}(\bigvee_{1 \le i \le k} c_i) \Big)\Big)\Big)$$

Figure 5 Structure $\mathcal{K}_{n,k}$ for the resources distribution problem.

For experimental results, we consider the grid $\mathcal{K}_{n,m}$ described at Figure 5. Note that for this example, reductions UU and FP are similar because there is no Until in the formula (**AG** is treated as a conjunction). From detailed results in appendix, one can see:

- Only small models have been successfully verified.

- The nesting of **EX**s operators give an advantage to the reduction based on flattening (FBV and FFP): the size of the QBF formula increases more slowly.

- The reduction FBV is the most efficient on this example. Since there is no Until modality (except behind **AG** which is treated separately), the difference with FFP is due to the encoding of $\exists_1$ operator with a unique bit vector in FBV, this choice seems to be more efficient in this example.

5 Conclusion

We have presented several reductions from QCTL model-checking to QBF. This provides a first tool for QCTL model-checking. Of course, this is an ongoing-work, and many improvements are possible: the reduction strategies are still naive and could be significantly improved, and a better understanding of QBF-solvers would also be helpful to produce more efficient formulas (we have not yet tried to normalise formulas in a specific form which is often a crucial aspect in SAT/QBF-solving). Still, these first results are rather interesting and encouraging. They show the importance of writing "good" QCTL formulas for which the solver will be able to provide a result (this problem already exists for classical temporal logics, but it is more significant here due to the complexity induced by the quantifications). The examples also show that there is no "one best strategy": it depends on the structure of the considered formula, and then offering several reduction strategies seems to be necessary in a QBF-based model-checker for QCTL. Finally this work is also interesting because it could easily be adapted for other logics (like Sabotage logics [17]). In particular, we plan to figure how it could be adapted for LTL. In the future, we plan to continue to work on reduction strategies, and to use other QBF-solvers.

References

1   Armin Biere, Alessandro Cimatti, Edmund M. Clarke, Ofer Strichman, and Yunshan Zhu. Bounded model checking. *Advances in Computers*, 58:117–148, 2003.

2   Armin Biere, Alessandro Cimatti, Edmund M. Clarke, and Yunshan Zhu. Symbolic Model Checking without BDDs. In *Tools and Algorithms for Construction and Analysis of Systems, 5th International Conference, TACAS '99, Amsterdam, The Netherlands, March 22-28, 1999, Proceedings*, volume 1579 of *Lecture Notes in Computer Science*, pages 193–207. Springer, 1999.

3   Edmund M. Clarke and E. Allen Emerson. Design and Synthesis of Synchronization Skeletons using Branching-Time Temporal Logic. In Dexter C. Kozen, editor, *Proceedings of the 3rd Workshop on Logics of Programs (LOP'81)*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71. Springer-Verlag, 1982.

4   Arnaud Da Costa, François Laroussinie, and Nicolas Markey. Quantified CTL: Expressiveness and Model Checking. In Maciej Koutny and Irek Ulidowski, editors, *Proceedings of the 23rd International Conference on Concurrency Theory (CONCUR'12)*, volume 7454 of *Lecture Notes in Computer Science*, pages 177–192. Springer-Verlag, September 2012.

5   Leonardo Mendonça de Moura and Nikolaj Bjørner. Z3: an efficient SMT solver. In *Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Budapest, Hungary, 2008. Proceedings*, volume 4963 of *Lecture Notes in Computer Science*, pages 337–340. Springer, 2008.

6   Nachum Dershowitz, Ziyad Hanna, and Jacob Katz. Bounded Model Checking with QBF. In *Theory and Applications of Satisfiability Testing, 8th International Conference, SAT 2005, St. Andrews, UK, June 19-23, 2005, Proceedings*, volume 3569 of *Lecture Notes in Computer Science*, pages 408–414. Springer, 2005.

7   Tim French. Decidability of Quantified Propositional Branching Time Logics. In Markus Stumptner, Dan Corbett, and Mike Brooks, editors, *Proceedings of the 14th Australian Joint Conference on Artificial Intelligence (AJCAI'01)*, volume 2256 of *Lecture Notes in Computer Science*, pages 165–176. Springer-Verlag, December 2001.

8   Tim French. Quantified Propositional Temporal Logic with Repeating States. In *Proceedings of the 10th International Symposium on Temporal Representation and Reasoning and of the 4th International Conference on Temporal Logic (TIME-ICTL'03)*, pages 155–165. IEEE Comp. Soc. Press, July 2003.

9   Orna Kupferman. Augmenting Branching Temporal Logics with Existential Quantification over Atomic Propositions. In Pierre Wolper, editor, *Proceedings of the 7th International Conference on Computer Aided Verification (CAV'95)*, volume 939 of *Lecture Notes in Computer Science*, pages 325–338. Springer-Verlag, July 1995.

10   François Laroussinie and Nicolas Markey. Quantified CTL: Expressiveness and Complexity. *Logical Methods in Computer Science*, 10(4), 2014.

11   François Laroussinie and Nicolas Markey. Augmenting ATL with strategy contexts. *Inf. Comput.*, 245:98–123, 2015.

12   Kenneth L. McMillan. Applying SAT Methods in Unbounded Symbolic Model Checking. In *Computer Aided Verification, 14th International Conference, CAV 2002, Copenhagen, Denmark, July 27-31, 2002, Proceedings*, volume 2404 of *Lecture Notes in Computer Science*, pages 250–264. Springer, 2002.

13   Anindya C. Patthak, Indrajit Bhattacharya, Anirban Dasgupta, Pallab Dasgupta, and P. P. Chakrabarti. Quantified Computation Tree Logic. *Information Processing Letters*, 82(3):123–129, 2002.

14   Amir Pnueli. The Temporal Logic of Programs. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS'77)*, pages 46–57. IEEE Comp. Soc. Press, October-November 1977.

15   Jean-Pierre Queille and Joseph Sifakis. Specification and verification of concurrent systems in CESAR. In Mariangiola Dezani-Ciancaglini and Ugo Montanari, editors, *Proceedings of the 5th International Symposium on Programming (SOP'82)*, volume 137 of *Lecture Notes in Computer Science*, pages 337–351. Springer-Verlag, April 1982.

16 A. Prasad Sistla. *Theoretical Issues in the Design and Verification of Distributed Systems.* PhD thesis, Harvard University, Cambridge, Massachussets, USA, 1983.

17 Johan van Benthem. An Essay on Sabotage and Obstruction. In *Mechanizing Mathematical Reasoning, Essays in Honor of Jörg H. Siekmann on the Occasion of His 60th Birthday*, volume 2605 of *Lecture Notes in Computer Science*, pages 268–276. Springer, 2005.

A   Proof of Theorem 4

Proof. We assume Φ to be fixed and we prove the property by structural induction over φ. Boolean operators are omitted.

- $\varphi = p$: if $\mathcal{K}, x \models_\varepsilon p$, then either p is a quantified proposition and it belongs to $\mathsf{dom}(\varepsilon)$ and x belongs to $\varepsilon(p)$ and thus $v_\varepsilon \models p^x$ by def. of v_ε, or p belongs to $\ell(x)$. In both cases we have $v_\varepsilon \models \widehat{p}^{x,\mathsf{dom}(\varepsilon)}$. The converse is similar.

- $\varphi = \mathbf{EX}\psi$: $\mathcal{K}, x \models_\varepsilon \mathbf{EX}\psi$ iff there exists $(x, x') \in E$ s.t. $\mathcal{K}, x' \models_\varepsilon \psi$, iff (by i.h.) there exists $(x, x') \in E$ s.t. $v_\varepsilon \models \widehat{\psi}^{x',\mathsf{dom}(\varepsilon)}$ which is equivalent to $v_\varepsilon \models \widehat{\mathbf{EX}\psi}^{x,\mathsf{dom}(\varepsilon)}$.

- $\varphi = \exists p.\psi$. We have $\mathcal{K}, x \models_\varepsilon \exists p.\psi$ iff there exists $V' \subseteq V$ s.t. $\mathcal{K}, x \models_{\varepsilon[p \mapsto V']} \psi$, iff (i.h.) $v_{\varepsilon[p \mapsto V']} \models \widehat{\psi}^{x,\mathsf{dom}(\varepsilon) \cup \{p\}}$ which is equivalent to $v_\varepsilon \models \exists p^{v_1} \ldots p^{v_n}.\widehat{\psi}^{x,\mathsf{dom}(\varepsilon) \cup \{p\}}$ (by def. of v_ε).

- $\varphi = \mathbf{E}\psi_1 \mathbf{U}\psi_2$: The definition of $\widehat{\varphi}^{x,\mathsf{dom}(\varepsilon)}$ corresponds to a finite unfolding of the expansion law that characterizes the $\mathbf{EU}$ modality. Assume $\mathcal{K}, x \models_\varepsilon \varphi$. There exists a path $\rho \in \mathsf{Path}_\mathcal{K}^\omega(x)$ and a position $i \geq 0$ s.t. $\rho(i) \models_\varepsilon \psi_2$ and $\rho(k) \models_\varepsilon \psi_1$ for any $0 \leq k < i$. The finite prefix $x = \rho(0) \cdots \rho(k)$ can be assumed to be simple, and then $k < |V|$. By using i.h., we get $v_\varepsilon \models \widehat{\psi_2}^{\rho(i),\mathsf{dom}(\varepsilon)}$, and $v_\varepsilon \models \widehat{\psi_1}^{\rho(k),\mathsf{dom}(\varepsilon)}$ for any $0 \leq k < i$. From this point, the reader can easily verify by induction (starting at i, down to 0) that $v_\varepsilon \models \overline{\mathbf{E}\psi_1\mathbf{U}\psi_2}^{\rho(k),dom(\varepsilon),\{\rho(j)|j \leq k\}}$ for all $k \leq i$. This makes $\widehat{\varphi}^{x,\mathsf{dom}(\varepsilon)}$ to be satisfied by v_ε. Conversely, assume $v_\varepsilon \models \widehat{\varphi}^{x,\mathsf{dom}(\varepsilon)}$. Let us build a finite path $x = x_0, x_1, \cdots x_i$ that satisfies $\psi_1\mathbf{U}\psi_2$. The first vertex is of course x, and we have $v_\varepsilon \models \overline{\mathbf{E}\psi_1\mathbf{U}\psi_2}^{x,dom(\varepsilon),\{x\}}$. Fix i so that for every $0 \leq k \leq i$, x_k is built, $v_\varepsilon \models \overline{\mathbf{E}\psi_1\mathbf{U}\psi_2}^{x_k,dom(\varepsilon),\{x_j|j \leq k\}}$, $v_\varepsilon \not\models \widehat{\psi_2}^{x_k,dom(\varepsilon)}$, and $(x_k, x_{k+1}) \in E$ when $k < i$. Then, there must be $(x_i, y) \in E$ so that $y \notin \{x_j | j \leq i\}$, and $v_\varepsilon \models \overline{\mathbf{E}\psi_1\mathbf{U}\psi_2}^{y,dom(\varepsilon),\{x_j|j \leq i\} \cup \{y\}}$, so we set $x_{i+1} = y$. The sequence eventually stops, because V is finite and the path is simple. If x_i is its last vertex, then we must have $v_\varepsilon \models \widehat{\psi_2}^{x_i,dom(\varepsilon)}$, and $v_\varepsilon \models \widehat{\psi_1}^{x_k,dom(\varepsilon)}$ for every $0 \leq k < i$. By using i.h., we get $\mathcal{K}, x \models_\varepsilon \varphi$.

- $\varphi = \mathbf{A}\psi_1\mathbf{U}\psi_2$: this case is similar to the previous one, except that we have to consider loops. Assume $\mathcal{K}, x \models_\varepsilon \varphi$. Then any path issued from x satisfies $\psi_1\mathbf{U}\psi_2$. If x contains a self-loop, then ψ_2 has to be satisfied at x (this is ensured by the first case in the def. of $\widehat{\varphi}^{x,\mathsf{dom}(\varepsilon)}$). Otherwise we consider all the paths from x: either there is a simple prefix witnessing $\psi_1\mathbf{U}\psi_2$, or there is a loop from some point. In the latter case, one of the state in the loop has to verify ψ_2. In both cases, the definition of $\widehat{\varphi}^{x,\mathsf{dom}(\varepsilon)}$ gives the result. ◀

B   Proof of Proposition 12

Proof. Consider w.l.o.g. a QCTL formula Φ in Prenex normal form and NNF. We can define Φ_0 and the basic formulas θ_is approximately as in Proposition 9, except that Φ_0 contains only $\mathbf{EX}$, $\mathbf{AX}$ or $\mathbf{AG}$ modalities, and every other modality gives rise to some quantified proposition κ and a subformula $\mathbf{AG}(\ldots)$ in the main conjunction of Ψ. Every θ_i starts with a modality in S_{tmod}. Let φ_i be the original Φ-subformula associated with θ_i. Note that Ψ is in NNF, and κ_i occurs only once in Ψ in the scope of a negation, and it happens in the

subformula $\mathbf{AG}(\kappa_i \Rightarrow \theta_i)$. We now have to show that Φ is equivalent to Ψ. Consider the formula $\widetilde{\Psi}$ where every $\Rightarrow$ is replaced by $\Leftrightarrow$: by following the same arguments of Proposition 9, we clearly have $\Phi \equiv \widetilde{\Psi}$, and $\widetilde{\Psi} \Rightarrow \Psi$. It remains to prove the opposite direction.

To prove $\Psi \Rightarrow \widetilde{\Psi}$, it is sufficient to show that this is true for the empty $\mathcal{Q}$ (as equivalence is substitutive). Assume $\mathcal{K}, x \models_\varepsilon \Psi$. Then there exists an environment ε' from $\{\kappa_1, \ldots, \kappa_m\}$ to 2^V such that $\mathcal{K}, x \models_{\varepsilon \circ \varepsilon'} \Phi_0 \wedge \bigwedge_i \mathbf{AG}(\kappa_i \Rightarrow \theta_i)^9$. Now we have:

$$\forall i, \quad \mathcal{K}, x \models_{\varepsilon \circ \varepsilon'} \theta_i \quad \Rightarrow \quad \mathcal{K}, x \models_\varepsilon \varphi_i \quad \text{and} \quad \mathcal{K}, x \models_{\varepsilon \circ \varepsilon'} \Phi_0 \quad \Rightarrow \quad \mathcal{K}, x \models_{\varepsilon \circ \varepsilon'} \widetilde{\Phi_0}$$

Indeed, assume that it is not true and $\mathcal{K}, x \models_{\varepsilon \circ \varepsilon'} \theta_i$ and $\mathcal{K}, x \not\models_\varepsilon \varphi_i$. Consider such a formula φ_i with the smallest temporal height. The only atomic propositions κ_j occurring in θ_i are then associated with some θ_j and φ_j which verify the property and thus any state satisfying such a θ_j, also satisfies φ_j. Therefore any state labelled by such a κ_j is correctly labelled (and satisfies φ_j). And the states that are not labelled by κ_j cannot make θ_i to be wrongly evaluated to true (because κ_j is not in the scope of a negation). Therefore φ_i holds true at x. The same holds for Φ_0 and $\widetilde{\Phi_0}$. As a direct consequence, we have $\Psi \Rightarrow \widetilde{\Psi}$. ◀

C   Experimental results

Detailed results for the three examples are presented in this section. In every case, we distinguished the time required to build the QBF formula (the Z3 specification) and the time required to solve it. Times are given in seconds.

C.1 k-connectivity

n, m	3, 2	4, 3	4, 3	5, 4	35, 4
formula	Ψ_2	Ψ_4	Φ_4	Ψ_5	Ψ_4
# states	18	32	32	50	2450
res	sat	unsat	unsat	unsat	sat
Reduction UU					
Time to build z3 form.	0.01	38.45	70.15	–	–
Size of z3 form.	7682	7922175	29983034	–	–
Time to solve z3 form.	0.04	2.38	8.61	–	–
Reduction FP					
Time to build z3 form.	0	0.03	0.07	0.09	388
Size of z3 form.	1313	9097	11234	27416	38675099
Time to solve z3 form.	0.03	0.06	–	0.16	132.67
Reduction FFP					
Time to build z3 form.	0.03	0.1	0.25	0.1	–
Size of z3 form.	3666	20944	56196	61519	–
Time to solve z3 form.	0.12	0.28	–	57.96	–
Reduction FBV					
Time to build z3 form.	0.01	1	0.23	1.3	–
Size of z3 form.	6363	31243	107120	87681	–
Time to solve z3 form.	40.16	–	43.09	–	–

---

[9] We define the graph $G_{\varepsilon \circ \varepsilon'}$ of $\varepsilon \circ \varepsilon'$ to be $G_\varepsilon \uplus G_{\varepsilon'}$

n, m	$3, 1$	$3, 3$	$9, 2$	$27, 2$
formula	Ψ_2^g	Ψ_3^g	Ψ_3^g	Ψ_3^g
# states	18	18	162	1458
res	unsat	sat	unsat	unsat
Reduction UU				
Time to build z3 form.	$1, 08$	$9, 3$	$-$	$-$
Size of z3 form.	1195684	8632341	$-$	$-$
Time to solve z3 form.	$0, 43$	$8, 96$	$-$	$-$
Reduction FP				
Time to build z3 form.	0.02	$0, 04$	$1, 94$	$190, 94$
Size of z3 form.	13268	18337	1698025	139992889
Time to solve z3 form.	$0, 1$	$0, 23$	$45, 51$	$-$
Reduction FFP				
Time to build z3 form.	$0, 02$	0.04	2.07	192
Size of z3 form.	5180	6877	551447	44643959
Time to solve z3 form.	$0, 10$	$35, 38$	180	$-$
Reduction FBV				
Time to build z3 form.	0.1	0.05	0.95	195
Size of z3 form.	6285	8331	726681	59447253
Time to solve z3 form.	$0, 35$	$-$	$-$	$-$

C.2 Nim game

	$[2, 2]$	$[3, 2]$	$[4, 5, 2]$	$[3, 4, 5]$	$[2, 3, 4, 4]$	$[5, 4, 3, 6]$	$[2, 4, 8, 14]$
# states	16	31	280	328	398	1595	13556
res	unsat	sat	sat	sat	sat	sat	unsat
Reduction UU							
Time to build z3 form.	$0, 00$	$0, 00$	$0, 09$	$0, 31$	$8, 99$	$-$	$-$
Size of z3 form.	32	105	90095	285505	1011324	$-$	$-$
Time to solve z3 form.	$0, 02$	$0, 03$	$0, 09$	$0, 10$	$0, 39$	$-$	$-$
Reduction FP							
Time to build z3 form.	$0, 00$	$0, 00$	$0, 00$	$0, 00$	$0, 00$	$0, 19$	$0, 21$
Size of z3 form.	104	216	2283	2698	3225	13865	125486
Time to solve z3 form.	$0, 01$	$0, 02$	$0, 04$	$0, 4$	$0, 06$	$0, 15$	$62, 2$
Reduction FFP							
Time to build z3 form.	$0, 00$	$0, 00$	$0, 03$	$0, 05$	$0, 07$	$0, 23$	$6, 95$
Size of z3 form.	161	326	3323	3926	4703	19928	178158
Time to solve z3 form.	$0, 02$	$0, 03$	$0, 12$	$0, 10$	$0, 18$	$0, 18$	$26, 66$
Reduction FBV							
Time to build z3 form.	$0, 00$	$0, 02$	$0, 79$	$1, 05$	$1, 88$	$24, 18$	$-$
Size of z3 form.	2468	9371	844412	1171832	1696982	29013387	$-$
Time to solve z3 form.	$0, 05$	$0, 12$	$207, 6$	$341, 6$	$879, 8$	$-$	
Reduction FBV with bounded Until							
Bound for Until:	10	20	20	20	21	28	43
Time to build z3 form.	0	$0, 01$	$0, 08$	$0, 1$	$0, 1$	$0, 48$	$6, 43$
Size of z3 form.	1682	6291	65712	77816	97371	540997	7311052
Time to solve z3 form.	$0, 02$	$0, 1$	$2, 37$	$3, 17$	$4, 20$	$184, 94$	$-$

C.3   Resources distribution

$n \times m$	10×5	10×5	10×5	10×7	10×10	10×10
k, d	$2, 8$	$4, 4$	$4, 6$	$4, 5$	$2, 3$	$4, 7$
# states	50	50	50	70	100	100
res	sat	unsat	sat	unsat	unsat	?
Reduction $\mathsf{UU} - \mathsf{FP}$						
Time to build z3 form.	$0, 03$	$0, 08$	$0, 07$	$0, 10$	$0, 13$	$0, 22$
Size of z3 form.	112155	32357	70757	74417	45905	283907
Time to solve z3 form.	$0, 08$	$274, 00$	$0, 06$	$-$	$0, 09$	$-$
Reduction FFP						
Time to build z3 form.	$0, 05$	$0, 08$	$0, 10$	$0, 10$	$0, 17$	$0, 19$
Size of z3 form.	14515	23213	24715	44744	43510	90916
Time to solve z3 form.	$0, 06$	$308, 48$	$0, 09$	$-$	$0, 10$	$-$
Reduction FBV						
Time to build z3 form.	$0, 08$	$0, 02$	$0, 03$	$0, 03$	$0, 02$	$0, 09$
Size of z3 form.	9521	7633	11133	13123	7521	25733
Time to solve z3 form.	$0, 05$	$27, 16$	$0, 02$	155	$0, 09$	$-$