On the Verification of Weighted Kripke Structures Under Uncertainty

Giovanni Bacci(✉), Mikkel Hansen, and Kim Guldstrand Larsen

Department of Computer Science, Aalborg University, Aalborg, 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 performed with a prototype tool implementation 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 of modelling formalisms have 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) weighted transition relation [8].

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. [8] 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 [9].

Work supported by the Advanced ERC Grant nr. 867096 (LASSO).

A. McIver and A. Horvath (Eds.): QEST 2018, LNCS 11024, pp. 71–86, 2018.
https://doi.org/10.1007/978-3-319-99154-2_5

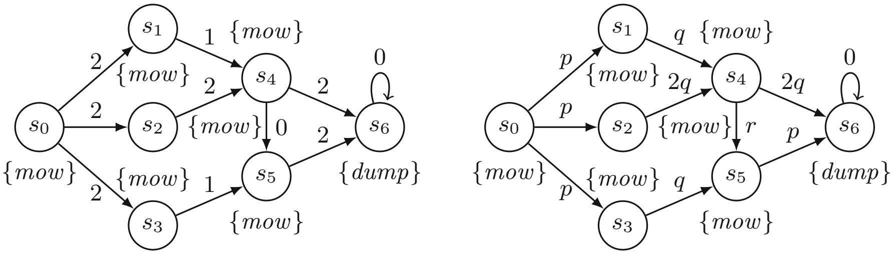


Fig. 1. (Left) A lawn mower example from [9]; (Right) the lawn mower model with weights parametric in p, q and, r.

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 it 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 [14]) 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 of 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 [5] for pWKSs. In contrast with [5], our method supports negation and implements an efficient termination condition. In line with [5,6,9], our algorithm uses an extension of dependency graphs by Liu and Smolka [11] to model-check pWKSs. Specifically, we integrate cover-edges from [9] and negation-edges from [6] 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.

The proposed model checking framework has been implemented on a prototype tool. Experiments show that our approach considerably improves w.r.t. the PVTool from [5] and outperforms an adaptation of the WKTool from [9].

We refer to the full version of this paper [2] for the omitted proofs.

Related Work. Our paper fits within the area of weighted automata [7] where weights come as elements of a semi-ring. By combining the tropical and the probability semi-rings, one obtains probabilistic weighted automata (PWA) [1,4]. 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 [3]. In contrast to PWAs and Markov reward models, where 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. [8] 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 [8], we keep the classical boolean interpretation of WCTL and measure how likely is the model to be correct. In this respect, our approach resembles that of probabilistic LTL model checking for Markov chains [3,15].

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 subsets of $\mathcal{AP}$, where $\mathcal{AP}$ is a fixed set of 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 $\pi = (w_i, s_i)_{i \in I}$, such that $w_0 = 0$ and I is an interval of $\mathbb{N}$ containing 0 where, for all $i \in I \setminus \{0\}$, $s_{i-1} \xrightarrow{w_i} s_i$. The *accumulated weight* of a run $\pi = (w_i, s_i)_{i \in I}$ 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., $\pi[i] = s_i$. 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 upper-bounds on weights. 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

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

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:

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

As usual, we can derive the logical operators ff, $\vee$ and $\rightarrow$ as follows: $ff \overset{def}{=} \neg tt$, $\Phi \vee \Psi \overset{def}{=} \neg(\neg\Phi \wedge \neg\Psi)$ 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 present *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 $\mathbf{a} = (a_1, \dots, a_k) \in \mathbb{Q}^k_{\geq 0}$ and $b \in \mathbb{Q}_{\geq 0}$, 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 $\mathcal{P}\colon \mathbb{R}^k \rightharpoonup$ 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 [5].

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

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

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 as 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 [12]) we reduce (4) to $\epsilon \leq 0.1$, indicating that robustness for $\mathcal{P}$ is ensured iff ϵ does not exceed 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 [5] that can also accept WCTL formulas with negation.

4 Extended Parametric Dependency Graphs

Dependency graphs as originally introduced by Liu and Smolka [11] can be applied to model-checking of the alternation-free modal μ-calculus, including its sub-logics like CTL. Jensen et al. [9] proposed to extend the dependency graphs framework using *cover-edges* and weighted *hyper-edges* for the verification of WKSs against negation-free WCTL formulas. Later, Christoffersen et al. [5] 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 [6] for extending the applicability of dependency graphs w.r.t. CTL model checking of Petri Nets.

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 $\mathcal{G}$ and, $\rightsquigarrow^*$ for the reflexive and transitive closure of $\rightsquigarrow$.

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

$$d(v) \overset{def}{=} \max\big\{0, \sup\big\{d(v'') + 1 \mid v' \Rightarrow v'' \text{ or } v' \overset{q}{\dashrightarrow} v'' \text{ for } v', v'' \in V \text{ s.t.} v \rightsquigarrow^* v' \ \big\}\big\}.$$

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 $A_\bot$ and top element $A_\top$ respectively defined as $A_\bot(v)(\mathbf{v}) = \infty$ and $A_\top(v)(\mathbf{v}) = 0$ for all $v \in V$ and $\mathbf{v} \in \mathcal{V}_{\mathcal{G}}$. Given $E \subseteq \mathcal{A}^{\mathcal{G}}$ such that $E \neq \emptyset$ the greatest lower bound $\bigsqcap E$ and least upper bound $\bigsqcup E$ are defined, for arbitrary $v \in V$ and $\mathbf{v} \in \mathcal{V}_{\mathcal{G}}$, as[2]

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

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*

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

[2] As usual, $\bigsqcap \emptyset = A_\top$ and $\bigsqcup \emptyset = A_\top$.

where $\chi(p) = 0$ *if the predicate* p *holds,* ∞ *otherwise. We assume that* $\max \emptyset = 0$ *and* $\min \emptyset = \infty$*.*

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 we are 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 [5], 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* $\mathcal{O}\big(|V_i| \cdot (|H_i| + |N_i| + |C_i|)\big)$*.*

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.

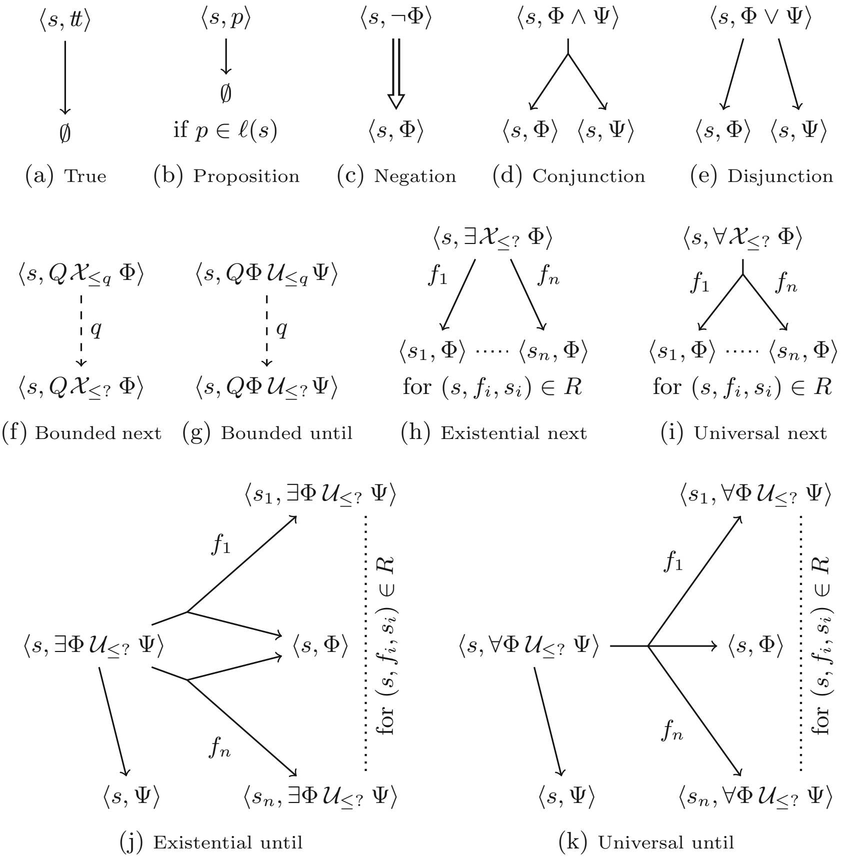
Given a pWKS $\mathcal{P} = (S, R, \ell)$, a state $s \in S$ and a WCTL formula Φ, we construct an EPDG $\mathcal{G}$ where every configuration is a pair consisting of a state and a formula. Starting from the initial pair $\langle s, \Phi \rangle$, $\mathcal{G}$ is constructed according to the rules given in Fig. 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 Sect. 4 can be applied to the EPDGs constructed according to the rules in Fig. 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.


$\langle s, tt \rangle$
$\emptyset$
$\langle s, p \rangle$
$\emptyset$
if $p \in \ell(s)$
$\langle s, \neg\Phi \rangle$
$\langle s, \Phi \rangle$
$\langle s, \Phi \wedge \Psi \rangle$
$\langle s, \Phi \rangle$ $\langle s, \Psi \rangle$
$\langle s, \Phi \vee \Psi \rangle$
$\langle s, \Phi \rangle$ $\langle s, \Psi \rangle$
$\langle s, Q\mathcal{X}_{\leq q}\, \Phi \rangle$
q
$\langle s, Q\mathcal{X}_{\leq ?}\, \Phi \rangle$
$\langle s, Q\Phi\, \mathcal{U}_{\leq q} \Psi \rangle$
q
$\langle s, Q\Phi\, \mathcal{U}_{\leq ?} \Psi \rangle$
$\langle s, \exists\mathcal{X}_{\leq ?}\, \Phi \rangle$
f_1
f_n
$\langle s_1, \Phi \rangle \cdots\cdots \langle s_n, \Phi \rangle$
for $(s, f_i, s_i) \in R$
$\langle s, \forall\mathcal{X}_{\leq ?}\, \Phi \rangle$
f_1
f_n
$\langle s_1, \Phi \rangle \cdots\cdots \langle s_n, \Phi \rangle$
for $(s, f_i, s_i) \in R$
$\langle s_1, \exists\Phi\, \mathcal{U}_{\leq ?}\, \Psi \rangle$
f_1
$\langle s, \exists\Phi\, \mathcal{U}_{\leq ?}\, \Psi \rangle$
$\langle s, \Phi \rangle$
f_n
$\langle s, \Psi \rangle$
$\langle s_n, \exists\Phi\, \mathcal{U}_{\leq ?}\, \Psi \rangle$
for $(s, f_i, s_i) \in R$
$\langle s_1, \forall\Phi\, \mathcal{U}_{\leq ?}\, \Psi \rangle$
f_1
$\langle s, \forall\Phi\, \mathcal{U}_{\leq ?}\, \Psi \rangle$
$\langle s, \Phi \rangle$
f_n
$\langle s, \Psi \rangle$
$\langle s_n, \forall\Phi\, \mathcal{U}_{\leq ?}\, \Psi \rangle$
for $(s, f_i, s_i) \in R$

(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 simplifications (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.

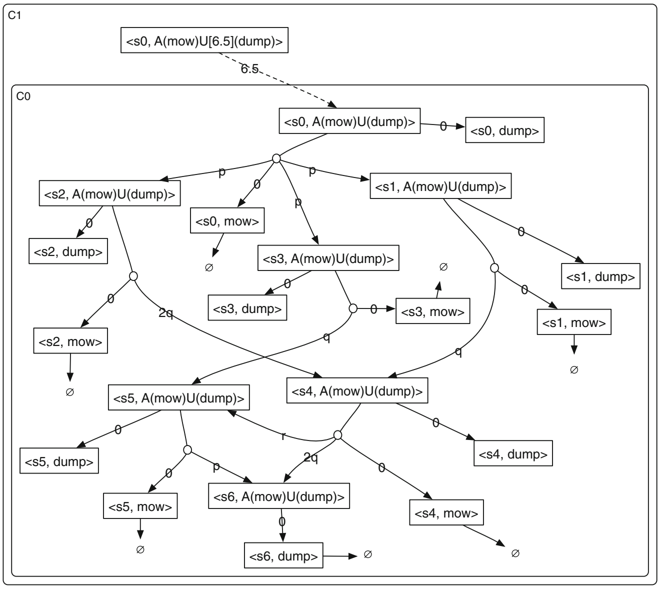


Fig. 3. EPDG rooted at $\langle s_0, \forall(mow\,\mathcal{U}_{\leq 6.5}\,dump)\rangle$ (*cf.* Example 16).

By Theorem 15 we are interested in a symbolic representation of the valuations $\mathbf{v}$ such that $A^{\mathcal{G}}_{min}(\langle s, \Phi\rangle)(\mathbf{v}) \leq 0$. 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 a formula is obtained as $\Gamma\big(A^{\mathcal{G}}_{min}(\langle s, \Phi\rangle) \leq 0\big)$ where Γ is defined by cases as follows[3], for $\bowtie \in \{\leq, >\}$, $\mathrm{m} \in \{\min, \max\}$ and, $q \in \mathbb{Q}_{\geq 0}$

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

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

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

The above expression can be then turned into the following formula

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

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

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

6 Weight-Uncertain Kripke Structures

In Sect. 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 value 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 we 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 we start let us 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, formalised by the Lebesgue integral $\int_{\Omega} X \, \mathrm{d}P$ (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 $Y \colon \Omega \to \mathbb{R}$.

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* $\ell \colon S \to 2^{\mathcal{AP}}$ *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 as $\mathcal{J}$. 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 Borel 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 $\mathcal{J} \colon \Omega \to \mathrm{WKS}_{\mathcal{J}}$, where $\mathcal{J}(\omega)$ is the WKS associated with $\omega \in \Omega$, justifying the intuition the it represents 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, denoted by $P[\mathcal{J}, s \models \Phi]$.

We address the above problems for a subclass of WUKSs having random variables $(Y \colon \Omega \to \mathbb{R}) \in \mathcal{E}_{\mathbf{X}}$ 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 $\mathbf{X} = (X_1, \ldots, X_k)$ is vector of pairwise *independent* non-negative real-valued random variables[4]. 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 $i \in \{1, \ldots, m\}$ where $\omega_i \colon \mathrm{WKS}_{\mathcal{J}} \to \mathbb{R}_{\geq 0}$ 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 a 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 formalised as follows

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

For the above definition 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. $\{\mathcal{K} \in \mathit{WKS}_{\mathcal{J}} \mid \mathcal{K}, s \models \Phi\} \in \Sigma_{\mathcal{J}}$

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

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 number of required simulations based on the inequality $P(|\tilde{p} - p| \geq \varepsilon) \leq \delta$ where $\delta = e^{-2n\varepsilon^2}$ (*cf.* [10,13]). 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[5] and a front-end, called UVTool[6]. UVTool supports the verification of pWKSs and WUKSs as described in Sects. 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.

Evaluation of the PVTool2. We compared the performance of the PVTool2 with the PVTool from [5]. For a fair comparison we used as benchmarks the vacuum cleaner models from [5] 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.

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

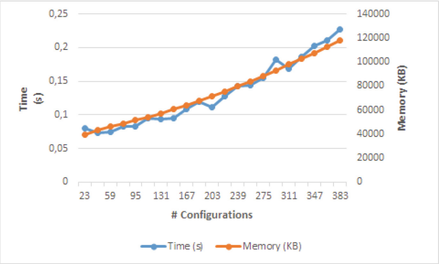
[6] The UVTool is implemented using Mathematica [16] and is available at http://people.cs.aau.dk/~giovbacci/tools.html.

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 worse than the PVTool on small models but it scales way better than the PVTool both in terms of computation time and memory consumption. This may be due to the fact that our algorithm does not perform any comparison of the symbolic assignments during the fixed-point computation. The improvement is measured when the overhead of the symbolic comparison exceeds the cost of the additional iterations required by the PVTool2.

Figure 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	0.073	1,004	43,616
13	77	0.017	0.083	1,504	48,686
19	113	0.190	0.095	3,808	54,274
25	149	0.250	0.096	14,264	60,780
31	185	35	0.119	60,548	67,806
34	203	781	0.112	263,832	71,667
40	239	N/A	0.142	N/A	79,530
46	275	N/A	0.155	N/A	88,252
52	311	N/A	0.168	N/A	97,889
58	347	N/A	0.202	N/A	107,574
64	383	N/A	0.227	N/A	118,044

(a) Comparison with the PVTool from [5]



(b) Performance of the PVTool2

Fig. 4. Experiments on an Intel i7 (5th gen.) 2.6 GHz processor with 12 GB 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 given WUKS and then estimate the relative number of models that satisfy the specification. In the second approach one could employ the WKTool[7] and exploit the efficient local algorithm from [9].

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 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.

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

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

Error ε	Confidence δ	# samples	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

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 [5], 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 satisfying the specification in estimated via Monte Carlo simulation techniques (*cf.* Example 22).

In the future we plan to consider an alternative semantic interpretation for WUKSs where the random weights are dynamically sampled while unfolding the model, thus modelling WKSs with an infinite state space. This alternative semantics would fit well in the contexts of reactive systems that respond to external stimuli whose values are uncertain. Another direction for future work would be to consider the model checking of weighted LTL properties under uncertainty.

References

1. Avni, G., Kupferman, O.: Stochastization of weighted automata. In: Italiano, G.F., Pighizzini, G., Sannella, D.T. (eds.) MFCS 2015. LNCS, vol. 9234, pp. 89–102. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48057-1_7
2. Bacci, G., Hansen, M., Larsen, K.G.: On the verification of weighted Kripke structures under uncertainty. Full version, Aalborg University (2018). http://people.cs.aau.dk/~giovbacci/papers/uncertwks-full.pdf
3. Baier, C., Katoen, J.-P.: Principles of model checking. MIT Press, Cambridge (2008)
4. Chatterjee, K., Doyen, L., Henzinger, T.A.: Probabilistic weighted automata. In: Bravetti, M., Zavattaro, G. (eds.) CONCUR 2009. LNCS, vol. 5710, pp. 244–258. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04081-8_17

5. Christoffersen, P., Hansen, M., Mariegaard, A., Ringsmose, J.T., Larsen, K.G., Mardare, R.: Parametric verification of weighted systems. In: 2nd International Workshop on Synthesis of Complex Parameters, SynCoP 2015, April 11, 2015, London, United Kingdom, pp. 77–90 (2015)
6. Dalsgaard, A.E., et al.: Extended dependency graphs and efficient distributed fixed-point computation. In: van der Aalst, W., Best, E. (eds.) PETRI NETS 2017. LNCS, vol. 10258, pp. 139–158. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-57861-3_10
7. Droste, M., Kuich, W., Vogler, H. (eds.): Handbook of Weighted Automata, 1st edn. Springer Publishing Company, Incorporated, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01492-5
8. Fahrenberg, U., Larsen, K.G., Thrane, C.: A quantitative characterization of weighted Kripke structures in temporal logic. Comput. Inf. **29**, 1311–1324 (2010)
9. Jensen, J.F., Larsen, K.G., Srba, J., Oestergaard, L.K.: Efficient model-checking of weighted CTL with upper-bound constraints. STTT **18**(4), 409–426 (2016)
10. Singh Kambo, N., Kotz, S.: On exponential bounds for binomial probabilities. Ann. Inst. Stat. Math. **18**(1), 277 (1966)
11. Liu, X., Smolka, S.A.: Simple linear-time algorithms for minimal fixed points (extended abstract). In: Proceedings of the Automata, Languages and Programming, 25th International Colloquium, ICALP 1998, Aalborg, Denmark, July 13-17, 1998, pp. 53–66 (1998)
12. Monniaux, D.: Quantifier elimination by Lazy model enumeration. In: Touili, T., Cook, B., Jackson, P. (eds.) CAV 2010. LNCS, vol. 6174, pp. 585–599. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14295-6_51
13. Okamoto, M.: Some inequalities relating to the partial sum of binomial probabilities. Ann. Inst. Stat. Math. **10**(1), 29–35 (1959)
14. Srkk, S.: Bayesian Filtering and Smoothing. Cambridge University Press, New York (2013)
15. Vardi, M.Y.: Automatic verification of probabilistic concurrent finite-state programs. In: 26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21–23 October 1985, pp. 327–338. IEEE Computer Society (1985)
16. Wolfram Research, Inc., Mathematica, Version 11.2. Champaign, IL (2017)