A Logic for Accumulated-Weight Reasoning on Multiweighted Modal Automata

Sebastian Bauer
Institut für Informatik
Ludwig-Maximilians-Universität München
München, Germany
Email: bauerse@pst.ifi.lmu.de

Line Juhl, Kim G. Larsen, Jiří Srba
Department of Computer Science
Aalborg University
Aalborg, Denmark
Email: {linej,kgl,srba}@cs.aau.dk

Axel Legay
INRIA/IRISA
Rennes, France
and Aalborg University
Email: axel.legay@inria.fr

***Abstract*—Multiweighted modal automata provide a specification theory for multiweighted transition systems that have recently attracted interest in the context of energy games. We propose a simple fragment of CTL that is able to express properties about accumulated weights along maximal runs of multiweighted modal automata. Our logic is equipped with a game-based semantics and guarantees both soundness (formula satisfaction is propagated to the modal refinements) as well as completeness (formula non-satisfaction is propagated to at least one of its implementations). We augment our theory with a summary of decidability and complexity results of the generalized model checking problem, asking whether a specification—abstracting the whole set of its implementations—satisfies a given formula.**

***Keywords*-weighted modal automata; model checking; decidability;**

I. Introduction

Modal transition systems [1] (MTS) have been recently studied as a suitable specification formalism in connection with step-wise design of component-based systems [2], [3]. This model is essentially a labelled transition system with two kinds of transition relations: a *may* (allowed) and *must* (compulsory) transition relation. During the system design process the must-transitions must be preserved, while the may-transitions may be omitted.

One of the key elements in model-based design is the notion of *refinement*. Specifications are gradually refined into more concrete ones until we arrive at the most concrete specification (called *implementation*) that cannot be refined any more. Taking the point of view that implementations can be seen as (abstractions of) the final products like executable systems we want to implement, we get the natural notion of so-called *thorough refinement*: specification S_1 thoroughly refines S_2 if any implementation of S_1 is also an implementation of S_2. Unfortunately, the thorough refinement preorder is computationally hard and deciding thorough refinement between two specifications is EXPTIME-complete [4]. Hence, for feasibility reasons, thorough refinement is often approximated by *modal refinement* (defined directly on specifications in a bisimulation-like manner). Deciding modal refinement is possible in deterministic polynomial time [5] and modal refinement is a sufficient condition for concluding thorough refinement, though not a necessary condition. [6].

On the logical counter-part of the theory, already the first work on MTS [1] introduced Hennessy-Milner logic for MTS with a model checking procedure on the specifications to decide whether all of its implementations satisfy the formula. Later, (3-valued) extensions of the problem were considered for several branching and linear time logics [7], [8], [9], [10], [11], [12], [13]. In this paper we study the model checking problem for multiweighted modal automata. In comparison to other related works, we do not deal with Kripke MTS, but we consider multiweighted modal automata, that are finite MTS with vectors of integer intervals as labels. These (multi-)weighted automata have recently been in the focus of the research community but treated mainly as reachability/infinite runs problems with cost/energy objectives without any logical characterization. The theory of multiweighted modal automata constitutes an abstraction theory for multiweighted automata which we already studied in the context of energy games [14].

As a first contribution we investigate model checking formulae of a simple fragment of CTL with atomic propositions referring explicitly to *accumulated (multi-)weights* along maximal runs in the automata. This logic is inspired by the one we recently proposed in [15], but with the crucial addition of a game semantics that is needed to prove completeness with respect to refinement. More precisely, the semantics of the logic is defined according to a game-based interpretation with two players, the *must-player* and the *may-player*. The intuition is that the selections of the must-player can be realized in *every* possible implementation, whereas the may-player can choose between all the design choices of the specification that are not yet fixed. The fact that our semantics guarantees that *all* implementations satisfy a formula[1], immediately implies preservation of formula

This work is partially supported by MT-LAB, VKR Centre of Excellence and the Sino-Danish Center IDEA4CPS.

Sebastian Bauer is partially supported by the German Academic Exchange Service (DAAD), grant D/10/46169.

Axel Legay is partially supported by the CREATIVE Project ESTASE and the EU projects DANSE and DALI.

[1] Other works seek for the existence of one such implementation [11].

satisfaction under refinement. This leads to a new aspect not seen in previous works, namely that of *completeness* of our logic: if all implementations of a specification satisfy a formula, so does the specification. Completeness is also of practical interest as it allows for counterexample generation.

As our second contribution, we provide an overview of decidability and complexity analysis of the model checking problems for our proposed accumulated-weight logic, showing that in their full generality the problems are undecidable but by imposing some natural restrictions we get decidability and the problems in many cases specialize to the well studied problems in the theory. Throughout the paper, we restrict ourselves to the four EF, EG, AF and AG fragments of CTL as they provide a good balance between the expressiveness of the logic for practical applications and on the other hand allow us to conclude at least partial decidability results.

Related work: Weighted models have been widely studied over the past years [16], [17]. Other works on quantitative models with quantitative reasoning include [18], [19], [20]. The main difference with our model is that they do not consider formalisms capable of a step-wise refinement process like we do through the modalities.

In a very recent paper [21], the branching time logic CTL has been extended to quantitative objectives that allow reasoning on several accumulated weights. The underlying model used in this work is the one of quantitative Kripke structures and the paper presents a largest decidable fragment of this logic. The logic we propose is only a fragment of CTL extended with accumulative reasoning on multiweights, however, we use multiweighted *modal* transition systems as the underlying specification model, not just the implementations. Hence, contrary to others, our model is able to express both allowed and required behaviours, and also looseness of the quantitative information, not possible in [21].

II. Multiweighted Modal Automata

We define $[a, b] = \{n \in \mathbb{Z} \mid a \leq n \leq b\}$ for $a \leq b$, $a \in \mathbb{Z} \cup \{-\infty\}$, $b \in \mathbb{Z} \cup \{\infty\}$ to denote the interval with lower bound a and upper bound b, and we use $\mathbb{W}$ to denote the set of all such intervals. A *k-weight interval*, for a natural number $k \geq 1$, is an element $\overline{W} \in \mathbb{W}^k$, in other words a vector consisting of k intervals. Projection on the i-th interval, $1 \leq i \leq k$, is denoted by $\overline{W}[i]$. Moreover, we write $\overline{W} \subseteq \overline{V}$ where $\overline{W}, \overline{V} \in \mathbb{W}^k$ iff $\overline{W}[i] \subseteq \overline{V}[i]$ for all i, $1 \leq i \leq k$. The set of all singleton intervals of the form $[a, a]$ is denoted by $\mathbb{W}_1$ and by misusing the notation the set $\mathbb{W}_1^k$ will often be identified with $\mathbb{Z}^k$ and we refer to its elements as *k-weights* or just *weights*. The addition of two weights $\overline{w}_1, \overline{w}_2 \in \mathbb{Z}^k$ is defined by $(\overline{w}_1 \oplus \overline{w}_2)[i] = \overline{w}_1[i] + \overline{w}_2[i]$.

A k-weighted modal automaton is a variant of the classical modal transition systems [1] where transitions are labeled by k-weight intervals (instead of actions). A k-weighted modal automaton differs from the usual weighted automaton by distinguishing two transition relations: the *may*-transition relation expresses which transitions are optional for any implementation, and the *must*-transition relation contains those transitions that are mandatory for any implementation.

Definition 1: A *k-weighted modal automaton* is a tuple $\mathcal{S} = (S, s_0, \overline{w}_0, \dashrightarrow, \longrightarrow)$ where S is a set of states, $s_0 \in S$ is an initial state, $\overline{w}_0 \in \mathbb{Z}^k$ is an initial weight, and $\longrightarrow \subseteq \dashrightarrow \subseteq S \times \mathbb{W}^k \times S$ are the must- and may-transition relations, respectively.

The set of all k-weighted modal automata is denoted by $\mathbb{M}$. Given a k-weighted modal automaton $\mathcal{S} = (S, s_0, \overline{w}_0, \dashrightarrow, \longrightarrow) \in \mathbb{M}$, a state $s \in S$ and a weight $\overline{w} \in \mathbb{Z}^k$, we write $\mathcal{S}_{(s,\overline{w})}$ for the k-weighted modal automaton $(S, s, \overline{w}, \dashrightarrow, \longrightarrow)$ where the initial state s_0 and the initial weight $\overline{w}_0$ are replaced by s and $\overline{w}$, respectively.

A k-weighted modal automaton $\mathcal{S} = (S, s_0, \overline{w}_0, \dashrightarrow, \longrightarrow)$ is an *implementation* iff all labels are singleton intervals from $\mathbb{W}_1^k$ and $\dashrightarrow = \longrightarrow$. In other words, all allowed transitions are also implemented and all choices of concrete weights from the interval are realized.

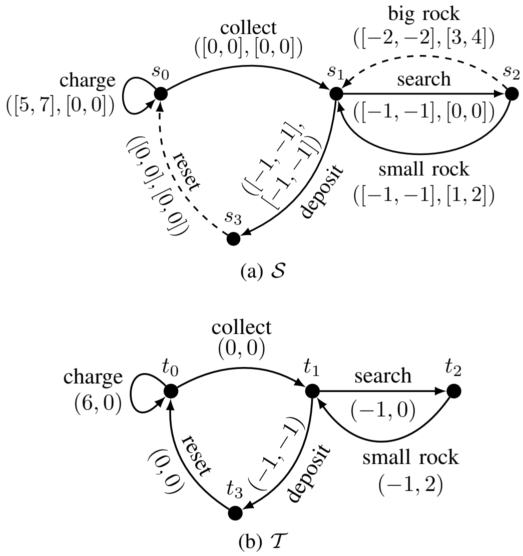
We shall now introduce the classical notion of modal refinement [1] extended with the interval refinement, defined similarly as in [15].

Definition 2: Let $\mathcal{S}_i = (S_i, s_{0,i}, \overline{w}_{0,i}, \dashrightarrow_i, \longrightarrow_i)$ for $i \in \{1, 2\}$ be two k-weighted modal automata. We say that $\mathcal{S}_1$ *modally refines* $\mathcal{S}_2$, written as $\mathcal{S}_1 \leq_m \mathcal{S}_2$, if $\overline{w}_{0,1} = \overline{w}_{0,2}$ and there is a binary relation $R \subseteq S_1 \times S_2$ such that $(s_{0,1}, s_{0,2}) \in R$ and for all $(s_1, s_2) \in R$:

1) whenever $s_1 \overset{\overline{W}_1}{\dashrightarrow}_1 s_1'$ then there exists $s_2 \overset{\overline{W}_2}{\dashrightarrow}_2 s_2'$ such that $\overline{W}_1 \subseteq \overline{W}_2$ and $(s_1', s_2') \in R$,
2) whenever $s_2 \overset{\overline{W}_2}{\longrightarrow}_2 s_2'$ then there exists $s_1 \overset{\overline{W}_1}{\longrightarrow}_1 s_1'$ such that $\overline{W}_1 \subseteq \overline{W}_2$ and $(s_1', s_2') \in R$.

Clearly, modal refinement is a preorder. The set of all implementations of a modal automaton $\mathcal{S}$ is then defined by $[\![\mathcal{S}]\!]_{\mathrm{impl}} = \{\mathcal{I} \mid \mathcal{I} \leq_m \mathcal{S} \text{ and } \mathcal{I} \text{ is an implementation }\}$.

Example 1: Examples of 2-weighted modal automata are drawn in Figure 1. Figure 1a shows a specification $\mathcal{S}$ of a Mars vehicle. The vehicle has a battery and a container for carrying rocks that it collects. The first weight denotes changes in the battery level, while the second weight denotes changes in the accumulated volume of rocks. In s_0 the battery can be charged, while state s_1 enables the search for new rocks or the deposit of a rock. The abilities to collect a big rock and to reset are not required behaviours in a possible implementation. The automaton $\mathcal{T}$ given in Figure 1b is a refinement of $\mathcal{S}$, demonstrated by the refinement relation $\{(t_0, s_0), (t_1, s_1), (t_2, s_2), (t_3, s_3)\}$. This refinement is furthermore an implementation of $\mathcal{S}$ as it implements two of the three proper may-transitions present in $\mathcal{S}$ (and all must-transitions) and has all transitions labelled with k-weights from the corresponding interval in $\mathcal{S}$.



(a) $\mathcal{S}$

(b) $\mathcal{T}$

Figure 1. Examples of 2-weighted modal automata

III. Games on Multiweighted Modal Automata and Logic $\mathcal{L}$

As motivated in the introduction, the semantics of our logic will be based on a game characterization in order to be able to argue for its completeness. We shall now introduce this game.

Let $\mathcal{S} = (S, s_0, \overline{w}_0, \dashrightarrow, \longrightarrow)$ be a k-weighted modal automaton. The game is played in rounds, with two players called the *must-* and *may-player*. The set of states S is partitioned into *must-* and *may-states* in which it is the turn of the must- and may-player, respectively. A *must-state* is a state in which there is at least one outgoing must-transition, otherwise the state is a *may-state*. *Configurations* are of the form $(s, \overline{w})$ where $s \in S$ and $\overline{w} \in \mathbb{Z}^k$ is the so far accumulated weight. Each round starts from the current configuration $(s, \overline{w})$, initially $(s_0, \overline{w}_0)$, and has two steps: (i) selection of a transition and (ii) choosing a specific weight from the weight interval of the chosen transition. Formally,

(i) a) If s is a must-state, then the must-player chooses some must-transition $(s, \overline{W}, s') \in \longrightarrow$.

b) If s is a may-state, then the may-player chooses some may-transition $(s, \overline{W}, s') \in \dashrightarrow$ or decides to stop the game.

(ii) Afterwards, if the game was not stopped and a transition $(s, \overline{W}, s')$ was selected, the may-player chooses a weight $\overline{w}' \in \overline{W}$.

The pair $(s', \overline{w} \oplus \overline{w}')$ now becomes the new current configuration and the game continues with a next round.

The intuition is that the must-player can only select must-transitions in must-states and thus the selections (or moves) of the must-player can be realized in *every* possible implementation. The may-player can choose between all the design choices of $\mathcal{S}$ which are not yet fixed, i.e. the may-player can choose whether to take any may-transition (and in this case, which one) or not, and which weight to pick from a weight interval (both for may- as well as must-transitions).

Any maximal sequence (finite that cannot be prolonged using a must-transition or infinite) of configurations $(s_0, \overline{w}_0)(s_1, \overline{w}_1)\ldots$ such that for all $i \geq 0$, we have $s_i \overset{\overline{W}_i}{\dashrightarrow} s_{i+1}$ and $\overline{w}_{i+1} = \overline{w}_i \oplus \overline{v}_i$ where $\overline{v}_i \in \overline{W}_i$, is called a *run* on $\mathcal{S}$. Let $runs(s, \overline{w})$ denote the set of all runs starting from $(s, \overline{w})$ on $\mathcal{S}_{(s,\overline{w})}$. Any run complying with the above rules (i) and (ii) is called a *play* in $\mathcal{S}$. A *strategy of the must-player* is a function σ that maps every finite prefix $(s_0, \overline{w}_0)(s_1, \overline{w}_1)\ldots(s_r, \overline{w}_r)$ of a play, ending in a must-state s_r, to a must-transition $(s_r, \overline{W}, s') \in \longrightarrow$. For a fixed strategy σ of the must-player, we define the set $plays(\sigma, (s, \overline{w}))$ of all plays starting from $(s, \overline{w})$ on $\mathcal{S}_{(s,\overline{w})}$ in which the choice of the next must-transition is according to σ. For a run $\gamma \in runs(s, \overline{w})$ the projection to the i-th configuration is denoted by γ_i.

We are now ready to define a fragment $\mathcal{L}$ of the logic CTL to express properties about accumulated weights of maximal runs in multiweighted modal automata. The satisfaction relation will be defined via the games introduced above and we will show that the logic is sound and complete w.r.t. refinement. In what follows let k implicitly represent the number of weight coordinates.

The set of *linear expressions* is given by the abstract syntax $e ::= \langle i \rangle \cdot c \mid e + e$ where $1 \leq i \leq k$ and $c \in \mathbb{Z}$. The *$\mathcal{L}$-formulae* are given by the abstract syntax

$$\varphi, \varphi_1, \varphi_2 ::= e \bowtie b \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2$$
$$\psi ::= \mathsf{AG}\varphi \mid \mathsf{AF}\varphi \mid \mathsf{EG}\varphi \mid \mathsf{EF}\varphi$$

where e is a linear expression, $\bowtie \in \{<, \leq, =, \neq, \geq, >\}$, $b \in \mathbb{Z} \cup \{-\infty, \infty\}$.

In order to give the semantics, we first define, for a linear expression e and a weight $\overline{w} \in \mathbb{Z}^k$, its denotational semantics $[\![e]\!]_{\overline{w}} \in \mathbb{Z}$ by $[\![\langle i \rangle \cdot c]\!]_{\overline{w}} = \overline{w}[i] \cdot c$ and $[\![e_1 + e_2]\!]_{\overline{w}} = [\![e_1]\!]_{\overline{w}} + [\![e_2]\!]_{\overline{w}}$. The satisfaction of an $\mathcal{L}$-formula ψ in a configuration $(s, \overline{w})$ is now given in Figure 2. Notice that the semantics of the AF and AG fragments can also be defined using strategies. However, as these reduce essentially to one-player games, we prefer to give their direct definitions for clarity.

By the EF, AF, EG and AG fragments we refer to the four main fragments of $\mathcal{L}$, namely formulae of the form $\mathsf{EF}\varphi$, $\mathsf{AF}\varphi$, $\mathsf{EG}\varphi$ or $\mathsf{AG}\varphi$, respectively. If φ moreover does not contain any disjunction (conjunction), we call the fragment *disjunction-free* (*conjunction-free*). In the following, for $\mathcal{S} = (S, s_0, \overline{w}_0, \dashrightarrow, \longrightarrow) \in \mathbb{M}$, we shortly write $\mathcal{S} \models \psi$ whenever $(s_0, \overline{w}_0) \models \psi$.

Example 2: As an example of an $\mathcal{L}$-formula consider $\psi = \mathsf{EG}\big(\langle 1 \rangle \geq 0 \wedge \langle 1 \rangle \leq 10 \wedge \langle 2 \rangle \geq 0 \wedge \langle 2 \rangle \leq 6\big)$ in connection with the modal automata from Figure 1. The formula claims the existence of a strategy for the must-

$$\begin{array}{lll}
(s,\overline{w}) \models e \bowtie b & \text{iff} & [\![e]\!]_{\overline{w}} \bowtie b \\
(s,\overline{w}) \models \varphi_1 \wedge \varphi_2 & \text{iff} & (s,\overline{w}) \models \varphi_1 \text{ and } (s,\overline{w}) \models \varphi_2 \\
(s,\overline{w}) \models \varphi_1 \vee \varphi_2 & \text{iff} & (s,\overline{w}) \models \varphi_1 \text{ or } (s,\overline{w}) \models \varphi_2 \\
(s,\overline{w}) \models \mathsf{AG}\varphi & \text{iff} & \forall \gamma \in \mathit{runs}(s,\overline{w})\ \forall i : \gamma_i \models \varphi \\
(s,\overline{w}) \models \mathsf{AF}\varphi & \text{iff} & \forall \gamma \in \mathit{runs}(s,\overline{w})\ \exists i : \gamma_i \models \varphi \\
(s,\overline{w}) \models \mathsf{EG}\varphi & \text{iff} & \text{there exists a strategy } \sigma \\
 & & \text{of the must-player such that} \\
 & & \forall \gamma \in \mathit{plays}(\sigma,(s,\overline{w}))\ \forall i : \gamma_i \models \varphi \\
(s,\overline{w}) \models \mathsf{EF}\varphi & \text{iff} & \text{there exists a strategy } \sigma \\
 & & \text{of the must-player such that} \\
 & & \forall \gamma \in \mathit{plays}(\sigma,(s,\overline{w}))\ \exists i : \gamma_i \models \varphi
\end{array}$$

Figure 2. Semantics of the logic $\mathcal{L}$

In s_0: **if** $\langle 1 \rangle = 0 \ \vee\ (\langle 1 \rangle = 1 \wedge \langle 2 \rangle = 0)$
then charge **else** collect
In s_1: **if** $(\langle 1 \rangle \geq 4 \wedge \langle 2 \rangle = 0) \ \vee\ (\langle 1 \rangle = 3 \wedge \langle 2 \rangle \in \{0,1\})$
then search **else** deposit

Figure 3. A strategy for the must-player

player such that the battery level is between 0 and 10 and the volume of accumulated rocks is between 0 and 6 in any configuration. By consulting Figure 1 we can see that $\mathcal{S} \models \psi$. No matter what weight the may-player chooses within the intervals and regardless whether the may-player chooses to stop or not in s_3, the must-player can always keep the two accumulated weights between the bounds. A sufficient strategy for the must-player is seen in Figure 3. Notice that the choice for the must-player in the state s_2 is always to go for a small rock, thus this is omitted in the strategy. As we will see shortly, the logic $\mathcal{L}$ is sound, implying that an $\mathcal{L}$-formula satisfied by $\mathcal{S}$ is also satisfied by all refinements of $\mathcal{S}$, thus $\mathcal{T} \models \psi$ as well.

On the other hand, the formula $\psi' = \mathsf{AG}\big(\langle 1 \rangle + \langle 2 \rangle \leq 20\big)$ can be easily seen *not* to be satisfied by $\mathcal{S}$. Since the logic $\mathcal{L}$ in this paper is proved complete, it allows us to generate counter-examples: implementations of $\mathcal{S}$ that do not satisfy the formula ψ'. Indeed, the implementation $\mathcal{T}$ does not satisfy ψ' as a run may consist of continuously charging using the selfloop in t_0.

Remark 1: Note that the formula $\mathsf{EF}\psi$ cannot in general be expressed as the negation of $\mathsf{AG}\neg\psi$ as opposed to the classical CTL. Consider for example a 1-weighted modal automaton with just one transition $s_0 \overset{[2,2]}{\dashrightarrow} s_1$. Clearly, $(s_0, \overline{0}) \not\models \mathsf{EF}\ \langle 1 \rangle = 2$ while also $(s_0, \overline{0}) \not\models \mathsf{AG}\ \langle 1 \rangle \neq 2$. Similarly, AF is not dual with EG. On the other hand, as expected, the classical duality laws for EF and AG, as well as AF and EG, hold on implementations.

Now we can show that satisfaction of any $\mathcal{L}$-formula is preserved by modal refinement. In particular this means that if $\mathcal{S} \models \psi$ then any implementation of $\mathcal{S}$ also satisfies ψ. The proof can be found in the full version of the paper.

Theorem 1: Let $\mathcal{S} \in \mathbb{M}$ be a k-weighted modal automaton and ψ be an $\mathcal{L}$-formula. Then

$$\mathcal{S} \models \psi \quad \Longrightarrow \quad \big(\forall\, \mathcal{T} \in \mathbb{M} : \mathcal{T} \leq_m \mathcal{S} \implies \mathcal{T} \models \psi\big)\ .$$

We shall now argue for the completeness of our logic. The proof is more straightforward for the formulae $\mathsf{AG}\varphi$ and $\mathsf{AF}\varphi$, and for the formulae $\mathsf{EG}\varphi$ and $\mathsf{EF}\varphi$ the proof relies on the fact that in our turn-based games the absence of a winning strategy implies the existence of a spoiling strategy for the opponent.

Theorem 2: Let $\mathcal{S} \in \mathbb{M}$ be a k-weighted modal automaton and ψ be an $\mathcal{L}$-formula. Then

$$\big(\forall \mathcal{I} \in [\![\mathcal{S}]\!]_{\mathrm{impl}} : \mathcal{I} \models \psi\big) \quad \Longrightarrow \quad \mathcal{S} \models \psi\ .$$

Proof (sketch): Let $\mathcal{S} = (S, s_0, \overline{w}_0, \dashrightarrow, \longrightarrow) \in \mathbb{M}$ be a k-weighted modal automaton. Let φ be a logical composition of atomic propositions of the form $e \bowtie b$. Since any k-weighted modal automaton has an implementation $\mathcal{I}$ and $\mathcal{I} \models \varphi$ by assumption, necessarily also $\mathcal{S} \models \varphi$.

Now consider the case $\psi \in \{\mathsf{AG}\varphi, \mathsf{AF}\varphi\}$. Let $\gamma_{\mathcal{S}}$ be a run in $\mathcal{S}$. Then, clearly, there exists an implementation $\mathcal{I} \in [\![\mathcal{S}]\!]_{\mathrm{impl}}$ and a run $\gamma_{\mathcal{I}}$ in $\mathcal{I}$ such that both runs have the same length, and for all i, the state of $(\gamma_{\mathcal{I}})_i$ is in a modal refinement relation with $(\gamma_{\mathcal{S}})_i$, i.e. $\mathcal{I}_{(\gamma_{\mathcal{I}})_i} \leq_m \mathcal{S}_{(\gamma_{\mathcal{S}})_i}$, and the accumulated weights of $(\gamma_{\mathcal{I}})_i$ and $(\gamma_{\mathcal{S}})_i$ coincide. So essentially both runs have the same sequence of weights, and we can conclude that $\mathcal{S} \models \psi$.

Let us now consider the case $\psi \in \{\mathsf{EG}\varphi, \mathsf{EF}\varphi\}$. Assume that $\mathcal{S}$ does *not* satisfy ψ, i.e. $\mathcal{S} \not\models \psi$, then there does not exist any strategy σ for the must-player such that all plays $\gamma \in \mathit{plays}(\sigma, (s_0, \overline{w}_0))$ satisfy the respective property. We can infer the existence of a spoiling strategy of the may-player as follows. First, we define $\mathit{win} = \{(s,\overline{w}) \mid (s,\overline{w}) \text{ is a configuration in } \mathcal{S}$ $\text{and must-player has a strategy in } \mathcal{S}_{(s,\overline{w})} \text{ witnessing } \psi\}$. Given a configuration $(s,\overline{w})$ in $\mathcal{S}$ such that $(s,\overline{w}) \notin \mathit{win}$ and s is a must-state, then for all choices of the must-player we know that there is a next choice of the may-player (selection of a weight from the interval) such that the next configuration is not in win. Similarly, if s is a may-state, we also know that there exists at least one choice of the next (may-)transition and weight such that the next configuration is not in win; otherwise it would contradict with $(s,\overline{w}) \notin \mathit{win}$.

Obviously, we can construct an implementation $\mathcal{I}$ of $\mathcal{S}$, in the form of a tree, for which the weights and the presence of transitions are chosen according to the above choices of the must- and may-players. It is clear that the must-player does not have any strategy witnessing the formula ψ since any play in $\mathcal{I}$ does not satisfy the respective property. This contradicts our assumption that all implementations satisfy ψ. Hence $\mathcal{S} \models \psi$. ■

IV. Decidability and Complexity of the Logic $\mathcal{L}$

We shall now study the *model checking problem* for $\mathcal{L}$: given a finite k-weighted modal automaton $\mathcal{T}$ and an $\mathcal{L}$-formula ψ, the question is to decide whether $\mathcal{T} \models \psi$. As we will show, the problem is undecidable in general. Fortunately, there are several practically usable fragments of the logic for which we show decidability. In the rest of this section we shall implicitly assume that all input modal automata are finite. The first part concentrates on undecidability results, while the two following parts study the decidable fragments of the logic.

A. Undecidability of EF*,* EG *and* AF

We start by showing that in general the model checking problem is undecidable for three (EF, EG, AF) of the four fragments of the logic.

The authors of [21] propose a CTL logic on Kripke structures that can reason about multiple accumulated weights and they show undecidability of an unnested EG formula. As our semantics of $\mathcal{L}$ corresponds to normal CTL semantics when interpreted on implementations (the may-player has no choices, thus the existence of a strategy corresponds to the existence of a run), the undecidable formula constructed in [21] can be expressed using an EG formula from our logic $\mathcal{L}$. Since the operators are dual on implementations (see Remark 1), the model checking problems of the EG and AF fragments of $\mathcal{L}$ are undecidable, even for implementations.

Theorem 3 ([21]): The model checking problems of the EG and AF fragments are undecidable, even for implementations.

For the EF fragment we will describe a reduction from the halting problem of a 2-counter Minsky machine to the model checking problem of the EF fragment of the logic. Recall that a *Minsky machine* [22] consists of a finite number of instructions and two nonnegative integer counters initially both set to 0. Each instruction either increases one of the counters (an *increment* instruction) or tests for zero and decreases a counter (a *test-and-decrement* instruction). We say that a Minsky machine halts if it is possible to reach the last instruction called *halt* when starting from the first instruction. Otherwise it *loops*. It is well-known that the halting problem for Minsky machines is undecidable [22].

We now describe the reduction for the EF fragment where the problem becomes undecidable even if we restrict ourselves only to specifications where the may- and must-transitions coincide (though intervals are not necessarily singletons).

Let $1 : \mathsf{inst}_1;\ 2 : \mathsf{inst}_2;\ \ldots;\ n-1 : \mathsf{inst}_{n-1};\ n : \mathsf{halt}$ be a Minsky machine over the nonnegative integer counters c_1 and c_2. We construct a 9-weighted modal automaton $\mathcal{S} = (S, s_1, \overline{0}, \dashrightarrow, \longrightarrow) \in \mathbb{M}$ where every may-transition is also a must-transition and an $\mathcal{L}$-formulae $\mathsf{EF}\varphi$ such that $(s_1, \overline{0}) \models \mathsf{EF}\varphi$ iff the Minsky machine halts. The intuition behind the coordinates is as follows: $\langle 1 \rangle$ represents the first counter, if $\langle 3 \rangle$ is set to 1 then the may-player is testing if the first counter is empty, if $\langle 5 \rangle$ is set to 1 then the may-player is testing if the first counter is nonempty, if $\langle 7 \rangle$ is set to 1 then the may-player indicates that an increment instruction is not allowed. The role of the coordinates $\langle 2 \rangle$, $\langle 4 \rangle$, $\langle 6 \rangle$ and $\langle 8 \rangle$ is dual and corresponds to the second counter. Finally, if $\langle 9 \rangle$ is nonzero, the halt instruction has been reached.

Let $S = \{s_i \mid 1 \le i \le n\}$ be the set of states. The transitions are of the following types, depending on the instructions of the Minsky machine (here $1 \le j \le 2$, $1 \le i, k, \ell \le n$).

1) For each instruction i: `c`$_j$ `:= c`$_j$ `+ 1; goto` k, we add the transitions
   - $s_i \xrightarrow{(1,0,0,0,0,0,[0,1],0,0)} s_k$ if $j = 1$, and
   - $s_i \xrightarrow{(0,1,0,0,0,0,0,[0,1],0)} s_k$ if $j = 2$.
2) For each instruction i: `if c`$_j$ `= 0 then goto` k `else (c`$_j$ `:= c`$_j$ `- 1; goto` ℓ`)`, we add the transitions
   - $s_i \xrightarrow{(0,0,[0,1],0,0,0,0,0,0)} s_k$ and $s_i \xrightarrow{(-1,0,0,0,[0,1],0,0,0,0)} s_\ell$ if $j = 1$, and
   - $s_i \xrightarrow{(0,0,0,[0,1],0,0,0,0,0)} s_k$ and $s_i \xrightarrow{(0,-1,0,0,0,[0,1],0,0,0)} s_\ell$ if $j = 2$.
3) Finally, we add the transition $s_n \xrightarrow{(0,0,0,0,0,0,0,0,1)} s_n$.

Let now

$$\begin{aligned}
\varphi_1 &= \langle 1 \rangle = 0 \ \wedge\ \langle 3 \rangle = 1 \ \wedge\ \langle 5 \rangle = 0\\
\varphi_2 &= \langle 2 \rangle = 0 \ \wedge\ \langle 4 \rangle = 1 \ \wedge\ \langle 6 \rangle = 0\\
\varphi_3 &= \langle 1 \rangle \ge 0 \ \wedge\ \langle 5 \rangle = 1 \ \wedge\ \langle 7 \rangle = 0\\
\varphi_4 &= \langle 2 \rangle \ge 0 \ \wedge\ \langle 6 \rangle = 1 \ \wedge\ \langle 8 \rangle = 0\\
\varphi_5 &= \langle 5 \rangle = 0 \ \wedge\ \langle 7 \rangle = 1\\
\varphi_6 &= \langle 6 \rangle = 0 \ \wedge\ \langle 8 \rangle = 1\\
\varphi_7 &= \langle 9 \rangle = 1 \ \wedge\ \langle 3 \rangle = 0 \ \wedge\ \langle 4 \rangle = 0 \wedge\ \langle 5 \rangle = \ 0\ \wedge\\
&\quad \langle 6 \rangle = 0 \ \wedge\ \langle 7 \rangle = 0 \ \wedge\ \langle 8 \rangle = 0
\end{aligned}$$

and $\varphi = \varphi_1 \ \vee\ \varphi_2 \ \vee\ \varphi_3 \ \vee\ \varphi_4 \ \vee\ \varphi_5 \ \vee\ \varphi_6 \ \vee\ \varphi_7$.

We now argue that $(s_1, \overline{0}) \models \mathsf{EF}\varphi$ iff the Minsky machine halts. Assume first that the machine halts. Then the must-player can reach the state s_n with some accumulated weight $\overline{w}$ by faithfully simulating the Minsky machine. If the may-player picks 0 in all intervals, φ_7 is satisfied in $(s_n, \overline{w})$, and thus also φ. If, on the other hand, the may-player picks 1 in any interval, this will force φ_1, φ_2, φ_3, φ_4, φ_5 or φ_6 to be true. Therefore $(s_0, \overline{0}) \models \mathsf{EF}\varphi$ holds regardless of the choices of the may-player.

Assume now that the Minsky machine does not halt. If the must-player does not cheat, s_n can never be reached and φ_7 can thus never be true. If the may-player chooses 0 in every interval, neither φ_1, φ_2, φ_3, φ_4, φ_5 nor φ_6 can be true in any configuration, hence $(s_0, \overline{0}) \models \mathsf{EF}\varphi$ can never be true.

We investigate what happens if the must-player cheats. This is possible either by (1) taking the transition $s_i \xrightarrow{(0,0,[0,1],0,0,0,0,0,0)} s_k$ while $c_1 > 0$ or (2) taking $s_i \xrightarrow{(-1,0,0,0,[0,1],0,0,0,0)} s_\ell$ while $c_1 = 0$ (similarly for c_2).

In case (1), the may-player sets $\langle 3 \rangle = 1$. Since the accumulated weight in coordinates 3-8 can never be lowered, φ_1 can still hope to be true at some point. This would be possible if $\langle 1 \rangle = 0$, but for this not to happen the may-player sets $\langle 5 \rangle = 1$ should the other player try to decrement $\langle 1 \rangle$. This ensures that φ_1 cannot be true, so $(s_0, \overline{0}) \models \mathsf{EF}\varphi$ can never be true if the may-player chooses 0 in the remaining intervals (unless the must-player cheats in the other counter; here the may-player behaves analogously).

In case (2), the may-player sets $\langle 5 \rangle = 1$. Now φ_3 can still be true at some point. However, that would require $\langle 1 \rangle = 0$, and when incrementing $\langle 1 \rangle$, the may-player can set $\langle 7 \rangle = 1$, making sure that $\mathsf{EF}\varphi$ cannot become true. Similarly for c_2.

The above construction leads to the following theorem.

Theorem 4: The model checking problem of the EF fragment is undecidable, even for specifications with intervals but where the may- and must-transition relations coincide.

Notice that in the proof of Theorem 4 we use intervals to argue for undecidability. We will see that this is exactly the issue causing undecidability.

B. Decidability with Restricted Fragments

In order to obtain decidability results, we first restrict our model to contain only singleton intervals and notice that the EF fragment becomes decidable. Notice that this restriction is not sufficient for the two remaining undecidable fragments, EG and AF, since these are undecidable even for implementations.

Notice that a formula $\mathsf{EF}\varphi$ is true on a k-weighted modal automaton iff it is true on its *must-projection* that is obtained by removing all the may-transitions that do not have a corresponding must-transition. The reason is that the may-player can, in any may-state, decide to stop the game and hence the formula φ must be satisfied in some configuration reachable via must-transitions only. So in the following we assume that $\dashrightarrow = \longrightarrow$. Since the intervals are assumed singletons, we therefore need to consider only implementations. The EF fragment was shown to be decidable for implementations in [21] (even if we allow nesting). We thus have the following decidability result.

Theorem 5 ([21]): The model checking problem of the EF fragment of $\mathcal{L}$ against weighted modal automata with singleton intervals is decidable.

We shall now argue that by restricting the formulae to contain only conjunctions, some fragments of the logic become decidable.

First we show that with singleton intervals (though still allowing modalities) the model checking problem for any $\mathcal{L}$-formula reduces to the same problem for a formula with so-called simple linear expressions. A linear expression is *simple* if it is of the form $\langle i \rangle \bowtie b$, where $\bowtie \in \{\leq, \geq\}$ and $b \in \mathbb{Z}$.

Lemma 1: Model checking of an $\mathcal{L}$-formula against a weighted modal automaton with singleton intervals is polynomial time reducible to model checking an $\mathcal{L}$-formula where all linear expressions are simple.

Proof: Let $\mathcal{S} = (S, s_0, \overline{w}_0, \dashrightarrow, \longrightarrow)$ be a k-weighted modal automaton with only singleton intervals. Consider any linear expression of the form $e = \langle i_1 \rangle \cdot c_1 + \ldots + \langle i_n \rangle \cdot c_n$, $i_1, \ldots, i_n \in \{1, \ldots, k\}$. Notice that for any modal automaton with singleton intervals, step (ii) in the game can be ignored, since the weights are already uniquely given.

We now construct a $(k+1)$-weighted automaton $\mathcal{T} = (S, s_0, \overline{w}_0', \dashrightarrow_{\mathcal{T}}, \longrightarrow_{\mathcal{T}})$, where $\overline{w}_0' = (\overline{w}_0[1], \ldots, \overline{w}_0[k], \overline{w}_0[i_1] \cdot c_1 + \ldots + \overline{w}_0[i_n] \cdot c_n)$. For each $s \overset{\overline{w}}{\dashrightarrow} s'$ in $\dashrightarrow$ we add $s \overset{\overline{v}}{\dashrightarrow} s'$ to $\dashrightarrow_{\mathcal{T}}$, where $\overline{v} = (\overline{w}[1], \ldots, \overline{w}[k], \overline{w}[i_1] \cdot c_1 + \ldots + \overline{w}[i_n] \cdot c_n)$. The set $\longrightarrow_{\mathcal{T}}$ is constructed similarly. Now any linear expression $e \bowtie b$, $\bowtie \in \{<, \leq, =, \neq, \geq, >\}$, $b \in \mathbb{Z}$, is satisfied in $\mathcal{S}$ if and only if the state formula $\langle k+1 \rangle \bowtie b$ is satisfied in $\mathcal{T}$ for $\bowtie \in \{\leq, \geq\}$. The relations $<$, $=$, $\neq$ and $>$ can be modelled using only $\leq$ and $\geq$. For $e \bowtie b$, where $\bowtie \in \{<, >\}$ we instead use $e \leq b-1$ and $e \geq b+1$, respectively. For $\bowtie$, where $\bowtie \in \{=, \neq\}$ we use $e \leq b \wedge e \geq b$ and $e < b \wedge e > b$, respectively. ■

We now prove that the disjunction-free EG fragment is decidable, by showing a reduction to the so-called *multi-weighted energy games* [14].

Energy games are played on k-weighted state-based games with weights from $\mathbb{Z}^k$, where the objective is to find a strategy for the existential player such that no matter what the universal player does, all infinite runs maintain the accumulated weights nonnegative. For a formal definition of energy games along with a proof of the theorem below we refer to the full version of the paper.

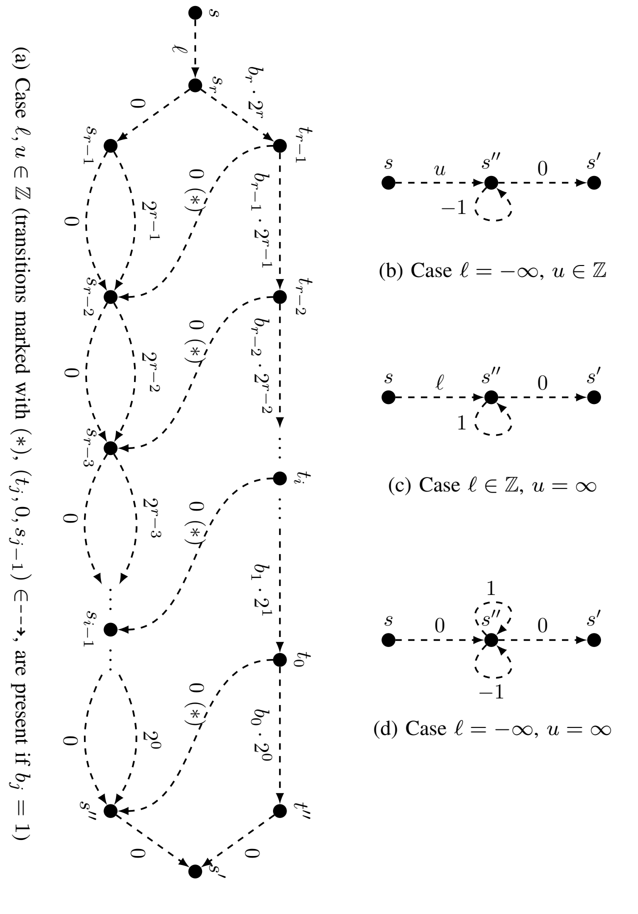
Theorem 6: Model checking the disjunction-free EG fragment is polynomial time equivalent to deciding the winner of energy games with lower bound.

Determining the winner of an energy game with only lower bounds is decidable and EXPSPACE-hard [14]. This yields the following corollary.

Corollary 1: Model checking the disjunction-free EG fragment on multiweighted modal automata is decidable and EXPSPACE-hard.

Another complexity result is obtained by reducing the model checking problem of the AG fragment to energy games. This time to the universal version, where all runs must maintain the accumulated weights nonnegative at all times, giving us a polynomial time algorithm. The details of the proof can be found in the full version of the paper.

Theorem 7: Model checking the disjunction-free AG fragment is in P.



(a) Case $\ell, u \in \mathbb{Z}$ (transitions marked with $(*)$, $(t_i, 0, s_{i-1}) \in \dashrightarrow$, are present if $b_i = 1$)

(b) Case $\ell = -\infty$, $u \in \mathbb{Z}$

(c) Case $\ell \in \mathbb{Z}$, $u = \infty$

(d) Case $\ell = -\infty$, $u = \infty$

Figure 4. Translation of $(s, ([\ell, u]), s') \in \dashrightarrow$ into singleton weighted transitions

Let us now prove the decidability of the full AG fragment.

C. *Decidability of* AG

In order to prove that the AG fragment of $\mathcal{L}$ is decidable, we need the following lemma, stating that the weight intervals can be reduced to singleton intervals when model checking a formula in the EG or AG fragment.

Lemma 2: Let $\mathcal{S} \in \mathbb{M}$ be a k-weighted modal automaton and ψ be a formula from the EG or AG fragment. We can in polynomial time construct a k-weighted modal automaton $\mathcal{T} \in \mathbb{M}$ with singleton intervals such that

$$\mathcal{S} \models \psi \quad \text{iff} \quad \mathcal{T} \models \psi.$$

Proof: In order to bypass the exponential blow up that the straightforward reduction of a transition $s \overset{[\ell,u]}{\dashrightarrow} s'$, $\ell, u \in \mathbb{Z}$, (even in the case of $k = 1$) to $u - \ell$ different singleton weighted transitions from s to s' would give, we instead use the following construction. Each transition $s \overset{\overline{W}}{\dashrightarrow} s'$ in $\mathcal{S}$ is translated into a number of transitions in $\mathcal{T}$. For each coordinate i of $\overline{W}$ one of the gadgets depicted in Figure 4 is chosen. Notice that the figure is shown for $k = 1$. Otherwise zeros are put into coordinates different from i. In case $s \xrightarrow{\overline{W}} s'$ in $\mathcal{S}$ the transition from s to s'' in each gadget must be a must-transition as well. The k selected gadgets (Figure 4a-4d) for each coordinate are then connected to each other in series in any order. What gadget to use depends on the specific interval, and we thus distinguish between the following three cases.

Both bounds are integers: In this case coordinate i of $\overline{W}$ is bounded by the interval $[\ell, u]$, $\ell, u \in \mathbb{Z}$. We can assume that the size of the interval, $u - \ell$, is written in binary using $r + 1$ bits as $b_r b_{r-1} \dots b_1 b_0$ such that $b_r = 1$. The corresponding gadget is given in Figure 4a. We add the lower bound of the interval first ($s \overset{\ell}{\dashrightarrow} s_k$), and using the remaining construction we let the may-player construct a value $j \in \{0, \dots, u - \ell\}$, simulating the game semantics of ψ, where the may-player chooses a weight in each weight interval. To see this we observe that taking the path $s_r \overset{b_r \cdot 2^r}{\dashrightarrow} t_{r-1} \overset{b_{r-1}\cdot 2^{r-1}}{\dashrightarrow} t_{r-2} \overset{b_{r-2}\cdot 2^{r-2}}{\dashrightarrow} \dots \overset{b_1 \cdot 2^1}{\dashrightarrow} t_0 \overset{b_0 \cdot 2^0}{\dashrightarrow} t'' \overset{0}{\dashrightarrow} s'$ (the uppermost path) corresponds to the may-player constructing $u - \ell$ and thus picking the weight u from $\overline{W}$. Taking the lowermost path along transitions with weight 0 corresponds to picking the weight ℓ, while any other path from s to s' builds a weight between ℓ and u. Notice that the may-player cannot construct a weight larger than $u - \ell$, since he only moves to the lowermost part of the figure (where he can construct any number for the remaining bits) if he chooses 0 for any of the bits in $u - \ell$ set to 1.

One bound is an integer, one bound is not: In the second case, where coordinate i of $\overline{W}$ is bounded below by $-\infty$ or above by ∞ (but not both) we use either the gadget in Figure 4b or the gadget in Figure 4c. We start by adding the upper bound (in case the lower bound is $-\infty$) or the lower bound (in case the upper bound is ∞) and then decrease (or increase) the first coordinate by an arbitrary number.

Both bounds are not integers: The last case, where the interval equals the set $\mathbb{Z}$, we use the gadget depicted in Figure 4d in order to decrease or increase the weight arbitrarily.

Now $\mathcal{S} \models \psi$ if and only if $\mathcal{T} \models \psi$. This is true since the may-player cannot choose the weights differently in $\mathcal{T}$ than he could have done in $\mathcal{S}$. The may-player can however choose to stop anywhere inside the gadget, but this cannot break the satisfiability of ψ due to the G quantifier. ■

Let us now prove the decidability of the AG fragment.

Theorem 8: The model checking problem of the AG fragment of $\mathcal{L}$ is decidable.

Proof: Let $\mathcal{S} \in \mathbb{M}$ and let AGφ be a formula from the AG fragment. By Lemma 2 we can reduce $\mathcal{S}$ to a weighted modal automaton with only singleton intervals. Furthermore we may consider only the *may-projection* of $\mathcal{S}$ obtained by turning all may-transitions into must-transitions, thus obtaining the most permissible implementation. Now φ must be satisfied in every reachable configuration. Since $\mathcal{S}$ is an implementation, $\mathsf{AG}\varphi$ corresponds to checking the negation of $\mathsf{EF}\neg\varphi$ and hence it is decidable by Theorem 5. ■

V. Conclusion

We studied multiweighted modal automata and proposed a fragment of CTL in order to reason about the accumulated weights gathered along the execution of the system, a practically motivated problem where one of its particular instances called energy games has recently become an active research topic. The semantics of the logic is given in terms of two-player games and the definitions were justified by showing that the fragment is both sound and, contrary to the previous attempts, also complete. We believe that a game-semantics is necessary for achieving the completeness of the logic, and the paper takes a first step in this direction. We showed that the logic is in general undecidable, but there are reasonable fragments that are practically interesting and remain decidable.

There are various directions for future works. Clearly, larger fragments of CTL (or generally of the μ-calculus), should be identified for which both soundness and completeness can be obtained. The corresponding AU/EU and AR/ER fragments of CTL may also be investigated. As the model checking problem is in general undecidable, a possible way to attack the problem can be to extend our framework to a three-valued formalism with refinement, like in the spirit of [12]. Another direction for future work is to extend the results to a more general setting using lattices, as in [23].

References

[1] K. Larsen and B. Thomsen, "A modal process logic," in *Proc. of LICS'88*. IEEE Computer Society, 1988, pp. 203–210.

[2] K. Larsen, "Modal specifications," in *Automatic Verification Methods for Finite State Systems*, ser. Lecture Notes in Computer Science, J. Sifakis, Ed., vol. 407. Springer, 1989.

[3] J.-B. Raclet, E. Badouel, A. Benveniste, B. Caillaud, A. Legay, and R. Passerone, "A modal interface theory for component-based design," *Fundamenta Informaticae*, vol. 108, no. 1-2, pp. 119–149, 2011.

[4] N. Beneš, J. Křetínský, K. Larsen, and J. Srba, "Checking thorough refinement on modal transition systems is EXPTIME-complete," in *Proc. of ICTAC'09*, ser. LNCS, vol. 5684. Springer, 2009, pp. 112–126.

[5] N. Beneš, J. Křetínský, K. Larsen, and J. Srba, "On determinism in modal transition systems," *Theor. Comput. Sci.*, vol. 410, no. 41, pp. 4026–4043, 2009.

[6] K. Larsen, U. Nyman, and A. Wasowski, "On modal refinement and consistency," in *Proc. of CONCUR'07*, ser. LNCS, vol. 4703. Springer, 2007, pp. 105–119.

[7] M. Huth, R. Jagadeesan, and D. Schmidt, "Modal transition systems: A foundation for three-valued program analysis," in *Proc. of ESOP'01*, ser. LNCS, vol. 2028. Springer, 2001, pp. 155–169.

[8] G. Bruns and P. Godefroid, "Generalized model checking: Reasoning about partial state spaces," in *Proc. of CONCUR'00*, ser. LNCS, vol. 1877. Springer, 2000, pp. 168–182.

[9] P. Godefroid, M. Huth, and R. Jagadeesan, "Abstraction-based model checking using modal transition systems," in *Proc. of CONCUR'01*, ser. LNCS, vol. 2154. Springer, 2001, pp. 426–440.

[10] A. Gurfinkel and M. Chechik, "How thorough is thorough enough?" in *Proc. of CHARME'05*, ser. LNCS, vol. 3725. Springer, 2005, pp. 65–80.

[11] O. Wei, A. Gurfinkel, and M. Chechik, "Mixed transition systems revisited," in *Proc. of VMCAI'09*, ser. LNCS, vol. 5403. Springer, 2009, pp. 349–365.

[12] P. Godefroid and N. Piterman, "LTL Generalized Model Checking Revisited," in *Proc. of VMCAI'09*, ser. LNCS, vol. 5403. Springer, 2009, pp. 89–104.

[13] N. Beneš, I. Černá, and J. Křetínský, "Disjunctive Modal Transition Systems and Generalized LTL Model Checking," Faculty of Informatics, Masaryk University, Tech. Rep. FIMU-RS-2010-12, 2010.

[14] U. Fahrenberg, L. Juhl, K. Larsen, and J. Srba, "Energy games in multiweighted automata," in *Proc. of ICTAC'11*, ser. LNCS, vol. 6916. Springer, 2011, pp. 95–115.

[15] L. Juhl, K. Larsen, and J. Srba, "Modal transition systems with weight intervals," *Journal of Logic and Algebraic Programming*, 2012, in print.

[16] M. Droste and P. Gastin, "Weighted automata and weighted logics," *Theor. Comput. Sci.*, vol. 380, no. 1-2, pp. 69–86, 2007.

[17] M. Droste, W. Kuich, and H. Vogler, Eds., *Handbook of Weighted Automata*, 1st ed. Springer-Verlag, 2009.

[18] M. Droste and I. Meinecke, "Describing average- and longtime-behavior by weighted MSO logics," in *Proc. of MFCS'10*, ser. LNCS, vol. 6281. Springer, 2010, pp. 537–548.

[19] B. Bollig and P. Gastin, "Weighted versus probabilistic logics," in *Proc. of DLT'09*, ser. LNCS, vol. 5583. Springer, 2009, pp. 18–38.

[20] A. Kučera and O. Stražovský, "On the controller synthesis for finite-state markov decision processes," *Fundam. Inform.*, vol. 82, no. 1-2, pp. 141–153, 2008.

[21] U. Boker, K. Chatterjee, T. Henzinger, and O. Kupferman, "Temporal specifications with accumulative values," in *Proc. of LICS'11*. IEEE Computer Society, 2011, pp. 43–52.

[22] M. Minsky, *Computation: Finite and Infinite Machines*. Prentice-Hall, 1967.

[23] O. Kupferman and Y. Lustig, "Lattice automata," in *Proc. of VMCAI'07*, ser. LNCS, vol. 4349. Springer, 2007, pp. 199–213.

[24] P. Bouyer, U. Fahrenberg, K. Larsen, N. Markey, and J. Srba, "Infinite runs in weighted timed automata with energy constraints," in *Proc. of FORMATS'08*, ser. LNCS, vol. 5215. Springer, 2008, pp. 33–47.

Appendix

The appendix contains proofs that are omitted in the main part of the paper due to space limitations.

Remark. We note that although we have omitted actions from the definition, this is not a real restriction: given a finite set of actions $\Sigma = \{a_1, \ldots, a_n\}$ and a k-weighted modal automaton, we can encode the actions in a $(k+n)$-weighted modal automaton by introducing for every i-th action $a_i \in \Sigma$, $1 \leq i \leq n$, a new weight coordinate $k+i$ which equals 1 iff the transition is labelled by action a, and 0 otherwise.

Theorem 1. Let $\mathcal{S} \in \mathbb{M}$ be a k-weighted modal automaton and ψ be an $\mathcal{L}$-formula. Then

$$\mathcal{S} \models \psi \quad \Longrightarrow \quad \left(\forall\, \mathcal{T} \in \mathbb{M} : \mathcal{T} \leq_m \mathcal{S} \implies \mathcal{T} \models \psi\right) .$$

Proof: It is clear that the theorem holds for any formula φ which is a logical combination of atomic propositions, because it only refers to the current accumulated weight (which is $\overline{0}$ in the initial configuration). Let φ be a logical combination of atomic propositions, and let ψ be a formula of the form $\mathsf{EF}\varphi$, $\mathsf{EG}\varphi$, $\mathsf{AF}\varphi$, $\mathsf{AG}\varphi$. Let $\mathcal{T} \in \mathbb{M}$ be a k-weighted modal automaton such that $\mathcal{T} \leq_m \mathcal{S}$.

For the case $\psi \in \{\mathsf{AG}\varphi, \mathsf{AF}\varphi\}$, consider a run $\gamma_{\mathcal{T}}$ in $\mathcal{T}$. It is clear that from the modal refinement $\mathcal{T} \leq_m \mathcal{S}$ it follows that there exists a run $\gamma_{\mathcal{S}}$ in $\mathcal{S}$ such that both runs have the same length, and for all i, the state of $(\gamma_{\mathcal{T}})_i$ is in a modal refinement relation with $(\gamma_{\mathcal{S}})_i$, i.e. $\mathcal{T}_{(\gamma_{\mathcal{T}})_i} \leq_m \mathcal{S}_{(\gamma_{\mathcal{S}})_i}$, and the accumulated weights of $(\gamma_{\mathcal{T}})_i$ and $(\gamma_{\mathcal{S}})_i$ coincide. Hence $\mathcal{T} \models \psi$.

Consider now the case $\psi \in \{\mathsf{EG}\varphi, \mathsf{EF}\varphi\}$. By assumption we know that there is a strategy σ for the must-player that witnesses $\mathcal{S} \models \psi$. We can iteratively construct a strategy σ' for the must-player on $\mathcal{T}$ as follows. Assume a configuration $(t, \overline{w})$ with a must-state t that is related (the states are related by modal refinement and the weights are the same) to a configuration $(s, \overline{w})$ of $\mathcal{S}$ such that there is a strategy for the must-player witnessing $\mathcal{S}_{(s,\overline{w})} \models \psi$. So for $(s, \overline{w})$ there exists a choice of the next must-transition such that for any choice of the weight of the may-player, in the next configuration there is again a strategy for the must-player which witnesses the satisfaction of ψ. This must-transition is also present in $\mathcal{T}$, due to modal refinement, hence the choice of the must-transition by σ can be simulated by the strategy σ' in $\mathcal{T}$. Finally, note that the may-player in $\mathcal{T}$ has at most the choices (for weights and may-transitions) as there are in $\mathcal{S}$. Hence $\mathcal{T} \models \psi$. ■

Definition of an energy game: A k-weighted energy game is a four tuple $G = (S_1, S_2, s_0, \longrightarrow)$ where S_1 and S_2 are finite disjoint sets of existential and universal states, respectively, $s_0 \in S_1 \cup S_2$ is the start state and $\longrightarrow \subseteq (S_1 \cup S_2) \times \mathbb{Z}^k \times (S_1 \cup S_2)$ is a finite multiweighted transition relation. Furthermore a k-weighted game is non-blocking, meaning that all states have some outgoing transition.

Configurations, plays and strategies are defined similarly to the same terms on k-weighted modal automata. A configuration is a pair $(s, \overline{w})$, where $s \in S_1 \cup S_2$ and $\overline{w} \in \mathbb{Z}^k$, while a play is an infinite sequence of configurations $(s_0, \overline{w}_0)(s_1, \overline{w}_1)\ldots$ such that $(s_i, \overline{v}_i, s_{i+1}) \in \longrightarrow$ and $\overline{w}_i + \overline{v}_i = \overline{w}_{i+1}$ for all $i \geq 0$. A strategy for the existential player is a mapping σ from each finite prefix of a play $(s_0, \overline{w}_0)\ldots(s_n, \overline{w}_n)$ such that $s_n \in S_1$ to a configuration $(s_{n+1}, \overline{w}_{n+1})$ such that $(s_0, \overline{w}_0)\ldots(s_n, \overline{w}_n)(s_{n+1}, \overline{w}_{n+1})$ is a prefix of a play in G. Given a strategy σ, the set of all plays in G of the form $(s_0, \overline{w}_0)(s_1, \overline{w}_1)\ldots$, where $\overline{w}_0 = \overline{0}$ and $\sigma((s_0, \overline{w}_0)\ldots(s_n, \overline{w}_n)) = (s_{n+1}, \overline{w}_{n+1})$ for all $s_n \in S_1$ is called $plays(\sigma, G)$. Given a k-weighted game G and a $\overline{b} \in \mathbb{N}^k$, the energy game with upper bound asks whether there exists a strategy σ for the existential player such that all $(s_0, \overline{w}_0), (s_1, \overline{w}_1)\ldots \in plays(\sigma, G)$ satisfy $\overline{0} \leq \overline{w}_i \leq \overline{b}$ for all i. If we have only the requirement of $\overline{0} \leq \overline{w}_i$ for all i, we call it an energy game with lower bound.

Theorem 6. Model checking the disjunction-free EG fragment is polynomial time equivalent to deciding the winner of energy games with lower bound.

Proof: Let $\mathcal{S} = (S, s_0, \overline{w}, \dashrightarrow, \longrightarrow)$ be a k-weighted modal automaton. We will first reduce the model checking problem of EG on $\mathcal{S}$ to an energy game with only lower bounds.

Due to Lemma 2 we can assume that $\mathcal{S}$ is a weighted modal automaton with singleton intervals. This ensures that we can apply Lemma 1 and assume that all linear expressions in φ are simple. First, we notice that any such φ can be rewritten to an equivalent form

$$\varphi = \left(\bigwedge_{i=1}^{k} \langle i\rangle \geq \ell_i\right) \wedge \left(\bigwedge_{i=1}^{k} \langle i\rangle \leq u_i\right) , \tag{1}$$

where $\ell_i, u_i \in \mathbb{Z} \cup \{-\infty, \infty\}$ and $\ell_i = -\infty$ implies $u_i = \infty$. This follows since a coordinate with no upper bound can safely be bounded above by ∞ and a coordinate with no lower bound can safely be bounded below by $-\infty$. In addition a coordinate with an upper bound and no lower bound can be simulated using only a lower bound by multiplying all weights on transitions and the bound in this coordinate by -1. It is clear that there exits a unique ℓ_i giving the largest lower bound and a unique u_i giving the smallest upper bound for each $i \in \{1, \ldots, k\}$ should there be multiple constrains related to the coordinate i.

We shall now reduce the model checking problem for $\mathsf{EG}\varphi$ to a k-weighted energy game. The game $G_{\mathcal{S}} = (S_1, S_2, s_G, \longrightarrow_G)$ is constructed from $\mathcal{S}$ by splitting S into two sets, such that $s_G = s_0$, $S_1 = \{s \in S \mid$

$s \longrightarrow s'$ for some $s' \in S\}$ (the existential states) and $S_2 = S \setminus S_1 \cup \{s'\}$ (the universal states). The state s' is added to S_2 in order to capture the fact that any may-transition can be dropped. In any universal state we therefore add a transition to s' with $\overline{0}$ as weight. Furthermore s' has a selfloop also with $\overline{0}$ as weight vector in order to ensure a non-blocking game. Hence we define

$$\longrightarrow_G = \longrightarrow \cup \dashrightarrow \cup \{s \xrightarrow{\overline{0}}_G s' \mid s \in S_2\} \cup \{s' \xrightarrow{\overline{0}}_G s'\}.$$

Now $\mathcal{S}$ satisfies $\mathsf{EG}\varphi$ iff there exists a strategy σ for the existential player in $G_{\mathcal{S}}$ such that any infinite play that proceeds according to σ and starts in s_G with initial weight $(\overline{w}[1]-\ell_1, \ldots, \overline{w}[k]-\ell_k)$ has accumulated weights which are always nonnegative and do not exceed $(u_1-\ell_1, \ldots, u_k-\ell_k)$. As proved in [14] it is possible to remove the upper bounds by doubling the number of weights, hence we get an equivalent instance of the lower bound energy game with $2k$ weights.

To conclude the other direction of the proof, we can realize that any k-weighted energy game $G = (S_1, S_2, s_G, \longrightarrow)$ with only lower bounds can be reduced to a modal automaton $\mathcal{S}$ with singleton intervals by turning all transitions $s \xrightarrow{\overline{w}} s'$, where $s \in S_2$ into only may-transitions. Clearly, the existential player wins the energy game with initial weight $\overline{w}_0$ iff $\mathcal{S} \models \mathsf{EG}(\langle 1 \rangle \geq 0 \wedge \ldots \wedge \langle k \rangle \geq 0)$. ■

Theorem 7. Model checking the disjunction-free AG fragment is decidable in P.

Proof: By using the same reductions as in the proof of Theorem 6, we find an equivalence with the universal energy game with only lower bounds. The only modification is that after constructing the game $G_{\mathcal{S}} = (S_1, S_2, s_G, \longrightarrow_G$) we make another game $G_{\mathcal{S}'} = (\emptyset, S_1 \cup S_2, s_G, \longrightarrow_G)$, such that all states belong to the universal player. Deciding a universal energy game (regardless of the bounds) can be done in polynomial time [24]. ■