On the Verification of Weighted Kripke Structures Under Uncertainty

Bacci, Giovanni; Hansen, Mikkel; Larsen, Kim Guldstrand

Published in:
Quantitative Evaluation of Systems - 15th International Conference, QEST 2018, Proceedings

DOI (link to publication from Publisher):
10.1007/978-3-319-99154-2_5

Publication date:
2018

Document Version
Accepted author manuscript, peer reviewed version

Link to publication from Aalborg University

Citation for published version (APA):
Bacci, G., Hansen, M., & Larsen, K. G. (2018). On the Verification of Weighted Kripke Structures Under Uncertainty. In A. Horvath, & A. McIver (Eds.), *Quantitative Evaluation of Systems - 15th International Conference, QEST 2018, Proceedings: 15th International Conference, QEST 2018, Beijing, China, September 4-7, 2018, Proceedings* (pp. 71-86). Springer. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) Vol. 11024 LNCS https://doi.org/10.1007/978-3-319-99154-2_5

On the Verification of Weighted Kripke Structures Under Uncertainty[⋆]

Giovanni Bacci, Mikkel Hansen, and Kim G. Larsen

Department of Computer Science, Aalborg University, Denmark
{giovbacci,mhan,kgl}@cs.aau.dk

Abstract. We study the problem of checking *weighted CTL* properties for weighted Kripke structures in presence of imprecise weights. We consider two extensions of the notion of weighted Kripke structures, namely (i) *parametric weighted Kripke structures*, having transitions weights modelled as affine maps over a set of parameters and, (ii) *weight-uncertain Kripke structures*, having transition labelled by real-valued random variables as opposed to precise real valued weights.
We address this problem by using *extended parametric dependency graphs*, a symbolic extension of dependency graphs by Liu and Smolka. Experiments show that our approach outperforms by orders of magnitude an adaptation of a state-of-the-art tool for WKSs.

1 Introduction

The rapid diffusion of cyber-physical systems (CPSs) poses the challenge of handling their growing complexity, while meeting requirements on correctness, predictability, performance without compromising time- and cost-to-market. In this respect model-driven development is a promising approach that allows for early design and verification and may be used as the basis for systematic testing of a final product. The verification of cyber-physical systems should not only address functional properties but also a number of non-functional properties related to the quantitative aspects that are typical of such systems.

In the area of model checking a number modelling formalisms has emerged, allowing for quantitative aspects to be expressed. Among these, Weighted Kripke structures (WKSs) were proposed as a natural extension of the usual notion of Kripke structures with a (real-valued) weighed transition relation [7].

Interesting properties of WKSs may be expressed by means of quantitative extensions of CTL. There are different ways of extending CTL with quantitative information. Fahrenberg et al. [7] proposed to generalise the classical Boolean interpretation of CTL to a map that assigns to states and temporal formulas a real-valued distance describing the degree of satisfaction. This paper considers weighted CTL (WCTL), an extension of CTL with weight-constrained modalities, because it is an expressive logic with efficient tool support for WKSs [8].

[⋆] Work supported by. . .

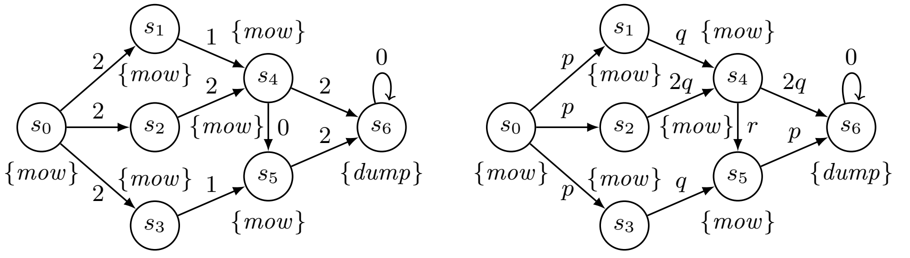


Fig. 1: (Left) A lawn mower example from [8]; (Right) the lawn mower model with weights parametric in p, q and, r.

Consider the WKS in Fig. 1(left) representing a grass field with different routes a lawn mower can take from the starting state s_0 to s_6 where the grass can be dumped. The weights on the transitions represent the amount of grass that is accumulated in the container when selecting a particular route. Assume that the lawn mower breaks when is forced to store more than 6.5 units of grass, then the property "the grass is always dumped before the lawn mower breaks, irrelevant of the selected route" is expressed in WCTL as $\forall(\mathit{mow}\,\mathcal{U}_{\leq 6.5}\,\mathit{dump})$.

The above example models the accumulated grass by means of precise weight values. This is an unrealistic simplification, since the amount of mowed grass may vary depending on different factors (e.g., distribution of the grass in the field, meteorologic conditions, etc.) that cannot be modelled with precise values. The same argument applies to CPSs, that typically rely on sensor measurements which are inherently imprecise.

Typically, there are two ways for dealing with uncertain sensor measurements: (i) determine the precision of the instrument and associate an error ϵ with each measurement, or (ii) perform estimation statistics (e.g., by recursive Bayesian estimation [13]) and associate a probability distribution with each measurement.

In this paper we aim at providing adequate formal basis and tool-support for the verification of WKSs in presence of imprecise weights. We consider two extensions of the notion of WKS: (i) *parametric weighted Kripke structures* (pWKSs), having weights depending on a set of parameters (*cf.* Fig. 1(right)) and, (ii) *weight-uncertain Kripke structures* (WUKSs), having as weights real-valued random variables as opposed to precise real values. On the one hand, verification of pWKSs is done by inferring constraints over its parameters characterising the valuations that ensure correctness then, verify the robustness of the model within the given precision. On the other hand, verifying WUKSs consists in measuring the degree of satisfaction of the model w.r.t. the given specification.

Our contribution is twofold. First, we extend and improve the model checking algorithm of [4] for pWKSs. In contrast with [4] our method supports negation and implements an efficient termination condition. In line with [4,8,5], our algorithm uses an extension of dependency graphs by Liu and Smolka [10] to model-check pWKSs. Specifically, we integrate cover-edges from [8] and negation-edges from [5] and, lift the computation of fixed points from the boolean domain to that of non-negative real-valued maps to cope with parametric weights and the non-monotonic reasoning necessary to deal with negation.

As for our second contribution, we introduce the notion of weight-uncertain Kripke structures and address two natural problems related to their analysis: (i) checking whether the expected behaviour of the model satisfies a given specification and, (ii) measuring the probability that a concrete realisation of the model satisfies a given WCTL formula.

Related Work. Our paper fits within the area of weighted automata [6] where weights come as elements of a semi-ring. By combining the tropical and the probability semi-rings, one obtains probabilistic weighted automata (PWA) [3,1]. There, transitions are labelled with a cost and a probability and the weight that the PWA assigns to a word is the expected accumulated costs of the runs producing the word. A similar approach is seen with Markov reward models whose analysis consider the computation of the expected reward for reachability properties or their verification against probabilistic reward CTL [2]. In contrast to PWAs and Markov reward models, were transitions are executed probabilistically and the weights are fixed, WUKSs choose transitions non-deterministically and generate weights according to the given probability distributions.

Fahrenberg et al. [7] consider the verification of WKSs with respect to two interpretations of WCTL where the satisfaction of a formula by a model is no longer interpreted in the Boolean domain, but rather assigns to a state a truth value in the domain of extended non-negative reals where a smaller value means a better match of the specified weights in the formula. Differently from [7], we keep the classical boolean interpretation of WCTL measure the expected correctness of the model. In this respect, our approach resembles that of probabilistic LTL model checking for Markov chain [14,15,2].

2 Weighted Kripke Structures and Weighted CTL

In this section we present *weighted Kripke structures* (WKSs) as an expressive modelling formalism for quantitative systems, and *weighted CTL* (WCTL), an extension of computation tree logic (CTL) with weight-constrained modalities, interpreted with respect to WKSs.

We denote by $\mathbb{R}$, $\mathbb{Q}$, and $\mathbb{N}$ respectively the sets of real numbers, rational numbers, and natural numbers. We write $\mathbb{R}_{\geq 0}$ (resp. $\mathbb{Q}_{\geq 0}$) to denote the set of non negative real (resp. rational) numbers.

Definition 1 (WKS). *A* weighted Kripke structure *is a tuple $\mathcal{K} = (S, R, \ell)$, where S is a finite nonempty set of states, $R \subseteq S \times \mathbb{R}_{\geq 0} \times S$ is a finite weighted transition relation and $\ell \colon S \to 2^{\mathcal{AP}}$ is a function labelling the states with atomic propositions.*

Let $\mathcal{K} = (S, R, \ell)$ be a WKS. We write $s \xrightarrow{w} s'$ to indicate that $(s, w, s') \in R$ and, we denote with $\omega(\mathcal{K}) \in \mathbb{R}^m_{\geq 0}$ the vector of weighs of $\mathcal{K}$, where $m = |R|$. A run in $\mathcal{K}$ from $s_0 \in S$ is a (finite or infinite) sequence $\pi = (w_i, q_i)_{i \in I}$, such that

$q_0 = s_0$, $w_0 = 0$ and I is an interval of $\mathbb{N}$ containing 0 where, for all $i \in I \setminus \{0\}$, $q_{i-1} \xrightarrow{w_i} q_i$. The *accumulated weight* of a run $\pi = (w_i, q_i)_{i \in I}$ at position $j \in I$ is defined as $\mathcal{W}(\pi, j) = \sum_{i=0}^{j} w_i$.

We write $|\pi|$ for the length of π (the cardinal of I); and, for $i \in I$, we write $\pi[i]$ for the i-th state in π, i.e., $\pi[i] = q_i$. A run is *maximal* if it has infinite length ($|\pi| = \omega$) or its last state has no outgoing transitions. $\mathit{Run}(\mathcal{K}, s_0)$ denotes the set of all maximal runs from s_0 in $\mathcal{K}$.

We can now define WCTL with weights upper-bounds. WCTL allows for *state formulas* describing properties about states in the system and *path formulas* describing properties about runs in a WKS. State formulas Φ, Ψ and path formluae φ are constructed over the following abstract syntax

$$\Phi, \Psi ::= \mathit{tt} \mid p \mid \neg\Phi \mid \Phi \wedge \Psi \mid \exists\varphi \mid \forall\varphi \,. \qquad\qquad \varphi ::= \mathcal{X}_{\leq q}\Phi \mid \Phi \, \mathcal{U}_{\leq q} \, \Psi$$

where $a \in \mathcal{AP}$ and $q \in \mathbb{Q}_{\geq 0}$.

Given a WKS $\mathcal{K} = (S, R, \ell)$, a state $s \in S$, and a run $\pi \in \mathit{Run}(\mathcal{K}, s)$, we denote by $\mathcal{K}, s \models \Phi$ (resp. $\mathcal{K}, \pi \models \varphi$) the fact that the state s satisfies the state formula Φ (resp. the path π satisfies the path formula φ). Formally, the *satisfiability relation* $\models$ is inductively defined as:

$$\begin{array}{lll}
\mathcal{K}, s \models \mathit{tt} & & \text{always holds} \\
\mathcal{K}, s \models p & \text{if} & p \in \ell(s) \\
\mathcal{K}, s \models \neg\Phi & \text{if} & \mathcal{K}, s \not\models \Phi \\
\mathcal{K}, s \models \Phi \wedge \Psi & \text{if} & \mathcal{K}, s \models \Phi \text{ and } \mathcal{K}, s \models \Psi \\
\mathcal{K}, s \models \exists\varphi & \text{if} & \text{there exists } \pi \in \mathit{Run}(\mathcal{K}, s) \text{ such that } \mathcal{K}, \pi \models \varphi \\
\mathcal{K}, s \models \forall\varphi & \text{if} & \text{for all } \pi \in \mathit{Run}(\mathcal{K}, s) \text{ it holds that } \mathcal{K}, \pi \models \varphi \\
\mathcal{K}, \pi \models \mathcal{X}_{\leq q}\Phi & \text{if} & |\pi| > 0,\ \mathcal{W}(\pi, 1) \leq q, \text{ and } \mathcal{K}, \pi[1] \models \Phi \\
\mathcal{K}, \pi \models \Phi \, \mathcal{U}_{\leq q} \, \Psi & \text{if} & \text{there exists } j \leq |\pi| \text{ such that } \mathcal{K}, \pi[j] \models \Psi, \\
 & & \mathcal{W}(\pi, j) \leq q, \text{ and } \mathcal{K}, \pi[j'] \models \Phi \text{ for all } j' < j
\end{array}$$

As usual, we derive derive logical operators ff, $\vee$ and $\rightarrow$ as follows: $\mathit{ff} \overset{def}{=} \neg\mathit{tt}$, $\Phi \vee \Psi \overset{def}{=} \neg(\neg\Phi \wedge \neg\Psi)$ and, $\Phi \rightarrow \Psi \overset{def}{=} \neg\Phi \vee \Psi$.

Example 2. Consider the WKS $\mathcal{K}$ in Fig. 1(left) described before. The WCTL state formulas $\Phi = \forall(\mathit{mow} \, \mathcal{U}_{\leq 6} \, \mathit{dump})$ and $\Phi' = \exists(\mathit{mow} \, \mathcal{U}_{\leq 4} \, \mathit{dump})$ express respectively the properties "the grass is always dumped before the lawn accumulates more that 6 grass units, irrelevant of the selected route" and "there exists a mowing route that accumulates at most 4 grass units before dumping". Clearly $\mathcal{K}, s_0 \models \Phi$ holds true because all paths from s_0 to s_6 accumulate at most 6 grass units, whereas $\mathcal{K}, s_0 \models \Phi'$ doesn't hold true, because each path from s_0 to s_6 accumulates at least 5 grass units. □

3 Parametric weighted Kripke structures

In this section we introduce the notion of *parametric weighted Kripke structures* and demonstrate how they can be employed for verifying the robustness of WKSs in presence of imprecise weights.

Parametric weighted Kripke structures (pWKSs) model families of WKSs that rely on the same graph structure, but differ in the concrete transition weights, which are specified as expressions built over a set of parameters.

Let $\mathbf{x} = (x_1, \dots, x_k)$ be a vector of real-valued parameters. We denote by $\mathcal{E}$ the set of affine maps $f \colon \mathbb{R}^k \to \mathbb{R}$ of the form $f(\mathbf{x}) = \mathbf{a} \cdot \mathbf{x} + b$, with $\mathbf{a} = (a_1, \dots, a_k) \in \mathbb{N}^k$ and $b \in \mathbb{N}$, i.e., $f(x_1, \dots, x_k) = (\sum_{i=1}^{k} a_i x_i) + b$. Hereafter we may denote the map f by means of the augmented vector[1] $(\mathbf{a}, b) \in \mathbb{N}^{k+1}$. Accordingly, for $f, g \in \mathcal{E}$ the map addition $(f + g)(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x})$ is encoded as the vector addition.

Definition 3. *A* parametric weighted Kripke structure *is a tuple $\mathcal{P} = (S, R, \ell)$, where S is a finite nonempty set of states, $R \subseteq S \times \mathcal{E} \times S$ is a finite parametric weighted transition relation and $\ell \colon S \to 2^{\mathcal{AP}}$ is a labelling function.*

Intuitively, a pWKS $\mathcal{P} = (S, R, \ell)$ defines a family of WKSs arising by plugging in concrete values for the parameters. A parameter valuation $\mathbf{v} \in \mathbb{R}^k$ is said to be *admissible* for $\mathcal{P}$ if for each transition $(s, f, s') \in R$ we have $f(\mathbf{v}) \geq 0$. Let $\mathcal{V}_{\mathcal{P}}$, or just $\mathcal{V}$ when $\mathcal{P}$ is clear from the context, denote the set of admissible valuations for $\mathcal{P}$. Given $\mathbf{v} \in \mathcal{V}$, we denote $\mathcal{P}(\mathbf{v})$ the WKS associated with $\mathbf{v}$. In this respect, it will be convenient to think at $\mathcal{P}$ as a partial function $\mathcal{P} \colon \mathbb{R}^k \rightharpoonup \mathrm{WKS}$ with domain $\mathcal{V}_{\mathcal{P}}$. The semantics of $\mathcal{K}$, written $[\mathcal{P}]$, is defined as the image of $\mathcal{P}$, i.e., $[\mathcal{P}] = \{\mathcal{P}(\mathbf{v}) \mid \mathbf{v} \in \mathcal{V}\}$.

A task typically addressed in the analysis of parametric Kripke structures is that of finding symbolic representations of the set of parameter valuations for which a given WCTL formula holds [4].

Formally, given a pWKS $\mathcal{P} = (S, R, \ell)$, a state $s \in S$ and a state formula Φ, the set of admissible valuations for which Φ holds at s is

$$[\![\mathcal{P}, s \models \Phi]\!] \overset{def}{=} \{\mathbf{v} \in \mathcal{V} \mid \mathcal{P}(\mathbf{v}), s \models \Phi\} \,. \tag{1}$$

Example 4. Consider the pWKS $\mathcal{P}$ depicted in Fig. 1(right) representing a family of lawn mower models parametric in p, q and, r. Its parameters represent the amount of grass measured in different parts of the field. The admissible valuations for $\mathcal{P}$, i.e., $\mathcal{V}_{\mathcal{P}}$, are represented by the constraint

$$\alpha(p, q, r) = p \geq 0 \wedge q \geq 0 \wedge r \geq 0 \,. \tag{2}$$

Let $\Phi = \forall(mow\,\mathcal{U}_{\leq 6.5}\, dump)$ be our specification. The set of valuations satisfying Φ, i.e., $[\![\mathcal{P}, s \models \Phi]\!]$, is represented by the following constraint

$$\beta(p, q, r) = \alpha(p, q, r) \wedge p + 4q \leq 6.5 \wedge 2p + 2q + r \leq 6.5 \,. \tag{3}$$

[1] Our is a special case of the so called *affine transformation matrix* (or *projective transformation matrix*) representation for generic affine tranformations.

Assume that we have measured $p \cong 2 \pm \epsilon$, $q \cong 1 \pm \epsilon$ and, $r \cong 0 \pm \epsilon$ where $\epsilon > 0$ is the measurement error. One can determine if $\mathcal{P}$ is robust w.r.t. Φ by checking that all possible measurement values lay in $[\![\mathcal{P}, s \models \Phi]\!]$, formally

$$\mathcal{V}_{\mathcal{P}} \cap \{(p,q,r) \mid |p-2| \leq \epsilon,\ |q-1| \leq \epsilon,\ |r| \leq \epsilon\} \subseteq [\![\mathcal{P}, s \models \Phi]\!] \,.$$

The above can be expressed a first-order formula in theory of linear real arithmetic

$$\forall p \in [0, 2+\epsilon].\, \forall q \in [0, 1+\epsilon].\, \forall r \in [0, \epsilon].\, \beta(p,q,r)\,. \tag{4}$$

By performing quantifier elimination (e.g., using MJOLLNIR [11]) we can reduce (4) to $\epsilon \geq 0.1$, indicating that the least error ensuring robustness for $\mathcal{P}$ is 0.1. □

In Example 4 we showed how to exploit pWKSs to verify a simple WKSs against a given specification up-to some error.

Clearly, with an increasing complexity of the model (or the formula) it becomes necessary to have an automatic procedure to resolve (1). The following two sections are devoted to present a generalization of the model checking algorithm presented in [4] that can also accept WCTL formulas with negation.

4 Extended Parametric Dependency Graphs

Dependency graphs as originally introduced by Liu and Smolka [10] can be applied to model-checking of the alternation-free modal μ-calculus, including its sub-logics like CTL. Jensen et al. [8] proposed to extend the dependency graphs framework using *cover-edges* and weighted *hyper-edges* for the verification of WKSs against the negation-free WCTL formulas. Later, Christoffersen et al. [4] further generalised their approach to pWKSs by using parametric *hyper-edges* and *cover-edges*.

In this section we present an extension of the parametric dependency graph framework by incorporating a new type of edges, called *negation-edges*. Negation-edges were originally used in [5] for dealing with negation.

Definition 5. *An* Extended Parametric Dependency Graph (EPDG) *is a tuple* $\mathcal{G} = (V, H, N, C)$ *where* V *is a nonempty set of configurations and*

- $H \subseteq V \times 2^{\mathcal{E} \times V}$ *is a set of* hyper-edges*,*
- $N \subseteq V \times V$ *is a set of* negation-edges*, and*
- $C \subseteq V \times \mathbb{Q}_{\geq 0} \times V$ *is a set of* cover-edges.

For $v, u \in V$, we write $v \xrightarrow{f} u$ if $(v, T) \in H$ and $(f, u) \in T$; $v \Rightarrow u$ if $(v, u) \in N$; $v \overset{q}{\dashrightarrow} u$ if $(v, q, u) \in C$ and v and u are said resp. the *source* and the *target* configurations of the edge. We write $v \rightsquigarrow u$ if v and u are respectively the source and target configurations of some edge in $\mathcal{G}$.

We identify a class of EPDGs having some convenient structural properties.

Definition 6. *Let* $\mathcal{G} = (V, H, N, C)$ *be an EPDG.* $\mathcal{G}$ *is* safe *if* *(i) its components are finite and for all* $(v, T) \in H$, T *is finite.*

(ii) for all $v \in V$ $|\{(v,u) \in N\} \cup \{(v,q,u) \in C\}| \leq 1$ *and if* $|\{(v,u) \in N\} \cup \{(v,q,u) \in C\}| = 1$ *then* $\{(v,T) \in H\} = \emptyset$.

(iii) there are no $u, v \in V$ *such that* $v \overset{q}{\dashrightarrow} u$ *and* $u \rightsquigarrow^* v$, *or* $v \Rightarrow u$ *and* $u \rightsquigarrow^* v$.

Intuitively, to be *safe* an EPDG $\mathcal{G}$ needs to have (i) finitely many configurations and edges, and each hyper-edge needs to be finitely branching; (ii) each of its configurations admits at most one type of outgoing edges and no cover edges or negation edges share the same source configuration; (iii) finally, no loop in $\mathcal{G}$ shall have any cover- or negation-edges.

In the rest of the section we fix $\mathcal{G} = (V, H, N, C)$ to be a safe EPDG.

We assign to each configuration $v \in V$ a distance $d(v) \in \mathbb{N}$ counting the maximum number of negation- and cover-edges in the paths starting from v

$$d(v) \overset{def}{=} \sup\{d(v'') + 1 \mid v' \Rightarrow v'' \text{ or } v' \overset{q}{\dashrightarrow} v'' \text{ for } v', v'' \in V \text{ s.t. } v \rightsquigarrow^* v'\}.$$

Notice that the distance is bounded because $\mathcal{G}$ is assumed to be safe.

We define $d(\mathcal{G}) = \max_{v \in V} d(v)$. The distance value is used to identify some *components* $\mathcal{C}_0, \ldots, \mathcal{C}_{d(\mathcal{G})}$, where $\mathcal{C}_i = (V_i, H_i, N_i, C_i)$ is the sub-EPDG of $\mathcal{G}$ induced by the configurations $V_i = \{v \in V \mid d(v) \leq i\}$. Note that by construction $N_0 = C_0 = \emptyset$.

A valuation $\mathbf{v} \in \mathbb{R}^k$ is said *admissible for* $\mathcal{G}$ if whenever $v \xrightarrow{f} u$ we have $f(\mathbf{v}) \geq 0$. We denote by $\mathcal{V}_{\mathcal{G}}$ the set of admissible valuations for $\mathcal{G}$.

Definition 7. *An* assignment A *of* $\mathcal{G}$ *is a function* $A : V \to (\mathcal{V}_{\mathcal{G}} \to \overline{\mathbb{R}}_{\geq 0})$ *where* $\overline{\mathbb{R}}_{\geq 0} = \mathbb{R}_{\geq 0} \cup \{\infty\}$. *The set of all assignments of* $\mathcal{G}$ *is denoted* $\mathcal{A}^{\mathcal{G}}$.

We equip $\mathcal{A}^{\mathcal{G}}$ with the partial order $\sqsubseteq \subseteq \mathcal{A}^{\mathcal{G}} \times \mathcal{A}^{\mathcal{G}}$ defined as

$$A_1 \sqsubseteq A_2 \qquad \text{iff} \qquad \forall v \in V.\, \forall \mathbf{v} \in \mathcal{V}_{\mathcal{G}}.\, A_1(v)(\mathbf{v}) \geq A_2(v)(\mathbf{v})\,.$$

$(\mathcal{A}^{\mathcal{G}}, \sqsubseteq)$ forms a complete lattice, with bottom element $A_\perp$ and top element $A_\top$ respectively defined as $A_\perp(v)(\mathbf{v}) = \infty$ and $A_\top(v)(\mathbf{v}) = 0$ for all $v \in V$ and $\mathbf{v} \in \mathcal{V}_{\mathcal{G}}$. Given $E \subseteq \mathcal{A}^{\mathcal{G}}$ the the greatest lower boud $\bigsqcap E$ and least upper bound $\bigsqcup E$ are defined, for arbitrary $v \in V$ and $\mathbf{v} \in \mathcal{V}_{\mathcal{G}}$, as

$$(\bigsqcap E)(v)(\mathbf{v}) = \sup_{A \in E} A(v)(\mathbf{v})\,, \qquad (\bigsqcup E)(v)(\mathbf{v}) = \inf_{A \in E} A(v)(\mathbf{v})\,.$$

We are now ready to define the least fixed-point assignment of an EPDG $\mathcal{G}$.

Definition 8. *The* least fixed-point assignment *for* $\mathcal{G}$, *denoted* $A^{\mathcal{G}}_{min}$, *is defined inductively on its components* $\mathcal{C}_0, \ldots, \mathcal{C}_{d(\mathcal{G})}$. *For* $0 \leq i \leq d(\mathcal{G})$, $A^{\mathcal{C}_i}_{min}$ *is the least fixed-point of the function* $F_i : \mathcal{A}^{\mathcal{C}_i} \to \mathcal{A}^{\mathcal{C}_i}$, *defined as*

$$F_i(A)(v)(\mathbf{v}) = \begin{cases} \chi(A^{\mathcal{C}_{i-1}}_{min}(u)(\mathbf{v}) > 0) & \text{if } v \Rightarrow u \\ \chi(A^{\mathcal{C}_{i-1}}_{min}(u)(\mathbf{v}) \leq q) & \text{if } v \overset{q}{\dashrightarrow} u \\ \min\limits_{(v,T) \in H_i} \max\limits_{(f,u) \in T} A(u)(\mathbf{v}) + f(\mathbf{v}) & \text{otherwise} \end{cases}$$

where $\chi(p) = 0$ *if the predicate* p *holds,* ∞ *otherwise.*

Lemma 9. *Let $i \in \{0, \dots, d(\mathcal{G})\}$ and $\{A_j\}_{j \in \mathbb{N}} \subseteq \mathcal{A}^{\mathcal{C}_i}$ be an ascending chain. Then, $F_i(\bigsqcup_{j \in \mathbb{N}} A_j) = \bigsqcup_{j \in \mathbb{N}} F_i(A_j)$, i.e., F_i is ω-continuous.*

Corollary 10. *F_i is monotonic for all $i \in \{0, \dots, d(\mathcal{G})\}$.*

By Knaster-Tarski's fixed-point theorem, $A^{\mathcal{C}_i}_{min}$ exists for all $i \leq d(\mathcal{G})$, moreover, by Kleene's fixed-point theorem, it is the limit of the ascending chain $A_\perp \sqsubseteq F_i(A_\perp) \sqsubseteq F_i(F_i(A_\perp)) \sqsubseteq \cdots \sqsubseteq F_i^n(A_\perp) \sqsubseteq \cdots$, i.e., $\bigsqcup_{n \in \mathbb{N}} F_i^n(A_\perp)$.

The following result states that the limit of the above chain is reached within $|V_i|$ steps. This result is essential for our algorithm.

Lemma 11. *Let $i \in \{0, \dots, d(\mathcal{G})\}$ and $k = |V_i|$. Then, $F_i^k(A^{\mathcal{C}_i}_\perp) = A^{\mathcal{C}_i}_{min}$.*

By Lemma 11, we can compute $A^{\mathcal{G}}_{min}$ symbolically by repeated application of F until it is sure that the fixed-point has been reached. It is worth noting that our termination condition only depends on the number of configurations of the EPDG. Therefore, in contrast with [4], we don't need to perform any symbolic comparison of the assignments to check whether a fixed-point has been reached. Not only does it simplifies the algorithm, but it also reduces the overhead caused by symbolic comparison.

Lemma 12. *For any safe EPDG $\mathcal{G} = (V, H, N, C)$ and component $\mathcal{C}_i$ of $\mathcal{G}$, the symbolic computation of the least fixed-point assignment, $A^{\mathcal{C}_i}_{min}$, by repeated application of the function F_i on $A^{\mathcal{C}_i}_\perp$ runs in time $\mathcal{O}(|V_i| \cdot (|H_i| + |N_i| + |C_i|))$.*

5 Model checking Parametric WKSs using EPDGs

In this section we present a reduction from the model checking problem of WCTL on pWKSs to the computation of least fixed-point assignments for EPDGs. Then, we show how to obtain from those assignments a symbolic representation of (1) as a (quantifier-free) first-order formula in the linear theory of the reals.

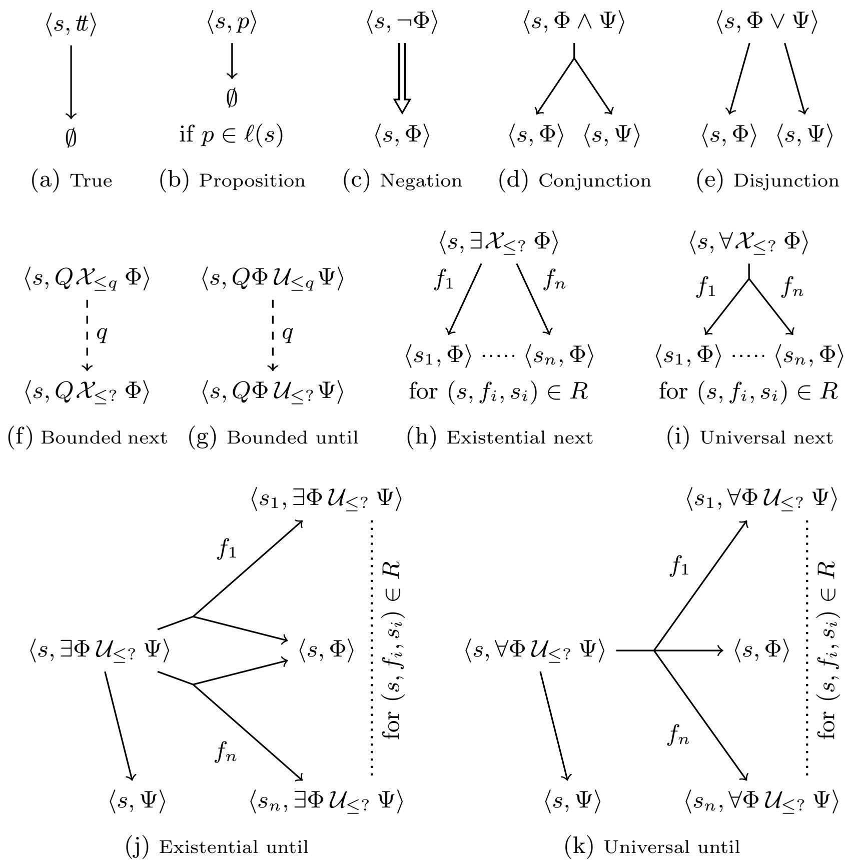
Given a pWKS $\mathcal{P} = (S, R, \ell)$, a state $s \in S$ and a WCTL formula Φ, we construct an EPDG $\mathcal{G}$ where every configuration is a pair consisting of a state and a formula. Starting from the initial pair $\langle s, \Phi \rangle$, $\mathcal{G}$ is constructed according to the rules given in Figure 2.

It is worth noting that the size of $\mathcal{G}$ does not depend on the actual weight values of Φ or $\mathcal{P}$ but only on the size of $\mathcal{P}$ and the number of sub-formulas of Φ.

The following result ensures that the EPDG framework described in Section 4 can be applied to the EPDGs constructed according to the rules in Figure 2.

Lemma 13. *The EPDG $\mathcal{G}$ rooted at $\langle s, \Phi \rangle$ is safe.*

In $\mathcal{G}$ we distinguish two types of configurations: *concrete* configurations have concrete WCTL formulas, while *symbolic* configurations have *symbolic formulas* of the form $Q\,\mathcal{X}_{\leq ?}\,\Phi$ or $Q \Phi\,\mathcal{U}_{\leq ?}\,\Psi$ where $Q \in \{\exists, \forall\}$ and Φ, Ψ are concrete WCTL formulas. Given a symbolic formula Φ and $q \in \mathbb{Q}_{\geq 0}$, we denote by Φ_q the corresponding concrete formula with bound q.



(a) True (b) Proposition (c) Negation (d) Conjunction (e) Disjunction

(f) Bounded next (g) Bounded until (h) Existential next (i) Universal next

(j) Existential until (k) Universal until

Fig. 2: EPDG construction rules. Here $Q \in \{\exists, \forall\}$ and hyper-edges without labels shall be assumed to be labelled with the constant weight map $\mathbf{0}$.

Lemma 14. *Let $v = \langle s, \Phi \rangle$ be a concrete configuration of $\mathcal{G}$ and $\mathbf{v} \in \mathcal{V}_{\mathcal{G}}$ an admissible valuation. Then, $A^{\mathcal{G}}_{min}(v)(\mathbf{v}) \in \{0, \infty\}$.*

The next theorem states that the set of correct valuations $[\![\mathcal{P}, s \models \Phi]\!]$ corresponds to the set $\{\mathbf{v} \in \mathcal{V}_{\mathcal{G}} \mid A^{\mathcal{G}}_{min}(\langle s, \Phi \rangle)(\mathbf{v}) \leq 0\}$. This reduces the model checking problem to the computation of least fixed-point assignments for EPDGs.

Theorem 15. *Let $v = \langle s, \Phi \rangle$ be a configuration of $\mathcal{G}$ and $\mathbf{v} \in \mathcal{V}_{\mathcal{G}}$ an admissible valuation. Then, the following hold*

1) *if v is concrete, then $A^{\mathcal{G}}_{min}(v)(\mathbf{v}) = 0$ iff $\mathcal{P}(\mathbf{v}), s \models \Phi$ and,*
2) *if v is symbolic, then for all $q \in \mathbb{Q}$, $A^{\mathcal{G}}_{min}(v)(\mathbf{v}) \leq q$ iff $\mathcal{P}(\mathbf{v}), s \models \Phi_q$.*

We showed that $A^{\mathcal{G}}_{min}(\langle s, \Phi \rangle)$ can be computed symbolically as a partially evaluated expression. During the computation one can perform some simplifica-

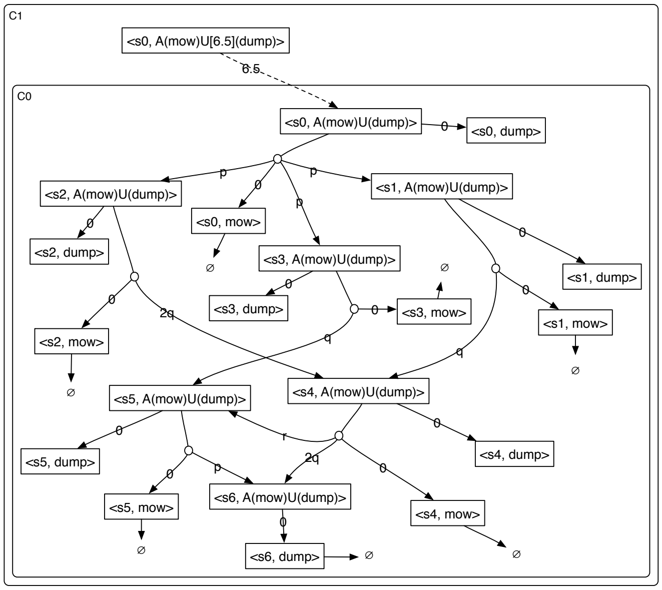


Fig. 3: EPDG rooted at $\langle s_0, \forall(mow\, \mathcal{U}_{\leq 6.5}\, dump)\rangle$ (*cf.* Example 16).

tions (e.g., $\min \emptyset = \infty$ or $\max \emptyset = 0$), nevertheless, the parts of the expression that depend on the actual value of the parameters are left unevaluated.

By Theorem 15 we are interested having a symbolic representation of the valuations $\mathbf{v}$ that satisfying $A^{\mathcal{G}}_{min}(\langle s, \Phi\rangle)(\mathbf{v}) \leq 0$. As anticipated in Example 4, this can be done by means of a (quantifier-free) first-order formula in the linear theory of the reals. In practice, such formula is obtained as $\Gamma(A^{\mathcal{G}}_{min}(\langle s, \Phi\rangle) \leq 0)$ where Γ is defined as follows[2], for $\bowtie \in \{\leq, >\}$, $\mathrm{m} \in \{\min, \max\}$ and, $q \in \mathbb{Q}_{\geq 0}$

$$\begin{aligned}
\Gamma(\max\{e_1, \dots, e_n\} \bowtie q) &= \Gamma(e_1 \bowtie q) \wedge \dots \wedge \Gamma(e_1 \bowtie q) \\
\Gamma(\min\{e_1, \dots, e_n\} \bowtie q) &= \Gamma(e_1 \bowtie q) \vee \dots \vee \Gamma(e_1 \bowtie q) \\
\Gamma(\chi(b) \leq q) = \Gamma(b) \qquad & \qquad \Gamma(\chi(b) > q) = \neg\Gamma(b) \\
\Gamma(e + \mathrm{m}\{e_1, \dots, e_n\} \bowtie q) &= \Gamma(\mathrm{m}\{e + e_1, \dots, e + e_n\} \bowtie q) \\
\Gamma(e \bowtie q) = e \bowtie q\,. \qquad & \qquad \text{(if } e \text{ has no occurrence of min, max or } \chi)
\end{aligned}$$

Example 16. Consider the pWKS $\mathcal{P}$ and the formula $\Phi = \forall(mow\, \mathcal{U}_{\leq 6.5}\, dump)$ from Example 4. In Fig. 3 is depicted the EPDG $\mathcal{G}$ rooted at $\langle s_0, \Phi\rangle$. By running our symbolic algorithm we obtain the following expression

$$A^{\mathcal{G}}_{min}(\langle s_0, \Phi\rangle) = \chi(\max\{p{+}q{+}\max\{p{+}r, 2q\}, p{+}2q{+}\max\{p{+}r, 2q\}, 2p{+}q\} \leq 6.5)\,.$$

[2] To simplify the exposition, here unevaluated expressions are assumed to be modulo commutativity and associativity of $+$.

The above expression can be then turned into the following formula

$$2p + q + r \leq 6.5 \land p + 3q \leq 6.5 \land 2p + 2q + r \leq 6.5 \land p + 4q \leq 6.5 \land 2p + q \leq 6.5\,,$$

that, in conjunction with $p \geq 0 \land q \geq 0 \land r \geq 0$ (*cf.* (2)) simplifies to (3). □

6 Weight-uncertain Kripke structures

In Section 3 we have seen how to use pWKSs for modelling and verifying the robustness of WKSs when the imprecision of the weights is quantified by means of an absolute accuracy error ϵ. However, for an experimental weight w, not all values in the interval $w \pm \epsilon$ are equally likely to occur in practice.

It's common practice to model experimental measurements by means of real-valued random variables distributed according to well studied family of distribution (e.g., normal or student's T). In this section introduce the notion of *weight-uncertain Kripke structures* (WUKSs), where weights are modelled as random variables and present a WCTL model checking framework for them.

Before to start we need to recall some notions from measure theory.

Measure Theory. Let Ω be a set. A family $\Sigma \subseteq 2^\Omega$ is called *σ-algebra* if it contains the empty set $\emptyset$ and is closed under complement and countable unions, in this case (Ω, Σ) is said *measurable space* and elements of Σ *measurable sets.* If Ω is given a topology then $\mathcal{B}(\Omega)$ denotes the Borel σ-algebra of Ω, i.e., the smallest σ-algebra having all open subsets of Ω. We say that Ω is a *Borel space* to indicate the measurable space $(\Omega, \mathcal{B}(\Omega))$, and elements of $\mathcal{B}(\Omega)$ are called *Borel sets.* As an example, $\mathbb{R}$ is assumed to have the usual Euclidean topology and $\mathcal{B}(\mathbb{R})$ denotes the induced Borel σ-algebra which makes $\mathbb{R}$ a Borel space.

A *measure* on (Ω, Σ) is a σ-additive function $\mu \colon \Sigma \to \mathbb{R}$, i.e, a map satisfying $\mu(\bigcup_{i \in I} E_i) = \sum_{i \in I} \mu(E_i)$ for any countable family of pairwise disjoint measurable sets $(E_i)_{i \in I}$, in this case (Ω, Σ, μ) is said *measure space.* If μ additionally satisfies $\mu(\Omega) = 1$, it is called *probability measure* and (Ω, Σ, μ) *probability space.*

For (Ω, Σ) and (Y, Θ) measurable spaces, the map $f \colon \Omega \to Y$ is *measurable* if for all $E \in \Theta$, $f^{-1}(E) = \{x \mid f(x) \in E\} \in \Sigma$. Given a measurable map $f \colon \Omega \to Y$ and a measure μ on (Ω, Σ) we define the measure $\mu[f]$ on (Y, Θ) as $\mu[f](E) = \mu(f^{-1}(E))$, for $E \in \Theta$, a.k.a. the *push forward of μ under f.*

A real-valued *random variable* $X \colon \Omega \to \mathbb{R}$ is a measurable function from a probability space (Ω, Σ, P) to the Borel space $\mathbb{R}$. Intuitively, X can be understood as the outcome value of an experiment (e.g., measuring some sensor value). Given a "test" $A \in \mathcal{B}(\mathbb{R})$, we write $P[X \in A]$ for the probability that X has value in A, i.e., $P[X \in A] = P[X](A)$. A random variable X is associated with its *cumulative distribution function* (CDF) $F_X \colon \mathbb{R} \to [0,1]$ defined as $F_X(x) = P[X \in (\infty, x]]$; and a *probability density function* (PDF) f_X, a non-negative Lebesgue-integrable function satisfying $P[X \in [a,b]] = \int_a^b f_X(x)\mathrm{d}x$. The expected value of X, written $\mathrm{E}[X]$ is intuitively understood as the long-run average value of repetitions of the experiment X, formally $\int_\Omega X \,\mathrm{d}P$ (corresponding to $\int_\mathbb{R} f_X(x)\mathrm{d}x$ when X admits density function f_X).

In the rest of the section we fix the probability space (Ω, Σ, P) representing the environment where the experiments are performed, and we use $\mathbb{Y}$ to denote the set of real-valued random variables of the form $Y : \Omega \to \mathbb{R}$.

We are now ready to define the concept of weight-uncertain Kripke structure.

Definition 17. *A* weight-uncertain Kripke structure *is a tuple $\mathcal{J} = (S, R, \ell)$, where S is a finite nonempty set of states, $R \subseteq S \times \mathbb{Y} \times S$ is a finite* random *weighted transition relation and $\ell : S \to 2^{\mathcal{AP}}$ is a labelling function.*

Consider the WUKS $\mathcal{J} = (S, R, \ell)$. We denote by $\mathrm{WKS}_{\mathcal{J}}$ the set of all WKSs having the same underlying graph than $\mathcal{J}$. We construct the σ-algebra $\Sigma_{\mathcal{J}}$ as the family of sets $A \subseteq \mathrm{WKS}_{\mathcal{J}}$ whose corresponding set of weights is measurable in $\mathbb{R}^m$ ($m = |R|$). Formally,

$$A \in \Sigma_{\mathcal{J}} \quad \text{iff} \quad A \subseteq \mathrm{WKS}_{\mathcal{J}} \text{ and } \{\omega(\mathcal{K}) \mid \mathcal{K} \in A\} \in \mathcal{B}(\mathbb{R}^m) \,.$$

$\mathcal{J}$ can be seen as a measurable function $\mathcal{J} : \Omega \to \mathrm{WKS}_{\mathcal{J}}$, where $\mathcal{J}(\omega)$ is the WKS associated with $\omega \in \Omega$, justifying the intuition the it represents as an experiment whose outcomes are WKSs. Accordingly, the semantics of $\mathcal{J}$ is the probability space $(\mathrm{WKS}_{\mathcal{J}}, \Sigma_{\mathcal{J}}, P[\mathcal{J}])$.

Given a WUKS $\mathcal{J}$, a state $s \in S$, and a WCLT property Φ, two natural model checking questions are (i) whether the expected behaviour of $\mathcal{J}$ satisfies Φ at s, informally "$E[\mathcal{J}], s \models \Phi$", (ii) and how likely is that a concrete instance of $\mathcal{J}$ satisfies Φ at s, that is, $P[\mathcal{J}](\{\mathcal{K} \in \mathrm{WKS}_{\mathcal{J}} \mid \mathcal{K}, s \models \Phi\})$.

We address the above problems for a subclass of WUKSs having random variables $(Y : \Omega \to \mathbb{R}) \in \mathcal{E}_{\mathbf{X}}$ of the form $Y(\omega) = \mathbf{a} \cdot \mathbf{X}(\omega) + b$, with $\mathbf{a} \in \mathbb{Q}^k_{\geq 0}$, $b \in \mathbb{Q}_{\geq 0}$ and, where $\mathbf{X} = (X_1, \dots, X_k)$ is vector of pairwise *independent* non-negative real-valued random variables[3]. Observe that, elements in $\mathcal{E}_{\mathbf{X}}$ may not be independent from each other.

From here on we consider the WUKS $\mathcal{J} = (S, \mathcal{E}, R, \ell)$ with $R \subseteq S \times \mathcal{E}_{\mathbf{X}} \times S$, and we use $\mathcal{P}$ to refer to the pWKS obtained by replacing the random variables X_i in $\mathcal{J}$ with the parameters x_i (for $i = 1..k$).

Let's consider the first question, namely "$E[\mathcal{J}], s \models \Phi$". There, $E[\mathcal{J}]$ was informally denoting the WKS obtained by replacing each transition weight in $\mathcal{J}$ with the corresponding expected value. Formally, $E[\mathcal{J}]$ is defined as the unique $\mathcal{K} \in \mathrm{WKS}_{\mathcal{J}}$ such that $\omega_i(\mathcal{K}) = \int_{\mathrm{WKS}_{\mathcal{J}}} \omega_i \, \mathrm{d}P[\mathcal{J}]$ for all $i \in \{1, \dots, m\}$ where $\omega_i : \mathrm{WKS}_{\mathcal{J}} \to \mathbb{R}_{\geq 0}$ is the function that returns the i-th weight from a given WKS.

The assumption made on the weights in $\mathcal{J}$ allows us to rephrase $E[\mathcal{J}], s \models \Phi$ as model checking problem for $\mathcal{P}$.

Lemma 18. *$E[\mathcal{J}], s \models \Phi$ if and only if $E[\mathbf{X}] \in [\![\mathcal{P}, s \models \Phi]\!]$.*

We are now ready to address the second question, that is computing

$$P[\mathcal{J}, s \models \Phi] \overset{def}{=} P[\mathcal{J}](\{\mathcal{K} \in \mathrm{WKS}_{\mathcal{J}} \mid \mathcal{K}, s \models \Phi\}) \,. \tag{5}$$

[3] In fact, the vector $\mathbf{X}$ is a multivariate random variable $\mathbf{X} : \Omega \to \mathbb{R}^n$ with marginals $X_i : \Omega \to \mathbb{R}_{\geq 0}$ ($i = 1..n$).

For the above formula to be well-defined the set $\{\mathcal{K} \in \mathrm{WKS}_{\mathcal{J}} \mid \mathcal{K}, s \models \Phi\}$ needs to be a measurable event in $\Sigma_{\mathcal{J}}$. The following result ensures that.

Lemma 19. $\{\mathcal{K} \in \mathit{WKS}_{\mathcal{J}} \mid \mathcal{K}, s \models \Phi\} \in \Sigma_{\mathcal{J}}$

The following theorem characterizes the model checking problem for the WUKS $\mathcal{J}$ in terms of the model checking problem of its associated pWKS $\mathcal{P}$.

Theorem 20. $P[\mathcal{J}, s \models \Phi] = P[\mathbf{X} \in [\![\mathcal{P}, s \models \Phi]\!]]$.

Remark 21. For the sake of clarity, so far we have assumed that $\mathbf{X}$ is non-negative real-valued random vector. However, provided that $P[\mathbf{X} \in \mathcal{V}_{\mathcal{P}}] > 0$, the non-negativity assumption can be dropped by replacing the probability distribution $P[\mathbf{X}]$ with the conditional probability $P[\mathbf{X} | \mathbf{X} \in \mathcal{V}_{\mathcal{P}}]$.

By Theorem 20 we can estimate the value p of (5) by applying Monte Carlo simulation techniques. For this, we sample n independent repetitions of $\mathbf{X}$, associating with each repetition a Bernoulli random variable B_i. A realisation b_i of B_i is 1 if the corresponding sampled value of $\mathbf{X}$ lays in $[\![\mathcal{P}, s \models \Phi]\!]$, and 0 otherwise. Finally, we estimate p by means of the observed relative success rate $\tilde{p} = (\sum_{i=1}^{n} b_i)/n$. The absolute error ε of the estimation can be bound with a certain degree of confidence $\delta \in (0, 1]$ by tuning the the number of required simulations based on the inequality $P(|\tilde{p} - p| \geq \varepsilon) \leq \delta$ where $\delta = e^{-2n\varepsilon^2}$ (*cf.* [12,9]). Therefore the required number n of samples is obtained as

$$n = \left\lceil -\frac{\ln(\delta)}{2\varepsilon^2} \right\rceil . \tag{6}$$

Example 22. Consider the WUKS $\mathcal{J}$ depicted in Fig. 1(right), where p, q and, r shall now be interpreted as real-valued random variables distributed as $p \sim \mathcal{N}(2, \epsilon)$, $q \sim \mathrm{unif}(1 - \epsilon, 1 + \epsilon)$, and $r \sim \mathcal{N}(0, \epsilon)$ for $\epsilon = 0.1$. We can estimate $P[\mathcal{J}, s_0 \models \Phi] = 0.959$ with an error $\varepsilon = 0.003$ and confidence of $99,9\%$ (i.e., $\delta = 0.001$) by generating $n = 383765$ samples.

7 Experimental Results

To evaluate the performance of the algorithms discussed in this paper, we developed a prototype tool suite for WCTL model checking of WKSs under uncertain weights. The tool suite consists of two parts: a back-end, called PVTool2[4] and a front-end, called UVTool[5]. UVTool supports the verification of pWKSs and WUKSs as described in Sections 5 and 6 making use of the PVTool2 which implements the EPDG construction and the symbolic fixed-point computation.

We have evaluated the PVTool2 and the UVTool separately.

[4] The PVTool2 is available at `https://github.com/AcId9381/PVTool`.

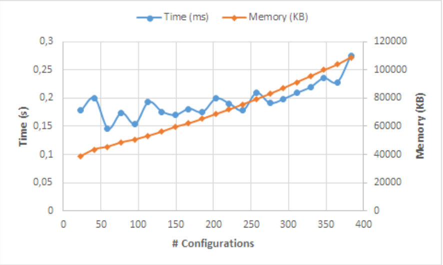
[5] The UVTool is implemented using Mathematica®and is available at `http://people.cs.aau.dk/~giovbacci/tools.html`.

Evaluation of the PVTool2. We compared the performance of the PVTool2 with the PVTool from [4]. For a fair comparison we used as benchmarks the vacuum cleaner models from [4] checking them against the WCTL formula $\exists(\forall \mathit{dirty}\, \mathcal{U}_{\leq 10}\, \mathit{clean})\mathcal{U}_{\leq 1000}\, \mathit{done}$. The table depicted in Fig. 4a reports the results obtained by increasing the number of rooms in the vacuum cleaning model. The first and the second columns respectively present the number of states of the model and the number of configurations of the resulting EPDG, while the last two columns present respectively the computation time and the memory consumption of the two tools. The results of the experiments show that the PVTool2 performs slightly worse than the PVTool on small models but it scales way better than the PVTool both in terms of computation time and memory consumption. We believe that this is due to the fact that our algorithm does not need to perform any comparison of the symbolic assignments during the fixed-point computation.

Fig 4b shows how the computation time and memory consumption of the PVTool2 grows linearly in the number of configurations of the EPGD.

Model	**EPGD**	**Time (s)**		**Memory (KB)**	
# states	# conf.	v1	v2	v1	v2
7	41	0.0015	0.200	1,004	43,540
13	77	0.017	0.174	1,504	48,548
19	113	0.190	0.193	3,808	53,104
25	149	0.250	0.170	14,264	59,520
31	185	35	0.175	60,548	65,256
34	203	781	0.199	263,832	68,560
40	239	N/A	0.178	N/A	75,536
46	275	N/A	0.191	N/A	82,972
52	311	N/A	0.209	N/A	91,020
58	347	N/A	0.235	N/A	99,872
64	383	N/A	0.275	N/A	108,976

(a) Comparison with the PVtool from [4]



(b) Performance of the PVTool2

Fig. 4: Experiments on an Intel i7 (5^{th} gen.) 2.6GHz processor with 12GB RAM

Evaluation of the UVTool. For the verification of WUKS, our algorithm first samples valuations from $\mathbf{X}$, then estimates the relative number of valuation-samples that are correct in the sense of (1). Alternatively, one could first sample WKSs from the WUKSand then estimate the relative number of models that satisfy the given specification. In the second approach one could employ the WKTool[6] and exploit the efficient local algorithm from [8].

We compared the two approaches on the WUKS of Example 22 and performed the evaluation with increasing precision and accuracy of the estimation. The results are presented in Table 1. The first three columns report the error, the confidence and the number of generated samples (*cf.* Eq. (6)), and the last two columns present the computation time respectively for the UVTool and the

[6] The WKTool is available at `https://github.com/jonasfj/WKTool`.

adaptation of the WKTool. It is worth mentioning that the values reported in the last column do not consider the time required to sample and generate the models, but only the total time used for the model checking. The results clearly show that our approach outperforms the second one by several orders of magnitude, showing that computing the symbolic representation of the correct valuations in advance gives a huge speed-up in the overall computation time.

ε	δ	n	UVTool (s)	WKTool (s)
0.02	0.01	5,757	0.137	181.009
0.01	0.01	23,026	0.533	724.206
0.01	0.001	34,539	0.828	1086.88
0.005	0.001	138,156	3.231	4,347.96
0.003	0.001	383,756	8.876	5,886.670

Table 1: Experiments on an Intel Core i5 3.1 GHz with 8GB RAM.

8 Conclusion and Future Work

We addressed the model checking problem of weighted Kripke structures under uncertainty. We proposed to employ parametric weighted Kripke structures and weight-uncertain Kripke structures for modelling WKSs with imprecise real-valued weights. For the verification of pWKSs against WCTL formulas we developed a model checking algorithm that, compared with [4], implements an improved termination condition and accepts formulas with negation. The algorithm, given a pWKS and a WCTL formula, and produces a quantifier free first-order formula in the linear theory of the reals representing the set of parameter valuations satisfying the specification. The outcome formula is then used as underlying ingredient for verifying the robustness of WKSs. If the imprecision of the weights by means of an absolute accuracy error the verification can be performed via quantifier elimination (*cf.* Example 4). Otherwise, if the imprecision is quantified by mean of random variables, the probability of satifiying the specification in estimated via Monte Carlo simulation techniques (*cf.* Example 22).

In the future we plan to consider an alternative semantic interpretation of WUKSwhere the random weights are dynamically sampled while unfolding the model, thus representing infinite state WKSs. This alternative representation would fit well in the contexts of reactive systems that respond to uncertain external stimuli. Another direction of future work would be to consider the model checking of weighted LTL property under uncertainty.

References

1. Guy Avni and Orna Kupferman. Stochastization of Weighted Automata. In Giuseppe F. Italiano, Giovanni Pighizzini, and Donald T. Sannella, editors, *Math-*

ematical Foundations of Computer Science 2015, pages 89–102, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

2. Christel Baier and Joost-Pieter Katoen. *Principles of model checking.* MIT Press, 2008.
3. Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Probabilistic Weighted Automata. In Mario Bravetti and Gianluigi Zavattaro, editors, *CONCUR 2009 - Concurrency Theory*, pages 244–258, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
4. Peter Christoffersen, Mikkel Hansen, Anders Mariegaard, Julian Trier Ringsmose, Kim Guldstrand Larsen, and Radu Mardare. Parametric verification of weighted systems. In *2nd International Workshop on Synthesis of Complex Parameters, SynCoP 2015, April 11, 2015, London, United Kingdom*, pages 77–90, 2015.
5. Andreas Engelbredt Dalsgaard, Søren Enevoldsen, Peter Fogh, Lasse S. Jensen, Tobias S. Jepsen, Isabella Kaufmann, Kim G. Larsen, Søren M. Nielsen, Mads Chr. Olesen, Samuel Pastva, and Jirí Srba. Extended dependency graphs and efficient distributed fixed-point computation. In *Application and Theory of Petri Nets and Concurrency - 38th International Conference, PETRI NETS 2017, Zaragoza, Spain, June 25-30, 2017, Proceedings*, pages 139–158, 2017.
6. Manfred Droste, Werner Kuich, and Heiko Vogler. *Handbook of Weighted Automata.* Springer Publishing Company, Incorporated, 1st edition, 2009.
7. Uli Fahrenberg, Kim G. Larsen, and Claus Thrane. A Quantitative Characterization of Weighted Kripke Structures in Temporal Logic. *Computing and Informatics*, 29:1311–1324, 2010.
8. Jonas Finnemann Jensen, Kim Guldstrand Larsen, Jirí Srba, and Lars Kaerlund Oestergaard. Efficient model-checking of weighted CTL with upper-bound constraints. *STTT*, 18(4):409–426, 2016.
9. N. Singh Kambo and Samuel Kotz. On exponential bounds for binomial probabilities. *Annals of the Institute of Statistical Mathematics*, 18(1):277, Dec 1966.
10. Xinxin Liu and Scott A. Smolka. Simple linear-time algorithms for minimal fixed points (extended abstract). In *Automata, Languages and Programming, 25th International Colloquium, ICALP'98, Aalborg, Denmark, July 13-17, 1998, Proceedings*, pages 53–66, 1998.
11. David Monniaux. Quantifier elimination by lazy model enumeration. In Tayssir Touili, Byron Cook, and Paul B. Jackson, editors, *Computer Aided Verification, 22nd International Conference, CAV 2010, Edinburgh, UK, July 15-19, 2010. Proceedings*, volume 6174 of *Lecture Notes in Computer Science*, pages 585–599. Springer, 2010.
12. Masashi Okamoto. Some inequalities relating to the partial sum of binomial probabilities. *Annals of the Institute of Statistical Mathematics*, 10(1):29–35, Mar 1959.
13. Simo Srkk. *Bayesian Filtering and Smoothing.* Cambridge University Press, New York, NY, USA, 2013.
14. Moshe Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*, pages 327–338. IEEE Computer Society, 1985.
15. Moshe Y. Vardi. Probabilistic linear-time model checking: An overview of the automata-theoretic approach. In *Formal Methods for Real-Time and Probabilistic Systems, 5th International AMAST Workshop, ARTS'99, Bamberg, Germany, May 26-28, 1999. Proceedings*, volume 1601 of *Lecture Notes in Computer Science*, pages 265–276. Springer, 1999.