

Responsibility and verification: Importance value in temporal logics

Corto Mascle*, Christel Baier†, Florian Funke†, Simon Jantsch†, Stefan Kiefer‡

*ENS Paris-Saclay, France

†Technische Universität Dresden, Germany

‡University of Oxford, UK

Abstract—We aim at measuring the influence of the nondeterministic choices of a part of a system on its ability to satisfy a specification. For this purpose, we apply the concept of Shapley values to verification as a means to evaluate how important a part of a system is. The importance of a component is measured by giving its control to an adversary, alone or along with other components, and testing whether the system can still fulfill the specification. We study this idea in the framework of model-checking with various classical types of linear-time specification, and propose several ways to transpose it to branching ones. We also provide tight complexity bounds in almost every case.

I. INTRODUCTION

Classical model-checking algorithms try to detect undesired behaviors in a formal system with reference to a given specification, and the system is deemed correct if they cannot find one. However, simply knowing that the system satisfies the specification is in practice often unsatisfactory: we also want to know *why* it does, or does not. Especially in the case that the specification is violated, knowing *where* in the system to look for a potential model repair can significantly reduce troubleshooting times for both engineers and users.

To this end, Chockler, Halpern and Kupferman defined a notion of *causality* aimed at explaining which parts of a system are relevant for the satisfaction of a specification φ [1]. More specifically, a state s is considered a cause for φ with respect to an atomic proposition p if the value of p can be swapped in a subset of the states T such that further swapping the value of p in s turns φ from being satisfied to being violated (we say that (s, T) is *critical*). Counterfactual reasoning in this spirit (i.e., had the cause not occurred, then the event would not have happened) has a rich history in the philosophy and moral responsibility literature, and had been formalized in the framework of *structural equation models* [2], [3], on which the work [1] is based. Causes are further assigned a *degree of responsibility* by taking the inverse of the size of the smallest set $T \cup \{s\}$ such that (s, T) is critical. This numerical value, adapted from [4], is designed to measure the impact of the state on the specification: Causes with high degree of responsibility

point to small changes of the system that have the power to crucially alter its behavior.

In this paper we define a novel measure for the influence of a state on a specification, called the *importance*. While it is related to the degree of responsibility of [1], a significant difference appears in how the counterfactuality principle is invoked. The degree of responsibility relies on hypothetical modifications of the structure and answers the question “Is the system still working if the truth value of this atomic proposition in that subset of states is switched?” In contrast, we never modify the system, but look at how its nondeterministic choices are resolved, thus tackling the question “Does the system yield a satisfying run if the subset of states is under control (i.e., behaving in a manner conducive to the functioning) while the others are not (i.e., behaving antagonistically)?” Hence, our definition of importance relies on a new viewpoint of what constitutes a critical pair, based on capturing the specific nondeterministic choices available in the states.

The approach above determines the impact of a subset of states on the satisfaction of a specification. In order to turn this information into the *individual* importance of a state (or a component) we employ a solution concept from cooperative game theory, called the *Shapley value* [5]. In a context of collaborative multi-agent interaction, Shapley values aim at measuring how beneficial the participation of a specific agent is in reaching some objective. Translated to Kripke structures, the idea is to compute the probability that taking control over a particular state makes the system work as intended, where the control over states is taken in a (uniformly) random order. The importance distills those parts of the system whose choices are *crucial* for its functioning.

As an example, consider a system testing a server S by sending regular requests. If the server does not respond correctly, the system retries to send a request; if it does respond correctly, then the system may wait before testing again. We represent this system by a Kripke structure, displayed on the left in Figure 1. Consider the specification stating that the system should make infinitely many tests and receive only finitely many incorrect answers (modeled by the LTL formula $\varphi = GF \text{check} \wedge FG \neg \text{fail}$). The system fails this condition if S malfunctions and fails infinitely often or if ok waits indefinitely from some point on without rechecking the server. As the other states cannot enforce breaching φ without S and

This work was funded by DFG grant 389792660 as part of TRR 248 (see <https://perspicuous-computing.science>), the Cluster of Excellence EXC 2050/1 (CeTI, project ID 390696704, as part of Germany’s Excellence Strategy), DFG-projects BA-1679/11-1 and BA-1679/12-1, the Research Training Group QuantLA (GRK 1763). Kiefer is supported by a Royal Society University Research Fellowship.

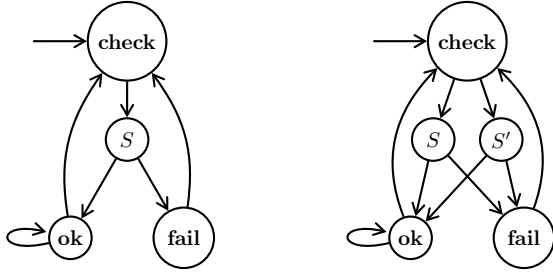


Fig. 1. Two simple systems, as used in the introductory example.

ok, the importance of these two states is $1/2$ and that of the other states is 0.

Let us now add a backup server S' with the same role as S (as displayed on the right in Figure 1). Then the system succeeds if it loops infinitely often between **ok**, **check** and the set $\{S, S'\}$, which is only possible if **ok** and at least two of **check**, S and S' behave well. In this case we get an importance of $1/2$ for **ok** and $1/6$ for **check**, S and S' . This is a numerical interpretation of the fact that control over the behavior of **ok** is more critical to the functioning of the system: unfortunate choices made in **ok** (i.e., avoiding further tests forever) instantly make the system fail. The equal importance of **check**, S , and S' reflects the fact that – although their actual roles in the system differ – they play interchangeable parts when only the functioning of the system is concerned: any two of them are needed to make the system work.

It is noteworthy that a variant of the degree of responsibility based on our notion of critical pair (and applied in reverse fashion, i.e., from violation of φ to satisfaction of φ) would not be able to distinguish **check**, **ok**, S , and S' as it evaluates to $1/3$ for each of these states. Roughly speaking, the degree of responsibility only takes a *minimal* critical pair into account, whereas the importance computes a weighted average over the size of *all* critical pairs that a state belongs to. The rationale for this is that belonging to many critical pairs makes the state less dependent on behavior outside of its control, and hence more powerful.

The construction of the importance value as outlined above gives rise to the following three complexity problems that we study in this paper for a wide range of specifications. The *value problem* consists in determining if a subset of states of a given Kripke structure can guarantee that the specification is respected when the other states act in an adversarial way. The *importance problem* is the problem of computing the actual importance value. Finally, the *usefulness problem* asks whether a state of a system has positive importance, i.e., whether its behavior has any influence at all on the satisfaction of the specification. In fact we define the importance value in the presence of a prescribed partition of the state space, and study the complexity problems in this generalized setting. This allows us to capture more realistic scenarios such as the importance of a *system component* in a composite architecture.

Table I summarizes the complexity results obtained through-

out the paper. We write $\in \mathcal{C}$ when the problem is in class \mathcal{C} and we do not have a matching lower bound, and just \mathcal{C} when the problem is \mathcal{C} -complete. Since our examinations spread over a wide range of specifications, our results crucially rely on a diverse game-theoretic toolkit.

The paper is split into three parts: In the first part we define the notions in the general setup of turn-based two-player games on finite graphs. Then we apply these notions in order to define the importance on Kripke structures with respect to LTL specifications, and finally we look at the case of CTL specifications on modal transition systems. The proofs missing in the main document due to space constraints can be found in the appendix.

A. Related work

The complexity of computing the aforementioned degree of responsibility was examined for the general class of structural equation models in [4] and for Boolean circuits in [1]. They are closely related to the complexity results about deciding causality [12], [13].

Our work ties into a ubiquitous quest for powerful *explanations* of model-checking results. If a system satisfies a specification, then *coverage estimation* has been used to analyze which parts of the system are essential for the successful verification result [14]–[17]. As in the definition of the degree of responsibility, the idea is to apply small changes to the system (*mutants*) and check the resulting effect on the specification. *Vacuity detection*, on the other hand, applies the principle of small changes to the specification [18]–[20]. This strand of research aims at checking whether the specification is satisfied in an undesired, trivial fashion (typically due to insufficient modeling of the system). Coverage and vacuity have been shown to exhibit a formal duality [21], and recent work on the subject is dedicated to analyzing network formation games [22].

In the case of an unsuccessful verification process, one of the powerful features of many model checking approaches is the ability to generate a counterexample [23]. In order to extract further diagnostic information, there has been extensive work on *localizing* errors in faulty traces [24]–[29]. Typically, one compares an erroneous trace with a successful one that lies nearby with respect to a suitable metric. Early detection of error traces has been investigated in [30], where a game-like description close to ours between a system module and its environment has been used. Work on explaining counterexamples using the notion of causality from [1] has been presented in [31].

Although the Shapley value is a classical solution concept in economics, it has recently received considerable attention in the computer science literature. Shapley-like values have been used as explanations for machine learning models, where they estimate the impact of the input parameters on the outcome [32]–[34]. They have also been employed as a means by which centrality in networks can be measured [35] or responsibilities can be assigned in game-like structures [36]. Computational approaches for the Shapley value are given in [37]–[40]. For

TABLE I
A SUMMARY OF THE RESULTS ON THE COMPLEXITY OF THE VALUE, USEFULNESS
AND IMPORTANCE PROBLEMS FOR VARIOUS TYPES OF SPECIFICATIONS.

	Büchi	Rabin	Streett	Parity	Explicit Muller
Value	P [6]	NP [7]	coNP [7]	$\in \text{NP} \cap \text{coNP}$ [8]	P [9] [6]
Usefulness	NP (Prop. IV.5)	Σ_2^P (Prop. IV.8)	Σ_2^P (Cor. IV.10)	NP (Prop. IV.5)	NP (Prop. IV.5)
Importance	#P (Thm. IV.6)	#P ^{NP} (Thm. IV.9)	#P ^{NP} (Cor. IV.10)	#P (Thm. IV.6)	#P (Thm. IV.6)
	Emerson-Lei	LTL	2-turn CTL	Concurrent CTL	
Value	PSPACE [10]	2EXPTIME [11]	Σ_2^P (Prop. V.2)	$\in \text{EXPTIME}$ (Rmk. 8)	
Usefulness	PSPACE (Thm. IV.7)	2EXPTIME (Thm. IV.3)	Σ_3^P (Prop. V.3)	$\in \text{EXPTIME}$ (Rmk. 8)	
Importance	PSPACE (Thm. IV.7)	2EXPTIME (Thm. IV.3)	#P ^{Σ_2^P} (Thm. V.4)	$\in \text{EXPTIME}$ (Rmk. 8)	

a variety of recent results and applications of Shapley values we refer to [41].

II. PRELIMINARIES

A. Words and structures

a) Words and trees: Let A be an alphabet. We denote by A^* (resp. A^ω) the set of finite (resp. infinite) words over A . Given a word w , we write $|w|$ for its length and, for all $i < |w|$, we write w_i for the i th letter of w .

An infinite tree t over A is a prefix-closed subset of A^* such that for all $p \in t$, there exists $a \in A$ such that $pa \in t$. The set of sons of a node p of the tree t is denoted by $\text{Sons}_t(p) = pA \cap t$.

b) Kripke structures: A Kripke structure \mathcal{K} is a 5-tuple $(S, \text{AP}, \Delta, \text{init}, \lambda)$ where S is a finite set of states, AP is a finite set of atomic propositions, $\Delta \subseteq S \times S$ is a set of transitions, init is an initial state and $\lambda : S \rightarrow 2^{\text{AP}}$ is a labeling function. For every $s \in S$, we define its image by Δ as $\Delta(s) = \{t \in S \mid (s, t) \in \Delta\}$, and we always assume $\Delta(s)$ to be nonempty for all s . A run of a Kripke structure $\mathcal{K} = (S, \text{AP}, \Delta, \text{init}, \lambda)$ is an infinite sequence $r \in S^\omega$ such that for all $i \in \mathbb{N}$, we have $(r_i, r_{i+1}) \in \Delta$. To every run r we can associate a trace, which is the sequence of labelings $\lambda(r_0)\lambda(r_1)\dots$. The set of runs of \mathcal{K} is denoted by $\mathcal{R}(\mathcal{K})$ while the set of traces it generates is called $\mathcal{L}(\mathcal{K})$.

c) Modal transition systems: A modal transition system (MTS) [42] \mathcal{M} is a 6-tuple $(S, \text{AP}, \Delta_{\text{may}}, \Delta_{\text{must}}, \text{init}, \lambda)$ where S is a finite set of states, AP is a set of atomic propositions, $\Delta_{\text{must}}, \Delta_{\text{may}} \subseteq S \times S$ are sets of transitions such that $\Delta_{\text{must}} \subseteq \Delta_{\text{may}}$, $\text{init} \in S$ is an initial state and $\lambda : S \rightarrow 2^{\text{AP}}$ is a labeling function. We assume $\Delta_{\text{must}}(s)$ to be nonempty for every state s . We call a Kripke structure $\mathcal{K} = (S, \text{AP}, \Delta, \text{init}, \lambda)$ an *implementation* of \mathcal{M} if $\Delta_{\text{must}} \subseteq \Delta \subseteq \Delta_{\text{may}}$. This is in contrast to other works in the modal transition system literature which usually consider a more general notions of implementation based on refinement relations (see [43] for a recent overview).

B. Temporal logics

We now define the syntax of the two logics we will consider in this paper, LTL and CTL. For the semantics and basic properties of these logics we refer the reader to [44] or [45].

a) Linear temporal logic: The formulas of LTL are given by the grammar

$$\varphi ::= a \mid \varphi \vee \varphi \mid \neg \varphi \mid X\varphi \mid \varphi U \varphi$$

with a ranging over a finite set of atomic propositions AP .

LTL formulas are evaluated on infinite words over 2^{AP} . We extend the set of operators with $\top, \perp, \wedge, F, G$ and R in the usual way.

b) Computation tree logic: The syntax of CTL is defined by the grammar

$$\varphi ::= a \mid \varphi \vee \varphi \mid \neg \varphi \mid EX\varphi \mid E\varphi U \varphi \mid A\varphi U \varphi$$

with a ranging over a finite set of atomic propositions AP .

CTL formulas are evaluated on infinite trees over 2^{AP} . We extend the set of operators with $\top, \perp, \wedge, EF, EG, ER, AX, AR, AF$ and AG in the usual way.

C. Games on graphs

A directed graph G is a pair (V, E) with V a set of vertices and $E \subseteq V^2$ a set of edges. An arena is a tuple $(G, V_{\text{Sat}}, V_{\text{Unsat}})$ with G a graph and $V_{\text{Sat}}, V_{\text{Unsat}}$ a partition of its vertices. We say that the vertices of V_{Sat} belong to player Sat, or are controlled by player Sat (and similarly for Unsat).

A game \mathcal{G} is defined by an arena $((V, E), V_{\text{Sat}}, V_{\text{Unsat}})$, an initial vertex $\text{init} \in V$ and a winning condition (also called objective) $\Omega \subseteq V^\omega$. As we will often use Kripke structures and modal transition systems to construct games, we will often refer to vertices as states and edges as transitions.

For more information on infinite games played on finite graphs we refer to [46]. In particular, we will use several classical winning conditions on such games, whose definitions can be found in [46, Chapter 2].

D. Complexity classes

We consider mostly well-known and classical complexity classes, a description of which can be found, e.g., in [47]. We use logarithmic space reductions for the decision problems and Turing reductions for the counting complexity classes.

III. A GENERAL DEFINITION OF IMPORTANCE IN TWO-PLAYER GAMES

Let \mathcal{G} be a two-player game between Sat and Unsat on an arena $((V, E), V_{\text{Sat}}, V_{\text{Unsat}})$. Let $\Omega \subseteq V^\omega$ be Sat's objective (i.e. the set of plays of \mathcal{G} she wins). In order for the game to be determined, we assume Ω to be a Borel set.

We start by defining a general notion of *importance* of a state (or a set of states, in a given partition), which is a measure of how much a state contributes towards Sat winning the game. In other words, if Sat is restricted to controlling only some of her states (for example, due to resource constraints) she should opt to control the ones with high importance in order to win the game.

Definition III.1. For all sets of states $V'_{\text{Sat}} \subseteq V_{\text{Sat}}$, we define $\mathcal{G}_{V'_{\text{Sat}}}$ as the game between Sat and Unsat played on the arena $(G, V'_{\text{Sat}}, V \setminus V'_{\text{Sat}})$, with G the graph of \mathcal{G} and the same objective Ω for Sat.

Definition III.2 (Value of state subset). For all sets of states $V'_{\text{Sat}} \subseteq V_{\text{Sat}}$, we define the value of the set V'_{Sat} as

$$\text{val}(V'_{\text{Sat}}) = \begin{cases} 1 & \text{if Sat has a winning strategy for } \mathcal{G}_{V'_{\text{Sat}}} \\ 0 & \text{if Unsat has a winning strategy for } \mathcal{G}_{V'_{\text{Sat}}} \end{cases}$$

Note that the value is defined with respect to a game, but that game does not appear in the notation as we will always make it clear from context. The value is well-defined for all V'_{Sat} as we assumed the objective to be a Borel set, thus the game is determined.

With the definition of *value* of a subset of states, we are in the position of defining the importance of a state. This definition corresponds to the classical formula for the Shapley value [5]. In our context it can be explained as follows: for a given state v , it counts the number of orderings of the states in V_{Sat} such that if Sat gives up control of her states one by one in that order, then Sat loses the game for the first time after giving up v . The number obtained is then divided by the total number of such orderings. We can also look at this definition from a probabilistic point of view: The importance of a state v is the probability that, if Sat gives up control of the states sequentially in an order drawn uniformly at random, the first time Sat is no longer able to win the game is when she gives up control of v . This is what we call switching in Definition III.3 below.

Definition III.3 (Importance). The *importance* for Sat of a state $v \in V_{\text{Sat}}$ with respect to a game \mathcal{G} on an arena $((V, E), V_{\text{Sat}}, V_{\text{Unsat}})$ is defined as

$$\mathcal{I}(v) = \frac{1}{n!} \sum_{\pi \in \Pi_{V_{\text{Sat}}}} \text{val}(V_{\geq v}^\pi) - \text{val}(V_{\geq v}^\pi \setminus \{v\})$$

where $n = |V_{\text{Sat}}|$, $\Pi_{V_{\text{Sat}}}$ is the set of bijections from V_{Sat} to $\{1, \dots, n\}$, and $V_{\geq v}^\pi = \{v' \in V_{\text{Sat}} \mid \pi(v') \geq \pi(v)\}$.

Given a bijection $\pi : V_{\text{Sat}} \rightarrow \{1, \dots, n\}$ we say that v is *switching the value* in π if we have $\text{val}(V_{\geq v}^\pi) = 1$ and $\text{val}(V_{\geq v}^\pi \setminus \{v\}) = 0$

An equivalent definition, obtained by deleting the null terms from the sum, is obtained through the notion of *critical pair*. A pair $(v, T) \in V_{\text{Sat}} \times 2^{V_{\text{Sat}}}$ is *critical* if $\text{val}(T \cup \{v\}) = 1$ and $\text{val}(T) = 0$. Then we set

$$\mathcal{I}(v) = \frac{1}{n!} \sum_{(v, T) \text{ critical}} (|T|)!(n - |T| - 1)!$$

as $(|T|)!(n - |T| - 1)!$ is the number of $\pi \in \Pi_{V_{\text{Sat}}}$ such that $V_{\geq v}^\pi \setminus \{v\} = T$.

Remark 1. Let v, v' be two states of \mathcal{G} . If for all $T \subseteq V_{\text{Sat}}$ such that $v \in T$ and $v' \notin T$, we have $\text{val}(T) = 0$ then $\mathcal{I}(v) \leq \mathcal{I}(v')$.

We now assume that we are given a partition V_1, \dots, V_n of V_{Sat} . We generalize the previous definitions in a straightforward manner. In all of our complexity proofs we will show the lower bounds for the previous case (in which states are partitioned in singletons) and the upper bounds for the general case. Thus all complexity results hold for both cases.

Definition III.4 (Importance for partitions). The *importance* for Sat of a set of states V_i with $1 \leq i \leq n$ is defined as

$$\mathcal{I}(V_i) = \frac{1}{n!} \sum_{\pi \in \Pi_n} \text{val}(V_{\geq i}^\pi) - \text{val}(V_{\geq i}^\pi \setminus V_i)$$

where Π_n stands for the set of permutations of $\{1, \dots, n\}$, and

$$V_{\geq i}^\pi = \bigcup_{\substack{1 \leq j \leq n \\ \pi(j) \geq \pi(i)}} V_j$$

We define a pair $(i, J) \in \{1, \dots, n\} \times 2^{\{1, \dots, n\}}$ to be *critical* if $\text{val}(\bigcup_{j \in J \cup \{i\}} V_j) = 1$ and $\text{val}(\bigcup_{j \in J} V_j) = 0$. Then we have:

$$\mathcal{I}(V_i) = \frac{1}{n!} \sum_{(i, J) \text{ critical}} |J|!(n - |J| - 1)!$$

Now let us show some basic results stating that parts with importance 0 can be ignored in the computation of the importance of the other parts.

Remark 2. Let $1 \leq i \leq n$. If $\mathcal{I}(V_i) = 0$ then there is no $J \subseteq \{1, \dots, n\}$ such that (i, J) is critical. As a consequence, for all $J \subseteq \{1, \dots, n\}$, $\text{val}(\bigcup_{j \in J} V_j) = \text{val}(\bigcup_{j \in J \cup \{i\}} V_j)$. This means that if $\mathcal{I}(V_i) = 0$, then Sat can always give up control of states V_i without any effect on whether she wins the game.

Lemma III.5 (Restriction to useful parts). Let $I \subseteq \{1, \dots, n\}$ be such that for all $j \notin I$, $\mathcal{I}(V_j) = 0$. Then we have for all $i \in I$

$$\mathcal{I}(V_i) = \frac{1}{|I|!} \sum_{\pi \in \Pi_I} \text{val}(V_{\geq i}^\pi) - \text{val}(V_{\geq i}^\pi \setminus V_i),$$

with Π_I the set of bijections from I to $\{1, \dots, |I|\}$.

Proof. For all $\pi \in \Pi_n$, let us denote by $\pi|_I : I \rightarrow \{1, \dots, |I|\}$ the bijection such that for all $i, j \in I$, $\pi(i) < \pi(j)$ if and only if $\pi|_I(i) < \pi|_I(j)$.

Note that for all $i \in I$ and $\pi \in \Pi$ we have

$$V_{\geq i}^\pi \setminus V_{\geq i}^{\pi|_I} \subseteq \bigcup_{j \in \{1, \dots, n\} \setminus I} V_j.$$

As a consequence, $\text{val}(V_{\geq i}^\pi) = \text{val}(V_{\geq i}^{\pi|_I})$, using Remark 2. Similarly we get $\text{val}(V_{\geq i}^\pi \setminus V_i) = \text{val}(V_{\geq i}^{\pi|_I} \setminus V_i)$. This allows us to rewrite the importance of V_i as

$$\begin{aligned} \mathcal{I}(V_i) &= \frac{1}{n!} \sum_{\pi \in \Pi_n} \text{val}(V_{\geq i}^\pi) - \text{val}(V_{\geq i}^\pi \setminus V_i) \\ &= \frac{1}{n!} \sum_{\pi \in \Pi_n} \text{val}(V_{\geq i}^{\pi|_I}) - \text{val}(V_{\geq i}^{\pi|_I} \setminus V_i) \\ &= \frac{1}{n!} \sum_{\pi' \in \Pi_I} \frac{n!}{|I|!} \cdot (\text{val}(V_{\geq i}^{\pi'}) - \text{val}(V_{\geq i}^{\pi'} \setminus V_i)) \\ &= \frac{1}{|I|!} \sum_{\pi' \in \Pi_I} \text{val}(V_{\geq i}^{\pi'}) - \text{val}(V_{\geq i}^{\pi'} \setminus V_i) \end{aligned}$$

as for all $\pi' \in \Pi_I$ there are $\frac{n!}{|I|!}$ permutations $\pi \in \Pi_n$ such that $\pi|_I = \pi'$. \square

Corollary III.6. *Just as in Definition III.4, by deleting the null terms from the sum we can rewrite the sum from Lemma III.5. Let $I \subseteq \{1, \dots, n\}$ be such that for all $j \notin I$, $\mathcal{I}(V_j) = 0$. Then we have*

$$\mathcal{I}(V_i) = \frac{1}{|I|!} \sum_{(i, J) \text{ critical}, J \subseteq I} (|J|)!(n - |J| - 1)!$$

for all $i \in I$.

We will also need the following lemma, stating that the importance of a part of a system remains unchanged when the specification is replaced with its complement.

Lemma III.7 (Complement objective). *Let $\overline{\mathcal{G}}$ be the game with the same arena and transitions as \mathcal{G} but the complement objective $\overline{\Omega} = V^\omega \setminus \Omega$. Then for all $1 \leq i \leq n$, the importance of V_i is the same for games \mathcal{G} and $\overline{\mathcal{G}}$.*

Proof. For all $S \subseteq V$ let $\overline{\text{val}}(S)$ be the value of S in $\overline{\mathcal{G}}$ and let $\overline{\mathcal{I}}(V_i)$ be the importance of V_i in $\overline{\mathcal{G}}$. For all permutations $\pi \in \Pi_n$ let $\tilde{\pi}$ be the mirror permutation, such that for all $1 \leq i \leq n$, $\tilde{\pi}(i) = \pi(n + 1 - i)$. As the function associating its mirror to each permutation is a bijection from Π_n to itself, we can rewrite $\mathcal{I}(V_i)$ as

$$\mathcal{I}(V_i) = \frac{1}{n!} \sum_{\pi \in \Pi_n} \text{val}(V_{\geq i}^\pi) - \text{val}(V_{\geq i}^\pi \setminus V_i)$$

As $V_{\geq i}^{\tilde{\pi}} = V \setminus (V_{\geq i}^\pi \setminus \{i\})$, we have

$$\text{val}(V_{\geq i}^{\tilde{\pi}}) = 1 - \overline{\text{val}}(V_{\geq i}^\pi \setminus V_i)$$

and

$$\text{val}(V_{\geq i}^{\tilde{\pi}} \setminus V_i) = 1 - \overline{\text{val}}(V_{\geq i}^\pi)$$

As a result, for all $\pi \in \Pi_n$,

$$\overline{\text{val}}(V_{\geq i}^\pi) - \overline{\text{val}}(V_{\geq i}^\pi \setminus V_i) = \text{val}(V_{\geq i}^\pi) - \text{val}(V_{\geq i}^\pi \setminus V_i)$$

Finally, we obtain

$$\mathcal{I}(V_i) = \frac{1}{n!} \sum_{\pi \in \Pi_n} \overline{\text{val}}(V_{\geq i}^\pi) - \overline{\text{val}}(V_{\geq i}^\pi \setminus V_i) = \overline{\mathcal{I}}(V_i) \quad \square$$

We now define the four computational problems which we will study throughout this paper. The three first are decision problems, the fourth is a counting one:

Value problem

$$\begin{cases} \text{Input:} & \text{A game } \mathcal{G}, \text{ a subset } V'_{\text{Sat}} \subseteq V_{\text{Sat}} \\ \text{Output:} & \text{Do we have } \text{val}(V'_{\text{Sat}}) = 1? \end{cases}$$

Usefulness problem

$$\begin{cases} \text{Input:} & \text{A game } \mathcal{G}, \text{ a partition } V_1, \dots, V_n \\ & \text{of the states, an index } i \\ \text{Output:} & \text{Do we have } \mathcal{I}(V_i) > 0? \end{cases}$$

Importance threshold problem

$$\begin{cases} \text{Input:} & \text{A game } \mathcal{G}, \text{ a partition } V_1, \dots, V_n \\ & \text{of the states, an index } i, \eta \in \mathbb{Q} \\ \text{Output:} & \text{Do we have } \mathcal{I}(V_i) > \eta? \end{cases}$$

Importance computation problem

$$\begin{cases} \text{Input:} & \text{A game } \mathcal{G}, \text{ a partition } V_1, \dots, V_n \\ & \text{of the states, an index } i \\ \text{Output:} & n! \cdot \mathcal{I}(V_i) \end{cases}$$

The way the game is encoded is left open at this point, as it will depend on the specific kind of game in question, especially when it comes to the encoding of the objective.

The two importance problems characterize the complexity of computing the importance of a state in a game. We will generally use the counting problem, except in cases where the complexity class obtained is more natural for the threshold version. For instance, if verifying some condition is already EXPTIME-complete, then we want to say that the problem of computing how many elements of a set of exponential size respect that condition is also EXPTIME-complete. However in order to do that we have to formulate the problem as a decision one. For the importance computation problem, the multiplication by $n!$ ensures that the output is always an integer, which is necessary in order for this to be a counting problem.

The usefulness problem is a restricted version of the importance threshold problem, only focusing on whether some part of the system may become necessary to the satisfaction of the specification when some other parts malfunction. A similar

problem for voting games, called the pivot problem, has been studied in [48].

IV. IMPORTANCE VALUES IN LTL

We now apply the theory developed in the preceding section to linear time specifications in Kripke structures. It turns out that the three decision problems defined above are 2EXPTIME-complete for LTL specifications. As this renders practical applications essentially impossible, we then go on to investigate the problems when specifications are restricted to fragments of LTL, for which we obtain more tractable complexity classes.

A. The full logic

Let $\mathcal{K} = (S, AP, \Delta, init, \lambda)$ be a Kripke structure and φ an LTL formula over AP.

Definition IV.1. Given a subset of states $V_{\text{Sat}} \subseteq S$, let $\mathcal{G}_{V_{\text{Sat}}}$ be the game between players Sat and Unsat over the arena $((S, \Delta), V_{\text{Sat}}, V_{\text{Unsat}})$ with $V_{\text{Unsat}} = S \setminus V_{\text{Sat}}$. The winning condition for player Sat is the set of runs of \mathcal{K} whose labeling satisfies φ , i.e. $\{r \in \mathcal{R}(\mathcal{K}) \mid \lambda(r) \models \varphi\}$. The *value* $val(V_{\text{Sat}})$ of $V_{\text{Sat}} \subseteq S$ is then defined as the value of V_{Sat} in the game $\mathcal{G}_{V_{\text{Sat}}}$ (see Definition III.2).

Note that if one of the players owns all the states, then the game comes down to that player selecting a run in the structure. As a consequence, $val(S) = 1$ if and only if \mathcal{K} has a run satisfying φ , and $val(\emptyset) = 1$ if and only if all runs in \mathcal{K} satisfy φ .

Definition IV.2. Given a partition S_1, \dots, S_n of S , we define the *importance* of a set of states S_i with respect to LTL formula φ as the importance of S_i in game \mathcal{G}_S under the same partition (see Definition III.4).

A straightforward telescope sum argument shows that $\sum_{i=1}^n I(S_i) = val(S) - val(\emptyset)$. Therefore we have $\sum_{i=1}^n I(S_i) = 1$ if and only if there exists a run in \mathcal{K} that satisfies φ , but not all runs satisfy φ . Otherwise the sum is 0.

The intuition behind these definitions is that the value of a subset of states is 1 if its elements can cooperate to guarantee the satisfaction of the specification no matter how the other states behave. The importance of a state is high if it is critical in small teams, or numerous teams. We now illustrate our importance notion with a number of examples.

Example 1. Let us first consider the examples given in the introduction and depicted in Figure 1, with states partitioned into singletons. Again we consider the specification $\varphi = GF \text{check} \wedge FG \neg \text{fail}$, and we begin with the left-hand system involving only a single server S . Then Sat wins the game $\mathcal{G}_{V_{\text{Sat}}}$ if and only if $\{S, \text{ok}\} \subseteq V_{\text{Sat}}$: if Sat is not in control of S , then she can respond **fail** forever, and if Sat is not in control of **ok**, then she can avoid further checks forever. Thus $(S, \{\text{ok}\})$ and $(\text{ok}, \{S\})$ are the only critical pairs, and it is straightforward to compute $\mathcal{I}(\text{ok}) = \mathcal{I}(S) = 1/2$.

Next consider the right-hand example of Figure 1 involving two servers S and S' . In this case Sat wins the game $\mathcal{G}_{V_{\text{Sat}}}$

if and only if $\text{ok} \in V_{\text{Sat}}$ and $|\{S, S', \text{check}\} \cap V_{\text{Sat}}| \geq 2$. Namely, in this case **ok** can initiate infinitely many checks; if both servers can be controlled to respond correctly, then this automatically results in infinitely many successful checks, and if one server and **check** can be controlled, then **check** can choose the functioning server infinitely often. From this it is obvious that $\mathcal{I}(\text{fail}) = 0$, so **fail** can be ignored by Lemma III.5, and hence $n = 4$. Each $v \in \{S, S', \text{check}\}$ belongs to two critical sets (v, T) , where $|T| = 2$, and so $\mathcal{I}(v) = 1/6$. On the other hand, **ok** belongs to three such critical sets and one calculates $\mathcal{I}(\text{ok}) = 1/2$.

Example 2. In the three following examples we consider $\varphi = aUb$, and the states are partitioned into singletons.

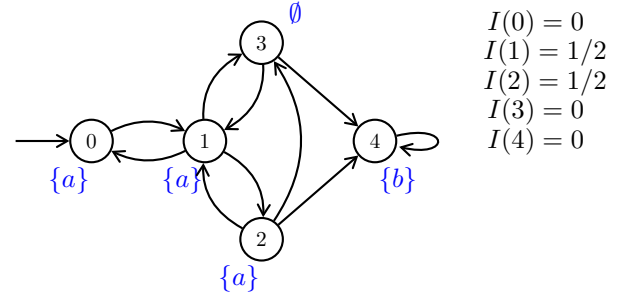


Fig. 2. Kripke structure of Example 2 (1), where atomic propositions are displayed in blue, and importance values for $\varphi = aUb$

(1) In the example of Figure 2 if 1 and 2 belong to Sat then as every game starts with the transition from 0 to 1, she can then go from 1 to 2 and then to 4, satisfying the specification.

However if 1 belongs to Unsat, then Unsat can win by indefinitely going back to 0 from 1. Similarly, if 2 belongs to Unsat, then he can win by going from 2 to 3 if the game reaches 2, leaving no possibility for Sat to satisfy aUb .

As a result, a set of states will allow Sat to win if and only if it contains 1 and 2, thus 1 will be the one switching the value from 1 to 0 whenever it appears before 2 in a permutation. This happens in half of the permutations, thus state 1 has importance 1/2 (see Definition III.3 for what we mean by switching the value). Similarly, 2 also has importance 1/2.

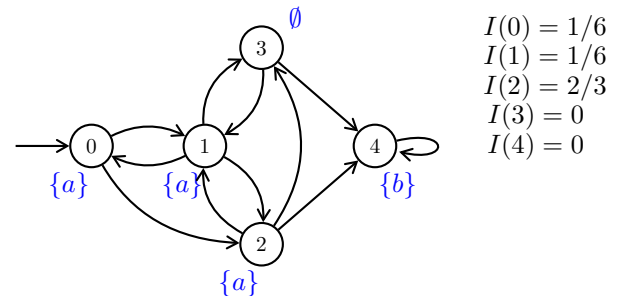


Fig. 3. Kripke structure of Ex. 2 (2) and importance values for $\varphi = aUb$

(2) In the example of Figure 3 one can check that a set of states is allowing Sat to win if and only if it contains 2

and either 0 or 1. Then 2 will be the one switching the value in permutations where it appears before either 1 or 0, i.e. in 2/3 of the permutations. In the other permutations the one switching the value is the second one to appear between 0 and 1.

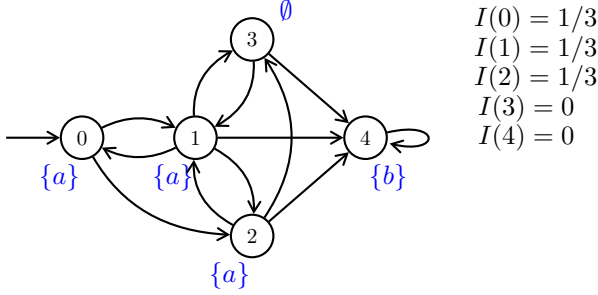


Fig. 4. Kripke structure of Ex. 2 (3) and importance values for $\varphi = aUb$

(3) In the example of Figure 4 one can check that a set of states allows Sat to win if and only if it contains at least two out of the three states 0, 1, 2. Then, the one switching the value will be the second one to appear in a permutation. As a result each one of the three will be the one switching the value in 1/3 of the permutations.

We start our complexity results with the general case of an LTL specification. The complexities of the problems we consider is inferred from the 2EXPTIME-completeness of solving LTL games [11], which is inherited by the value problem.

Theorem IV.3. *The usefulness and importance threshold problems for LTL with respect to Kripke structures are 2EXPTIME-complete. Further, one can compute the importance of a set of states in doubly exponential time.*

Proof sketch. The upper bound comes from the 2EXPTIME upper bound on solving LTL games and the fact that enumerating exponentially many permutations still stays within that class. The idea for the lower bound is to reduce the problem of solving an LTL game to the usefulness problem (with states partitioned into singletons). We consider an LTL game with states split between V_{Sat} and V_{Unsat} . We add states c_s, c_u and t which are visited at the beginning of the game, and we add transitions from c_s to states of V_{Unsat} and c_u to states of V_{Sat} . Finally, we add a sink state and a transition to it from every state. See Figure 5 for an illustration.

Let T be a set of states of the game and assume that one of the states of V_{Sat} is not in T . Then we encode in the specification that Unsat can win by jumping from c_u to that state and then to *sink*, making Sat lose with both T and $T \cup \{t\}$. Similarly we ensure that in order for (t, T) to be critical, T has to be disjoint from V_{Unsat} . The only case in which (t, T) can be critical is then the case where states are correctly distributed between the players, and the usefulness of t is then equivalent to Sat winning the original game. \square

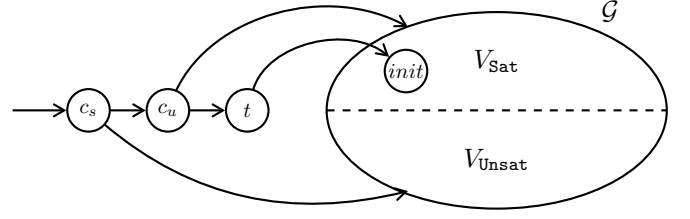


Fig. 5. Illustration for the proof of Theorem IV.3. Every state has a transition to a sink state which is not shown here.

B. Fragments of LTL

Considering the high complexity of the computation of the importance in the case of LTL, we now look at fragments of the logic in order to get more tractable problems. We therefore explore several classical winning conditions which can be expressed as LTL formulas. The value problem over Kripke structures with respect to some kind of specification is precisely the problem of deciding the winner of a game on a finite graph with such a specification as winning condition.

For the usefulness and importance problems, if the value problem has a complexity at least PSPACE, we can enumerate permutations of states while keeping the same complexity. However, if the value problem is for instance in P or NP, then the complexity of the usefulness and importance problems is more involved.

Below, we study various types of winning conditions. We start with the basic case of reachability conditions, which allows us to also prove tight complexity bounds for Büchi, Muller and parity conditions. We consider here explicit Muller conditions, i.e., the condition is encoded as a list of sets of states. Muller conditions are sometimes encoded in more concise forms, such as a coloring function. We will give the complexity of that version as a consequence of the Emerson-Lei case, studied later in the paper.

Proposition IV.4. *The value problems for reachability, Büchi and explicit Muller conditions are P-complete.*

Proof. Reachability, Büchi and explicit Muller conditions are all known to be in P [9]. Furthermore, solving reachability games is known to be P-hard [6].

As we can encode the reachability condition reaching f in all three winning conditions we consider here, we obtain P-hardness for those conditions. \square

Remark 3. Solving games with parity conditions is in $\text{NP} \cap \text{CONP}$ [8], but tight complexity bounds are not known, thus the same can be said about the value problem for parity conditions.

Proposition IV.5. *The usefulness problems for reachability, Büchi, parity and explicit Muller conditions with respect to Kripke structures are NP-complete.*

Proof. The problem is clearly in NP in the case of reachability, Büchi or Muller conditions as one can nondeterministically

guess $J \subseteq \{1, \dots, n\}$ and check in polynomial time whether $\text{val}(\bigcup_{j \in J} V_j) = 0$ and $\text{val}(\bigcup_{j \in J \cup \{i\}} V_j) = 1$ hold.

For parity conditions we also have to guess positional strategies for Sat and Unsat along with J and check in polynomial time that those strategies allow Sat to win when she owns $\bigcup_{j \in J \cup \{i\}} V_j$ and Unsat to win when Sat owns $\bigcup_{j \in J} V_j$.

We obtain NP-hardness through a reduction from 3SAT. Let $\psi = C_1 \wedge C_2 \wedge \dots \wedge C_k$ be a 3SAT instance, with $C_j = (\ell_j^1 \vee \ell_j^2 \vee \ell_j^3)$ for all j , and let $\{x_1, \dots, x_n\}$ be the set of variables appearing in ψ .

We consider the Kripke structure $\mathcal{K} = (S, \text{AP}, \Delta, c_1, \lambda)$ with states partitioned into singletons, and

- $S = \{f, s, \text{sink}\} \cup \{c_i \mid 1 \leq i \leq k\} \cup \{\ell_i^p \mid 1 \leq i \leq k, 1 \leq p \leq 3\} \cup \{x'_j, \neg x'_j \mid 1 \leq j \leq n\}$
- $\text{AP} = \{f\}$
- $\lambda(f) = \{f\}$ and $\lambda(q) = \emptyset$ for all $q \neq f$

$$\begin{aligned} \Delta = & \{(c_i, \ell_i^p) \mid 1 \leq p \leq 3, 1 \leq i \leq k\} \\ & \cup \{(\ell_i^p, c_{i+1}) \mid 1 \leq p \leq 3, 1 \leq i \leq k-1\} \\ & \cup \{(\ell_k^p, s) \mid 1 \leq p \leq 3\} \cup \{(s, x'_1), (s, \neg x'_1)\} \\ & \cup \{(x'_j, x'_{j+1}), (\neg x'_j, \neg x'_{j+1}) \mid 1 \leq j \leq n-1\} \\ & \cup \{(x'_j, \neg x'_{j+1}), (\neg x'_j, x'_{j+1}) \mid 1 \leq j \leq n-1\} \\ & \cup \{(x'_n, f), (\neg x'_n, f)\} \cup \{(q, \text{sink}) \mid q \in S\} \\ & \cup \{(\ell_j^p, x'_m) \mid \ell_j^p \equiv \neg x_m\} \end{aligned}$$

Note that every literal in the clauses has a transition towards **its negation** in the variables. Player Sat wins if and only if f is reached, which can be expressed as a reachability, Büchi or Muller condition. The construction can be done in logarithmic space. See Figure 6 for an illustration of the construction.

We are now going to show that state s is useful if and only if the 3SAT formula is satisfiable, thus proving NP-hardness of the usefulness problem for all four types of winning conditions.

As every state has a transition to a sink state, if at some point a state belonging to Unsat is reached before reaching f , then Sat loses. As a consequence, Sat wins with a set of states if and only if there is a path in this set of states from c_1 to f (possibly not including f).

Suppose there exists a valuation ν satisfying ψ . We extend ν to literals in the natural way, i.e. $\nu(\neg x_i) = \perp$ if $\nu(x_i) = \top$ and $\nu(\neg x_i) = \top$ otherwise. Then we set

$$\begin{aligned} T = & \{x'_m \mid \nu(x_m) = \top\} \cup \{\neg x'_m \mid \nu(x_m) = \perp\} \\ & \cup \{\ell_j^p \mid \nu(\ell_j^p) = \top\} \cup \{c_i \mid 1 \leq i \leq k\} \end{aligned}$$

Clearly there is a path from c_1 to f in $T \cup \{s\}$, as for all $1 \leq i \leq k$ there is at least one ℓ_i^p satisfied by ν (and thus in T), and for all $1 \leq i \leq n$ one of $x'_i, \neg x'_i$ is in T . However, for all $(\ell_i^p, x'_j) \in \Delta$ (resp. $(\ell_i^p, \neg x'_j)$), if $\ell_i^p \in T$ then $\nu(\ell_i^p) = \top$ thus, as $\ell_i^p \equiv \neg x_j$ (resp. x_j), $\nu(x_j) = \perp$ (resp. \top) and $x'_j \notin T$ (resp. $\neg x'_j$). Therefore there is no path in T from c_1 to f .

Now suppose there exists T such that there is a path from c_1 to f in $T \cup \{s\}$ but not in T . Then there is a path in $T \cup \{s\}$

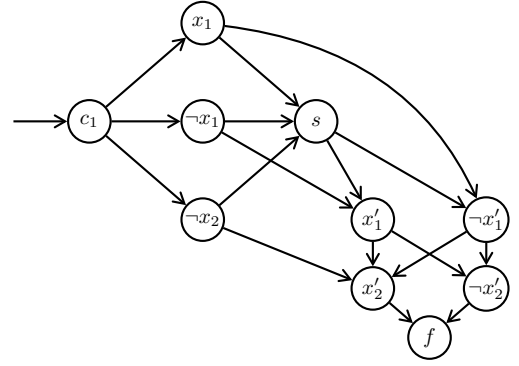


Fig. 6. Construction for $(x_1 \vee \neg x_1 \vee \neg x_2)$. All states have a transition to a sink state, not shown here.

from c_1 to f going through s . In particular for all $1 \leq i \leq n$ at least one of $x'_i, \neg x'_i$ is in T . Let ν be a valuation such that for all i , if $\nu(x_i) = \top$ then $x'_i \in T$ and $\neg x'_i \in T$ otherwise. There is also a path from c_1 to s in T , hence for all i there is a p_i such that $\ell_i^{p_i} \in T$. Then for all $\ell_i^{p_i}$ of the form x_j for some j , the state $\neg x'_j$ cannot be in T as otherwise there would be a path from c_1 to $\ell_i^{p_i}$ then to $\neg x'_j$ and finally to f in T , not going through s . As a result we have $x'_j \in T$ and thus $\nu(\ell_i^{p_i}) = \nu(x_j) = \top$. By a similar argument, if $\ell_i^{p_i} = \neg x_j$ then $\nu(x_j) = \perp$. Hence for every i there is a literal in the i th clause satisfied by ν , thus the 3SAT instance is satisfiable. \square

Theorem IV.6. *The importance computation problems for reachability, Büchi, parity and explicit Muller conditions with respect to Kripke structures are #P-complete.*

Proof sketch. The idea is to reduce the problem of counting the valuations satisfying exactly one literal of every clause of a 3SAT formula φ , known to be #P-complete. First we transform the formula φ into another one ψ that is satisfied by a valuation ν if and only if ν satisfies one literal per clause in φ . We then reuse the construction of the usefulness proof, and notice that the set of teams of states T making (s, T) critical in the structure can be split into parts of (up to some details) equal size, each one matching a valuation satisfying the formula. Further, all such teams are of (again, up to some details) the same size. This allows us to compute the number of valuations satisfying the formula from the importance of s . \square

Remark 4. One can show with nearly identical proofs that those problems keep the same complexity with co-Büchi, safety or co-safety conditions.

Now we consider not only Büchi conditions, but Boolean combinations of them, called Emerson-Lei conditions. As expected, we get an intermediate complexity between those for Büchi and LTL conditions.

Theorem IV.7. *The value, usefulness and importance threshold problems for Emerson-Lei conditions are PSPACE-*

complete. Further, one can compute the importance of a set of states in polynomial space.

Proof sketch. As Emerson-Lei games are known to be in PSPACE, the upper bound follows easily [10]. We prove the lower bound by reduction of QSAT. We construct a structure encoding a sequence of choices of the values of the variables. We ensure that for all T , (s, T) can only be critical if T contains the states choosing the values of the existential variables and not the other ones, by making one of the players win without using s for sets T not satisfying this condition.

We also ensure that Unsatisfiable wins if he owns s , thus s is useful if and only if Satisfiable wins with s . We make players choose valuations of the variables infinitely many times, and we encode in the specification that the player owning the first variable x_i such that x_i and $\neg x_i$ are chosen infinitely often loses. If both players play consistently, the game is decided by the satisfaction of the QSAT formula. \square

Remark 5. It was proven by Hunter and Dawar that Emerson-Lei conditions are more succinct than Muller conditions encoded with a coloring of the states and a list of sets of colors [10]. As a result, the PSPACE lower bounds we obtained for Emerson-Lei transfer to these succinct Muller conditions. Hunter and Dawar also show that solving games with those Muller conditions is PSPACE-complete, from which we can easily infer the PSPACE-completeness of the value, usefulness and importance threshold problem for this type of condition.

We continue our exploration with a more complicated case, the Rabin and Streett conditions. We treat both cases simultaneously as they are symmetric.

Remark 6. As solving Rabin (resp. Streett) games is NP-complete (resp. CONP-complete), so is the value problem for Rabin (resp. Streett) conditions [7].

Proposition IV.8. *The usefulness problem for Rabin conditions is Σ_2^P -complete.*

Proof sketch. The complete proof is in the appendix. We reduce the dual of the $\forall\exists 3SAT$ problem to the usefulness problem in the case when states are partitioned in singletons. The set of states T witnessing the usefulness of the state s will encode the valuation of the first set of variables, with a trick similar to the one used in the proof of Proposition IV.5 to ensure that the encoded valuation is correct.

As Sat plays for a Rabin objective, she has a positional strategy, with which she has to choose for each clause a satisfied literal. We use the Rabin condition to make sure that Sat does not pick a literal and its negation. We also ensure that Sat wins automatically with $T \cup \{s\}$ as soon as T encodes a correct valuation, and then s is useful if and only if there exists a set of states T (i.e. a valuation of the first variables) such that for all positional strategy of Sat over T (i.e. valuation of the second variables), Sat loses the game (i.e. the formula is not satisfied). \square

The theorem below uses the complexity class $\#P^{NP}$, which is the class of counting problems P such that there exists a

nondeterministic polynomial-time Turing machine with an NP oracle such that the answer of P on an input is the number of accepting runs of the machine on that input.

Theorem IV.9. *The importance computation problem for Rabin conditions is $\#P^{NP}$ -complete.*

Proof sketch. The idea is simply to observe that in the construction for Proposition IV.8, the sets of states witnessing the usefulness of s are in bijection with the valuations of the universal variables witnessing the non-validity of the $\forall\exists 3Sat$ formula (up to some technical details). In the appendix we show that counting such valuations is $\#P^{NP}$ -complete, from which one can infer $\#P^{NP}$ -completeness of the importance computation problem. \square

Corollary IV.10. *As Streett conditions are exactly the complements of Rabin ones, by Lemma III.7 and Proposition IV.8, the usefulness problem for Streett conditions is Σ_2^P -complete.*

By the same argument, by Lemma III.7 and Theorem IV.9, the importance computation problem for Streett conditions is $\#P^{NP}$ -complete.

V. IMPORTANCE VALUES IN CTL

We now adapt the definitions to deal with CTL specifications. A notion of degree of responsibility of a state in a Kripke structure for the satisfaction of a CTL formula was already given by Chockler, Halpern, and Kupferman [1]. While in their approach the responsibility of a state was based on the set of atomic propositions it chooses to satisfy, in ours it is based on the set of outgoing transitions it chooses to allow.

In contrast to the previous sections, CTL has the additional challenge that the formulas are evaluated on trees and not on words. The first question that arises is the nature of the nondeterministic choices in this setting. Our definitions rely on the fact that the nondeterminism of a state may be resolved in different ways by the two players. However, due to the branching time nature of CTL, directly applying this methodology does not make sense, as CTL formulas already take the nondeterminism into account. This is why we consider *modal transition systems* (MTS), in which there is another layer of choice: namely determining the subset of *may* transitions that are present in any state. Modal transition systems have been widely studied as a formalism to capture the refinement of processes from abstract specifications to concrete implementations [42], [43]. They have been extended in various ways, and the corresponding synthesis and verification problems have been considered [49]–[51].

The second, and related, issue is that letting the players construct the tree turn-by-turn runs into the problem that the order in which different branches are considered will often make a difference. In Section V-A we explain the difficulties of defining a game which allows both players to construct a tree generated by an MTS in a turn-by-turn way.

In the Section V-B we define a notion of importance for CTL on MTS, which we call *two-turn CTL*, where both players choose once in the beginning which *may*-transitions they

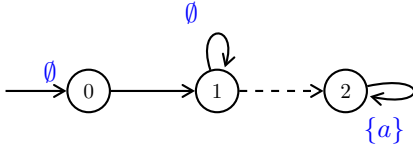


Fig. 7. A modal transition system with *must*-transitions depicted as solid lines, and *may*-transitions depicted as dashed lines. Even this simplistic example illustrates the problem that has to be faced when defining turn-based CTL values (cf. Example 3).

allow in the states under their control. This choice induces a Kripke structure on which the CTL formula can be evaluated.

However the order in which the choices are made affects the importance values. Therefore, in Section V-C we consider also the concurrent setting in which randomized strategies become important, which we call *Concurrent CTL*.

Throughout, let $\mathcal{M} = (S, AP, \Delta_{must}, \Delta_{may}, init, \lambda)$ be a modal transition system and let φ be a CTL formula.

A. Importance in an MTS with respect to a CTL specification

It is appealing to define a notion of importance that relies on the ability of a set of states to *guarantee* the satisfaction of a specification. However, in the case of CTL the fact that formulas are evaluated on trees and not on runs forbids us to make the players construct the run turn by turn, as the winning player would heavily depend on the order in which branches are constructed. We could define a success value where some sets of states are seen as neutral, meaning that this group of states cannot guarantee that the specification is satisfied, nor can its complement guarantee that the specification is unsatisfied, similarly to what was done in [52]. However we wish to define the importance as a numerical value, thus it is more practical that the success value can only be 0 or 1.

One could try to use formulations of CTL model-checking in terms of turn-based games, as described for instance by Lange [53]. However this faces a major issue, illustrated by the following example.

Example 3. Consider the tautological formula $EFa \vee AG \neg a$ and the MTS in Figure 7. Say state 1 belongs to *Unsat*. In a model-checking game, *Sat* would be expected to choose one of the sides of the disjunction at some point in the game and prove it. But if we do not fix the structure and allow *Unsat* to choose transitions after *Sat* has chosen a subformula, then *Unsat* will be able to react to the choice of *Sat* by allowing or not the transition from 1 to 2. As a result, *Unsat* wins the game even though the specification is a tautology.

B. Two-turn CTL importance

The idea of two-turn importance values is that a set of states has value one if it can choose sets of outgoing transitions such that the specification is satisfied no matter which sets of outgoing transitions are chosen by the other states. This definition puts more burden on the satisfier, but matches a vision of the MTS as a way to represent a set of Kripke

structures (possible implementations of a system) rather than a language of trees.

Definition V.1 (Two-turn importance values). Let $\mathcal{M} = (S, AP, \Delta_{must}, \Delta_{may}, init, \lambda)$ be a modal transition system, let $V_{Sat} \subseteq S$ and let φ be a CTL specification. A *pure strategy* for *Sat* is a function $\sigma_{Sat} : V_{Sat} \rightarrow \Delta_{may}$ such that for all $v \in V_{Sat}$, $\Delta_{must}(v) \subseteq \sigma_{Sat}(v) \subseteq \Delta_{may}(v)$. We define pure strategies σ_{Unsat} for *Unsat* symmetrically.

Two pure strategies $\sigma_{Sat}, \sigma_{Unsat}$ yield a Kripke structure, whose states are the ones of \mathcal{M} and transitions from a state are given by the strategy of the player owning that state. We call that Kripke structure $\mathcal{K}(\sigma_{Sat}, \sigma_{Unsat})$.

The value $val_{2turn}(V_{Sat})$ of V_{Sat} is defined as 1 if there exists a pure strategy σ_{Sat} of *Sat* such that for all pure strategies σ_{Unsat} of *Unsat*, $\mathcal{K}(\sigma_{Sat}, \sigma_{Unsat})$ satisfies φ , and 0 otherwise.

The *importance* is defined analogously to Definition III.4: Given a partition S_1, \dots, S_n of S , the importance of S_i is defined as

$$\mathcal{I}_{2turn}(S_i) = \frac{1}{n!} \sum_{\pi \in \Pi_n} val_{2turn}(S_{\geq i}^{\pi}) - val_{2turn}(S_{\geq i}^{\pi} \setminus S_i)$$

Example 4. (1) Consider the formula $\varphi = A(EFa)Ub$ and the modal transition system displayed in Figure 8. Observe that the two ways in which φ may be violated are

- *Unsat* owns 0, 2 and 5 and allows transitions from 0 to 5, 5 to 2 but not 2 to 4, so that there is a path labeled $\{a\}\emptyset\{b\}$ to 2 and then a single possible path looping on 2.
- *Unsat* owns 1 and 2 and chooses transitions so that there is no transition from 1 to 3 or from 2 to 4.

The importance is therefore distributed as follows:

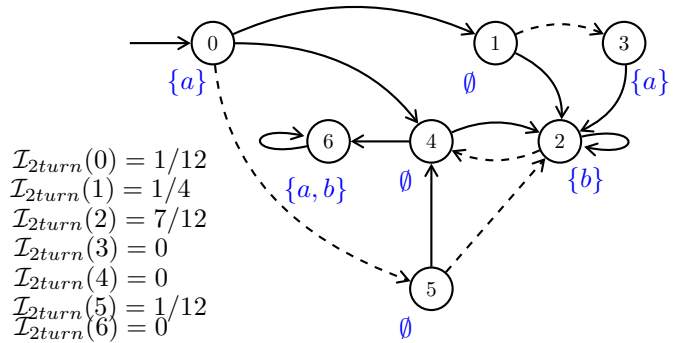


Fig. 8. MTS of Ex. 4 (1) and 2-turn importance values for $\varphi = A(EFa)Ub$

(2) In the example of fig. 9 we want to illustrate a limitation of this notion with respect to what was discussed in Section V-A. Such a mechanism can be illustrated by trying to prove $AG(a \Rightarrow EX(EFb))$ on the following structure:

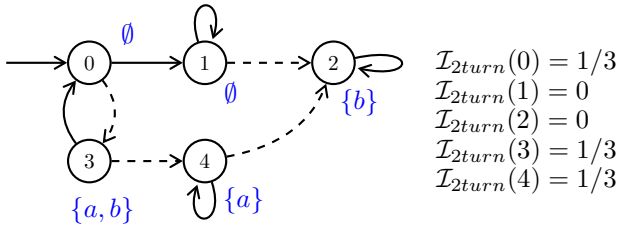


Fig. 9. MTS of Ex. 4 (2) and 2-turn importance values for $\varphi = AG(a \Rightarrow EX(EFb))$

In the 2-turn CTL framework, Unsat wins if and only if he owns 0, 3 and 4, in order to create a path to 4 but no transition from 4 to 2. Thus in any ordering of the states the last one between 0, 3 and 4 will be the one switching the value.

However one might want to design a richer model in which we would also give the victory to Unsat when he owns either 0, 1 and 3 or 0, 1 and 4. The reason is that after allowing the transition from 0 to 3 at the start, we would like to let Unsat delete it. Then Unsat can not allow the transition from 1 to 2, and ensure that there is no path from 3 to 2. Therefore there is no path from a successor of 3 reaching a state labeled b . This version would yield an importance of $1/2$ for 0 and $1/6$ for 1, 3 and 4.

This observation motivates the study of turn-based definitions of CTL importance in MTS for restricted sets of formulas. We will not investigate it further in this paper, focusing instead on the two definitions of CTL importance we propose here.

In the appendix we prove the following results. The hardness proofs consist in encoding choices of valuations of variables in SAT formulas as the players' choices of transitions.

Proposition V.2. *The value problem for two-turn CTL is Σ_2^P -complete.*

Proposition V.3. *The usefulness problem for two-turn CTL is Σ_3^P -complete.*

Theorem V.4. *The importance computation problem for two-turn CTL is $\#P^{\Sigma_2^P}$ -complete.*

Remark 7. We can define a dual game, in which Unsat plays first, and then Sat. While in the former game Sat was at a disadvantage, in this version Unsat is, as he is the one who has to choose his strategy without knowing the adversary's.

Let \mathcal{M} be an MTS with a set of states S , let $V_{\text{Sat}} \subseteq S$ and let φ be a specification, the value of V_{Sat} with respect to φ in the game where Unsat starts is $1 - \text{val}_{2\text{turn}}(S \setminus V_{\text{Sat}})$ with $\text{val}_{2\text{turn}}(V_{\text{Sat}})$ the value of V_{Sat} with respect to $\neg\varphi$ in the game where Sat starts. From this one infers easily that the value problem for the game where Unsat starts is Π_2^P -complete and that, by an argument similar to the proof of III.7, the usefulness problem is Σ_3^P -complete and the importance computation problem $\#P^{\Sigma_2^P}$ -complete.

C. Concurrent CTL importance

The previous version of the game breaks the symmetry between the two players: We have to pick either Sat or Unsat to play first (we chose Sat in the definition above). One may prefer a version of this game in which we do not give any such advantage to a player.

We now introduce a concurrent game, in which both players choose a mixed strategy, in the form of a distribution over all the possible choices of sets of transitions from their respective states. The value of a set of states is the highest probability such a mixed strategy can guarantee for Sat with this set of states. The Nash theorem guarantees the existence of a Nash equilibrium, which means that the highest probability a set of states can achieve for Sat is one minus the highest probability its complement can achieve for Unsat. For an introductory account on non-cooperative concurrent games, we refer to [54].

Definition V.5 (Concurrent game induced by CTL formula). Let $\mathcal{M} = (S, AP, \Delta_{\text{must}}, \Delta_{\text{may}}, \text{init}, \lambda)$ be a modal transition system, let $V_{\text{Sat}} \subseteq S$ and let φ be a CTL specification. Let $C_S(\mathcal{M}, V_{\text{Sat}})$ be the set of pure strategies for Sat. A *mixed strategy* for Sat is a probability distribution $p_S : C_S(\mathcal{M}, V_{\text{Sat}}) \rightarrow [0, 1]$. We define $C_U(\mathcal{M}, V_{\text{Unsat}})$ and p_U in a similar way. Let M_S and M_U denote respectively the set of mixed strategies of Sat and Unsat.

We consider the concurrent game with the payoff functions

$$\rho_{\text{Sat}}(\sigma_{\text{Sat}}, \sigma_{\text{Unsat}}) = \begin{cases} 1 & \text{if } \mathcal{K}(\sigma_{\text{Sat}}, \sigma_{\text{Unsat}}) \text{ satisfies } \varphi \\ 0 & \text{otherwise} \end{cases}$$

and $\rho_{\text{Unsat}}(\sigma_{\text{Sat}}, \sigma_{\text{Unsat}}) = 1 - \rho_{\text{Sat}}(\sigma_{\text{Sat}}, \sigma_{\text{Unsat}})$, for all $\sigma_{\text{Sat}} \in C_S(\mathcal{M}, V_{\text{Sat}})$, $\sigma_{\text{Unsat}} \in C_U(\mathcal{M}, V_{\text{Unsat}})$. Given two mixed strategies p_S, p_U , the expected payoff of Sat is

$$E_{\text{Sat}}(\mathcal{M}, \varphi, p_S, p_U) = \sum_{\substack{\sigma \in C_S(\mathcal{M}, V_{\text{Sat}}) \\ \sigma' \in C_U(\mathcal{M}, V_{\text{Unsat}})}} p_S(\sigma) p_U(\sigma') \rho_{\text{Sat}}(\sigma, \sigma')$$

The expected payoff of Unsat is

$$E_{\text{Unsat}}(\mathcal{M}, \varphi, p_S, p_U) = 1 - E_{\text{Sat}}(\mathcal{M}, \varphi, p_S, p_U)$$

. Finally, we define the value of a set of states V_{Sat} as

$$\text{val}_{\text{concur}}(V_{\text{Sat}}) = \sup_{p_S \in M_S} \inf_{p_U \in M_U} E_{\text{Sat}}(\mathcal{M}, \varphi, p_S, p_U)$$

It is a direct consequence of Nash's theorem [55] that $\text{val}_{\text{concur}}(V_{\text{Sat}})$ is the payoff of Sat obtained in any Nash equilibrium of the concurrent game defined above. In particular we have

$$\text{val}_{\text{concur}}(V_{\text{Sat}}) = \inf_{p_U \in M_U} \sup_{p_S \in M_S} E_{\text{Sat}}(\mathcal{M}, \varphi, p_S, p_U)$$

Definition V.6 (Concurrent importance values). Given a partition of the states S_1, \dots, S_n , we define the importance of a set of states S_i as usual:

$$\mathcal{I}_{\text{concur}}(S_i) = \frac{1}{n!} \sum_{\pi \in \Pi_n} \text{val}_{\text{concur}}(S_{\pi_i}^\pi) - \text{val}_{\text{concur}}(S_{\pi_i}^\pi \setminus S_i)$$

Remark 8. Computing the value of a set of states in this framework amounts to solving a linear optimization problem with exponential input [56]. As the latter problem can be solved in polynomial time, the former is in EXPTIME [57].

Lemma V.7. *For each set of states V_{Sat} , we have $\text{val}_{\text{concur}}(V_{\text{Sat}}) = 1$ if and only if $\text{val}_{2\text{turn}}(V_{\text{Sat}}) = 1$. In particular, as $\text{val}_{2\text{turn}}(V_{\text{Sat}}) \in \{0, 1\}$ for all V_{Sat} , the value $\text{val}_{2\text{turn}}(V_{\text{Sat}})$ is entirely determined by $\text{val}_{\text{concur}}(V_{\text{Sat}})$ (it is its integer part).*

Proof. Suppose $\text{val}_{2\text{turn}}(V_{\text{Sat}}) = 1$, then Sat has a winning pure strategy, thus wins with probability 1 if she applies it in the concurrent game. Hence $\text{val}_{\text{concur}}(V_{\text{Sat}}) = 1$.

Now suppose $\text{val}_{2\text{turn}}(V_{\text{Sat}}) = 0$, then for every pure strategy σ of Sat, Unsats has a winning strategy against σ . As a result, by taking a uniform distribution over its strategies, Unsats can achieve a positive probability to win. As a result, $\text{val}_{\text{concur}}(V_{\text{Sat}}) < 1$. \square

Remark 9. We can make a similar statement about the dual of the two-turn CTL game, described in Remark 7. For all sets of states and specifications the value given by the dual game is 0 if and only if the concurrent value is.

Proposition V.8 (Comparing 2-turn and concurrent importance values). *Let S_1, \dots, S_n be a partition of the states of an MTS. If a set of states S_i is useful with respect to the 2-turn Definition V.1, then it is useful with respect to the concurrent Definition V.6.*

Proof. Suppose $\mathcal{I}_{\text{concur}}(S_i) = 0$. Then for all $J \subseteq \{1, \dots, n\}$ we have $\text{val}_{\text{concur}}(\bigcup_{j \in J} S_j) = \text{val}_{\text{concur}}(\bigcup_{j \in J \cup \{i\}} S_j)$. Then by Lemma V.7, for all J we have

$$\text{val}_{2\text{turn}}(\bigcup_{j \in J} S_j) = \text{val}_{2\text{turn}}(\bigcup_{j \in J \cup \{i\}} S_j)$$

and thus $\mathcal{I}_{2\text{turn}}(S_i) = 0$. \square

The converse of Proposition V.8 does not hold as shown by the following example.

Example 5. We consider the MTS displayed in Figure 10 with states partitioned into singletons, and the formula $\varphi_1 \vee \varphi_2 \vee \varphi_3$ with

$$\begin{aligned} \varphi_1 &= EX(b \wedge EXc) \wedge AX(\neg c \wedge \neg(a \wedge EXc)) \\ \varphi_2 &= AXa \wedge EXEXc \\ \varphi_3 &= EXc \wedge EX(b \wedge EXc) \wedge EX(a \wedge EXc) \end{aligned}$$

In this system, φ_1 expresses that the only path from 0 to 3 is through 2, φ_2 that the only path is through 1 and φ_3 that all three paths exist.

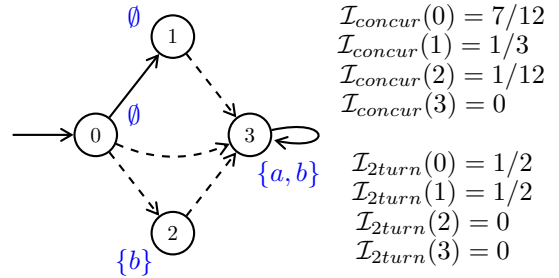


Fig. 10. MTS of Ex. 5

The computation of the concurrent game importance values is lengthy, but straightforward. We observe that Sat has a pure winning strategy whenever she has states 0 and 1, and Unsats has a pure winning strategy whenever he has either 0 or 1 and 2. The remaining case is when Sat has 0 and 2 and Unsats has 1, so Sat can choose to allow or not the paths 0, 3 and 0, 2, 3, and Unsats can choose to allow or not path 0, 1, 3.

Then one can observe that the case where Unsats allows path 0, 1, 3 with probability 1/2 and Sat never allows 0, 3 and allows 0, 2, 3 with probability 1/2 is a Nash equilibrium, thus the set $\{1, 3\}$ has value 1/2. This example shows that in some cases some sets of states may be useless from the 2-turn CTL point of view but not from the Concurrent CTL one.

VI. CONCLUSION

We have introduced a new measure of the influence that a part of a system has on its ability to satisfy a given specification. We have considered two model-checking frameworks, LTL formulas against Kripke structures and CTL formulas against modal transition systems. We have provided tight complexity bounds in most of those cases, except for the complexity of computing the importance in the concurrent case, which we leave open. A general conclusion is that the notion of importance value is natural, but still costly in terms of complexity, especially in the case of LTL. This problem can be mitigated by considering sets of states rather than single states, and formulas from weaker logics.

We expect that the principle of designing a game and computing the importance of a part of the system by shifting its control from one player to the other can be easily adapted to many model-checking problems. We have studied here classical and basic logics, but one could try to find or design logics more well-suited to the computation of the importance, yielding lower complexities.

Another continuation of this work would be a fairer definition of the importance in the case of CTL model-checking. Some subsets of CTL formulas may allow us to design a game in which the players can simultaneously choose transitions on the structure and prove the formula without disadvantaging one of the two. This could be related to the notion of good-for-games automata.

Finally we can extend the definition of value to probabilistic games, by defining the value as the maximal probability of

success that Sat can achieve. This gives us a natural notion of importance in probabilistic games that calls for study.

REFERENCES

- [1] H. Chockler, J. Y. Halpern, and O. Kupferman, “What causes a system to satisfy a specification?” *ACM Transactions on Computational Logic*, vol. 9, no. 3, Jun. 2008. [Online]. Available: <https://doi.org/10.1145/1352582.1352588>
- [2] J. Y. Halpern and J. Pearl, “Causes and Explanations: A Structural-Model Approach. Part I: Causes,” *The British Journal for the Philosophy of Science*, vol. 56, no. 4, pp. 843–887, 2005.
- [3] —, “Causes and Explanations: A Structural-Model Approach. Part II: Explanations,” *The British Journal for the Philosophy of Science*, vol. 56, no. 4, pp. 889–911, 2005.
- [4] H. Chockler and J. Y. Halpern, “Responsibility and blame: A structural-model approach,” in *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, 2003. [Online]. Available: <http://ijcai.org/Proceedings/03/Papers/021.pdf>
- [5] L. S. Shapley, “A value for n -person games,” in *Kuhn, H., Tucker, A. (Eds.), Contributions to the Theory of Games. Vol. II*. Princeton University Press, 1953, pp. 307–317.
- [6] N. Immerman, “Number of quantifiers is better than number of tape cells,” *Journal of Computer and System Sciences*, vol. 22, no. 3, pp. 384–406, 1981.
- [7] E. A. Emerson and C. S. Jutla, “The complexity of tree automata and logics of programs (extended abstract),” in *29th Annual Symposium on Foundations of Computer Science*, 1988, pp. 328–337. [Online]. Available: <https://doi.org/10.1109/SFCS.1988.21949>
- [8] E. A. Emerson, C. S. Jutla, and A. P. Sistla, “On model-checking for fragments of μ -calculus,” in *Computer Aided Verification, 5th International Conference, (CAV)*, 1993, pp. 385–396. [Online]. Available: https://doi.org/10.1007/3-540-56922-7_32
- [9] F. Horn, “Explicit Muller Games are PTIME,” in *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, 2008, pp. 235–243. [Online]. Available: <https://doi.org/10.4230/LIPIcs.FSTTCS.2008.1756>
- [10] P. Hunter and A. Dawar, “Complexity bounds for regular games,” in *Mathematical Foundations of Computer Science, 30th International Symposium (MFCS)*, 2005, pp. 495–506. [Online]. Available: https://doi.org/10.1007/11549345_43
- [11] R. Rosner, “Modular synthesis of reactive systems,” Ph.D. dissertation, PhD thesis, Weizmann Institute of Science, 1992.
- [12] T. Eiter and T. Lukasiewicz, “Causes and explanations in the structural-model approach: Tractable cases,” *Artificial Intelligence*, vol. 170, no. 6, pp. 542 – 580, 2006. [Online]. Available: <https://doi.org/10.1016/j.artint.2005.12.003>
- [13] —, “Complexity results for structure-based causality,” *Artificial Intelligence*, vol. 142, no. 1, pp. 53 – 89, 2002. [Online]. Available: [https://doi.org/10.1016/S0004-3702\(02\)00271-0](https://doi.org/10.1016/S0004-3702(02)00271-0)
- [14] Y. Hoskote, T. Kam, P.-H. Ho, and X. Zhao, “Coverage estimation for symbolic model checking,” in *Proceedings of the 36th Annual ACM/IEEE Design Automation Conference (DAC)*, 1999, p. 300–305. [Online]. Available: <https://doi.org/10.1145/309847.309936>
- [15] H. Chockler, O. Kupferman, R. P. Kurshan, and M. Y. Vardi, “A practical approach to coverage in model checking,” in *Computer Aided Verification*, 2001, pp. 66–78. [Online]. Available: https://doi.org/10.1007/3-540-44585-4_7
- [16] H. Chockler, O. Kupferman, and M. Y. Vardi, “Coverage metrics for temporal logic model checking,” in *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, 2001, pp. 528–542. [Online]. Available: https://doi.org/10.1007/3-540-45319-9_36
- [17] H. Chockler, O. Kupferman, and M. Vardi, “Coverage metrics for formal verification,” *International Journal on Software Tools for Technology Transfer*, vol. 8, no. 4–5, p. 373–386, Aug. 2006.
- [18] I. Beer, S. Ben-David, C. Eisner, and Y. Rodeh, “Efficient detection of vacuity in actl formulas,” in *Proceedings of the 9th International Conference on Computer Aided Verification (CAV)*, 1997, p. 279–290. [Online]. Available: https://doi.org/10.1007/3-540-63166-6_28
- [19] O. Kupferman and M. Y. Vardi, “Vacuity detection in temporal model checking,” in *Proceedings of the 10th IFIP WG 10.5 Advanced Research Working Conference on Correct Hardware Design and Verification Methods (CHARME)*, 1999, p. 82–96.
- [20] M. Purandare and F. Somenzi, “Vacuum cleaning CTL formulae,” in *Computer Aided Verification*, 2002, pp. 485–499. [Online]. Available: https://doi.org/10.1007/3-540-45657-0_39
- [21] O. Kupferman, W. Li, and S. A. Seshia, “A theory of mutations with applications to vacuity, coverage, and fault tolerance,” in *Formal Methods in Computer-Aided Design (FMCAD)*, 2008, pp. 1–9. [Online]. Available: <https://doi.org/10.1109/FMCAD.2008.ECP.29>
- [22] G. Bielous and O. Kupferman, “Coverage and Vacuity in Network Formation Games,” in *28th EACSL Annual Conference on Computer Science Logic (CSL)*, 2020. doi: 10.4230/LIPIcs.CSL.2020.10 pp. 10:1–10:18. [Online]. Available: <https://doi.org/10.4230/LIPIcs.CSL.2020.10>
- [23] E. M. Clarke, O. Grumberg, K. L. McMillan, and X. Zhao, “Efficient generation of counterexamples and witnesses in symbolic model checking,” in *Proceedings of the 32nd Annual ACM/IEEE Design Automation Conference*, ser. DAC ’95. New York, NY, USA: Association for Computing Machinery, 1995. doi: 10.1145/217474.217565. ISBN 0897917251 p. 427–432.
- [24] T. Ball, M. Naik, and S. K. Rajamani, “From symptom to cause: Localizing errors in counterexample traces,” *SIGPLAN Not.*, vol. 38, no. 1, p. 97–105, 2003. doi: 10.1145/640128.604140
- [25] A. Zeller, “Isolating cause-effect chains from computer programs,” in *Proceedings of the 10th ACM SIGSOFT Symposium on Foundations of Software Engineering*, ser. SIGSOFT ’02/FSE-10. New York, NY, USA: Association for Computing Machinery, 2002. doi: 10.1145/587051.587053. ISBN 1581135149 p. 1–10.
- [26] A. Groce and W. Visser, “What went wrong: Explaining counterexamples,” in *Model Checking Software*, T. Ball and S. K. Rajamani, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 121–136.
- [27] M. Reniers and S. P. Reiss, “Fault localization with nearest neighbor queries,” in *18th IEEE International Conference on Automated Software Engineering*, 2003. *Proceedings.*, 2003. doi: 10.1109/ASE.2003.1240292 pp. 30–39.
- [28] A. Groce, “Error explanation with distance metrics,” in *Tools and Algorithms for the Construction and Analysis of Systems*, K. Jensen and A. Podolski, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 108–122.
- [29] A. Groce, S. Chaki, D. Kroening, and O. Strichman, “Error explanation with distance metrics,” *International Journal on Software Tools for Technology Transfer*, vol. 8, no. 3, pp. 229–247, 2006. [Online]. Available: <https://doi.org/10.1007/s10009-005-0202-0>
- [30] L. de Alfaro, T. A. Henzinger, and F. Y. C. Mang, “Detecting errors before reaching them,” in *Computer Aided Verification*, E. A. Emerson and A. P. Sistla, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 186–201.
- [31] I. Beer, S. Ben-David, H. Chockler, A. Orni, and R. J. Treffer, “Explaining counterexamples using causality,” *Formal Methods in System Design*, vol. 40, no. 1, pp. 20–40, 2012.
- [32] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NeurIPS’17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 4768–4777.
- [33] S. M. Lundberg, G. G. Erion, and S.-I. Lee, “Consistent individualized feature attribution for tree ensembles,” Tech. Rep. arxiv:1802.03888, 2018.
- [34] M. Sundararajan and A. Najmi, “The many shapley values for model explanation,” in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 2020, pp. 9269–9278.
- [35] M. K. Tarkowski, T. P. Michalak, T. Rahwan, and M. Wooldridge, “Game-theoretic network centrality: A review,” 2017.
- [36] V. Yazdanpanah, M. Dastani, W. Jamroga, N. Alechina, and B. Logan, “Strategic responsibility under imperfect information,” in *Proceedings of the 18th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. International Foundation for Autonomous Agents and Multiagent Systems, 2019, p. 592–600.
- [37] X. Deng and C. Papadimitriou, “On the complexity of cooperative solution concepts,” *Mathematics of Operations Research*, vol. 19, no. 2, pp. 257–266, 1994.
- [38] S. S. Fatima, M. J. Wooldridge, and N. R. Jennings, “A linear approximation method for the Shapley value,” *Artificial Intelligence*, vol. 172, no. 14, pp. 1673–1699, 2008. [Online]. Available: <https://doi.org/10.1016/j.artint.2008.05.003>
- [39] O. Skibski, T. Rahwan, T. P. Michalak, and M. J. Wooldridge, “Enumerating connected subgraphs and computing the Myerson and

- Shapley values in graph-restricted games,” *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 15:1–15:25, 2019. [Online]. Available: <https://doi.org/10.1145/3235026>
- [40] O. Skibski, T. P. Michalak, Y. Sakurai, M. J. Wooldridge, and M. Yokoo, “Partition decision trees: representation for efficient computation of the Shapley value extended to games with externalities,” *Autonomous Agents and Multi-Agent Systems*, vol. 34, no. 1, p. 11, 2020. [Online]. Available: <https://doi.org/10.1007/s10458-019-09429-7>
- [41] E. Algaba, V. Fragnelli, and J. Sánchez-Soriano, *Handbook of the Shapley value*. CRC Press, 2019.
- [42] K. G. Larsen and B. Thomsen, “A modal process logic,” in *[1988] Proceedings. Third Annual Symposium on Logic in Computer Science*, Jul. 1988. doi: 10.1109/LICS.1988.5119 pp. 203–210.
- [43] J. Křetínský, “30 Years of Modal Transition Systems: Survey of Extensions and Analysis,” in *Models, Algorithms, Logics and Tools: Essays Dedicated to Kim Guldstrand Larsen on the Occasion of His 60th Birthday*, L. Aceto, G. Bacci, G. Bacci, A. Ingólfssdóttir, A. Legay, and R. Mardare, Eds., 2017, pp. 36–74. [Online]. Available: https://doi.org/10.1007/978-3-319-63121-9_3
- [44] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model checking*. MIT Press, 2001. ISBN 978-0-262-03270-4. [Online]. Available: <http://books.google.de/books?id=Nmc4wEaLXFEC>
- [45] C. Baier and J. Katoen, *Principles of model checking*. MIT Press, 2008. ISBN 978-0-262-02649-9
- [46] E. Grädel, W. Thomas, and T. Wilke, Eds., *Automata, Logics, and Infinite Games: A Guide to Current Research*, ser. Lecture Notes in Computer Science, vol. 2500. Springer, 2002. [Online]. Available: <https://doi.org/10.1007/3-540-36387-4>
- [47] C. H. Papadimitriou, *Computational complexity*. Academic Internet Publ., 2007. ISBN 978-1-4288-1409-7
- [48] K. Prasad and J. S. Kelly, “NP-completeness of some problems concerning voting games,” *International Journal of Game Theory*, vol. 19, no. 1, p. 1–9, Apr. 1990. doi: 10.1007/BF01753703. [Online]. Available: <https://doi.org/10.1007/BF01753703>
- [49] N. Beneš, I. Černá, and J. Křetínský, “Modal Transition Systems: Composition and LTL Model Checking,” in *Automated Technology for Verification and Analysis (ATVA)*, 2011. ISBN 978-3-642-24372-1 pp. 228–242. [Online]. Available: https://doi.org/10.1007/978-3-642-24372-1_17
- [50] S. S. Bauer, U. Fahrenberg, L. Juhl, K. G. Larsen, A. Legay, and C. Thrane, “Quantitative Refinement for Weighted Modal Transition Systems,” in *Mathematical Foundations of Computer Science (MFCS)*, 2011, pp. 60–71. [Online]. Available: https://doi.org/10.1007/978-3-642-22993-0_9
- [51] A. Antonik, M. Huth, K. G. Larsen, U. Nyman, and A. Wąsowski, “Complexity of Decision Problems for Mixed and Modal Specifications,” in *Foundations of Software Science and Computational Structures (FOSSACS)*, 2008, pp. 112–126. [Online]. Available: https://doi.org/10.1007/978-3-540-78499-9_9
- [52] M. Huth, R. Jagadeesan, and D. A. Schmidt, “Modal transition systems: A foundation for three-valued program analysis,” in *Programming Languages and Systems, 10th European Symposium on Programming (ESOP)*, 2001, pp. 155–169. [Online]. Available: https://doi.org/10.1007/3-540-45309-1_11
- [53] M. Lange, “Games for modal and temporal logics,” *Ph.D. Thesis*, 2003.
- [54] G. Owen, *Game theory*. Emerald Group Publishing Limited, 2013. ISBN 978-1-781-90507-4
- [55] J. Nash, “Non-cooperative games,” *Annals of Mathematics*, vol. 54, no. 2, pp. 286–295, 1951. [Online]. Available: <http://www.jstor.org/stable/1969529>
- [56] X. Chen and X. Deng, “Settling the complexity of two-player nash equilibrium,” in *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS’06)*. IEEE, 2006, pp. 261–272.
- [57] L. Khachiyan, “A polynomial algorithm in linear programming,” *Doklady Akademii Nauk SSSR*, vol. 244, pp. 1093–1096, 1979.
- [58] M. Jurdziński, “Deciding the winner in parity games is in $UP \cap co-UP$,” *Information Processing Letters*, vol. 68, no. 3, pp. 119–124, 1998. [Online]. Available: [https://doi.org/10.1016/S0020-0190\(98\)00150-1](https://doi.org/10.1016/S0020-0190(98)00150-1)
- [59] N. Creignou and M. Hermann, “Complexity of generalized satisfiability counting problems,” *Information and Computation*, vol. 125, no. 1, pp. 1 – 12, 1996. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0890540196900164>
- [60] L. J. Stockmeyer, “The polynomial-time hierarchy,” *Theoretical Computer Science*, vol. 3, no. 1, pp. 1 – 22, 1976. [Online]. Available: [https://doi.org/10.1016/0304-3975\(76\)90061-X](https://doi.org/10.1016/0304-3975(76)90061-X)
- [61] C. Wrathall, “Complete sets and the polynomial-time hierarchy,” *Theoretical Computer Science*, vol. 3, no. 1, pp. 23 – 33, 1976. [Online]. Available: [https://doi.org/10.1016/0304-3975\(76\)90062-1](https://doi.org/10.1016/0304-3975(76)90062-1)
- [62] N. Klarlund, “Progress measures, immediate determinacy, and a subset construction for tree automata,” *Annals of Pure and Applied Logic*, vol. 69, no. 2, pp. 243 – 268, 1994. [Online]. Available: [https://doi.org/10.1016/0168-0072\(94\)90086-8](https://doi.org/10.1016/0168-0072(94)90086-8)

A. The full logic

Proof of Theorem IV.3

The usefulness and importance threshold problems for LTL with respect to Kripke structures are 2EXPTIME-complete.

Proof. First, as one can solve LTL games in doubly exponential time, one can compute the value of any subset of the states of \mathcal{K} in doubly exponential time as well. There are exponentially many such subsets, thus the computation of all those values takes again doubly exponential time. The computation of the importance then comes down to enumerating orderings of the states and computing the sum along the way. As a result, one can compute the importance and compare it with τ in doubly exponential time, thus the importance threshold problem (and thus also the usefulness one) is in 2EXPTIME.

For the hardness, we prove that the usefulness problem is 2EXPTIME-hard in the case where the states are partitioned in singletons. The hardness of the usefulness and importance threshold problems follow directly. We reduce the problem of solving LTL games. Let $\mathcal{K} = (S, AP, \Delta, init, \lambda)$ be a Kripke structure, let φ be an LTL formula, and let $V_{\text{Sat}} \sqcup V_{\text{Unsat}} = S$ be a partition of S between states of Sat and Unsat. We consider the LTL game \mathcal{G} induced by those parameters.

Consider the Kripke structure $\mathcal{K}' = (S', AP', \Delta', c_s, \lambda')$ with $S' = S \cup \{c_s, c_u, sink, t\}$, $AP' = AP \cup S'$, and

$$\begin{aligned} \Delta' &= \Delta \cup \{(s, sink) \mid s \in S'\} \\ &\cup \{(c_s, s) \mid s \in V_{\text{Unsat}}\} \cup \{(c_u, s) \mid s \in V_{\text{Sat}}\} \\ &\cup \{(c_s, c_u), (c_u, t), (t, init)\} \end{aligned}$$

and for all $s \in S'$, $\lambda'(s) = \lambda(s) \cup \{s\}$ if $s \in S$ and $\lambda'(s) = \{s\}$ otherwise. In other words, every state is labeled with its own name. See Figure 5 for an illustration of the construction.

Let $\varphi' = \neg\varphi_{\text{checkUnsat}} \vee (\varphi_{\text{checkSat}} \wedge X^3\varphi)$ with

$$\begin{aligned} \varphi_{\text{checkSat}} &= \neg X sink \\ &\wedge X \neg c_u \Rightarrow X^2 sink \\ &\wedge X^2 t \Rightarrow X^3 init \\ &\wedge X^3 G \left(\bigvee_{s \in V_{\text{Sat}}} s \Rightarrow \neg X sink \right) \end{aligned}$$

$$\begin{aligned} \varphi_{\text{checkUnsat}} &= [X c_u \Rightarrow (\neg X^2 sink \wedge (\neg X^2 t \Rightarrow X^3 sink))] \\ &\wedge X^3 G \left(\bigvee_{s \in V_{\text{Unsat}}} s \Rightarrow \neg X sink \right) \end{aligned}$$

This construction can be done in logarithmic space. The intuition is that if some state in V_{Unsat} belongs to Sat then she can win by going from c_s to that state and then to $init$. Similarly if some state of V_{Sat} belongs to Unsat then he can win by going to that state from c_u and then to $init$. In both cases players win without using t . The remaining case is when Sat owns states of V_{Sat} and Unsat of V_{Unsat} . Then if Unsat owns t he can win by going from there to $init$, otherwise

the players have to play the original game \mathcal{G} from $init$. As a result t is useful if and only if Sat wins \mathcal{G} . We will now prove that the state t is useful with respect to φ' if and only if Sat wins the original LTL game.

First suppose that Sat wins \mathcal{G} , then we consider $T = \{c_s\} \cup V_{\text{Sat}}$. Player Sat loses with T :

- If she goes from c_s to $init$ she loses.
- If she goes from c_s to a state of V_{Unsat} Unsat can then go to some state different from $init$ and not satisfy $\varphi_{\text{checkSat}}$ while satisfying $\varphi_{\text{checkUnsat}}$ (recall that in our definition of Kripke structure we assume every state to have at least one outgoing transition).
- If she goes from c_s to c_u then Unsat can go to t then $init$ and not satisfy $\varphi_{\text{checkSat}}$ while satisfying $\varphi_{\text{checkUnsat}}$.

Moreover, player Sat wins with $T \cup \{t\}$, as she can start by going from c_s to c_u and:

- If Unsat goes to $init$ from c_u he loses.
- If Unsat goes from c_u to a state of V_{Sat} , Sat can then go to some state different from $init$ and not satisfy $\varphi_{\text{checkUnsat}}$.
- If Unsat goes from c_u to s then Sat can go to $init$ and then win by playing a winning strategy for \mathcal{G} , thus satisfying $\varphi_{\text{checkSat}} \wedge X^3\varphi$.

Thus t is useful.

Now suppose that t is useful, let $T \subseteq S'$ be a set of states such that (t, T) is critical. T has to contain c_s as otherwise Unsat can go from c_s to $init$ directly and make Sat lose with $T \cup \{t\}$. If Sat had a winning strategy with $T \cup \{t\}$ not going from c_s to c_u , then she would also win with just T by applying this strategy as t is then never reached.

As a result, Sat with $T \cup \{t\}$ has to go from c_s to c_u . As a consequence, T has to be disjoint from V_{Unsat} , as otherwise Sat with T could go from c_s to a state in $T \cap V_{\text{Unsat}}$ and from there to $init$, unsatisfying $\varphi_{\text{checkUnsat}}$. Further, c_u cannot be in T as otherwise Sat could win by going from c_u to $init$.

Finally, Unsat cannot win when Sat has T by going from c_u to a state different from t as otherwise he could win when Sat has $T \cup \{t\}$ with the same strategy. As a consequence, T has to contain V_{Sat} , as if not Unsat could go from c_u to a state in $V_{\text{Sat}} \setminus T$ and then $init$, winning the game.

Whether $init$ is in T is irrelevant to the game as there is only one outgoing transition from $init$. Thus we can assume that $T = \{c_s\} \cup V_{\text{Sat}}$. Suppose Unsat wins \mathcal{G} , and consider the game where Sat has $T \cup \{t\}$. As Sat has to go from c_s to c_u to win, Unsat can then go from c_u to s , and Sat has to go to $init$. Then Unsat can apply his winning strategy for \mathcal{G} , as Sat loses if she goes to $init$ and thus cannot go out of S . This makes Unsat win, contradicting the hypothesis that Sat wins with $T \cup \{t\}$. In conclusion, Sat wins \mathcal{G} .

As a result, the usefulness and importance threshold problems are 2EXPTIME-complete for LTL. \square

Proof of Theorem IV.6

The importance computation problem for reachability, Büchi, parity and explicit Muller conditions with respect to Kripke structures are #P-complete.

Proof. First the upper bound for reachability, Büchi and explicit Muller conditions is obtained by constructing a Turing machine guessing an ordering of $\{1, \dots, n\}$, and accepting if the set J of indices coming after i in the ordering is such that (i, J) is critical, which can be checked in polynomial time. The number of accepting runs is the number of permutations satisfying this condition, i.e., $n! \mathcal{I}(V_i)$. The problem is therefore in #P.

For parity conditions, we rely on the result by Jurdziński that solving parity games can be done by a polynomial-time unambiguous Turing machine, i.e., a nondeterministic machine that has at most one accepting run on every input [58].

This allows us to build a machine that takes as input a Kripke structure \mathcal{K} , a partition V_1, \dots, V_n of the states, an index i and a coloring c and guesses an ordering of $\{1, \dots, n\}$. Let J be the set of indices coming after i in the permutation, our machine can simulate the unambiguous Turing machine in order to check that Sat wins with $\bigcup_{j \in J \cup \{i\}} V_j$ and Unsat wins with $\bigcup_{j \in J} V_j$. The number of accepting runs of this machine is precisely $n! \mathcal{I}(V_i)$.

Our reduction to show #P-hardness is from the problem of counting solutions to a 1-in-3SAT instance, i.e., given a 3SAT formula, counting the number of valuations such that every clause has exactly one satisfied literal. This problem was shown to be #P-complete by Creignou and Hermann [59].

Let $\varphi = C_1 \wedge C_2 \wedge \dots \wedge C_k$ be a 3SAT formula, with $C_j = (\ell_j^1 \vee \ell_j^2 \vee \ell_j^3)$ for all j , and let $\{x_1, \dots, x_n\}$ be the set of variables appearing in φ . We first construct the formula $\psi = \bigwedge_{j=1}^k C_j \wedge \bigwedge_{j=1}^n (x_j \vee \neg x_j) \wedge \bigwedge_{j=1}^k C_j^{1,2} \wedge C_j^{2,3} \wedge C_j^{3,1}$ with $C_j^{i_1, i_2} = (\neg \ell_j^{i_1} \vee \neg \ell_j^{i_2})$.

One can check that a valuation $\nu : \{x_1, \dots, x_n\} \rightarrow \{\perp, \top\}$ satisfies ψ if and only if it satisfies exactly one literal per clause in φ .

Furthermore if a valuation satisfies ψ then it satisfies exactly one literal in every clause except for exactly one of $C_j^{1,2}, C_j^{2,3}, C_j^{3,1}$ for each j , in which it satisfies both literals.

We reuse the construction from the proof of Proposition IV.5, with ψ as our 3SAT instance. Recall that this construction used a reachability condition, easily expressible as a Büchi, parity or Muller condition, making the reduction work for all those winning conditions. As sink and f only have one outgoing transition, they have no influence on the satisfaction of a specification by a set of states, thus their importance is 0. As a consequence, by Corollary III.6 they can be ignored in the computation of the importance, thus we will only consider set of states containing neither. Then a team of states T makes (s, T) critical if and only if it contains all the c_i but not s and there exists a valuation ν satisfying ψ such that T contains exactly the states associated literals satisfied by ν , except in clauses $C_j^{i_1, i_2}$ in which ν satisfies both literals, in which T contains either one of the two states or both.

As a result, for every valuation ν satisfying ψ , we have exactly 3^k sets of states T making s critical and matching that valuation. Indeed, T is completely determined by ν except for one $C_j^{i_1, i_2}$ for each $1 \leq j \leq k$, in which it has three possibilities: contain the first literal, the second, or both.

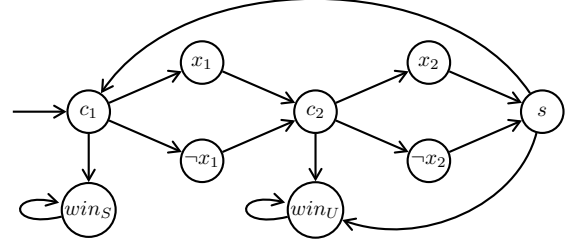


Fig. 11. Kripke structure corresponding to formula $\forall x_1, \exists x_2, x_1 \wedge \neg x_2$

A brief analysis shows that for each such valuation ν , there are, for each $0 \leq i \leq k$, $\binom{i}{k} 2^{k-i}$ corresponding teams of size $2i + k - i + 2k + 2n$ (those teams being the ones containing both literals in i out of the k clauses $C_j^{i_1, i_2}$ in which ν satisfies both literals), adding up to 3^k teams.

Let N be the total number of states in the Kripke structure. By Corollary III.6, the number of valuations satisfying φ with exactly one satisfied literal per clause is therefore $\frac{(N-2)!}{N!M} N! I(s)$, with

$$M = \sum_{i=1}^k \binom{i}{k} 2^{k-i} (i + 3k + 2n)! ([N - 2] - i - 3k - 2n)!$$

As M can be computed in polynomial time, the problem is therefore #P-complete. \square

Proof of Theorem IV.7

The value, usefulness and importance threshold problems for Emerson-Lei conditions are PSPACE-complete.

Proof. The upper bounds arise from the complexity of solving Emerson-Lei games, which are PSPACE-complete [10]. As enumerating permutations of the state can be done in linear space, one can compute the importance of a set of states in PSPACE.

For the lower bounds, we adapt a classic proof that Emerson-Lei games are PSPACE-hard to our framework. We only need to prove that the usefulness problem is PSPACE-hard as the importance threshold problem reduces to it. Further, we only use the particular case when the set of states is partitioned in singletons.

We reduce the QSAT problem. Let $Q_1 x_1 \dots Q_k x_k \psi$ be a QSAT instance, we consider the following Kripke structure:

- $\{c_i, x_i, \neg x_i \mid 1 \leq i \leq k\} \cup \{s, \text{win}_S, \text{win}_U\}$ is the set of states, c_1 is the only initial state.
- For all $1 \leq i \leq k$ there are transitions $(c_i, x_i), (c_i, \neg x_i), (x_i, c_{i+1}), (\neg x_i, c_{i+1})$, with $c_{k+1} = s$. There is also a transition (c_i, win_U) if $Q_i = \exists$ and (c_i, win_S) if $Q_i = \forall$. The remaining transitions are $(s, c_1), (s, \text{win}_U), (\text{win}_S, \text{win}_S), (\text{win}_U, \text{win}_U)$.

The labeling is irrelevant here. Figure 11 illustrates the construction.

For all $1 \leq i \leq k$ let

$$\varphi_i = \text{Inf}(x_i) \wedge \text{Inf}(\neg x_i) \wedge \bigwedge_{j=1}^i \neg(\text{Inf}(x_j) \wedge \text{Inf}(\neg x_j))$$

expressing that i is the minimal i such that both x_i and $\neg x_i$ are visited infinitely many times.

We take as winning condition for Sat the formula

$$(\psi' \vee \text{Inf}(\text{win}_S)) \vee \bigvee_{Q_i=\forall} \varphi_i \wedge \neg \text{Inf}(\text{win}_U) \wedge \bigwedge_{Q_i=\exists} \neg \varphi_i$$

where ψ' is ψ in which every x_i has been replaced with $\text{Inf}(x_i)$.

This construction can be done in logarithmic space. We will now prove that the QSAT formula is valid if and only if state s is useful.

Suppose the QSAT formula is valid, let $T = \{c_i \mid Q_i = \exists\}$. Clearly Sat loses with T as by taking the transition to win_U from s , Unsat can guarantee that every play reaches win_U and thus wins.

As the QSAT formula is valid, there exist functions $(f_i)_{Q_i=\exists}$ such that for all i $f_i : \{\top, \perp\}^{i-1} \rightarrow \{\top, \perp\}$ and for all $\nu : \{x_1, \dots, x_k\} \rightarrow \{\top, \perp\}$ such that for all f_i we have $\nu(x_i) = f_i(\nu(x_1), \dots, \nu(x_{i-1}))$, ν satisfies ψ .

Further, as Sat makes all the existential choices, if Sat chooses according to f_i from every c_i she owns, and takes the transition to c_1 from s . Suppose Sat takes the transitions to x_i and $\neg x_i$ infinitely many times, then as Sat plays according to functions f_i , it means there exists a $j < i$ such that x_j and $\neg x_j$ were visited infinitely many times.

As a consequence the minimal i , if it exists, such that x_i and $\neg x_i$ are visited infinitely many times is such that $Q_i = \forall$. If it exists then φ_i is satisfied, while φ_j is not satisfied for any other j , and as win_U is never visited, Sat wins.

If it does not exist, then for all j exactly one of $x_j, \neg x_j$ is visited infinitely many times, and as Sat plays according to the f_i , we have that ψ' is satisfied. As no φ_j is satisfied and win_U is never visited, Sat wins.

Now suppose the QSAT formula is not satisfiable, and suppose there exists T such that Sat wins with $T \cup \{s\}$ but not with T . If there exists $c_i \in T$ such that $Q_i = \forall$ or $c_i \notin T$ such that $Q_i = \exists$ then either Sat can reach win_S and win with T , or Unsat can reach win_U and win while Sat has $T \cup \{s\}$. Whether $\text{win}_S, \text{win}_U$ or the $x_i, \neg x_i$ belong to T is irrelevant as they have only one outgoing transition.

Thus we can assume that $T = \{c_i \mid Q_i = \exists\}$. By similar arguments as above, there exist functions $(f_i)_{Q_i=\forall}$ such that f_i associates to the $i-1$ values of the previous literals a valuation of x_i , and any valuation respecting those functions does not satisfy ψ . And again by similar arguments as above, playing according to those functions allows Unsat to win the game while Sat has $T \cup \{s\}$, contradicting the hypothesis that Sat wins with $T \cup \{s\}$. As a result, s is not useful. \square

Proof of Proposition IV.8

The usefulness problem for Rabin conditions is Σ_2^P -complete.

Proof. For the upper bound we simply consider a non-deterministic Turing machine guessing a set of indices J and calling an NP oracle twice to check that player Sat wins with $\bigcup_{j \in J \cup \{i\}} V_j$ as set of states and loses with just $\bigcup_{j \in J} V_j$.

For the lower bound, we reduce the dual of the $\forall\exists\text{SAT}$ problem, known to be Π_2^P -complete [60] [61]. Given a formula $\varphi = \forall(x_i)_{1 \leq i \leq n}, \exists(y_i)_{1 \leq i \leq p} \psi$ with $\psi = \bigvee_{i=1}^k Cl_i$ a $\forall\exists\text{SAT}$ instance, we are going to construct a Kripke structure \mathcal{K} (with states partitioned in singletons), a state s and a Rabin condition R such that s is useful to \mathcal{K} with respect to R if and only if this formula is *not* valid.

First of all note that we can assume that every clause contains an existential variable y_i . Indeed, any clause $(\ell_1 \vee \ell_2 \vee \ell_3)$ can be replaced by $(\ell_1 \vee \ell_2 \vee y) \wedge (\neg y \vee \ell_3) \wedge (\neg \ell_3 \vee y)$, with y a fresh variable which we add to the set of existential ones. One can check that we obtain a formula equisatisfiable to the previous one.

Consider the structure \mathcal{K} whose states are elements of

$$\begin{aligned} & \{c_i, x_i, \neg x_i, c'_i, x'_i, \neg x'_i \mid 1 \leq i \leq n\} \\ & \cup \{sk_{x_i}, sk_{\neg x_i}, ret_{x_i}, ret_{\neg x_i} \mid 1 \leq i \leq n\} \\ & \cup \{y_j, \neg y_j \mid 1 \leq j \leq p\} \cup \{Cl_i \mid 1 \leq i \leq k\} \cup \{s, sink\} \end{aligned}$$

whose initial state is c_1 and whose transitions are as follows:

- There are transitions from $init$ to itself, to c_1 and to every Cl_j .
- For all i there are transitions from c_i to x_i and $\neg x_i$ and from x_i and $\neg x_i$ to c_{i+1} , with $c_{n+1} = c'_1$.
- For all i , for all $\ell \in \{x_i, \neg x_i\}$, there are transitions $(\ell, sk_\ell), (sk_\ell, \neg \ell')$
- We have transitions $(c'_i, x'_i), (c'_i, \neg x'_i), (x'_i, c'_{i+1}), (\neg x'_i, c'_{i+1})$ for all $1 \leq i \leq n$, with the convention $c'_{n+1} = s$.
- For all ℓ of the form x_i or $\neg x_i$, for all clause Cl_j containing ℓ there are transitions (Cl_j, ret_ℓ) and (ret_ℓ, ℓ) .
- For all ℓ of the form y_i or $\neg y_i$, for all Cl_j containing ℓ , there is a transition (Cl_j, ℓ) and a transition (ℓ, c_1) .
- For all clause Cl_j there is a transition (s, Cl_j) .
- There are transitions from all $c_i, c'_i, x_i, \neg x_i, x'_i, \neg x'_i, Cl_j$ to $sink$.

Figure 12 illustrates the construction. There are transitions from the blue and white states to $skip$, which is omitted on the picture. The blue states are the ones hardcoded to belong to Sat, the grey ones are the ones that have only one outgoing transition, and the white ones are the ones encoding the valuation of the x_i .

As states $sink, ret_\ell, sk_\ell, y_i, \neg y_i$ have only one outgoing transition, whether they belong to Sat or Unsat has no consequence on the game. In the proof that follows we will ignore which player they belong to.

We take as Rabin condition

$$\begin{aligned} R = & \{(\{y_i\}, \{\neg y_i\}), (\{\neg y_i\}, \{y_i\}) \mid 1 \leq i \leq p\} \\ & \cup \{(\{sk_\ell\}, \emptyset), (\{ret_\ell\}, \emptyset) \mid 1 \leq i \leq n, \ell \in \{x_i, \neg x_i\}\} \\ & \cup \{(\emptyset, \{c'_1, sink\}), (\{init\}, \emptyset)\}. \end{aligned}$$

The construction can be done in logarithmic space. We will now show that the formula φ is not valid if and only if s

First let us justify that this problem is $\#P^{NP}$ -hard. Let M be a non-deterministic Turing machine with an oracle solving an NP-complete problem (say SAT), let w be an input.

We can assume without loss of generality that M only makes one query to the oracle, and accepts if and only if the answer is negative.

Indeed, say M has to make queries ψ_1, \dots, ψ_k to the oracle, all over existential variables y_1, \dots, y_p . It can nondeterministically guess the answers of the oracle and delay the verification to the end of the run.

Now let us define the order \leq on valuations of the existential variables as the lexicographic order, a valuation ν being seen as the tuple $(\nu(y_1), \dots, \nu(y_p))$ and with the convention $\perp \leq \top$. Then for the positive answers M can guess a *minimal* witness valuation for the y_i with respect to \leq , negate the formula. The problem, given a SAT instance and a valuation of the existential variables, of checking whether this is the minimal valuation witnessing the satisfiability of the formula, is clearly in coNP. As a result M can guess the minimal valuation of the y_i witnessing the satisfiability of the formula, and then turn it into a SAT formula unsatisfiable if and only if the guess is correct. As there is for every SAT formula an unique minimal valuation satisfying it, M can only make one correct guess, thus its number of runs is unchanged.

Finally, in the end M has to make the oracle check a disjunction of \exists formulas, it can rename variables in order to merge them all into one equivalent SAT instance, and accept if and only if the oracle rejects that formula.

We now use the classical encoding of Turing machines in 3CNF formulas to construct a 3CNF formula ψ over variables $x_1, \dots, x_n, y_1, \dots, y_p, r_1, \dots, r_k$ such that for all valuations of the x_i, y_i (encoding respectively the nondeterministic choices of M and the ones of the oracle), there is a non-accepting run of M on w if and only if there exists a valuation of the r_i (encoding the runs of M and the oracle) satisfying the formula along with these valuations of x_i, y_i .

As a result, there is an accepting run of M on w if and only if the formula $\forall(x_i)\exists(y_i), (r_i)\varphi_1$ is not valid, and the valuations of the x_i witnessing non-validity are in bijection with the runs of M .

Hence the problem is $\#P^{NP}$ -hard.

Now in order to prove the hardness for the importance computation problem for Rabin conditions, we use the same construction as in the proof of Proposition IV.8. Let φ be a 3CNF formula with k clauses over variables $x_1, \dots, x_n, y_1, \dots, y_p$, we consider the Kripke structure from that proof.

Furthermore, a set of states T makes (s, T) critical if and only if it contains the c_i, c'_i, Cl_i , and the $x_i, \neg x_i, x'_i \neg x'_i$ encoding a valuation of the x_i such that for all valuation of the y_i , the combination of the two valuations does not satisfy φ .

Note that states $sk_\ell, ret_\ell, y_i, \neg y_i$ all have one outgoing transition and thus have importance 0.

As those sets T all have the same size $k + 4n$, the formula from Corollary III.6 gives us that the number of valuations of the x_i such that for all valuation of the y_i ,

the combination of the two valuations does not satisfy φ is $\frac{P!}{N!(k+4n)!(P-k+4n)!} N! I(s)$ where N is the number of states in the Kripke structure and P the number of states minus the $sk_\ell, ret_\ell, y_i, \neg y_i$.

As $\frac{P!}{N!(k+4n)!(P-k+4n)!}$ can be computed in polynomial time, the importance computation problem for Rabin conditions is $\#P^{NP}$ -complete. \square

Proof of Proposition V.2

The value problem for two-turn CTL is Σ_2^P -complete.

Proof. One can reformulate the problem as the existence of a subset of outgoing transitions from V_{Sat} such that for all subsets of outgoing transitions from V_{Unsat} , the structure yielded by those subsets of transitions satisfies φ .

As those subsets of transitions are of polynomial size, and as the satisfaction of a Kripke structure by a CTL formula can be checked in polynomial time, the problem is in Σ_2^P .

We now prove the lower bound, by reducing $\exists\forall\text{SAT}$. Let $\exists(x_i)_{1 \leq i \leq n}, \forall(y_i)_{1 \leq i \leq k} \psi$ with ψ quantifier-free be a $\exists\forall\text{SAT}$ instance. Without loss of generality, we assume that all the negations in ψ have been pushed to the atomic propositions.

We consider the following modal transition system $\mathcal{M} = (S, AP, \Delta_{\text{must}}, \Delta_{\text{may}}, \text{init}, \lambda)$ with:

- $S = \{\text{sink}\} \cup \{c_i, x_i, \neg x_i \mid 1 \leq i \leq n+k\}$. The initial state is c_1 .
- $AP = \{x_i \mid 1 \leq i \leq n+k\}$ and $\lambda(x_i) = \{x_i\}$ for all $1 \leq i \leq n+k$ and $\lambda(s) = \emptyset$ for all other $s \in S$.
- $\Delta_{\text{must}} = \{(x_{n+k}, \text{sink}), (\neg x_{n+k}, \text{sink}), (\text{sink}, \text{sink})\} \cup \{(x_i, c_{i+1}), (\neg x_i, c_{i+1}) \mid 1 \leq i \leq n+k-1\}$.
- $\Delta_{\text{may}} = \{(c_i, x_i), (c_i, \neg x_i) \mid 1 \leq i \leq n+k\}$.

We split S into

$$V_{\text{Sat}} = \{\text{sink}\} \cup \{x_i, \neg x_i \mid 1 \leq i \leq n+k\} \cup \{c_i \mid 1 \leq i \leq n\}$$

$$\text{and } V_{\text{Unsat}} = \{c_i \mid n+1 \leq i \leq n+k\}.$$

Informally, we are going to make players choose valuations of the variables through their choices of transitions. The CTL formula will then ensure that the choices of transitions yield well-defined valuations, and that these valuations satisfy the SAT formula.

With that goal in mind, we define the specification as follows:

$$\varphi = (\varphi_{\text{SAT}} \wedge \varphi_{\text{checkSat}}) \vee \varphi_{\text{checkUnsat}}$$

$$\varphi_{\text{checkSat}} = \bigwedge_{i=1}^n EX^{2i-2} (AX(x_i) \vee AX(\neg x_i)) \wedge EX \top$$

$$\varphi_{\text{checkUnsat}} = EX^{2n} \bigvee_{i=1}^n EX^{2i-2} (EX(x_i) \wedge EX(\neg x_i)) \wedge AX \perp$$

and φ_{SAT} is ψ where every x_p has been replaced by $EX^{2p-1} x_p$ and every $\neg x_p$ replaced by $EX^{2p-1} \neg x_p$. Recall that we assumed that ψ only has negations in front of atomic

propositions. This construction can be done in logarithmic space.

The idea is that φ_{SAT} mimics ψ in order to check that there exists a path in the structure obtained through the game matching a valuation satisfying ψ . Meanwhile, formulas $\varphi_{checkSat}$ and $\varphi_{checkUnsat}$ ensure that players never pick both x_i or neither.

Now for the formal proof, suppose there exists a valuation $\nu_1 : \{x_1, \dots, x_n\} \rightarrow \{\top, \perp\}$ such that for every valuation $\nu_2 : \{x_{n+1}, \dots, x_{n+k}\} \rightarrow \{\top, \perp\}$, the combination of ν_1 and ν_2 satisfies ψ .

Then let $\sigma_1(c_i) = \begin{cases} x_i & \text{if } \nu_1(x_i) = \top \\ \neg x_i & \text{otherwise} \end{cases}$ for $1 \leq i \leq n$

and let σ_2 be a pure strategy for Unsat. Clearly as $|\sigma_1(c_i)| = 1$ for all i , the resulting structure satisfies $\varphi_{checkSat}$. If $|\sigma_2(c_i)| = 0$ for some i , then $\varphi_{checkUnsat}$ is satisfied, thus so is φ . If Unsat gives every c_i a successor, then there is a path from c_1 to $sink$, representing a valuation whose projection to $\{x_1, \dots, x_n\}$ matches ν_1 . As a result, ψ is satisfied by this valuation, thus φ_{SAT} is satisfied by the structure yielded by σ_1 and σ_2 , hence so is φ .

Now suppose there exists a pure strategy σ_1 for Sat such that for all pure strategy σ_2 for Unsat, σ_1, σ_2 yield a structure satisfying φ . For all $1 \leq i \leq n$, if we had $|\sigma(c_i)| = 0$ then neither $\varphi_{checkSat}$ nor $\varphi_{checkUnsat}$ would be satisfied, and if we had $|\sigma(c_i)| > 1$ then $\varphi_{checkSat}$ would not be satisfied, and Unsat could win by choosing one outgoing transition for each c_i he owns, thereby unsatisfying $\varphi_{checkUnsat}$. As a result, σ_1 selects exactly one of $\{x_i, \neg x_i\}$ for each i , thus we can define ν_1 the valuation such that

$$\nu_1(x_i) = \begin{cases} \top & \text{if } \sigma_1(c_i) = x_i \\ \perp & \text{otherwise} \end{cases} \quad \text{for } 1 \leq i \leq n$$

Let $\nu_2 : \{x_{n+1}, \dots, x_{n+k}\} \rightarrow \{\top, \perp\}$, we define a corresponding strategy for Unsat as

$$\sigma_2(c_i) = \begin{cases} x_i & \text{if } \nu_2(c_i) = \top \\ \neg x_i & \text{otherwise} \end{cases} \quad \text{for } n+1 \leq i \leq n+k.$$

As σ_1, σ_2 yield a structure satisfying φ , either φ_{SAT} is satisfied or $\varphi_{checkUnsat}$ is. Further, as in that structure every state has exactly one successor, $\varphi_{checkUnsat}$ is not satisfied, thus φ_{SAT} is. As a consequence, the combination of ν_1 and ν_2 satisfies ψ .

We have constructed in logarithmic space a CTL formula, a modal transition system and a subset V_{Sat} of states such that Sat has a pure winning strategy on V_{Sat} if and only if ψ with set of existential variables $\{x_1, \dots, x_n\}$ is in $\exists\forall SAT$.

As a result the value problem corresponding to definition V.1 is Σ_2^P -complete. \square

Proof of Proposition V.3

The usefulness problem for two-turn CTL is Σ_3^P -complete.

Proof. Let $\mathcal{M} = (S, AP, \Delta_{must}, \Delta_{may}, init, \lambda)$ be an MTS, let V_1, \dots, V_n be a partition of S , let $1 \leq i \leq n$, let φ be a CTL formula.

In order to check the usefulness of s , we can guess a set of indices J and a pure strategy $\sigma_1 : \bigcup_{j \in J \cup \{i\}} V_j \rightarrow \Delta_{may}$,

make an adversary choose pure strategies $\sigma'_1 : \bigcup_{j \in J} V_j \rightarrow \Delta_{may}$ and $\sigma_2 : S \setminus (\bigcup_{j \in J \cup \{i\}} V_j) \rightarrow \Delta_{may}$, and then guess a pure strategy $\sigma'_2 : S \setminus \bigcup_{j \in J} V_j \rightarrow \Delta_{may}$ such that the structure yielded by σ_1, σ_2 satisfies φ but the one yielded by σ'_1, σ'_2 does not.

This shows that the problem is in Σ_3^P .

Now let us show hardness. We reduce the problem $\exists\forall\exists SAT$. Let $\exists x_1, \dots, x_n, \forall y_1, \dots, y_k, \exists z_1, \dots, z_p \psi$ with ψ quantifier-free be a $\exists\forall\exists SAT$ instance. We assume without loss of generality that all negations have been pushed to the atomic propositions.

We define the MTS $\mathcal{M} = (S, AP, \Delta_{must}, \Delta_{may}, init, \lambda)$ as follows :

$$\begin{aligned} S = & \{x_i, \neg x_i, c_i^x \mid 1 \leq i \leq n\} \\ & \cup \{y_i, \neg y_i, c_i^y \mid 1 \leq i \leq k\} \\ & \cup \{z_i, \neg z_i, c_i^z \mid 1 \leq i \leq p\} \\ & \cup \{x'_i, \neg x'_i \mid 1 \leq i \leq n\} \cup \{win_S, win_U, s\} \\ AP = & S \end{aligned}$$

$$\begin{aligned} \Delta_{must} = & \{(x_i, c_{i+1}^x), (\neg x_i, c_{i+1}^x) \mid 1 \leq i \leq n-1\} \\ & \cup \{(y_i, c_{i+1}^y), (\neg y_i, c_{i+1}^y) \mid 1 \leq i \leq k-1\} \\ & \cup \{(z_i, c_{i+1}^z), (\neg z_i, c_{i+1}^z) \mid 1 \leq i \leq p-1\} \\ & \cup \{(x_n, c_1^y), (\neg x_n, c_1^y), (y_k, c_1^z), (\neg y_k, c_1^z)\} \\ & \cup \{(x'_i, x'_{i+1}), (x'_i, \neg x'_{i+1}) \mid 1 \leq i \leq n-1\} \\ & \cup \{(\neg x'_i, x'_{i+1}), (\neg x'_i, \neg x'_{i+1}) \mid 1 \leq i \leq n-1\} \\ & \cup \{(x_i, \neg x'_i), (\neg x_i, x'_i) \mid 1 \leq i \leq n\} \\ & \cup \{(x'_n, win_S), (\neg x'_n, win_S)\} \\ & \cup \{(win_S, win_S), (win_U, win_U), (z_p, s), (\neg z_p, s)\} \\ & \cup \{(c_i^y, win_U) \mid 1 \leq i \leq k\} \\ & \cup \{(c_i^z, win_S) \mid 1 \leq i \leq p\} \end{aligned}$$

$$\begin{aligned} \Delta_{may} = & \{(c_i^x, x_i), (c_i^x, \neg x_i) \mid 1 \leq i \leq n\} \\ & \cup \{(c_i^y, y_i), (c_i^y, \neg y_i) \mid 1 \leq i \leq k\} \\ & \cup \{(c_i^z, z_i), (c_i^z, \neg z_i) \mid 1 \leq i \leq p\} \\ & \cup \{(x_i, win_U), (\neg x_i, win_U) \mid 1 \leq i \leq n\} \\ & \cup \{(x'_i, win_U), (\neg x'_i, win_U) \mid 1 \leq i \leq n\} \\ & \cup \{(s, x'_1), (s, \neg x'_1)\} \end{aligned}$$

$\lambda(t) = \{t\}$ for every state t , and the initial state is $init = c_1^x$.

We consider the formula

$$\begin{aligned} \varphi = & (\neg \varphi_{SAT} \wedge \varphi_{checkSat} \wedge AG \neg win_U) \\ & \vee (EF win_S \wedge AG \neg win_U) \\ & \vee \varphi_{checkUnsat} \end{aligned}$$

with

$$\varphi_{checkSat} =$$

$$AG(\bigwedge_{i=1}^k AX(y_i) \vee AX(\neg y_i)) \wedge (\bigwedge_{i=1}^k (EX^{2n+2i-2}) EX \top)$$

$$\varphi_{checkUnsat} =$$

$$EF(\bigvee_{i=1}^p EX(z_i) \wedge EX(\neg z_i)) \vee \bigvee_{i=1}^p (EX^{2n+2k+2i-2}) AX \perp$$

and φ_{SAT} is ψ where every x_i, y_i, z_i has been replaced by respectively EFx_i, Efy_i and EFz_i , and every $\neg x_i, \neg y_i, \neg z_i$ by respectively $AG\neg x_i, AG\neg y_i, AG\neg z_i$. This construction can be done in logarithmic space. The formulas $\varphi_{checkUnsat}$ and $\varphi_{checkSat}$ ensure that the players never allow transitions to both or neither variables from a c_i^x, c_i^y or c_i^z state.

Suppose there exists T such that Sat wins with $T \cup \{s\}$ but loses with T . As all $x_i, \neg x_i, x'_i, \neg x'_i$ have a may transition to win_U , there has to be either a path from c_1^x to c_1^y in T , or a path in T from c_1^x to some x_i or $\neg x_i$, from there a transition to some x'_i or $\neg x'_i$, and a path in T from there to win_S , otherwise Unsat wins both games. In the second case, Sat wins without s , thus we have to be in the first case. In particular for every $x_i \in T$, $x'_i \notin T$ and for every $\neg x_i \in T$, $\neg x'_i \notin T$.

As a result, there has to be a path in \mathcal{M} (using may and must transitions) to all c_i^x, c_i^y, c_i^z from c_1^x . In order for the games with T and $T \cup \{s\}$ to have different winners, every c_i^y has to be in T (as they have a may transition to win_U) and similarly every c_i^z has to not be in T . The formulas $\varphi_{checkSat}$ and $\varphi_{checkUnsat}$ force both players to pick exactly one outgoing transition from each c_i^x, c_i^y, c_i^z .

Now observe that the choice of transitions from s has no impact on the satisfaction of $\neg\varphi_{SAT}, \varphi_{checkSat}, AG\neg win_U$ or $\varphi_{checkUnsat}$. As Unsat has a winning strategy when Sat only has T , this same strategy will ensure that Sat can only win by satisfying $EFwin_S \wedge AG\neg win_U$ in the game with $T \cup \{s\}$. In order to satisfy $EFwin_S \wedge AG\neg win_U$, there has to be a path in T from s to win_S . As a result, at least one of $x'_i, \neg x'_i$ has to be in T . As we have seen before, for every $x_i \in T$, $x'_i \notin T$ and for every $\neg x_i \in T$, $\neg x'_i \notin T$, thus at most one of $x_i, \neg x_i$ can be in T for all $1 \leq i \leq n$. Further, we have seen that at least one of $x_i, \neg x_i$ has to be in T .

As a result, the set of x_i in T with $1 \leq i \leq n$ matches a valuation ν_1 of x_1, \dots, x_n . Let ν_2 be a valuation of y_1, \dots, y_k , suppose Sat picks transitions matching ν_2 from the c_i^y . As Unsat wins the game in which Sat owns only T , and as the satisfaction of both $\varphi_{checkSat}$ and $AG\neg win_U$ is guaranteed by the strategy of Sat, the only possibility is that $\neg\varphi_{SAT}$ is dissatisfied, which Unsat can only achieve by picking transitions matching a valuation ν_3 of z_1, \dots, z_p such that the combination of ν_1, ν_2 and ν_3 satisfies φ . As a result, the $\exists\forall\exists SAT$ instance is true.

Now for the converse, suppose there exists a valuation ν_1 such that for all ν_2 , there exists ν_3 such that their combination satisfies ψ . Let T be such that $T \cap \{x_1, \dots, x_n, \neg x_1, \dots, \neg x_n\}$

and $T \cap \{x'_1, \dots, x'_n, \neg x'_1, \dots, \neg x'_n\}$ both match ν_1 , T contains every c_i^x and c_i^y but does not contain c_i^z for any $1 \leq i \leq p$.

Let us first look at the game in which Sat has states $T \cup \{s\}$. As one of $\{x_i, \neg x_i\}$ belongs to T for all $1 \leq i \leq n$, Sat can choose transitions so that there is a path from c_1 to c_{n+k+1} , and a transition from s to win_S .

If Unsat gives no outgoing transition to one of the c_i with $n+k+1 \leq i \leq n+k+p$, then $\varphi_{checkUnsat}$ is satisfied, thus so is φ . As a result, there is a path from c_{n+k+1} to either win_S or s , and thus also win_S . Hence φ is satisfied in every case, Sat wins that game.

Now let us study the game in which Sat only owns T . No matter which strategy Sat chooses, Unsat can guarantee that $EFwin_S \wedge AG\neg win_U$ is not satisfied by allowing the transition to win_U from every $x_i, \neg x_i, x'_i, \neg x'_i$ it owns, and not allowing any transition from s . This way, all paths to win_S go through states with a transition towards win_U . Unsat can also ensure that $\varphi_{checkUnsat}$ is not satisfied by picking transitions matching some valuation from the c_i he owns.

Assume Sat has a winning strategy to ensure that $\varphi_{checkSat} \wedge \varphi_{checkUnsat} \wedge AG\neg win_U$ is satisfied. As $\varphi_{checkSat}$ and $AG\neg win_U$ are satisfied, the choices of transitions of Sat from the c_i have to match ν_1 and a valuation ν_2 of $\{x_{n+1}, \dots, x_{n+k}\}$. There exists a valuation ν_3 of $\{x_{n+k+1}, \dots, x_{n+k+p}\}$ such that the combination of ν_1, ν_2 and ν_3 satisfies ψ . Then if Unsat chooses transitions from the c_i he owns matching ν_3 , $\neg\varphi_{SAT}$ is not satisfied by the resulting structure, contradicting the existence of a winning strategy for Sat.

We have proven the proposition. \square

Proof of Theorem V.4

The importance computation problem associated to definition V.1 is $\#P^{\Sigma_2^P}$ -complete.

Proof. The upper bound is easily obtained by considering the machine which guesses an ordering π of the elements of the partition V_1, \dots, V_n , computes the set J of indices appearing after s in π , and then calls a Σ_2^P oracle twice to determine the winner when Sat owns $\bigcup_{j \in J} V_j$ and when Sat owns $\bigcup_{j \in J \cup \{i\}} V_j$. The number of accepting runs of the machine is then precisely the number of permutations π matching the above condition.

Now for the lower bound, we proceed in two steps. First we show that the following problem is $\#P^{\Sigma_2^P}$ -complete.

Count $\exists\forall\exists$ SAT

{	Input:	A SAT formula ψ over variables $\{x_1, \dots, x_n, y_1, \dots, y_m, z_1, \dots, z_r\}$.
	Output:	The number of valuations of the x_i such that for all valuations of the y_i there exists a valuation of the z_i such that the combination of those valuations satisfies ψ .

Then we show that the importance computation problem reduces to Count $\exists\forall\exists$ SAT.

For the first part, let M be a nondeterministic Turing machine with an oracle solving a Σ_2^P -complete problem (say $\exists\forall\text{SAT}$). We can assume without loss of generality that M only makes one query to the oracle, and accepts if and only if the answer is negative.

Indeed, say M has to make queries ψ_1, \dots, ψ_k to the oracle, all over existential variables y_1, \dots, y_m and universal variables z_1, \dots, z_r . It can nondeterministically guess the answers of the oracle and delay the verification to the end of the run.

Now let us define the order \leq on valuations of the existential variables as the lexicographic order, a valuation ν being seen as the tuple $(\nu(y_1), \dots, \nu(y_m))$ and with the convention $\perp \leq \top$. Then for the positive answers M can guess a *minimal* witness valuation for the y_i with respect to \leq , negate the formula. The problem, given a $\exists\forall\text{SAT}$ instance and a valuation of the existential variables, of checking whether this is the minimal valuation witnessing the satisfiability of the formula, is clearly in CONP , thus also in Π_2^P . As a result M can guess the minimal valuation of the y_i witnessing the satisfiability of the formula, and then turn it into a $\exists\forall\text{SAT}$ formula unsatisfiable if and only if the guess is correct.

Finally, in the end M has to make the oracle check a disjunction of $\exists\forall$ formulas, it can rename variables in order to merge them all into one equivalent $\exists\forall$ formula, and accept if and only if the oracle rejects that formula.

In all the above transformations, the number of accepting runs of the machine stays the same as the non-deterministic transitions we added (in order to guess minimal valuations witnessing satisfiability of $\exists\forall\text{SAT}$ formulas) yield at most one accepting run (as the existence of such a valuation is equivalent to the existence of a single minimal one).

An adaptation of the classical construction proving that $\exists\forall\text{SAT}$ is Σ_2^P -complete allows us to construct in polynomial time, given an input w for M , a formula $\varphi_1((x_i), (q_i), (s_i))$ such that the following conditions are equivalent for all valuations ν of the x_i, q_i and s_i :

- ν satisfies $\varphi_1((x_i), (q_i), (s_i))$
- the $\nu(x_i)$ encode a sequence of non-deterministic choices of M , the $\nu(s_i)$ encode a correct run of M following those choices, and the $\nu(q_i)$ encode the query made to the oracle at the end of this run

We can also construct in polynomial time a formula $\varphi_2((y_i)_{1 \leq i \leq m}, (z_i)_{1 \leq i \leq r}, (q_i)_{1 \leq i \leq p}, (u_i)_{1 \leq i \leq k})$ simulating the oracle such that a valuation of the q_i satisfies $\exists(y_i), \forall(z_i), \varphi_2((y_i), (z_i), (q_i), (u_i))$ if and only if the q_i encode a valid instance of $\exists\forall\text{SAT}$. As a result the formula

$$\forall(y_i), \exists(z_i), (x_i), (q_i), \\ \varphi_1((x_i), (q_i), (s_i)) \wedge \varphi_2((y_i), (z_i), (q_i), (u_i))$$

is satisfied by a valuation of the x_i if and only if M has a run accepting w following the choices encoded by this valuation. Thus the number of accepting runs of M is precisely the number of valuations of the s_i witnessing the validity of

$$\exists(x_i), \forall(y_i), \exists(z_i), (s_i), (q_i), \\ \varphi_1((x_i), (q_i), (s_i)) \wedge \varphi_2((y_i), (z_i), (q_i), (u_i))$$

The problem $\text{Count}\exists\forall\text{SAT}$ is therefore $\#\text{P}^{\Sigma_2^P}$ -hard.

Finally, $\text{Count}\exists\forall\text{SAT}$ can be reduced to the importance computation problem for 2-turn CTL using the same construction as in the proof of Proposition V.3. Note that the $x_i, \neg x_i$ for all $n+1 \leq i \leq n+k+p$, as well as $\text{win}_S, \text{win}_U$, all have no outgoing may transitions, thus have importance 0 and thus, by a similar argument as in Lemma III.5, can be ignored in the computation of the importance. We will now only consider sets of states containing none of those. Then one can observe that the teams T allowing player Sat to win with $T \cup \{s\}$ but not with T are exactly the teams T such that

- T contains all the c_i for $\leq n+k$ and no other c_i .
- $T \cap \{x_i, \neg x_i \mid 1 \leq i \leq n\}$ and $T \cap \{x'_i, \neg x'_i \mid 1 \leq i \leq n\}$ match a same valuation ν witnessing the validity of the $\exists\forall\text{SAT}$ formula.

Then we have that all the teams T such that (s, T) is critical (and containing none of the aforementioned states with importance 0) have the same size M .

We obtain that the number of valuations witnessing the validity of the $\exists\forall\text{SAT}$ formula is $\frac{P!}{N!M!(P-M-1)!} N! I(s)$, with N the number of states in the constructed MTS and P the number of states minus the $x_i, \neg x_i$ for $n+1 \leq i \leq n+k+p$, s , win_U and win_S . Hence we have a reduction from $\text{Counting}\exists\forall\text{SAT}$ to the importance problem for 2-turn CTL. As $\frac{P!}{N!M!(P-M-1)!}$ can be computed in polynomial time, the latter problem is $\#\text{P}^{\Sigma_2^P}$ -complete. \square