COUNTING CTL

FRANÇOIS LAROUSSINIE, ANTOINE MEYER, AND EUDES PETONNET

LIAFA, Université Paris Diderot – Paris 7 & CNRS UMR 7089, France
e-mail address: Francois.Laroussinie@liafa.jussieu.fr

LIGM, Université Paris Est – Marne-la-Vallée & CNRS UMR 8049, France
e-mail address: Antoine.Meyer@univ-mlv.fr

LIAFA, Université Paris Diderot – Paris 7 & CNRS UMR 7089, France
e-mail address: Eudes.Petonnet@liafa.jussieu.fr

Abstract. This paper presents a range of quantitative extensions for the temporal logic CTL. We enhance temporal modalities with the ability to constrain the number of states satisfying certain sub-formulas along paths. By selecting the combinations of Boolean and arithmetic operations allowed in constraints, one obtains several distinct logics generalizing CTL. We provide a thorough analysis of their expressiveness and succinctness, and of the complexity of their model-checking and satisfiability problems (ranging from P-complete to undecidable). Finally, we present two alternative logics with similar features and provide a comparative study of the properties of both variants.

1. Introduction

Among the existing approaches to the formal verification of automated systems, model checking [CE81, QS82] aims at automatically establishing the validity of a certain formal specification (modeled as a formula in a suitable logic) over the system under study (modeled for instance as a finite transition system). This set of techniques is now well established and successful, with several industrial applications.

To formalize the specification of temporal properties, for instance in the case of reactive systems, temporal logics (TL) were proposed thirty years ago [Pnu77] and widely studied since. They are today used in many model-checking tools. There exists a wide variety of temporal logics, differing for instance by the models over which formulas are interpreted or by the kind of available temporal modalities. Two well-known examples are LTL in the linear-time framework (where formulas are interpreted over infinite runs) and CTL for the branching-time case (where formulas are interpreted over states of Kripke structures). See [Eme90] for a survey of classical temporal logics for systems specification.

1998 ACM Subject Classification: Categories and Subject Descriptors: F.4.1 [Mathematical Logic and Formal Languages]: Mathematical Logic—temporal logic; D.2.4 [Software Engineering]: Software/Program Verification—model checking. General Terms: Algorithms, Verification.

Key words and phrases: branching time, counting, constraints, satisfiability, complexity.

LOGICAL METHODS IN COMPUTER SCIENCE DOI:10.2168/LMCS-???

Temporal logics have been extended in various ways in order to increase their expressive power. For example, while LTL and CTL only handle future operators, it is also possible to consider past-time modalities to express properties of the past of a run. One can also extend temporal logics with regular expressions (see for instance [Wol83, ET97]). Other extensions were proposed to handle *quantitative* aspects of systems. For example, some logics can contain timing constraints to specify that an event, say P_1, has to occur less than 10 time units before another event P_2. This kind of temporal logics, such as TCTL [ACD93, EMSS92], have been particularly studied in the framework of timed model checking. Another quantitative extension consists in *probabilistic* logics where one can specify probability bounds over the truth of some property (see for instance [BdA95]).

We propose several extensions of CTL with constraints over the number of states satisfying certain sub-formulas along runs. For example, considering a model for an ATM, we can express the property "whenever the PIN is locked, at least three erroneous attempts have been made" by: $\neg\mathsf{EF}_{[\sharp\mathtt{error}\leq 2]}\mathtt{lock}$ (one cannot reach a state where the PIN is locked but less than two errors have occurred). Similarly, $\neg\mathsf{EF}_{[\sharp\mathtt{error}\geq 3]}\mathtt{money}$ states that three mistakes forbid cash retrieval. We put a subscript on the temporal modality (as in TCTL) to constrain the runs over which the modality holds. Note that most properties of this kind can also be expressed in CTL by nesting E_U_ modalities, but the resulting formulas may be too large to be conveniently handled by the user of a model checker. This is discussed in more detail in Section 3, where we study the expressiveness of each of our fragments compared to CTL. In some cases, there exist natural translations into equivalent CTL formulas, implying that there is no strict gain in expressiveness. However, these translations are often at best *exponentially larger* than the original formula. In other cases, we show that our extensions strictly increases the expressive power of CTL.

We consider the model checking problem for various sets $\mathcal{C}$ of constraints. We show that polynomial-time algorithms exist when considering Until modalities with constraints of the form[1] $(\sum_i \sharp\varphi_i) \sim c$ with $\sim \in \{<, \leq, =, \geq, >\}$ and $c \in \mathbb{N}$. Additionally allowing Boolean combinations of such constraints or integer coefficients in the sum (or both) makes model checking Δ_2^{P}-complete. We also consider the case of "diagonal" constraints $(\sharp\varphi - \sharp\psi) \sim c$ and their more general form $(\sum_i \pm\sharp\varphi_i) \sim c$ with $c \in \mathbb{Z}$ and show that model checking can still be done in polynomial time. However, allowing Boolean combinations of such constraints leads to undecidability. We also investigate the complexity of the satisfiability problem, which is 2-EXPTIME-complete for all fragments without subtraction and undecidable otherwise. Finally, in order to investigate alternative definitions of counting logics generalizing CTL, we define another semantics for our logics (called cumulative semantics) and a logic with explicit variables. In both cases, we show that it induces a complexity blow-up for model checking, which becomes PSPACE-complete without subtraction and undecidable otherwise. The asymptotic complexity of satisfiability remains however 2-EXPTIME-complete in all decidable cases.

Several existing works provide related results. In [LMP10a], we presented a preliminary version of the current article. Proofs and constructions were since considerably refined, and are provided here in greater detail. This paper also provides new satisfiability results. In [LMP10b], we provided a similar study of counting extensions of LTL and CTL*. In [ET97], an extension of LTL with a kind of regular expressions containing quantitative constraints over the number of occurrences of sub-expressions is presented. This extension

[1]Unless stated otherwise, complexity results always assume a binary encoding of constants.

yields algorithms whose time complexity is exponential in the size of formulas and the *value* of integer constants. In [ET99], extensions of CTL including parameters in constraints are defined. One of these formalisms, namely GPCTL, allows one to express properties with constraints defined as positive Boolean combinations of sums of the form $\sum_i P_i \leq c$ where every P_i is an atomic proposition. Model-checking E_U_ formulas with such a constraint is shown to be NP-complete and a polynomial algorithm is given for a restricted logic (with parameters). Another interesting specification language is Sugar/PSL [psl03], which defines many additional operators above LTL and CTL*. These include in particular a kind of counting constraints used together with regular expressions, but to our knowledge, there is no accurate study of lower complexity bounds for these extensions [BFH05]. In [YMW97], a branching-time temporal logic with general counting constraints (using a variant of freeze variables) is defined to specify event-driven real-time systems. To obtain decidability, the authors restrict their analysis to systems verifying some bounded progress condition. In [BEH95a, BEH95b], extensions of LTL and CTL with Presburger constraints over the number of states satisfying a formula are considered, for a class of infinite state processes. The complexity of these problems is much higher than the cases we are concerned with. Finally there also exist timed extensions of CTL interpreted over Kripke structures (see for instance [EMSS92]).

The paper is organized as follows. In Section 2, we introduce the definitions of the main formalisms we will use. In Section 3, we show that several of our proposed extensions are not more expressive than classical CTL, yet exponentially more succinct. In Section 4, we address the model-checking problem and provide exact complexity results for almost all the logics we introduce. In Section 5 we study the complexity of the satisfiability problem. Finally we present in Section 6 a different logic with explicit counting variables, as well as an alternative semantics for our logics, together with the complexity of the related model-checking problems.

2. Definitions

2.1. **Models.** Let AP be a set of atomic propositions. In branching-time temporal logics, formulas are generally interpreted over states of Kripke structures.

Definition 2.1. A *Kripke structure* (or KS) $\mathcal{S}$ is a tuple $\langle Q, R, \ell \rangle$ where Q is a finite set of states, $R \subseteq Q \times Q$ is a total[2] transition relation and $\ell : Q \to 2^{\mathsf{AP}}$ is a labelling of states with atomic propositions.

A run ρ of $\mathcal{S}$ is an infinite sequence of states $q_0 q_1 q_2 \ldots$ such that $(q_i, q_{i+1}) \in R$ for every i. We use $\rho(i)$ to denote state q_i, $\rho_{|_i}$ to denote the prefix $q_0 \cdots q_i$ of ρ, and ϵ to represent the empty prefix. Notice that $\rho_{|_{-1}} = \epsilon$, but $\rho_{|_0} = q_0 \neq \epsilon$. Runs$(q)$ denotes the set of runs starting from some state $q \in Q$ and Runs$(\mathcal{S})$ (resp. Prefs$(\mathcal{S})$) the set of all runs (resp. finite prefixes of runs) in $\mathcal{S}$. The length $|\sigma|$ of a finite run prefix σ is defined as usual (i.e. $|\sigma| = 0$ if $\sigma = \epsilon$ and $|\sigma| = i + 1$ if $\sigma = q_0 \ldots q_i$). Note in particular that for any run ρ, $|\rho_{|_i}| = i + 1$. We write $\sigma \leq \rho$ when σ is a prefix of ρ.

We will also consider *durational Kripke structures* (DKS), where an integer duration is associated with every transition. A DKS $\mathcal{S} = \langle Q, R, \ell \rangle$ is defined similarly to a KS, except that $R \subseteq Q \times \mathbb{Z} \times Q$. The duration of a transition is also called a *weight* or a *cost*, especially

[2]By *total* relation, we mean a relation $R \subseteq Q \times Q$ such that $\forall p \in Q, \exists q \in Q, (p, q) \in R$.

when negative values are used to label a transition. We use DKS^{1} to denote the class of DKS in which every weight is 1, $\mathrm{DKS}^{0/1}$ when the weights belong to $\{0,1\}$, and $\mathrm{DKS}^{-1/0/1}$ when they belong to $\{-1,0,1\}$. The notion of weight is additively extended to finite runs of DKS. The existence of a transition of weight k between states p and q is sometimes denoted as $p \xrightarrow[R]{k} q$, that of a finite run of weight k as $p \xRightarrow[R]{k} q$. R may be omitted when it is clear from the context. The weight of a finite run ρ is also denoted as $\|\rho\|$.

2.2. **Counting CTL.** We define several extensions of CTL able to express constraints over the number of times certain sub-formulas are satisfied along a run.

Definition 2.2. Given a set of atomic propositions AP, we define the logic CCTL as the set of formulas

$$\varphi, \psi \ ::= \ P \mid \varphi \wedge \psi \mid \neg\varphi \mid \mathsf{E}\varphi\mathsf{U}_{[C]}\psi \mid \mathsf{A}\varphi\mathsf{U}_{[C]}\psi$$

where $P \in \mathsf{AP}$ and C is a constraint of the form

$$C \ ::= \ (\sum_{i=1}^{m} \alpha_i \cdot \sharp\varphi_i) \sim k$$

where $\varphi_i \in \mathsf{CCTL}$, $\alpha_i, k \in \mathbb{N}$ and $\sim \in \{<, \leq, =, \geq, >\}$.

We make use of the standard abbreviations $\vee, \Rightarrow, \Leftrightarrow, \bot, \top$, as well as the additional modalities $\mathsf{EF}_{[C]}\varphi = \mathsf{E}\top\mathsf{U}_{[C]}\varphi$, $\mathsf{AF}_{[C]}\varphi = \mathsf{A}\top\mathsf{U}_{[C]}\varphi$, and their duals $\mathsf{AG}_{[C]}\varphi = \neg\mathsf{EF}_{[C]}\neg\varphi$ and $\mathsf{EG}_{[C]}\varphi = \neg\mathsf{AF}_{[C]}\neg\varphi$. Any formula occurring in a constraint C associated with a modality in Φ is considered a sub-formula of Φ. The size $|\Phi|$ of Φ thus takes into account the size of these constraints and their sub-formulas, assuming that integer constants are encoded in *binary* (unless explicitly stated otherwise). The DAG-size of Φ is the number of distinct sub-formulas of Φ. As model-checking algorithms may be implemented in such a way that the truth value of each sub-formula is computed only once, for instance using dynamic programming, this is generally more relevant to the complexity of model-checking.

We also introduce several variants and extensions of CCTL:

- CCTL_1 is the restriction of CCTL where every coefficient α_i occurring in the constraints is equal to 1. Thus the constraints are of the form $(\sum_i \sharp\varphi_i) \sim k$. For example, $\mathsf{EF}_{[\sharp P + \sharp P' = 10]}P''$ belongs to CCTL_1.
- $\mathsf{CCTL}_\pm$ is an extension of CCTL where coefficients α_i are in $\mathbb{Z}$. The formula $\mathsf{EF}_{[\sharp P - 3\cdot\sharp P' = 10]}P''$ belongs to $\mathsf{CCTL}_\pm$.
- $\mathsf{CCTL}_\wedge$ extends CCTL by allowing Boolean combinations in the constraints. For example, $\mathsf{EF}_{[\sharp P < 4 \wedge \sharp P' > 8]}$ is in $\mathsf{CCTL}_\wedge$.

We can combine the previous variants and define the logics $\mathsf{CCTL}_{\pm 1}$, $\mathsf{CCTL}_{\wedge 1}$, $\mathsf{CCTL}_{\wedge\pm}$ and $\mathsf{CCTL}_{\wedge\pm 1}$. The semantics of our logics are defined over Kripke structures as follows:

Definition 2.3. The following clauses define the conditions for a state q of some KS $\mathcal{S} = \langle Q, R, \ell\rangle$ to satisfy a formula φ (written $q \models_{\mathcal{S}} \varphi$) by induction over the structure of φ :

$$\begin{array}{lll}
q \models_{\mathcal{S}} P & \text{iff} & P \in \ell(q) \\
q \models_{\mathcal{S}} \neg\varphi & \text{iff} & q \not\models_{\mathcal{S}} \varphi \\
q \models_{\mathcal{S}} \varphi \vee \psi & \text{iff} & q \models_{\mathcal{S}} \varphi \text{ or } q \models_{\mathcal{S}} \psi \\
q \models_{\mathcal{S}} \mathsf{E}\varphi\mathsf{U}_{[C]}\psi & \text{iff} & \exists \rho \in \mathrm{Runs}(q), \rho \models_{\mathcal{S}} \varphi\mathsf{U}_{[C]}\psi \\
q \models_{\mathcal{S}} \mathsf{A}\varphi\mathsf{U}_{[C]}\psi & \text{iff} & \forall \rho \in \mathrm{Runs}(q), \rho \models_{\mathcal{S}} \varphi\mathsf{U}_{[C]}\psi
\end{array}$$

where $\rho \models_{\mathcal{S}} \varphi \mathsf{U}_{[C]} \psi$ iff $\exists i \geq 0,\ \rho(i) \models_{\mathcal{S}} \psi,\ \rho_{|i-1} \models_{\mathcal{S}} C$ and $\forall 0 \leq j < i,\ \rho(j) \models_{\mathcal{S}} \varphi$.

For every finite run prefix $\sigma = q_0 \dots q_i$, the meaning of $\sigma \models_{\mathcal{S}} C$ is based on the interpretation of $\sharp\varphi$ over σ, which is the number of states among $q_0, \dots, q_i$ verifying φ, denoted by $|\sigma|_\varphi$ and defined as: $|\sigma|_\varphi = |\{j \mid 0 \leq j \leq i \wedge \sigma(j) \models_{\mathcal{S}} \varphi\}|$. Given these values, C is evaluated as an ordinary equation or inequation over integer expressions.

In the following we omit the subscript $\mathcal{S}$ for $\models$ when no confusion occurs. We use $\equiv$ to denote the standard equivalence between formulas.

Remark 2.4. It can be derived from the above definitions that formula $\mathsf{EF}_{[C]}\varphi$ holds from q if and only if there is a run ρ from q and an index i such that $\rho(i) \models \varphi$ and $\rho_{|i-1|} \models C$. Similarly, $\mathsf{EG}_{[C]}\varphi$ holds if and only if there exists a run ρ such that, whenever a finite prefix of ρ satisfies C, the next state must satisfy φ (in other words, for all $i \geq 0$, $\rho_{|i-1} \models C \implies \rho(i) \models \varphi$).

Remark 2.5. The above semantics imply that the truth value of a constraint only depends on the *strict* prefix of the run leading to (but not including) the current state. This is not an essential feature, and another definition would also be valid. However, this choice is consistent with the semantics of existing logics (in particular TCTL [ACD93]). It also allows us to express the classical X (or *next*) operator as $\mathsf{EX}\varphi = \mathsf{EF}_{[\sharp\top=1]}\varphi$. Moreover, under this semantics the formulas $\mathsf{E}\varphi\mathsf{U}\psi$, $\mathsf{E}\top\mathsf{U}_{[\sharp\neg\varphi=0]}\psi$ and $\mathsf{EF}_{[\sharp\neg\varphi=0]}\psi$ are all equivalent.

Remark 2.6. In all logics allowing Boolean connectives inside constraints, the modality F is sufficient to define U. Indeed, $\mathsf{E}\varphi\mathsf{U}_{[C]}\psi \equiv \mathsf{EF}_{[C\wedge\sharp(\neg\varphi)=0]}\psi$ (and similarly for A-quantified formulas). Thus every such logic can also be built from atomic propositions using Boolean operators and modalities $\mathsf{EF}_{[C]}\varphi$ and $\mathsf{AF}_{[C]}\varphi$ (or $\mathsf{EG}_{[C]}\varphi$). Note that all these translations are succinct (linear in the size of formulas) and thus do not have any impact on complexity results.

Remark 2.7. The related temporal logic TCTL, whose semantics is defined over *timed* models (in particular durational Kripke structures), allows one to label temporal modalities with duration constraints. For instance, one may write $\mathsf{A}\varphi\mathsf{U}_{<k}\psi$ to express the fact that φ is consistently true until, before k time units have elapsed, ψ eventually holds.

When all transitions in a DKS have duration 1 (i.e. the duration of any run is equal to its length), TCTL (or RTCTL in [EMSS92]) formulas can be directly expressed in any variant of CCTL using only the sub-formula $\top$ inside constraints. A similar coding is also possible when one uses a proposition *tick* to mark the elapse of time as in [LST03].

2.3. **Examples of CCTL formulas.** We now give several examples of natural quantitative properties that can be easily expressed with CCTL-like logics.

(1) First consider an engine or plant that has to be controlled every 10000 cycles. Suppose a warning is activated whenever the number of elapsed cycles since the last control belongs to the interval $[9900; 9950]$, and is maintained until the next control is done. Moreover, an alarm is raised when the number of cycles is above 10100 (unless a control was performed in-between) and is maintained until the next control. Such a specification could be expressed in CCTL as follows:
   (a) Either a control or a warning must occur in every period of 9950 cycles:

$$\mathsf{AG}\big(\mathsf{AF}_{[\sharp\mathtt{cycle}\leq 9950]}(\mathtt{control} \vee \mathtt{warning})\big)$$

where `cycle` (resp. `warning`, `control`) labels states corresponding to the end of a cycle (resp. a warning, a control action).

(b) A warning cannot occur before 9900 cycles after a control:

$$\mathsf{AG}\big(\texttt{control} \;\Rightarrow\; \neg\mathsf{EF}_{[\sharp\texttt{cycle}<9900]}\texttt{warning}\big).$$

(c) A control or an alarm occurs in every period of 10100 cycles:

$$\mathsf{AG}\big(\mathsf{AF}_{[\sharp\texttt{cycle}\leq 10100]}(\texttt{control} \vee \texttt{warning})\big).$$

(d) An alarm cannot occur strictly before 10100 cycles after a control:

$$\mathsf{AG}\big(\texttt{control} \;\Rightarrow\; \neg\mathsf{EF}_{[\sharp\texttt{cycle}<10100]}\texttt{alarm}\big).$$

(e) The warning and the alarm are maintained:

$$\mathsf{AG}\big(\texttt{warning} \;\Rightarrow\; \mathsf{A}\,\texttt{warning}\,\mathsf{U}\,(\texttt{alarm} \vee \texttt{control})\big)$$

and

$$\mathsf{AG}\big(\texttt{alarm} \;\Rightarrow\; \mathsf{A}\,\texttt{alarm}\,\mathsf{W}\,\texttt{control}\big).$$

Note that we use a *weak* Until modality in the latter formula because we cannot ensure the occurrence of a control.

(2) Consider a model for an ATM, whose atomic propositions include `money`, `reset` and `error`, with the obvious meaning. To specify that it is not possible to get money after three mistakes were made in the same session (*i.e.* with no intermediate reset), we can use the $\mathsf{CCTL}_{\wedge 1}$ formula

$$\mathsf{AG}\big(\neg\mathsf{EF}_{[\sharp\texttt{error}\geq 3\wedge\sharp\texttt{reset}=0]}\texttt{money}\big),$$

or the CCTL_1 formula

$$\mathsf{AG}\big(\neg\mathsf{E}(\neg\texttt{reset})\mathsf{U}_{[\sharp\texttt{error}\geq 3]}\texttt{money}\big).$$

(3) Consider a mutual exclusion algorithm with n processes trying to reach their critical section (CS). We can express a bounded waiting property with bound 10 (*i.e.* when a process P tries to reach its CS, then at most 10 other processes can reach theirs before P does) by the $\mathsf{CCTL}_{\wedge 1}$ formula

$$\mathsf{AG}\bigwedge_{i\in[1,n]}\big(\texttt{request}_i \Rightarrow \neg\mathsf{EF}_{[\sum_{j\neq i}\sharp\texttt{cs}_j>10\wedge\sharp\texttt{cs}_i=0]}\top\big).$$

As in the previous case, this can also be expressed in CCTL_1 using U instead of F.

(4) In a model for a communicating system with events for the emission and reception of messages, the $\mathsf{CCTL}_{\pm 1}$ formula $\mathsf{AG}_{[\sharp\texttt{send}-\sharp\texttt{receive}<0]}\bot$ states that along any finite run, the number of `receive` events cannot exceed the number of `send` events.

(5) Quantitative constraints can also be useful for fairness properties. For example the $\mathsf{CCTL}_{\wedge 1}$ formula $\mathsf{AG}\,\mathsf{AF}_{[\bigwedge_i 5\leq\sharp\varphi_i\leq 10]}\top$ states that each φ_i occurs infinitely often along every run (as does the CTL formula $\bigwedge_i(\mathsf{AG}\,\mathsf{AF}\,\varphi_i)$) but also ensures some constraint on the number of states satisfying formulas φ_i along every execution: for example, it is not possible to have a sub-run where φ_1 holds in 11 states and φ_2 in only 4 states.

(6) Note that $\mathsf{CCTL}_{\pm}$ can express properties about the ratio between the number of occurrences of two kinds of states along a run. For example, $\mathsf{EF}_{[100\cdot\sharp\mathtt{error}-\sharp\top<0]}P$ is true when there is a run leading to some state satisfying P along which the rate of `error` states is less than 1 percent. In fact any constraint of the form $\frac{\sharp P}{\sharp P'} \sim k$ can be expressed in this logic.
(7) Finally note that we can use any temporal formula inside a constraint (and not only atomic propositions). For example, $\mathsf{AG}(\mathsf{EF}_{[\sharp(\mathsf{EX}\mathtt{alarm})\leq 5]}\mathtt{init})$ states that it is always possible to reach `init` with a path along which at most 5 states have a successor satisfying `alarm`.

Note that expressing these properties is rather straightforward using counting constraints. When considering a classical temporal logic, such properties cannot easily be expressed directly. Unfolding the formula as it is done in the next section to prove expressiveness results cannot be achieved in practice even when the integer constraints are small: the formula would most of the time become too long and too complex to be handled. A possible pragmatic solution to avoiding counting constraints would be to add one or several counters to the model and to use additional atomic propositions to mark states (or rather, in such an extended model, configurations) where the constraints over the values of counters are satisfied. First note that this method may be less convenient or even inapplicable in some cases, as it requires modifying the model under verification. Moreover, this approach is difficult to use when counting constraints do not only refer to atomic propositions, but deal with nested temporal logic formulas (as in the last example above) or even other counting properties, as this would require even more drastic modifications to the model.

These examples illustrate the ability of our logics to state properties over the portion of a run leading to some state. A similar kind of properties could also be expressed with past-time modalities (like § or F^{-1}), but unlike these modalities our constraints cannot easily describe the ordering of events in the past: they "only" allow to count the number of occurrences of formulas. We will see in the next sections that our extensions do not always induce a complexity blow-up, while model-checking $\mathsf{CTL} + \mathsf{F}^{-1}$ is known to be PSPACE-complete [LS00].

3. Expressiveness and succinctness

When comparing two logics, the first question which comes to mind is the range of properties they can be used to define, in other words their *expressiveness*. When they turn out to be equally expressive, a natural way to distinguish them is then to ask *how concisely* each logic can express a given property. This is referred to as *succinctness*, and is also relevant when studying the complexity of model-checking for instance, since it may considerably influence the size of a formula required to express a given property, hence the time required to model-check it. In this section we study the expressiveness of the various logics defined in the previous section, and provide results and comments about their respective succinctness with respect to CTL.

3.1. Expressiveness.

We first show that only allowing Boolean combinations does not allow our logics to express more properties than CTL.

Proposition 3.1. *Any $\mathsf{CCTL}_{\wedge}$ formula Φ can be translated into an equivalent CTL formula of DAG-size $2^{O(|\Phi|^2)}$.*

Proof. A naive translation, using nested $\mathsf{E\_U\_}$ and $\mathsf{A\_U\_}$ modalities to precisely count the number of times each subformula inside a constraint is satisfied, is sufficient to show the result. However the size of a translated formula would in general be exponential in the *value* of all integer constants and in the DAG size of the original formula. We thus propose a more concise (yet more involved) translation, whose size will be useful later on.

Let Φ be a $\mathsf{CCTL}_\wedge$ formula. The proof is done by structural induction over Φ. The basic and Boolean cases are direct. By Remark 2.6, we only need to consider the cases $\Phi = \mathsf{EF}_{[C]}\varphi$ and $\Phi = \mathsf{AF}_{[C]}\varphi$. Assume C contains m atomic constraints of the form $(\sum_{j\in[1,n_i]} \alpha_j^i \sharp\varphi_j^i) \sim k_i$ for $i \in [1,m]$. We translate Φ to CTL by building a family of formulas whose intended meaning is as follows:

- If constraint C holds with $\sharp\varphi_j^i = 0$ for all j, i, then φ may be true immediately.
- Otherwise, successively check for every j, i whether φ_j^i holds in the current state, and if so then update C by decreasing the constant k_i by α_j^i.
- Once all φ_j^i have been scanned, proceed to the next state and re-evaluate C for the new values of the constants.

Let $decr(C, i, j)$ denote the constraint obtained from C by replacing k_i by $k_i - \alpha_j^i$. Note that in contrast with the formal definition of $\mathsf{CCTL}_\wedge$ constraints, we allow the *decr* operation to result in negative constants in the right-hand sides of atomic constraints.

Let $\bot$ and $\top$ be two special constraints satisfied by no (resp. any) finite path in any Kripke structure, we also define the constraint $C_\downarrow$ obtained from C by replacing any trivially true atomic constraint (such as $S \geq 0$ or $S > -3$) by $\top$ and any trivially false one (such as $S < 0$ or $S \leq -1$) by $\bot$, and normalizing the obtained constraint in the usual way ($C \vee \bot \to C$, ...). Note that due to this simplification step, $C_\downarrow$ is either reduced to $\top$ or $\bot$, or it does not contain $\top$ or $\bot$ as a sub-formula. Also note that C and $C_\downarrow$ are equivalent (i.e. satisfied by the same finite runs).

We now turn to the formal CTL translation $[\![\Phi]\!]$ of formula Φ, which is defined inductively on the structure of Φ. Boolean combinations and negation are left unchanged. In the case where $\Phi = \mathsf{EF}_{[C]}\varphi$, we proceed by unfolding the EF modality as follows:

$$[\![\mathsf{EF}_{[C]}\varphi]\!] = \begin{cases} \bot & \text{if } C_\downarrow = \bot \\ \mathsf{EF}[\![\varphi]\!] & \text{if } C_\downarrow = \top \\ \mathsf{E}\big(\bigwedge_{i,j} \neg[\![\varphi_j^i]\!]\big)\mathsf{U}\big([\![\varphi]\!] \vee \Psi\big) & \text{if } \epsilon \models C \\ \mathsf{E}\big(\bigwedge_{i,j} \neg[\![\varphi_j^i]\!]\big)\mathsf{U}\Psi & \text{if } \epsilon \not\models C \end{cases}$$

where Ψ is a CTL formula designed to be true in states where both $\mathsf{EF}_{[C]}\varphi$ and at least one formula φ_j^i hold. Indeed if C is trivially false, then $\mathsf{EF}_{[C]}\varphi$ is clearly not satisfiable. If C is trivially true, it is sufficient to check that φ eventually holds without any further checks on sub-formulas φ_j^i. The third case states that if C holds on the empty path then $\mathsf{EF}_{[C]}\varphi$ holds if, after a path prefix not affecting the satisfaction of C, either φ holds or some φ_j^i holds and we need to update C again. The last case is identical except that it does not check for φ in the current state. It then only remains to define Ψ. More generally, we describe a family of CTL formulas $\Psi_{C,i,j,C'}$, where $i \in [1,m]$, $j \in [1,n_i]$ with m and n_i as above, and C, C' are $\mathsf{CCTL}_\wedge$ constraints. For all $1 \leq i \leq m, 1 \leq j \leq n_i$, let

$$\Psi_{C,i,j,C'} = ([\![\varphi_j^i]\!] \wedge \Psi_{C,i,j+1,decr(C',i,j)}) \vee (\neg[\![\varphi_j^i]\!] \wedge \Psi_{C,i,j+1,C'}). \tag{3.1}$$

For all $1 \leq i < m$,

$$\Psi_{C,i,n_i+1,C'} = \Psi_{C,i+1,1,C'}. \tag{3.2}$$

Finally

$$\Psi_{C,m,n_m+1,C'} = \begin{cases} \bot & \text{if } C = C', \\ \mathsf{EX}[\![\mathsf{EF}_{[C'_\downarrow]}\varphi]\!] & \text{otherwise.} \end{cases} \tag{3.3}$$

We then set Ψ to denote $\Psi_{C,1,1,C}$. Formula $\Psi_{C,i,j,C'}$ implicitly assumes that, in the current state, a certain (potentially empty) subset of the formulas φ_1^1 up to (but not including) φ_j^i holds, and that C' is the constraint obtained by updating C with respect to these formulas. Then, it evaluates (the CTL translation of) formula φ_j^i, updating C' if necessary and moving on to the next sub-formula (Eq. (3.1)). Whenever the scanning of sub-formulas $\{\varphi_1^i, \ldots, \varphi_{n_i}^i\}$ relevant to the i-th atomic constraint is finished, we proceed with the next one (Eq. (3.2)). Finally, once all sub-formulas have been scanned and the constraint updated (Eq. (3.3)), if no progress was made at all (witnessed by the fact that $C = C'$), the formula is simply deemed false. Otherwise we move to the next state along a possible run using modality EX, and develop the translation of formula $\mathsf{EF}_{[C'_\downarrow]}\varphi$ with the last updated C'.

The above recursive definition characterizes finite formulas. Indeed, consider a formula $[\![\mathsf{EF}_{[C'_\downarrow]}\varphi]\!]$ occurring as a sub-formula of $[\![\mathsf{EF}_{[C_\downarrow]}\varphi]\!]$, and f the injection mapping each right-hand-side constant k' in C' to the corresponding constant in C. By definition, we have $f(k') \leq k'$ for every k' in C', and either f is not surjective (meaning that some constant k in C no longer appears in C' due to the simplification step in Eq. (3.3) above) or there exists k' such that $f(k') < k'$. This is guaranteed by the fact that developing $\Psi_{C,1,1,C}$ according to its definition into a formula containing $[\![\mathsf{EF}_{[C'_\downarrow]}\varphi]\!]$ resorts to at least one *decr* operation followed by a simplification operation. Since any negative constant appearing after a decrement is eliminated by the next simplification step, this process cannot repeat indefinitely and must therefore terminate.

The translation of $\mathsf{AF}_{[C]}\varphi$ is obtained by replacing each occurrence of the path quantifier E by A in the above. The correctness of the translation can be shown by induction on the nesting depth of until modalities in $[\![\Phi]\!]$ and quantities m and n_i.

We now turn to the worst-case DAG-size of the translation of the whole CCTL formula Φ. Let K be the largest integer constant in Φ, M the maximal number of atomic constraints in any constraint in Φ and N the maximal number of counting expressions in any atomic constraint in Φ. The number of distinct $\Psi_{C,i,j,C'}$ formulas involved in the translation of any sub-formula $\mathsf{EF}_{[C]}\varphi$ or $\mathsf{AF}_{[C]}\varphi$ of Φ is bounded by $K^M \cdot M \cdot N \cdot K^M$. This construction is repeated as many times as there are temporal modalities in Φ, which amounts to at most $|\Phi| \cdot K^M \cdot M \cdot N \cdot K^M$ distinct sub-formulas (this pessimistic upper bound clearly covers the case of Boolean connectives, whose translation is much simpler). Since $M, N \in O(|\Phi|)$ and $K \in O(2^{|\Phi|})$, we get a total DAG-size for $[\![\Phi]\!]$ in $O\big(|\Phi| \cdot (2^{|\Phi|})^{|\Phi|} \cdot |\Phi| \cdot |\Phi| \cdot (2^{|\Phi|})^{|\Phi|}\big) = O(|\Phi|^3.2^{2|\Phi|^2}) \subseteq 2^{O(|\Phi|^2)}$. □

Example 3.2. For any integer k and formula φ, we look at the translation of $\Phi_k = \mathsf{EF}_{[C_k]}\varphi$ where C_k denotes the constraint $\sharp p_1 + \sharp p_2 = k$ and φ is any formula:

$$[\![\Phi_k]\!] = \begin{cases} \mathsf{E}\big(\bigwedge_i \neg p_i\big)\mathsf{U}\varphi & \text{if } k = 0 \\ \mathsf{E}\big(\bigwedge_i \neg p_i\big)\mathsf{U}\Psi_k & \text{otherwise} \end{cases} \tag{3.4}$$

$$\text{with} \qquad \Psi_k = \Big(p_1 \wedge \big((p_2 \wedge \mathsf{EX}[\![(\Phi_{k-2})_\downarrow]\!]) \vee (\neg p_2 \wedge \mathsf{EX}[\![(\Phi_{k-1})_\downarrow]\!])\big)\Big) \\ \vee (\neg p_1 \wedge p_2 \wedge \mathsf{EX}[\![(\Phi_{k-1})_\downarrow]\!]),$$

where $(\Phi_k)_\downarrow = \Phi_k$ if $k \geq 0$ and $\bot$ otherwise. Note that some simplifications were performed in this translation: namely, $(\varphi \vee \Psi_0)$ is replaced by φ in the first case of Eq. (3.4) since $\Psi_0 \equiv \bot$, and a conjunct containing $\bot$ is removed from Ψ_k.

Note that we provided a parametric upper bound for the above translation which can be interpreted for all variants of CCTL below $\mathsf{CCTL}_\wedge$. In contrast to this result, introducing subtractions in constraints yields a strict increase in expressiveness.

Proposition 3.3. *The $\mathsf{CCTL}_{\pm 1}$ formula $\varphi = \mathsf{AG}_{[\sharp A - \sharp B < 0]} \bot$ cannot be translated into CTL.*

(sketch). Formula φ (already seen in Sec. 2.3 with different atomic propositions) states that the number of B-labeled states cannot exceed the number of A-labeled states along any path. As shown by [BVW94] and also presented in [Wil99], the set of models of any CTL formula can be recognized by a finite alternating tree automaton. Suppose there exists a CTL formula φ' equivalent to φ, and let $\mathcal{A}$ be the alternating tree automaton accepting its set of models. From $\mathcal{A}$, one can easily build a finite alternating automaton on words over $2^{\{A,B\}}$, whose accepted language is the set of all finite prefixes of branches in models of φ, namely words whose prefixes contain at most as many B's as A's. Since this language is clearly not regular, this leads to a contradiction. □

3.2. **Succinctness.** Our extensions of CTL come with three main potential sources of concision, which appear to be orthogonal: the encoding of constants in binary, the possibility to use Boolean combinations in constraints, and the use of sums. However, only the first two turn out to yield an exponential improvement in succinctness. First we consider the case of sums:

Proposition 3.4. *For every formula $\Phi \in \mathsf{CCTL}$ with unary encoding of integers, there exists an equivalent CTL formula of DAG-size polynomial in $|\Phi|$.*

Proof. This proposition is a direct consequence of the DAG-size computation presented in the proof of Prop. 3.1 where M, the number of atomic constraints in a constraint in Φ, is set to 1 to reflect the absence of Boolean connectives inside constraints, and where K, the maximal constant in Φ, is bounded by $|\Phi|$ due to the unary encoding. □

We now look at the succinctness gap due to the binary encoding of constants[3]:

Proposition 3.5. *CCTL_1 can be exponentially more succinct than CTL.*

Proof. In [LST03], it is shown that the logic TCTL, when interpreted over Kripke structures with a special atomic proposition *tick* used to mark the elapsing of time, can be exponentially more succinct than CTL[4]. More precisely, the TCTL formulas $\mathsf{EF}_{<n}A$ and $\mathsf{EF}_{>n}A$, which are of size $O(\log(n))$ since n is encoded in binary, do not admit any equivalent CTL formula of temporal height (and hence also size) less than n. These formulas express the existence of a path where A eventually holds and less (resp. more) than n clock ticks are

[3]Note that for real-time logics, it is already known that the binary encoding of integer constants induces a complexity blow-up for the decision procedures [AH93, AH94].

[4]This was also observed in [EMSS92] for the logic RTCTL over DKS^1.

seen until then. They are respectively equivalent to the $O(\log(n))$-size CCTL_1 formulas $\mathsf{EF}_{[\sharp tick<n]}A$ and $\mathsf{EF}_{[\sharp tick>n]}A$. □

Note that the proof of the previous proposition only uses the simplest kind of constraint: we do not need sums (and coefficients) or Boolean combinations in the constraints.

This exhibits a first aspect in which CCTL logics can be exponentially more succinct than CTL. However, as expressed in the next proposition, another orthogonal feature of the logic may yield a similar blow-up.

Proposition 3.6. *$CCTL_{\wedge 1}$ with unary encoding of integers can be exponentially more succinct than CTL.*

Proof. It was shown by [Wil99, AI03] that any CTL formula φ equivalent to the CTL^+ formula $\psi = \mathsf{E}(\mathsf{F}P_0 \wedge \ldots \wedge \mathsf{F}P_n)$ must be of length exponential in n. It turns out ψ is equivalent to the $\mathsf{CCTL}_{\wedge 1}$ formula $\psi' = \mathsf{EF}_{[\bigwedge_i \sharp P_i \geq 1]}\top$, which entails the result. Note that ψ' only contains the constant 1, which means that this gap cannot be imputed to the binary encoding. □

The intuitive reason for this blow-up is that a CTL formula expressing the property that atomic propositions P_1 to P_n are each seen at least once along a path would have to keep track of all possible interleavings of occurrences of P_i's.

To summarize, we showed that two different aspects of the extensions of CTL presented in this paper, while not increasing the overall expressiveness of the logic, may yield exponential improvements in succinctness. It would remain to study the succinctness of remaining CCTL fragments with respect to each other, in particular when these aspects are combined.

4. Model checking

4.1. **Polynomial-time model-checking.** Even though, as we discussed in the previous section, diagonal constraints lead to strictly more expressive logics than CTL, it turns out that model-checking $\mathsf{CCTL}_{\pm 1}$ is asymptotically not more difficult than model checking CTL itself. As a preliminary result of independent interest, we show that the existence of a polynomial-time algorithm for the model-checking of the logic TCTL over $\mathrm{DKS}^{0/1}$, as shown in [LST03], remains true when considering more general weighted graphs, namely DKS's with weights in $\{-1, 0, 1\}$. This result will be used to establish the complexity of model-checking for $\mathsf{CCTL}_{\pm 1}$, and as a corollary also for all weaker fragments.

Proposition 4.1. *The model-checking problem for TCTL over $DKS^{-1/0/1}$ is P-complete.*

P-hardness is inherited from CTL (see [Sch03] for a proof of the P-hardness of CTL). For membership in P, we consider a DKS $\mathcal{S} = \langle Q, R, \ell\rangle$ with $R \subseteq Q \times \{-1, 0, 1\} \times Q$, a state $q \in Q$ and a TCTL formula Φ, and show that deciding whether $q \models \Phi$ can be done in polynomial time. As usual, we inductively assume the set of states satisfying all strict sub-formulas of Φ to be known, and proceed from there. We distinguish several cases:

(1) $\Phi = \mathsf{E}\varphi\mathsf{U}_{\leq k}\psi$: We first determine the subset of states $Q|_{\mathsf{E}\varphi\mathsf{U}\psi}$ from which the CTL formula $\mathsf{E}\varphi\mathsf{U}\psi$ holds, and consider the restriction $\mathcal{S}'$ of $\mathcal{S}$ to $Q|_{\mathsf{E}\varphi\mathsf{U}\psi}$ in which outgoing edges of states labeled by $\psi \wedge \neg\varphi$ are removed. Φ holds over some state q in $\mathcal{S}$ if and only if $q \in Q|_{\mathsf{E}\varphi\mathsf{U}\psi}$ and there exists a path of weight at most k in $\mathcal{S}'$ from q to some other state q' where ψ holds. Considered paths are either simple,

or composed of a prefix from q to some state q'', a negative-weight cycle from q'' to itself repeated a certain number of times, and a suffix from q'' to q'.

Even though finding a *simple* path of weight less than k in a graph containing negative cycles is NP-complete, this is not exactly what we are considering since we allow paths containing repeated states. Our problem can thus be tested in polynomial time using the classical Floyd-Warshall algorithm (to compute all-pairs shortest paths) over $\mathcal{S}'$. The matrix $(\alpha)_{q,q'}$ of shortest-path weights computed by this algorithm gives us sufficient information: to decide whether a state q satisfies Φ: $q \models \Phi$ one simply need to check whether there exists q' satisfying ψ such that either $\alpha_{q,q'} \leq k$, or there exists q'' such that $\alpha_{q,q''} < \infty$, $\alpha_{q'',q''} < 0$ and $\alpha_{q'',q'} < \infty$.

(2) $\Phi = \mathsf{E}\varphi\mathsf{U}_{\geq k}\psi$: We build the DKS $\mathcal{S}'$ as in the previous case, and a new DKS $\mathcal{S}''$ isomorphic to $\mathcal{S}'$ but with opposite weights. Then, Φ is satisfied from q in $\mathcal{S}'$ (and thus also $\mathcal{S}$) if and only if the formula $\mathsf{E}\varphi\mathsf{U}_{\leq -k}\psi$ is satisfied from q in $\mathcal{S}''$.

(3) $\Phi = \mathsf{E}\varphi\mathsf{U}_{=k}\psi$: We build the DKS $\mathcal{S}'$ as in case 1, and compute the relation

$$R_k = \{(q, q') \in Q|_{\varphi \wedge \mathsf{E}\varphi\mathsf{U}\psi} \times Q|_{\mathsf{E}\varphi\mathsf{U}\psi} \mid q \xRightarrow[R]{k} q'\}.$$

For $k = 0$, R_0 can be seen as $\bigcup_{i\geq 0} X_i$ with:

$$\begin{cases} X_0 &= (\xrightarrow[R]{0})^* \\ X_{i+1} &= X_i \cup (X_i \cdot \xrightarrow[R]{1} \cdot X_i \cdot \xrightarrow[R]{-1} \cdot X_i) \cup (X_i \cdot \xrightarrow[R]{-1} \cdot X_i \cdot \xrightarrow[R]{1} \cdot X_i) \end{cases}$$

which can be obtained by a simple fixed-point computation requiring at most $|Q|^2$ iterations (since $|R_0| \leq |Q|^2$). For $k = 1$, we simply have $R_1 = R_0 \cdot \xrightarrow[R]{1} \cdot R_0$. For greater values of k, we use dichotomy to express this relation in terms of R_0 and R_1 in $O(\log(k))$ steps (*i.e.* $O(|\Phi|)$, since k is encoded in binary), by writing

$$R_k = R_{\lfloor k/2 \rfloor} \cdot R_{\lceil k/2 \rceil}.$$

Each of these relational compositions requires time at most cubic in the size of Q. It then suffices to test whether $(q, q') \in R_k$ for some q' verifying ψ.

(4) $\Phi = \mathsf{A}\varphi\mathsf{U}_{=0}\psi$: The procedure consists in defining a standard Kripke structure $\mathcal{S}'$ and a classical CTL formula Ψ such that $\mathcal{S}'$ satisfies Ψ if and only if $\mathcal{S}$ does not satisfy Φ.

Using fixed-point computations over $Q \times Q$, we compute the relations R_0^+ and R_0^- as the respective least solutions of

$$\begin{cases} X_0 &= (\xrightarrow[R]{0})^* \\ X_{i+1} &= X_i \cup (\xrightarrow[R]{1} \cdot X_i \cdot \xrightarrow[R]{-1}) \end{cases} \quad \text{and} \quad \begin{cases} X_0 &= (\xrightarrow[R]{0})^* \\ X_{i+1} &= X_i \cup (\xrightarrow[R]{-1} \cdot X_i \cdot \xrightarrow[R]{1}). \end{cases}$$

R_0^+ and R_0^- respectively express the reachability relation in $\mathcal{S}$ along paths of weight 0 with no prefix of strictly negative (resp. positive) weight. We also define the relation R_0^s (where s stands for *strict*) as:

$$R_0^s = \xrightarrow{0} \cup (\xrightarrow{1} \cdot R_0^+ \cdot \xrightarrow{-1}) \cup (\xrightarrow{-1} \cdot R_0^- \cdot \xrightarrow{1}),$$

which expresses reachability in $\mathcal{S}$ by 0-weight paths such that no intermediate state (other than the initial one) is reached with weight 0. Let Q^+, Q^- be two isomorphic

copies of Q (and q^+, q^- denote the copies in Q^+ and Q^- of some state $q \in Q$), we can now construct $\mathcal{S}' = (Q', R', \ell')$ with $Q' = Q \cup Q^+ \cup Q^-$, $\ell'(q) = \ell(q)$ if $q \in Q$ and $\ell'(q^\pm) = \ell(q) \cup \{\mathsf{ok}\}$, and

$$\begin{aligned} R' = &\{q_1 \to q_2 \mid (q_1, q_2) \in R_0^s\} \\ &\cup \{q_1 \to q_2^+ \mid q_1 \xrightarrow[R]{1} q_2\} \cup \{q_1^+ \to q_2^+ \mid q_1 \xrightarrow[R]{1} q_2 \vee (q_1, q_2) \in R_0^+\} \\ &\cup \{q_1 \to q_2^- \mid q_1 \xrightarrow[R]{-1} q_2\} \cup \{q_1^- \to q_2^- \mid q_1 \xrightarrow[R]{-1} q_2 \vee (q_1, q_2) \in R_0^-\}. \end{aligned}$$

In order to eliminate finite paths, we additionally complete $\mathcal{S}'$ with a dummy state $q_\perp$ and transitions from every state to $q_\perp$ and a loop from $q_\perp$ to itself. We let $\ell'(q_\perp) = \{\psi\}$, which will be explained in detail later on.

The set of states of $\mathcal{S}'$ is divided into four subsets: states in Q correspond to the states reachable with weight 0 in $\mathcal{S}$, and states in Q^+ and Q^- are the states reachable with weight strictly more or strictly less than 0. Paths in $\mathcal{S}'$ ending in the dummy state may not correspond to actual paths in $\mathcal{S}$, but they correspond to situations which are irrelevant to solving the problem. Since a path going from Q to Q^+, and then from Q^+ to Q is captured by the relation R_0^s, we can omit transitions going back to Q from Q^+ (and similarly for Q^-). Hence all runs of $\mathcal{S}'$ either stay forever in Q, eventually reach Q^+ or Q^- and stay there forever, or reach the dummy state and stay there forever.

We now define the CTL formula Ψ as $\mathsf{E}(\neg\psi \vee \mathsf{ok})\mathsf{W}(\neg\varphi \wedge (\neg\psi \vee \mathsf{ok}))$ [5] and claim that $q \models_{\mathcal{S}'} \Psi$ if and only if $q \models_{\mathcal{S}} \neg\Phi$. The idea of the proof is to show that if Φ is not satisfied from some state q in $\mathcal{S}$ then one can find a path from q in $\mathcal{S}'$ satisfying Ψ, and conversely that finding a path satisfying Ψ from q over $\mathcal{S}'$ is sufficient to disprove Φ from that state in $\mathcal{S}$.

Lemma 4.2. $q \models_{\mathcal{S}} \neg\Phi \implies q \models_{\mathcal{S}'} \Psi$.

Proof. There are several ways in which Φ may fail to hold over $\mathcal{S}$:

(a) There exists a path ρ in $\mathcal{S}$ along which a state $q_1 \models \neg\varphi$ appears strictly before the first state satisfying ψ and reached with weight 0. Let $\rho_1 q_1$ be the shortest prefix of ρ such that $q_1 \models \neg\varphi$ and either $q_1 \models \neg\psi$ or $\|\rho_1 q_1\| \neq 0$.
   (i) If $q_1 \models \neg\psi$ and $\|\rho_1 q_1\| = 0$, then by definition of R_0^s there must exist a path ρ_1' from q to q_1 in $\mathcal{S}'$ whose intermediate states all satisfy $\neg\psi$. Consequently, any infinite continuation of ρ' must satisfy $\neg\psi\mathsf{W}(\neg\varphi \wedge \neg\psi)$, which implies that $q \models_{\mathcal{S}'} \Psi$.
   (ii) If $\|\rho_1 q_1\| \neq 0$, then we can write $\rho_1 q_1 = \rho_2 q_2 \rho_3 q_1$ where $\rho_2 q_2$ is the longest prefix of ρ_1 of weight 0. By definition, $q_2\rho_3 q_1$ starts with a non-0 transition and has no prefix of weight 0, hence by definition of $\mathcal{S}'$ there must exist a finite path $\rho' = \rho_2' q_2 \rho_3' q_1^\pm$ in $\mathcal{S}'$ such that all intermediate states of $\rho_2' q_2$ satisfy $\neg\psi$ and all intermediate states of $\rho_3' q_1^\pm$ satisfy ok. Hence any continuation of ρ' must satisfy $(\neg\psi \vee \mathsf{ok})\mathsf{W}(\neg\varphi \wedge \mathsf{ok})$, which implies that $q \models_{\mathcal{S}'} \Psi$.

(b) There exists a path ρ in $\mathcal{S}$ along which no state satisfying ψ ever appears at the end of a prefix of weight 0. We assume that φ consistently holds along the

[5] W is called the *weak until* modality, and $\varphi\mathsf{W}\psi$ holds along a path if either $\mathsf{G}\varphi$ or $\varphi\mathsf{U}\psi$ does.

path, otherwise it comes down to the previous case. There are again two cases to consider:

(i) If ρ has infinitely many prefixes of weight 0, then by definition of R_0^s there must exist an infinite path ρ' in $\mathcal{S}'$ whose intermediate states never leave the set Q and all satisfy $\neg\psi$. Therefore ρ' satisfies $\mathsf{G}\neg\psi$, which implies that $q \models_{\mathcal{S}'} \Psi$.

(ii) If ρ has finitely many prefixes of weight 0, then using ideas similar to the above, one can decompose it as $\rho_1\rho_2$, with ρ_1 its longest finite prefix of weight 0, and ρ_2 an infinite path with no prefix of weight 0. This implies the existence of a corresponding path ρ' in $\mathcal{S}'$ with a finite prefix remaining in Q whose states all satisfy $\neg\psi$ and an infinite suffix remaining in Q^+ or Q^- whose states all satisfy ok. Therefore ρ' must satisfy $\mathsf{G}\neg\psi \vee \mathsf{ok}$, which implies that $q \models_{\mathcal{S}'} \Psi$. □

Lemma 4.3. $q \models_{\mathcal{S}'} \Psi \implies q \models_{\mathcal{S}} \neg\Phi$.

Proof. The proof is very similar to that of the previous lemma. Let us consider a path ρ' in $\mathcal{S}'$ satisfying Ψ. There are two main possibilities:

(a) The path ρ' consistently satisfies $\neg\psi \vee \mathsf{ok}$. We distinguish two cases.

(i) If ρ' never leaves the set Q (and thus consists only of edges representing the relation R_0^s), then there must exist a corresponding path ρ in $\mathcal{S}$ visiting at least the same states in the same order (since R_0^s is a restriction of the reachability relation of $\mathcal{S}$). Moreover, *all* states reached with weight 0 in ρ must appear in ρ' (by definition of R_0^s). Now whether or not the states in ρ satisfy φ, Φ cannot be satisfied in $\mathcal{S}$ from q since no state reached with weight 0 satisfies ψ along ρ.

(ii) If ρ' eventually leaves the set Q, and since there are no transitions out of Q^+ and Q^- except to the dummy state $q_\perp$ (which satisfies ψ but not ok), then necessarily ρ' can be decomposed into $\rho'_1 q_1 \rho'_2$ where $\rho'_1 q_1$ is a finite path in Q necessarily satisfying $\mathsf{G}\neg\psi$ and ρ_2 is an infinite path either in Q^+ or Q^- necessarily satisfying Gok. As previously, this implies the existence of a corresponding path ρ in $\mathcal{S}$, where the part corresponding to ρ'_1 never visits a state satisfying ψ with weight 0, and the part corresponding to ρ_2 never reaches weight 0 again. Thus Φ cannot be satisfied from q in $\mathcal{S}$.

(b) The other possibility is that ρ' can be written $\rho'_1 q_1 \rho'_2$, where ρ'_1 satisfies $\mathsf{G}\neg\psi \vee \mathsf{ok}$ and q_1 satisfies $\neg\varphi \wedge (\neg\psi \vee \mathsf{ok})$. Again there are two possible cases.

(i) If $q_1 \in Q$, then ρ'_1 only visits states satisfying $\neg\psi$, and $q_1 \models \neg\varphi \wedge \neg\psi$. As previously there must exist a corresponding path ρ_1 in $\mathcal{S}$ visiting at least the same states in the same order. Now since by definition of R_0^s all 0-weight prefixes of ρ_1 end in states appearing in ρ' and satisfying $\neg\psi$, and since q_1 satisfies $\neg\varphi \wedge \neg\psi$, no continuation of $\rho_1 q_1$ in $\mathcal{S}$ can satisfy $\varphi \mathsf{U}_{=0} \psi$.

(ii) If $q_1 \notin Q$, then necessarily $q_1 \in Q^+ \cup Q^-$ (since $q_\perp \not\models \neg\psi \vee \mathsf{ok}$) and $q_1 \models \neg\varphi \wedge \mathsf{ok}$. Consequently one can write $\rho'_1 = \rho'_2 q_2 \rho'_3$ such that $q_2 \in Q$, ρ'_2 never leaves Q and ρ'_3 never leaves either Q^+ or Q^-. Moreover, all states in $\rho'_2 q_2$ satisfy $\neg\psi$. One can thus build in $\mathcal{S}$ a finite path ρ from q to q_1 going through q_2, in which no state reached with weight 0 up to q_2

(and thus also up to q_1) satisfies ψ, and all states occurring after q_2 (in particular q_1) are reached with non-0 weight. Hence since $q_1 \models \neg\varphi$, this implies that no continuation of ρ can satisfy $\varphi \mathsf{U}_{=0} \psi$. □

(5) $\Phi = \mathsf{A}\varphi\mathsf{U}_{=k}\psi$: This case is similar to the previous one with slight modifications of the construction. We first assume k to be positive, otherwise we can replace S by an identical structure in which all weights are inverted and solve the formula with parameter $-k$. We then inductively compute

$$R_k^- = R_{\lfloor k/2 \rfloor}^- \cdot R_{\lceil k/2 \rceil}^- \quad \text{with} \quad R_1^- = R_0^- \cdot \xrightarrow{1}$$

and R_0^- defined as previously. R_k^- is the reachability relation in $G_{\mathcal{S}}$ by paths of weight k whose prefixes all have weight strictly less than k. We also compute, using for instance a modified Floyd-Warshall algorithm in which all integers greater than or equal to k are assimilated to ∞, the reachability relation $R_{<k}^- = \{(q, q') \mid \exists \sigma = q\sigma' q', \forall \rho \leq \sigma,\ \|\rho\| < k\}$.

We now construct a Kripke structure $\mathcal{S}'$ as in the previous case, except that $Q' = Q \cup Q^{init} \cup Q^+ \cup Q^- \cup \{q_\perp\}$ (where Q^{init} is yet another copy of Q and q^{init} denotes the copy of q in Q^{init}) and R' also contains $\{q_1^{init} \to q_2 \mid q_1 R_k^- q_2\} \cup \{q_1^{init} \to q_2^- \mid q_1 R_{<k}^- q_2\}$. We additionally label states in Q^{init} with the atomic proposition ok. With this new Kripke structure, we can show that $q \models_{\mathcal{S}} \neg\Phi$ if and only if $q^{init} \models_{\mathcal{S}'} \Psi$.

(6) $\Phi = \mathsf{A}\varphi\mathsf{U}_{\sim k}\psi$ with $\sim \in \{\leq, <, >, \geq\}$: Let us first treat the case where $\sim$ is $\leq$. We assume k to be greater than or equal to 0, otherwise we invert all weights in $\mathcal{S}$ and solve the problem using the procedure for $\Phi = \mathsf{A}\varphi\mathsf{U}_{\geq -k}\psi$. We essentially use the same procedure as for the previous case ($= k$), with a few modifications:
   (a) Relations R_x^- have to be computed over the restricted set of states $Q' = \{q \in Q \mid q \models \neg\psi\}$, because we have to make sure that no "hidden" intermediate state reached after a path of weight less than k satisfies ψ;
   (b) States in Q^- should no longer be labelled by atomic proposition ok, because paths which ultimately remain in Q^- may correspond to paths in $\mathcal{S}$ satisfying Φ, and thus should not satisfy Ψ unlike previously;
   (c) Similarly, we remove the label ok from states in Q^{init}, in other words $\forall q \in Q,\ \ell'(q^{init}) = \ell(q)$.

   In the case where $\sim$ is $\geq$, we simply need to re-label states in Q^{init} and Q^- with ok, and remove ok from the labelling of Q^+. Cases where $\sim$ is $<$ and $>$ are dealt with by adding the ok label on states in Q in the constructions for $\leq$ and $\geq$.

This concludes the proof that deciding the satisfaction of a TCTL formula from a given state of a $\mathrm{DKS}^{-1/0/1}$ is in P. □

Theorem 4.4. *The model-checking problem for $\mathsf{CCTL}_{\pm 1}$ is P-complete.*

Proof. As usual, P-hardness is inherited from CTL. Membership in P is done by reduction to TCTL model-checking over $\mathrm{DKS}^{-1/0/1}$.

We provide polynomial-time procedures to deal with the sub-formulas $\mathsf{E}\varphi\mathsf{U}_{[C]}\psi$ and $\mathsf{A}\varphi\mathsf{U}_{[C]}\psi$ with $C = \sum_{i=1}^{\ell} \alpha_i \sharp \varphi_i \sim k$ where $\alpha_i \in \{-1, 1\}$ and $k \in \mathbb{Z}$. Consider a Kripke structure $\mathcal{S} = (Q, R, \ell)$, and inductively assume that the truth values of φ, ψ and φ_i over each state of $\mathcal{S}$ are known: these sub-formulas will be seen as atomic propositions in the following.

To each state q occurring along a path, we associate a cost $|q|_C = \sum\{\alpha_i \mid q \models \varphi_i\}$, and note that the *value* of $|q|_C$ is in $O(|C|)$. This cost is additively extended to paths in the usual way. Deciding the truth value of the path formula $\varphi \mathsf{U}_{[C]} \psi$ then amounts to checking whether there exists a finite prefix $\rho' q$ of ρ such that $|\rho'|_C \sim k$, $q \models \psi$ and $\forall i \leq |\rho'|, \rho'(i) \models \varphi$.

Given the type of our counting constraints, each state contributes to the cost of a path by a certain positive or negative number whose absolute value is bounded by $d = \max(\sum\{\alpha_i \mid \alpha_i = 1\}, \sum\{|\alpha_i| \mid \alpha_i = -1\})$. The idea is to build a durational Kripke structure with weights in $\{-1, 0, 1\}$, by adding (at most $d+1$) copies of each state in the original Kripke structure.

Formally, we build from $\mathcal{S}$ a $\mathrm{DKS}^{-1/0/1}$ $\mathcal{S}' = (Q', R', \ell')$ as follows: for each state $q \in Q$ with $||q|_C| = n$, Q' contains $n+1$ additional states $q_0, \ldots, q_n$. R' is then defined as $\{q \xrightarrow{0} q_0 \mid q \in Q\} \cup \{q_n \xrightarrow{0} q' \mid (q, q') \in R, n = ||q|_C|\} \cup \{q_i \xrightarrow{\delta_q} q_{i+1} \mid q \in Q, i < ||q|_C|\}$ with $\delta_q = 1$ if $|q|_C > 0$ and $\delta_q = -1$ otherwise. Finally, we set $\ell'(q_i) = \varnothing$ for all $q_i \in Q' \setminus Q$ and $\ell'(q) = \ell(q) \cup \{\mathsf{ok}\}$ for all $q \in Q' \cap Q$, where ok is a new atomic predicate.

To each path $\rho = q\sigma$ in $\mathcal{S}$, we associate the path $\tilde{\rho} = qq_0 \ldots q_n \tilde{\sigma}$ in $\mathcal{S}'$. It can now be shown that ρ satisfies $\varphi \mathsf{U}_{[C]} \psi$ if and only if $\tilde{\rho}$ satisfies the TCTL path formula $(\mathsf{ok} \Rightarrow \varphi)\mathsf{U}_{[\sim k]}(\mathsf{ok} \wedge \psi)$, and consequently that some state q satisfies $\mathsf{A}\varphi \mathsf{U}_{[C]} \psi$ (resp. $\mathsf{E}\varphi \mathsf{U}_{[C]} \psi$) in $\mathcal{S}$ if and only if it satisfies $\mathsf{A}(\mathsf{ok} \Rightarrow \varphi)\mathsf{U}_{[\sim k]}(\mathsf{ok} \wedge \psi)$ (resp. $\mathsf{E}(\mathsf{ok} \Rightarrow \varphi)\mathsf{U}_{[\sim k]}(\mathsf{ok} \wedge \psi)$) in $\mathcal{S}'$.

Suppose $\rho \models_{\mathcal{S}} \varphi \mathsf{U}_{[C]} \psi$. We reason by induction on the least integer i such that $\rho_{|i-1} \models_{\mathcal{S}} C$, $\rho(j) \models_{\mathcal{S}} \varphi$ for all $j < i$ and $\rho(i) \models_{\mathcal{S}} \psi$. If $i = 1$, then $\rho(1) \models_{\mathcal{S}} \psi$ and thus $\tilde{\rho}(1) \models_{\mathcal{S}'} \psi$ (recall that ψ is seen as atomic). Otherwise, $\rho = q\rho'$ with $q \models_{\mathcal{S}} \varphi$ and $\rho' \models_{\mathcal{S}} \varphi \mathsf{U}_{[C']} \psi$ with $C' = \sum_{i=1}^{\ell} \sharp\varphi_i \sim k - |q|_C$, in other words $\rho'_{|i-2} \models_{\mathcal{S}} C'$ and $\rho'(i-1) \models_{\mathcal{S}} \psi$. By induction hypothesis, we have $\tilde{\rho}' \models_{\mathcal{S}'} (\mathsf{ok} \Rightarrow \varphi)\mathsf{U}_{[\sim k - |q|_C]}(\mathsf{ok} \wedge \psi)$. Hence $\tilde{\rho} = qq_0 \ldots q_{|q|_C} \tilde{\rho}' \models_{\mathcal{S}'} (\mathsf{ok} \Rightarrow \varphi)\mathsf{U}_{[\sim k]}(\mathsf{ok} \wedge \psi)$.

Conversely, consider a path ρ in $\mathcal{S}'$ starting with some state $q \in Q$ such that $\rho \models_{\mathcal{S}'} (\mathsf{ok} \Rightarrow \varphi)\mathsf{U}_{[\sim k]}(\mathsf{ok} \wedge \psi)$, and as previously let i be the least integer such that $|\rho_{|i-1}| \sim k$, $\rho(j) \models_{\mathcal{S}'} (\mathsf{ok} \Rightarrow \varphi)$ for all $j < i$ and $\rho(i) \models_{\mathcal{S}'} (\mathsf{ok} \wedge \psi)$. By construction of $\mathcal{S}'$, there must exist a unique path σ in $\mathcal{S}$ such that $\tilde{\sigma} = \rho$. We show by induction on i that $\sigma \models_{\mathcal{S}} \varphi \mathsf{U}_{[C]} \psi$. If $i = 1$, then $\rho(1) \models_{\mathcal{S}'} \mathsf{ok} \wedge \psi$, in which case $\sigma(1) \models_{\mathcal{S}} \psi$ holds in $\mathcal{S}$. Otherwise by construction of $\mathcal{S}'$ there must exist $q' \in Q$ such that $\rho = qq_0q_1 \ldots q_n q' \rho'$ and $q'\rho' \models_{\mathcal{S}'} (\mathsf{ok} \Rightarrow \varphi)\mathsf{U}_{[\sim k - |q|_C]}(\mathsf{ok} \wedge \psi)$. Let $\sigma = qq'\sigma'$, by induction hypothesis we have $q'\sigma' \models_{\mathcal{S}} \varphi \mathsf{U}_{[C']} \psi$ with $C' = \sum_{i=1}^{\ell} \sharp\varphi_i \sim k - |q|_C$. Hence $\sigma \models_{\mathcal{S}} \varphi \mathsf{U}_{[C]} \psi$. □

This result implies the following corollary on the complexity of model-checking for all fragments of intermediate expressiveness:

Corollary 4.5. *The model-checking problem for CCTL_1 is P-complete.*

Note that this weaker fragment allows considerable simplification of the proof presented above for $\mathsf{CCTL}_{\pm 1}$. Moreover, model-checking CCTL_1 can be done using the TCTL model-checking algorithm provided in [LST03] instead of the more involved construction used for Prop. 4.1.

4.2. **Model-checking $\mathsf{CCTL}_{\wedge 1}$, CCTL and $\mathsf{CCTL}_{\wedge}$.** We now establish the complexity of model-checking for the fragments $\mathsf{CCTL}_{\wedge 1}$, CCTL and $\mathsf{CCTL}_{\wedge}$ and show that these problems are all Δ_2^{P}-complete. Let us first recall the definition of the complexity class Δ_2^{P}, one of the classes of the polynomial hierarchy.

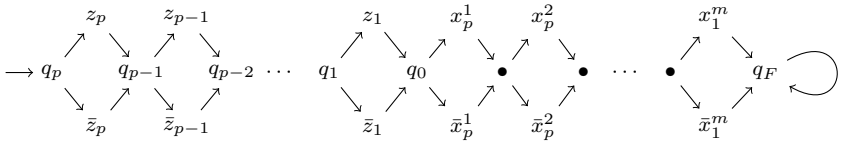


FIGURE 1. Kripke structure associated to an SNSAT problem.

Definition 4.6. $\Delta_2^{\mathsf{P}} = \mathsf{P}^{\mathsf{NP}}$ is the class of problems solvable in polynomial time with access to an oracle for some NP-complete problem.

We now prove Δ_2^{P}-hardness of the model-checking problem for $\mathsf{CCTL}_{\wedge 1}$.

Theorem 4.7. *The model-checking problem for $\mathsf{CCTL}_{\wedge 1}$ is Δ_2^{P}-hard.*

Proof. We proceed by reduction from the Δ_2^{P}-complete problem SNSAT (sequentially nested satisfiability of propositional logic) [LMS01].

Given p families of variables $X_1, \dots X_p$ with $X_i = \{x_i^1, ..., x_i^m\}$ and a set $Z = \{z_1, \dots, z_p\}$ of p variables, an instance $\mathcal{I}$ of SNSAT is defined as a collection of p propositional formulas $\varphi_1, \dots, \varphi_p$ under 3-conjunctive normal form (3-CNF), where each φ_i involves variables in $X_i \cup \{z_1, ..., z_{i-1}\}$, and the value of each z_i is defined as $z_i = \exists X_i.\ \varphi_i(z_1, ..., z_{i-1}, X_i)$. The instance $\mathcal{I}$ is positive iff the value of z_p is $\top$. We denote by $v_{\mathcal{I}}$ the unique valuation of variables in Z induced by $\mathcal{I}$.

From $\mathcal{I}$, we define the Kripke structure described in Figure 1. Every state z_i or x_i^j is labeled by its name, every state $\bar{z}_i$ is labeled by some new atomic proposition $\bar{z}$ and every state of the form q_i is labeled by q. We use X to denote the set $X_1 \cup \dots \cup X_p$ and $\mathcal{V}$ for $X \cup Z$. A path ρ from q_p to q_F describes the valuation v_ρ such that $v_\rho(y) = \top$ if ρ visits state y and $\bot$ if it visits $\bar{y}$ for every variable y in $\mathcal{V}$. We use a $\mathsf{CCTL}_{\wedge 1}$ formula to ensure that v_ρ coincides with $v_{\mathcal{I}}$ over Z, that is: $v_\rho(z_i) = \top$ iff $v_{\mathcal{I}}(z_i) = \top$ for any $i \in \{1, \dots, p\}$.

Let $\widetilde{\varphi_i}$ be the formula φ_i where every occurrence of the *literal* x is replaced by $\sharp x = 1$. We define the $\mathsf{CCTL}_{\wedge 1}$ formula Ψ_0 as $\top$ and for every $1 \leq k \leq p$, Ψ_k as $\mathsf{EX}\big(\mathsf{E}(\bar{z} \Rightarrow \neg\Psi_{k-1})\mathsf{U}_{[C_k]} q_F\big)$, with $C_k = \bigwedge_{\ell \leq k} \big((\sharp z_\ell = 1) \Rightarrow \widetilde{\varphi_\ell}\big) \wedge \bigwedge_{j=1}^{k} \big((\sharp q = j) \Rightarrow \widetilde{\varphi_j}\big)$. The first part of the constraint C_k aims at ensuring that $v_\rho(z_\ell) = \top$ is witnessed by a valuation for $\{z_1, \dots, z_{\ell-1}\} \cup X_\ell$ satisfying φ_ℓ. The second part ensures the formula φ_j is satisfied by v_ρ when Ψ_k is interpreted from z_j or $\bar{z}_j$ (*i.e.* when the number of q's along the path leading to q_F is j). The formula Ψ_j holds for a state q_i with $i \leq j$ when $v_{\mathcal{I}}(z_i)$ is $\top$. The embedding of Ψ_{j-1} inside Ψ_j is used to ensure that going through a $\bar{z}_m$ with $i \geq m$ is always necessary w.r.t. $\mathcal{I}$ (*i.e.* there is no way to satisfy the corresponding φ_m):

Lemma 4.8. *For any $i = 1, \dots, p$ and $i \leq j \leq p$, we have: $z_i \models \Psi_j \Leftrightarrow v_{\mathcal{I}}(z_i) = \top$ and $\bar{z}_i \not\models \Psi_j \Leftrightarrow v_{\mathcal{I}}(z_i) = \bot$*

Proof. First note that the truth value of Ψ_j at z_i and $\bar{z}_i$ is the same, due to the structure of paths and the fact that Ψ_j begins with operator EX. Therefore, both statements of the lemma are actually equivalent. Their proof is done by induction on i.

- $i = 1$: Any formula Ψ_j with $1 \leq j \leq p$ holds from z_1 iff q_0 satisfies $\mathsf{EF}_{[C_j]} q_F$. And given the definition of C_j and the structure of any path starting from q_0, this is equivalent to $q_0 \models \mathsf{EF}_{[\widetilde{\varphi_1}]} q_F$. And this last requirement is clearly equivalent to the existence of some valuation for X^1 to satisfy φ_1. Finally note that $\bar{z}_1 \not\models \Psi_j$ is equivalent to $q_0 \models \neg\mathsf{EF}_{[\widetilde{\varphi_1}]} q_F$ and then $v_{\mathcal{I}}(z_1) = \bot$.

- $i > 1$: Knowing whether $z_i \models \Psi_j$ is equivalent to $q_{i-1} \models \mathsf{E}(\bar{z} \Rightarrow \neg\Psi_{j-1})\mathsf{U}_{[C'_j]}\, q_F$ where C'_j is the constraint $C'_j = \bigwedge_{\ell \le i}(\sharp z_\ell = 1 \Rightarrow \widetilde{\varphi_\ell}) \wedge \widetilde{\varphi_i}$. This entails that there exists a path ρ leading to q_F and defining a valuation v_ρ such that:
  - for any visited z_ℓ with $\ell < i$, we have $v_\rho \models \varphi_\ell$;
  - for any visited $\bar{z}_\ell$ with $\ell < i$, $\bar{z}$ is true, and then $\neg\Psi_{j-1}$ holds from $\bar{z}_\ell$. By induction hypothesis we have $v_{\mathcal{I}}(z_\ell) = \bot$; and
  - $v_\rho \models \varphi_i$.

  These three conditions define a valuation v_ρ that coincides with $v_{\mathcal{I}}$ for $\{z_{i-1}, \dots, z_1\}$ and such that there exists a compatible valuation for satisfying φ_i, thus $v_{\mathcal{I}}(z_i) = \top$. Now if $\bar{z}_i \not\models \Psi_j$, then $q_{i-1} \models \neg\mathsf{E}\big(\bar{z} \Rightarrow \neg\Psi_{j-1}\big)\mathsf{U}_{[C'_j]}\, q_F$ and then $v_{\mathcal{I}}(z_i) \neq \top$. □

It is now sufficient to check whether q_0 satisfies Ψ_p or not, and then deduce the truth value of $v_{\mathcal{I}}(z_p)$. □

Note that in the previous proof, one does not use sums in the constraints to get the complexity lower bound.

Theorem 4.9. *The model-checking problem for* CCTL *is* Δ_2^P*-hard.*

Proof. We provide a reduction from the model checking problem for TCTL specifications over Durational Kripke structures. TCTL formulas allow to deal with the cost (or duration) of paths (*i.e.* the sum of the weight of every transition occurring along the path). This problem is Δ_2^{P}-complete [LMS06]. Let $\mathcal{S} = (Q, R_{\mathcal{S}}, \ell)$ be a DKS. Let W be the set of weights occurring in $\mathcal{S}$. We define the Kripke structure $\mathcal{S}' = (Q', R_{\mathcal{S}'}, \ell')$ as follows:

- $Q' = Q \cup \{(q, d, q') \mid \exists (q, d, q') \in R_{\mathcal{S}}\}$,
- for any $(q, d, q') \in R_{\mathcal{S}}$, we add $(q, (q, d, q'))$ and $((q, d, q'), q')$ in $R_{\mathcal{S}'}$; and
- $\ell' : Q' \to 2^{\mathsf{AP}'}$ with $\mathsf{AP}' = \mathsf{AP} \cup \{\mathsf{ok}\} \cup \{P_d \mid d \in W\}$, assuming $\mathsf{ok}, P_d \notin \mathsf{AP}$. And we have: $\ell'(q) = \ell(q) \cup \{\mathsf{ok}\}$ for any $q \in Q$, and $\ell'(q, d, q') = \{P_d\}$.

We also inductively define $\widetilde{\Phi}$ for any TCTL formula Φ as: $\widetilde{P} = P$, $\widetilde{\neg\psi} = \neg\widetilde{\psi}$, $\widetilde{\varphi \wedge \psi} = \widetilde{\varphi} \wedge \widetilde{\psi}$, $\widetilde{\mathsf{E}\varphi\mathsf{U}_{\sim c}\psi} = \mathsf{E}(\mathsf{ok} \Rightarrow \widetilde{\varphi})\mathsf{U}_{[C_{\sim c}]}(\mathsf{ok} \wedge \widetilde{\psi})$ and $\widetilde{\mathsf{A}\varphi\mathsf{U}_{\sim c}\psi} = \mathsf{A}(\mathsf{ok} \Rightarrow \widetilde{\varphi})\mathsf{U}_{[C_{\sim c}]}(\mathsf{ok} \wedge \widetilde{\psi})$ with $C_{\sim c} = \sum_{d \in W} d \cdot \sharp P_d \sim c$.

Now we can easily see that $q \models_{\mathcal{S}} \Phi$ with $\Phi \in$ TCTL is equivalent to $q \models_{\mathcal{S}'} \widetilde{\Phi}$. □

Theorem 4.10. *The model-checking problem for* $\mathsf{CCTL}_\wedge$ *is in* Δ_2^P*.*

Proof. Let $\mathcal{S} = \langle Q, R, \ell \rangle$ be a Kripke structure. For this proof, by definition of Δ_2^{P}, it is sufficient to provide NP procedures to deal with sub-formulas of the form $\mathsf{EF}_{[C]}\varphi$ and $\mathsf{EG}_{[C]}\varphi$ (Cf. Rem.2.6). First let $\{C_1, \dots, C_m\}$ be the set of atomic constraints occurring in C. Each C_i is of the form $\sum_{j \in [1,n_i]} \alpha_j^i \cdot \sharp\varphi_j^i \sim_i k_i$. And let k be the maximal integer constant occurring in C. We can now present the algorithms:

- $\Phi = \mathsf{EF}_{[C]}\psi$: If $q \models \Phi$, then there exists a run $\rho q'$ starting from q such that $q' \models \psi$ and $\rho \models C$. First note that we can assume that the length of ρ is bounded with respect to the model and formula (more specifically by $m \cdot |Q| \cdot (k+1)$): a sequence of $|Q|$ states contributes for at least 1 to some linear expressions in C (loops containing only 0-states can be avoided since they do not contribute to the satisfaction of C) and every atomic constraint in C needs at most to collect a total weight of $k+1$. Hence the length of ρ is in $O(|Q|.2^{|C|})$ due to the binary encoding of the constants.

An easy NP algorithm consists in guessing the Parikh image[6] $F_\rho : R \to \mathbb{N}$ of the sequence of transitions in ρ, where $F_\rho(r)$ with $r \in R$ is the number of occurrences of transition r in ρ. As the length of ρ is bounded by $m \cdot |Q| \cdot (k+1)$, F_ρ can be represented in polynomial size. Moreover one can check in polynomial time that:
- q' satisfies ψ,
- ρ satisfies C, since $\sharp\varphi^i_j = \sum_{r \mid r \models \varphi^i_j} \sum_{(r,r') \in R} F_\rho(r, r')$.
- F_ρ corresponds to a correct path in $\mathcal{S}$ (by verifying that the sub-graph induced by F_ρ is connected and then applying the Euler circuit theorem).

- $\Phi = \mathsf{EG}_{[C]}\psi$: For this case we have to find an infinite path ρ satisfying the property "whenever the current prefix satisfies C then the next state has to satisfy ψ".

  Every atomic constraint C_i in C may change its truth value at most twice along ρ. Therefore ρ can be decomposed in at most $3m$ parts $(\rho_j)_{j \in [1,3m]}$ along each of which the truth value of every C_i is constant. Of course a part can be empty (restricted to a single state) and the last part must contain a cycle to ensure that ρ is infinite.

  As previously, the length of every ρ_j is bounded and its Parikh image can be encoded in polynomial size. Moreover it is possible to ensure that each ρ_j ends at the starting state of ρ_{j+1}. Finally we can also compute the truth value of C over any sequence $\rho_1 \dots \rho_j$ and then verify whether ψ holds for any state in such a sequence if necessary. □

A direct corollary of Theorems 4.7, 4.9 and 4.10 is:

Corollary 4.11. *The model-checking problem for* CCTL, $\mathsf{CCTL}_{\wedge 1}$, $\mathsf{CCTL}_{\wedge}$ *is* Δ_2^P*-complete.*

4.3. **Undecidability.**

Theorem 4.12. *The model-checking problem for* $\mathsf{CCTL}_{\wedge \pm 1}$ *is undecidable.*

Proof. This is done by reduction from the halting problem of a two-counter machine $\mathcal{M}$ with counters C and D, and n instructions $I_1, \dots, I_n$. Each I_i is either a decrement $\langle$`if` X`=0 then` j `else` X`--,` $k\rangle$ where X stands for C or D, an increment $\langle X$`++,` $j\rangle$, or the halting instruction $\langle$`halt`$\rangle$. We define a Kripke structure $\mathcal{S}_\mathcal{M} = (Q, R, \ell)$, where $Q = \{q_1, \dots, q_n\} \cup \{r_i, s_i, t_i \mid I_i = \langle$`if` $\dots\rangle\}$. The transition relation is defined as follows:
- if $I_i = \langle X$`++,` $j\rangle$, then $(q_i, q_j) \in R$; and
- if $I_i = \langle$`if` X`=0 then` j `else` X`--,`$k\rangle$, then (q_i, r_i), (r_i, q_k), (q_i, s_i), (s_i, t_i), and (t_i, q_j) in R.

The labeling ℓ is defined over the set $\{\mathsf{halt}, \mathsf{C}^\oplus, \mathsf{C}^\ominus, \mathsf{C}^0, \mathsf{C}^{\bar{0}}, \mathsf{D}^\oplus, \mathsf{D}^\ominus, \mathsf{D}^0, \mathsf{D}^{\bar{0}}\}$ as $\ell(q_i) = \{X^\oplus\}$ if I_i is an increment of X, $\ell(r_i) = \{X^\ominus\}$, $\ell(s_i) = \{X^{\bar{0}}\}$ and $\ell(t_i) = \{X^0\}$ if I_i is a decrement for X, and $\ell(q_i) = \{\mathsf{halt}\}$ if I_i is the halting instruction.

A run going through s_i and t_i for some i will simulate the positive test "$X = 0$": we use the propositions $X^{\bar{0}}$ and X^0 to observe this fact. Indeed along any run in $\mathcal{S}_\mathcal{M}$, a prefix satisfies $\sharp X^{\bar{0}} > \sharp X^0$ if and only if that prefix ends in some state s_i, which witnesses the fact that the counter's value was deemed equal to zero. The propositions on the other states are self-explanatory, witnessing increments and decrements of counters.

[6]Recall that the Parikh image of a sequence u over some alphabet A is the function mapping each symbol in A to its number of occurrences in u. This is also equivalently seen as a vector of dimension $|A|$ called the Parikh vector of u.

Checking $\mathsf{CCTL}_{\wedge\pm1}$ on this structure solves the halting problem, since $\mathcal{M}$ does not halt *if and only if* $q_1 \models_{\mathcal{S}_\mathcal{M}} \mathsf{EG}_{[C]}\bot$ with the following constraint:

$$C = (\sharp\mathsf{halt} \geq 1) \ \vee \bigvee_{X\in\{\mathtt{C},\mathtt{D}\}} \Big((\sharp X^{\oplus} - \sharp X^{\ominus} < 0) \ \vee \ (\sharp X^{\oplus} - \sharp X^{\ominus} > 0 \wedge \sharp X^{\bar{0}} - \sharp X^{0} > 0) \Big)$$

This formula states that there exists a run where C is consistently false, where C is true either if the run terminates, or if the simulation of $\mathcal{M}$ is wrong because the number of decrements is at some point larger than the number of increments, or because some counter was incorrectly assumed to be zero while simulating a test. □

5. Satisfiability

Here we address the satisfiability problem: given a formula Φ, does there exist a Kripke structure $\mathcal{S} = \langle Q, R, \ell\rangle$ with a state $q \in Q$ such that $q \models \Phi$?

For branching-time temporal logics, satisfiability problems are often harder than model checking (contrary to linear-time temporal logics) [Eme90], this is also the case for our counting logics. As soon as diagonal constraints are allowed (as in $\mathsf{CCTL}_{\pm1}$ or $\mathsf{CCTL}_{\pm}$), satisfiability is undecidable: this can be easily shown by adapting the undecidability proof of $\mathsf{CCTL}_{\wedge\pm1}$ model checking:

Theorem 5.1. *The satisfiability problem for $\mathsf{CCTL}_{\pm1}$ is undecidable.*

Proof. As in the proof of Theorem 4.12, consider a two-counter machine $\mathcal{M}$ with counters C and D, and n instructions $I_1, \ldots, I_n$. We build a $\mathsf{CCTL}_{\pm1}$ formula $\Phi_\mathcal{M}$ that is satisfiable iff $\mathcal{M}$ halts.

We use the following set of atomic propositions: $\mathsf{AP} = \{q_1, \ldots, q_n, \mathsf{C}^{\oplus}, \mathsf{C}^{\ominus}, \mathsf{C}^{0}, \mathsf{D}^{\oplus}, \mathsf{D}^{\ominus}, \mathsf{D}^{0}, \mathsf{halt}\}$. The $\mathsf{CCTL}_{\pm1}$ formula $\Phi_\mathcal{M}$ describes a linear KS whose every state is labeled by exactly one q_i corresponding to the current state of $\mathcal{M}$ and one proposition in $\mathcal{P} = \{\mathsf{C}^{\oplus}, \mathsf{C}^{\ominus}, \mathsf{C}^{0}, \mathsf{D}^{\oplus}, \mathsf{D}^{\ominus}, \mathsf{D}^{0}, \mathsf{halt}\}$ that indicates the operation that has to be done ($X^{\oplus}$ and $X^{\ominus}$ are used to mark increment and decrement of X, and X^0 labels states corresponding to an instruction "if $X == 0 \ldots$" when the current value of X is 0). In the following we use $\mathcal{I}_\mathcal{M}(X)$ (resp. $\mathcal{T}_\mathcal{M}(X)$) to denote the set of instruction numbers corresponding to an increment (resp. a test) of counter X. $\Phi_\mathcal{M}$ is the conjunction of the following formulae:

(1) $\mathsf{AG}\Big(\bigvee_{i=1\ldots n} (q_i \wedge \bigwedge_{j\neq i} \neg q_j) \Big)$

(2) $\mathsf{AG}\Big(\bigvee_{p\in\mathcal{P}} (p \wedge \bigwedge_{p'\in\mathcal{P}\setminus\{p\}} \neg p') \Big)$

(3) for every instruction, we have a step formula Φ_i:

$$\Phi_i = \begin{cases} \mathsf{AG}\big(q_i \ \Rightarrow \ (X^{\oplus} \wedge \mathsf{AX}\, q_j)\big) & \text{if } I_i = \langle X\texttt{++}, \ j\rangle \\ \mathsf{AG}\Big(q_i \Rightarrow \big((X_0 \wedge \mathsf{AX}\, q_j) \vee (X^{\ominus} \wedge \mathsf{AX}\, q_k)\big)\Big) & \text{if } I_i = \langle \texttt{if } X\texttt{=0 then } j \texttt{ else } X\texttt{--,}k\rangle \\ \mathsf{AG}\big(q_i \Rightarrow (\mathsf{halt} \wedge \mathsf{AX}\, \mathsf{halt})\big) & \text{if } I_i = \langle \texttt{halt}\rangle \end{cases}$$

(4) no zero test succeeds when the actual value of the corresponding counter is strictly positive (i.e. after a prefix witnessing strictly more increments than decrements),

and no decrement is performed when that value is 0:

$$\bigwedge_{X\in\{C,D\}} \left(\mathsf{AG}_{[\sharp X^{\oplus}-\sharp X^{\ominus}>0]}(\neg X^0) \ \wedge \ \mathsf{AG}_{[\sharp X^{\oplus}=\sharp X^{\ominus}]}(\neg X^{\ominus})\right)$$

(5) AF halt

Clearly Φ is satisfiable by a finite KS iff $\mathcal{M}$ terminates. □

For logics with no diagonal constraints, satisfiability remains decidable, with an additional cost compared to classical CTL.

Theorem 5.2. *The satisfiability problems for logics ranging from CCTL_1 to $\mathsf{CCTL}_\wedge$ are 2-EXPTIME-complete .*

Proof. Hardness comes from the complexity of $RTCTL^=$ satisfiability [EMSS92]: this logic is an extension of CTL with an Until operator equipped with constraints of the form "$=k$" over the number of transitions leading to the state satisfying the right part of the Until. This result is based on an encoding of an exponential space alternating Turing machine by a $RTCTL^=$ formula. Clearly, $RTCTL^=$ is included in CCTL_1.

2-EXPTIME membership directly follows from the translation given in Lemma 3.1: any $\mathsf{CCTL}_\wedge$ formula can be translated into CTL and the resulting formula's DAG-size is in $O(2^{|\Phi|^2})$. It remains to use an exponential algorithm for CTL satisfiability to obtain a 2-EXPTIME procedure (note that considering DAG-size instead of standard size does not matter for the complexity of the CTL procedure: indeed, in [KVW00] for instance, the size of the alternating tree automaton built from a given CTL formula is its number of distinct subformulae). □

6. Extensions

In the semantics of CCTL modalities, each new path quantifier resets the counting along a run, or more precisely starts counting anew on the remaining portion of the run. This restriction is quite significant, and ensures in particular that CCTL is a *state-based* temporal logic. Under some circumstances (as well as for the sake of completeness), it could be useful to relax this hypothesis and consider logics in which nested modalities do not necessarily reset the counting process.

In this section, we define two logics that allow this behaviour. The first one, called $\mathsf{CCTL}^{\mathsf{v}}$, uses explicit variables to keep track of the number of times a sub-formula was made true along the current run since the variable was bound. The second logic, called $\mathsf{CCTL}^{\mathsf{c}}$ uses a special reset modality and a different, cumulative semantics for $\mathsf{U}_{[C]}$, where counting ranges over the whole portion of the run since the last reset (hence potentially since the very beginning of the run). This logic is interpreted over states with a history.

6.1. Explicit variables.

Instead of using counting constraints associated with temporal modalities, we now consider a logic equipped with explicit *variables* and constraints directly stated inside formulas.

Definition 6.1. Given a set of atomic propositions AP and a countable set of variables V, we denote by $\mathsf{CCTL}^{\mathsf{v}}$ the set of formulas of the form

$$\varphi, \psi \ ::= \ P \mid \varphi \vee \psi \mid \neg\varphi \mid \mathsf{E}\varphi\mathsf{U}\psi \mid \mathsf{A}\varphi\mathsf{U}\psi \mid z[\psi].\varphi \mid \sum_{i=1}^{\ell} \alpha_i \cdot z_i \sim c$$

where $P \in \mathsf{AP}$, $z, z_i \in V$, $\ell, \alpha_i, c \in \mathbb{N}$ and $\sim \in \{<, \leq, =, \geq, >\}$.

Intuitively $z[\psi].\varphi$ means that variable z is defined and may be used in formula φ, where it will stand for the number of times formula ψ was observed to be true along the current run since z was defined.

More precisely, when the above formula is evaluated in a certain state, (1) variable z is reset to zero and bound to the sub-formula ψ, (2) at each subsequent step of a run, z is assigned the number of states in which formula ψ has held along this run since z was bound (*i.e.* the value of z evolves like $\sharp\psi$ as in Definition 2.3) and (3) given this semantics for z, φ holds in the current state.

Remark 6.2. The logic $\mathsf{CCTL}^{\mathsf{v}}$ can easily express any $\mathsf{CCTL}_{\wedge}$ property. Indeed, any $\mathsf{CCTL}_{\wedge}$ formula $\mathsf{E}\varphi\mathsf{U}_{[C]}\psi$, where C is a boolean combination $f(C_1, \ldots, C_m)$ of atomic constraints $C_i = \sum_{j=1}^{n_i} \sharp\varphi_j^i \sim k_i$, is equivalent to the $\mathsf{CCTL}^{\mathsf{v}}$ formula

$$z_1^1[\varphi_1^1].z_2^1[\varphi_2^1] \ldots z_{n_m}^m[\varphi_{n_m}^m].\mathsf{E}\varphi\mathsf{U}\big(\psi \wedge f(C_1', \ldots, C_m')\big)$$

where $C_i' = \sum_{j=1}^{n_i} z_j^i \sim k_i$ (and similarly for the A-quantified modality). This translation yields formulas whose size is linear in that of the original formulas.

For example, the $\mathsf{CCTL}_{\wedge 1}$ formula $\mathsf{EF}_{[\sharp P \leq 5 \wedge \sharp P' > 2]} P''$, stating that there exists a run along which a state satisfying P'' is reached after at most 5 occurrences of P and more than 2 occurrences of P', can be expressed in $\mathsf{CCTL}^{\mathsf{v}}$ as $z_1^1[P].z_2^1[P'].\mathsf{EF}(z_1^1 \leq 5 \wedge z_2^1 > 2 \wedge P'')$.

We first introduce some notations. Given a function $f : E \to F$, we denote by $\mathrm{dom}(f) \subseteq E$ the domain of f, and by $\mathrm{ran}(f) \subseteq F$ its range. For $x \in E$ and $a \in F$, let $f[x \leftarrow a]$ be the function mapping x to a and every $y \in \mathrm{dom}(f) \setminus \{x\}$ to $f(y)$, and $f|_D$ be the restriction of f to some subset D of E. Moreover we let $\mathrm{cl}(\varphi)$ be the set of all sub-formulas of φ and $V(\varphi)$ denote the set of all variables occurring in φ. An occurrence of some $z \in V(\Psi)$ is *bound* if it occurs in the right-hand side φ of some sub-formula $z[\psi].\varphi \in \mathrm{cl}(\Psi)$, and *free* otherwise. A variable is free in Ψ if it has at least one free occurrence. A formula without any free variable is called *closed*. Formally, the set $FV(\Psi) \subseteq V(\Psi)$ of free variables of Ψ is

$$\begin{array}{ll} FV(\varphi_1 \vee \varphi_2) = FV(\mathsf{E}\varphi_1\mathsf{U}\varphi_2) = FV(\mathsf{A}\varphi_1\mathsf{U}\varphi_2) = FV(\varphi_1) \cup FV(\varphi_2) & \\ FV(P) = \varnothing & FV(\textstyle\sum_{i=1}^{\ell} \alpha_i \cdot z_i \sim c) = \{z_i \mid i \in [1, \ell]\} \\ FV(\neg\varphi) = FV(\varphi) & FV(z[\psi].\varphi) = FV(\psi) \cup (FV(\varphi) \setminus \{z\}) \end{array}$$

Remark 6.3. In order to define the formal semantics of $\mathsf{CCTL}^{\mathsf{v}}$, one must be able to determine, in a given context, which sub-formula ψ is bound to each variable z. For simplicity, we will henceforth make the following two assumptions on the syntax of formulas:

(1) In any formula, every variable is bound at most once. In other words, every subformula $z[\psi].\varphi$ deals with a distinct variable z.
(2) In any formula Φ, there exists a (strict) total ordering $\prec$ on $V(\Phi)$ such that any formula bound to some variable z only contains occurrences of variables less than z, or more formally, for any sub-formula $z[\psi].\varphi$ of Φ, $z' \in V(\psi)$ implies $z' \prec z$.

Note that neither assumption restricts the expressiveness of the logic, since one may easily rename variable occurrences in any formula to fulfill constraint 1, and order variables according to an infix traversal of a formula's syntax tree to fulfill constraint 2.

We call *environment* any partial function $\varepsilon : V \to \mathsf{CCTL}^{\vee}$. A pair (Φ, ε) where Φ is a $\mathsf{CCTL}^{\vee}$ formula and ε is an environment, is a called a *closure*. We distinguish a specific class of closures, called *consistent*, defined as follows:

Definition 6.4. A $\mathsf{CCTL}^{\vee}$ closure (Φ, ε) is said to be *consistent* if

(1) $\mathrm{dom}(\varepsilon) \cap V(\Phi) = FV(\Phi)$;
(2) for all $z \in \mathrm{dom}(\varepsilon)$ and $z' \in FV(\varepsilon(z))$, $z' \in \mathrm{dom}(\varepsilon)$;
(3) for all $z \in \mathrm{dom}(\varepsilon)$ and $z' \in V(\varepsilon(z))$, $z' \prec z$.

Condition (1) guarantees that the environment for Φ defines at least all free variables in Φ (and potentially some additional variables not occurring in Φ) and does not redefine any of Φ's variables, condition (2) that ε does not refer to undefined variables and condition (3) that there are no cyclic definitions. Note that for any closed formula Φ, $(\Phi, \varepsilon_{\varnothing})$ is consistent, where $\varepsilon_{\varnothing}$ is the empty environment.

A consistent $\mathsf{CCTL}^{\vee}$ closure (φ, ε) is interpreted over a state of a Kripke structure extended with a valuation $v : V \to \mathbb{N}$ such that $\mathrm{dom}(v) = \mathrm{dom}(\varepsilon)$. Given a consistent closure (φ, ε), a valuation v such that $\mathrm{dom}(v) = \mathrm{dom}(\varepsilon)$, and a finite run π of a Kripke structure, let $v +_{\varepsilon} \pi$ be the valuation describing the values of variables in $\mathrm{dom}(v)$ *after* following π (*i.e.* once the states of π have all been visited and belong to the past): at each step along π, the value of every variable $z \in \mathrm{dom}(v)$ is updated to take into account the truth value of $\varepsilon(z)$. Formally $v +_{\varepsilon} \pi$ is defined inductively as: $v +_{\varepsilon} \pi = v$ if $|\pi| = 0$ (*i.e.* π is the empty sequence), and $(v +_{\varepsilon} \pi \cdot r)(z) = v'(z) + 1$ if $(r, v', \varepsilon) \models \varepsilon(z)$ (the satisfaction relation $\models$ is defined below) and $(v +_{\varepsilon} \pi \cdot r)(z) = v'(z)$ otherwise, where v' is the valuation $v +_{\varepsilon} \pi$ and r is a state.

Definition 6.5. The following clauses define the satisfaction of a consistent $\mathsf{CCTL}^{\vee}$ closure (φ, ε) from the state q of some Kripke structure $\mathcal{S} = \langle Q, R, \ell \rangle$ under valuation v with $\mathrm{dom}(v) = \mathrm{dom}(\varepsilon)$ – written $(q, v, \varepsilon) \models_{\mathcal{S}} \varphi$ – by induction over the structure of φ (we omit the cases of Boolean modalities):

$$
\begin{array}{lll}
(q, v, \varepsilon) \models_{\mathcal{S}} z[\psi].\varphi & \text{iff} & (q, v[z \leftarrow 0], \varepsilon[z \leftarrow \psi]) \models_{\mathcal{S}} \varphi, \\
(q, v, \varepsilon) \models_{\mathcal{S}} \sum_{i=1}^{\ell} \alpha_i \cdot z_i \sim c & \text{iff} & \sum_{i=1}^{\ell} \alpha_i \cdot v(z_i) \sim c, \\
(q, v, \varepsilon) \models_{\mathcal{S}} \mathsf{E}\varphi\mathsf{U}\psi & \text{iff} & \exists \rho \in \mathrm{Runs}(q) \text{ s.t. } (\rho, v, \varepsilon) \models_{\mathcal{S}} \varphi\mathsf{U}\psi, \\
(q, v, \varepsilon) \models_{\mathcal{S}} \mathsf{A}\varphi\mathsf{U}\psi & \text{iff} & \forall \rho \in \mathrm{Runs}(q), \text{we have } (\rho, v, \varepsilon) \models_{\mathcal{S}} \varphi\mathsf{U}\psi,
\end{array}
$$

where

$$
\begin{array}{lll}
(\rho, v, \varepsilon) \models_{\mathcal{S}} \varphi\mathsf{U}\psi & \text{iff} & \exists i \geq 0 \text{ s.t. } (\rho(i), v +_{\varepsilon} \rho|_{i-1}, \varepsilon) \models_{\mathcal{S}} \psi \\
 & \text{and} & \forall\, 0 \leq j < i,\ (\rho(j), v +_{\varepsilon} \rho|_{j-1}, \varepsilon) \models_{\mathcal{S}} \varphi.
\end{array}
$$

When there is no risk of confusion, we may omit subscript $\mathcal{S}$, and simply write $(q, v, \varepsilon) \models \varphi$. For any closed formula Φ, only the state q is relevant and we will simply write $q \models_{\mathcal{S}} \Phi$, or directly $q \models \Phi$. Remark that, when evaluating a closed formula according to the above semantic rules, only consistent closures are built and considered.

Finally, as a technical tool for the following proofs, we consider the *set of relevant variables* of a closure, that is the set of variables whose current value is required to decide

whether the formula holds for a given state. Given a consistent closure (Φ, ε), we define $RV(\Phi, \varepsilon)$ as follows:

$$RV(z[\psi].\varphi, \varepsilon) = RV(\varphi, \varepsilon[z \leftarrow \psi])\backslash\{z\} \tag{6.1}$$

$$RV(\mathsf{E}\varphi_1\mathsf{U}\varphi_2, \varepsilon) = RV(\mathsf{A}\varphi_1\mathsf{U}\varphi_2, \varepsilon) = RV(\varphi_1 \vee \varphi_2, \varepsilon) = RV(\varphi_1, \varepsilon) \cup RV(\varphi_2, \varepsilon) \tag{6.2}$$

$$RV(\neg\varphi, \varepsilon) = RV(\varphi, \varepsilon) \tag{6.3}$$

$$RV(P, \varepsilon) = \varnothing \tag{6.4}$$

$$RV(z_i \sim c, \varepsilon) = \{z_i\} \cup RV(\varepsilon(z_i), \varepsilon) \tag{6.5}$$

Note that relevant variables in formula ψ are only added to $RV(z[\psi].\varphi, \varepsilon)$ when z_i occurs in formula φ, i.e. in case (6.5) above. Clearly $FV(\Psi) \subseteq RV(\Psi, \varepsilon) \subseteq V(\Psi)$. Moreover by Def. 6.4, for every $z' \in RV(\varepsilon(z), \varepsilon)$, $z' \prec z$.

Example 6.6. Consider the consistent closure (Ψ, ε) with

$$\Psi = z_4[P'].\mathsf{EF}(z_4 \geq 2 \wedge z_2 = 4) \quad \text{and} \quad \varepsilon = \{z_1 \mapsto P, z_2 \mapsto \mathsf{EX}(z_1 > 2), z_3 \mapsto P''\},$$

we have $FV(\Psi) = \{z_2\}$ and $RV(\Psi, \varepsilon) = \{z_2, z_1\}$ because z_1 occurs free in $\varepsilon(z_2)$, hence $RV(\varepsilon(z_2), \varepsilon) = \{z_1\}$ by Eq. (6.5). Of course z_3 belongs to neither set because it occurs nowhere, and z_4 because it is bound in Ψ and $RV(\varepsilon(z_4), \varepsilon) = \varnothing$.

Given a closure (Ψ, ε) and a valuation v, we denote by v_Ψ the restriction $v_{|RV(\Psi,\varepsilon)}$ of v to the domain $RV(\Psi, \varepsilon)$ (and ε_Φ is the corresponding restriction of ε). The set $RV(\Psi, \varepsilon)$ contains the relevant variables for evaluating Ψ as stated by the following lemma.

Lemma 6.7. *For any consistent closure (Ψ, ε), the closure (Ψ, ε_Ψ) is consistent. Moreover, let v be a valuation over* $\mathrm{dom}(\varepsilon)$ *and q a state,*

$$(q, v, \varepsilon) \models \Psi \iff (q, v_\Psi, \varepsilon_\Psi) \models \Psi.$$

The proof of this lemma is straightforward. In the remainder of this section, we will study the expressiveness of this logic, as well as the complexity of its model-checking and satisfiability problems.

6.1.1. *Expressiveness.* Similarly to CCTL formulas without diagonal constraints, we show in this section that any closed CCTL$^\vee$ formula can be translated into an equivalent CTL formula.

Proposition 6.8. *For every closed CCTL$^\vee$ formula Φ, there exists an equivalent CTL formula of dag-size $2^{O(|\Phi|^2)}$.*

Before presenting the actual translation, we show that variable values may be bounded without changing the satisfaction of a formula. For a valuation v and an integer K, let us denote by v_K the restriction of v to the domain $\{z \in \mathrm{dom}(v) \mid v(z) \leq K\}$.

Lemma 6.9. *Let (φ, ε) be a consistent CCTL$^\vee$ closure, and K the maximal constant occurring in a constraint in φ or ε. For all Kripke structure $\mathcal{S}$, state q of $\mathcal{S}$ and valuations v and v' over* $\mathrm{dom}(\varepsilon)$*, we have:*

$$v_K = v'_K \implies \big((q, v, \varepsilon) \models_{\mathcal{S}} \varphi \iff (q, v', \varepsilon) \models_{\mathcal{S}} \varphi\big).$$

Proof. A reformulation of $v_K = v'_K$ is $v(z) \leq K \Rightarrow v(z) = v'(z)$. For each free variable z whose value by v is greater than K, the truth value of any constraint where z occurs will be the same for $v(z)$ and any other value greater than K, in particular $v'(z)$, since the constant in the right-hand side of the constraint is at most K. This is true in q, and remains true along any run from q. □

For any consistent closure (φ, ε) and for some valuation v with $\mathrm{dom}(v) = \mathrm{dom}(\varepsilon) = RV(\varphi, \varepsilon)$, we define the CTL translation $[\![\varphi]\!]^v_\varepsilon$ by induction on the structure of φ. The case of boolean connectives and atomic formulas is trivial:

$$[\![\psi_1 \wedge \psi_2]\!]^v_\varepsilon = [\![\psi_1]\!]^{v_{\psi_1}}_{\varepsilon_{\psi_1}} \wedge [\![\psi_2]\!]^{v_{\psi_2}}_{\varepsilon_{\psi_2}} \qquad [\![P]\!]^v_\varepsilon = P \qquad [\![\neg\varphi]\!]^v_\varepsilon = \neg[\![\varphi]\!]^v_\varepsilon \tag{6.6}$$

Variable definitions and constraints are also straightforward. It suffices to update and use the valuation and environment suitably:

$$[\![z[\varphi].\psi]\!]^v_\varepsilon = [\![\psi]\!]^{v[z\leftarrow 0]}_{\varepsilon[z\leftarrow\varphi]} \qquad [\![\textstyle\sum_i \alpha_i \cdot z_i \sim c]\!]^v_\varepsilon = \begin{cases} \top & \text{if } \sum_i \alpha_i \cdot v(z_i) \sim c \\ \bot & \text{otherwise} \end{cases} \tag{6.7}$$

Dealing with temporal modalities is more complex, and justifies the introduction of auxiliary formulas. Similarly to the translation of CCTL to CTL, the idea is to successively evaluate each formula $\varepsilon(z)$ which is *relevant* to the truth value of the whole formula, and to update the valuation accordingly. However, since variable values strictly larger than K (where K is the largest constant occurring in the formula or the environment) are all equivalent according to the previous proposition, it is only useful to evaluate formulas $\varepsilon(z)$ such that $v(z) \leq K$.

$$[\![\mathsf{E}\varphi\mathsf{U}\psi]\!]^v_\varepsilon = \mathsf{E}\Big([\![\varphi]\!]^{v_\varphi}_{\varepsilon_\varphi} \wedge \Theta^v_\varepsilon\Big)\mathsf{U}\Big[[\![\psi]\!]^{v_\psi}_{\varepsilon_\psi} \vee \Big([\![\varphi]\!]^{v_\varphi}_{\varepsilon_\varphi} \wedge \Gamma^v_\varepsilon\big(\mathsf{E}\varphi\mathsf{U}\psi, \mathrm{dom}(v_K), v\big)\Big)\Big] \tag{6.8}$$

with $\Theta^v_\varepsilon = \bigwedge_{z\in\mathrm{dom}(v_K)} \left(\neg[\![\varepsilon(z)]\!]^{v_{\varepsilon(z)}}_{\varepsilon_{\varepsilon(z)}}\right)$ and, for $Z \neq \varnothing$, $z \in Z$ and v' a valuation:

$$\begin{aligned}\Gamma^v_\varepsilon(\mathsf{E}\varphi\mathsf{U}\psi, Z, v') = \Big(\neg[\![\varepsilon(z)]\!]^{v_{\varepsilon(z)}}_{\varepsilon_{\varepsilon(z)}} \wedge \Gamma^v_\varepsilon(\mathsf{E}\varphi\mathsf{U}\psi, Z\setminus\{z\}, v')\Big) \\ \vee \Big([\![\varepsilon(z)]\!]^{v_{\varepsilon(z)}}_{\varepsilon_{\varepsilon(z)}} \wedge \Gamma^v_\varepsilon(\mathsf{E}\varphi\mathsf{U}\psi, Z\setminus\{z\}, v'[z \leftarrow v(z)+1])\Big)\end{aligned} \tag{6.9}$$

and finally:

$$\Gamma^v_\varepsilon(\mathsf{E}\varphi\mathsf{U}\psi, \varnothing, v') = \begin{cases} \bot & \text{if } v = v', \\ \mathsf{EX}[\![\mathsf{E}\varphi\mathsf{U}\psi]\!]^{v'}_\varepsilon & \text{otherwise.} \end{cases} \tag{6.10}$$

Finally, given a closed CCTL$^\vee$ formula Φ, we define its CTL translation $[\![\Phi]\!]$ as $[\![\Phi]\!]^\varnothing_\varnothing$.

Intuitively, the above translation of *until* modalities with valuation v and environment ε works by distinguishing *interesting* states, in which the value of at least one variable in $\mathrm{dom}(v_K)$ changes, from uninteresting ones. The CCTL$^\vee$ formula $\mathsf{E}\varphi\mathsf{U}\psi$ then holds if, and only if, after a finite sequence of uninteresting states satisfying φ, either ψ holds or the run has reached an interesting state satisfying φ, after which $\mathsf{E}\varphi\mathsf{U}\psi$ holds with a suitably updated valuation.

Formula Θ^v_ε in Eq. (6.8) expresses the fact that the current state is uninteresting, and $\Gamma^v_\varepsilon(\mathsf{E}\varphi\mathsf{U}\psi, \mathrm{dom}(v_K), v)$ that the current state is interesting, in other words satisfies at least one of the formulas $\varepsilon(z)$ for z a variable with value at most K in v, and satisfies $\mathsf{EX}[\![\mathsf{E}\varphi\mathsf{U}\psi]\!]^v_\varepsilon$. For such a state it is necessary to know exactly which formulas $\varepsilon(z)$ are satisfied and this is done by scanning the set $\mathrm{dom}(v_K)$, updating the valuation v' for each z in turn whenever $\varepsilon(z)$ is attested to hold (Eq. (6.9)). If no $\varepsilon(z)$ holds in the current state, which is witnessed

by the fact that $v = v'$, the state is in fact uninteresting and the whole scanning fails, otherwise the unfolding process continues (Eq. (6.10)).

Note how v and ε are restricted to relevant variables at every recursive call to the above translation procedure (for instance in $[\![\varepsilon(z)]\!]^{v_{\varepsilon(z)}}_{\varepsilon_{\varepsilon(z)}}$). This precaution is used to avoid cycles in the update of variables. It is necessary, since simply translating $\varepsilon(z)$ with an environment and valuation containing z itself may generate an infinite formula. It is also sufficient, since by definition of consistent closures, $z \notin RV(\varepsilon(z), \varepsilon)$.

Formulas $[\![\mathsf{A}\varphi\mathsf{U}\psi]\!]^v_\varepsilon$ and $\Gamma^v_\varepsilon(\mathsf{A}\varphi\mathsf{U}\psi, Z, v')$ are defined similarly by replacing each occurrence of E with A in the above formulas.

Lemma 6.10. *The above inductive definition for $[\![\Phi]\!]$ is well-founded, in other words $[\![\Phi]\!]$ is a finite CTL formula. The DAG-size of $[\![\Phi]\!]$ is in $2^{O(|\Phi|^2)}$.*

Proof. In Equations (6.6) and (6.7), all inductive uses of the translation function are performed over strictly shorter formulas. Even though this is not the case in Eq. (6.10), no recursive call is made unless the valuation v' used in Eq. (6.10) is different from (hence necessarily strictly greater than) v. Since variables assigned a value greater than K do not belong to $\mathrm{dom}(v_K)$, this set will eventually become empty, meaning that no state is considered interesting after some point. Hence no infinite inductive "call" to $[\![\mathsf{E}\varphi\mathsf{U}\psi]\!]^v_\varepsilon$ is possible. Finally, the definition of $\Gamma^v_\varepsilon(\varphi, Z, v')$ only refers to formulas $\Gamma^v_\varepsilon(\varphi, Z', v')$ with Z' strictly included in Z.

The maximal number of distinct valuations v we need to consider is bounded by $(K+3)^n$ (since each of the n variables can assume a value between 0 and $K+1$ or be undefined). Since each $\Gamma^v_\varepsilon(\varphi, Z, v')$ is indexed by two valuations v and v', one sub-formula φ (of which there are at most $|\Phi|$) and a set of variables Z (at most 2^n possibilities), the total number of distinct such formulas to consider is less than $((K+3)^n)^2 \cdot |\Phi| \cdot 2^n$. Overall, since $K \in O(2^{|\Phi|})$ due to the binary encoding and $n \in O(\Phi)$, this yields a worst-case DAG-size for $[\![\Phi]\!]$ in $O(|\Phi| \cdot (2^{|\Phi|}+3)^{2|\Phi|} \cdot 2^{|\Phi|}) \subseteq 2^{O(|\Phi|^2)}$. □

We have the following correctness lemma:

Lemma 6.11. *Let (Φ, ε) be a consistent CCTL$^\vee$ closure, K the maximal constant in Φ and ε. For every Kripke structure $\mathcal{S}$, state q of $\mathcal{S}$ and $(K+1)$-bounded valuation v we have:*

$$(q, v, \varepsilon) \models_{\mathcal{S}} \Phi \iff q \models_{\mathcal{S}} [\![\Phi]\!]^{v_\Phi}_{\varepsilon_\Phi}.$$

Proof. The proof of the direct implication is done by structural induction over Φ. We only detail the cases of variable definition and temporal modalities.

- $\Phi = z[\varphi].\psi$: Assume $(q, v, \varepsilon) \models z[\varphi].\psi$. This is semantically equivalent to $(q, v[z \leftarrow 0], \varepsilon[z \leftarrow \varphi]) \models \psi$. By induction hypothesis $q \models [\![\psi]\!]^{v[z\leftarrow 0]_\psi}_{\varepsilon[z\leftarrow\varphi]_\psi}$, hence $q \models [\![\Phi]\!]^{v_\Phi}_{\varepsilon_\Phi}$.
- $\Phi = \mathsf{E}\varphi\mathsf{U}\psi$: Assume $(q, v, \varepsilon) \models \mathsf{E}\varphi\mathsf{U}\psi$. There exists a run $\rho = q_0 q_1 q_2 \dots$ with $q_0 = q$ and an index $i \geq 0$ such that $(q_i, v +_\varepsilon \rho|_{i-1}, \varepsilon) \models \psi$ and for all $0 \leq j < i$, $(q_j, v +_\varepsilon \rho|_{j-1}, \varepsilon) \models \varphi$. For every $0 \leq j < i$, let v_j be the valuation $v_\Phi +_{\varepsilon_\Phi} \rho|_{j-1}$, and Z_j be the set of variables z such that $v_j(z) \leq K$ and $v_{j+1}(z) = v_j(z) + 1$, i.e. the set of relevant variables whose value is incremented in state q_j.
  Let $j_1, \dots, j_\ell$ be the positions in $\{0, \dots, i-1\}$ along ρ where Z_{j_h} is non-empty. We reason by induction over ℓ. If $\ell = 0$, then clearly $q \models \mathsf{E}([\![\varphi]\!]^{v_\varphi}_{\varepsilon_\varphi} \wedge \Theta^{v_\Phi}_{\varepsilon_\Phi})\mathsf{U}[\![\psi]\!]^{v_\psi}_{\varepsilon_\psi}$, and

thus $q \models [\![\Phi]\!]^{v_\Phi}_{\varepsilon_\Phi}$. Now assume $\ell > 0$, we have: $Z_j = \varnothing$ for $0 \leq j < j_1$, $Z_{j_1} \neq \varnothing$, and:

$$q_{j_1} \models \underbrace{\bigwedge_{z \in Z_{j_1}} \varepsilon_\Phi(z)}_{\Phi_1} \wedge \underbrace{\bigwedge_{z \in \mathrm{dom}(\varepsilon_\Phi) \setminus Z_{j_1}} \neg \varepsilon_\Phi(z)}_{\Phi_2}$$

Moreover we have $(q_{j_1+1}, v_{j_1+1}, \varepsilon) \models \mathsf{E}\varphi\mathsf{U}\psi$. By induction hypothesis over ℓ we have $q_{j_1+1} \models [\![\mathsf{E}\varphi\mathsf{U}\psi]\!]^{v_{j_1+1}}_{\varepsilon_\Phi}$ and thus: $q \models \mathsf{E}([\![\varphi]\!]^{v_\varphi}_{\varepsilon_\varphi} \wedge \Theta^{v_\Phi}_{\varepsilon_\Phi})\mathsf{U}(\Phi_1 \wedge \Phi_2 \wedge \mathsf{EX}[\![\mathsf{E}\varphi\mathsf{U}\psi]\!]^{(v_{j_1+1})}_{\varepsilon_\Phi})$. Therefore we have $q \models [\![\Phi]\!]^{v_\Phi}_{\varepsilon_\Phi}$.

- $\Phi = \mathsf{A}\varphi\mathsf{U}\psi$: in this case, every run from q has to verify $\varphi\mathsf{U}\psi$. We can reuse the same approach as before. In the general case, every run starts with a prefix along which every state q_j is such that Z_j is empty, followed by some state q_{j_1} where $Z_{j_1} \neq \varnothing$, which satisfies $\mathsf{AX}[\![\mathsf{A}\varphi\mathsf{U}\psi]\!]^{(v_{j_1+1})_\Phi}_{\varepsilon}$.

The converse is also done by structural induction on Φ. The case where $\Phi = z[\varphi].\psi$ follows the same reasoning as above, only backwards. When $\Phi = \mathsf{E}\varphi\mathsf{U}\psi$, we reason by induction on the following (well-founded) ordering of valuations. We write $v' \trianglelefteq v$ whenever $\mathrm{dom}(v'_K) \subseteq \mathrm{dom}(v_K)$ and $\forall x \in \mathrm{dom}(v'_K), v'(x) \geq v(x)$, meaning that v' assigns greater values than v to all variables to which v' assigns a value less than or equal to K, and $v' \triangleleft v$ if additionally $v' \neq v$. Assume $q \models [\![\Phi]\!]^v_\varepsilon$, and consider the iterative unfolding of the definitions of subformula Γ in $[\![\Phi]\!]^v_\varepsilon$. For this formula to hold, there must exist a satisfied formula Ψ, obtained by replacing each disjunction by one of its operands, resulting in a "witness" for the satisfaction of $[\![\Phi]\!]^v_\varepsilon$. Ψ is of one of the forms:

$$\Psi = \mathsf{E}\Big([\![\varphi]\!]^{v_\varphi}_{\varepsilon_\varphi} \wedge \Theta^v_\varepsilon\Big)\mathsf{U}\Big([\![\varphi]\!]^{v_\varphi}_{\varepsilon_\varphi} \wedge \bigwedge_{z \in Z} [\![\varepsilon(z)]\!]^{v_{\varepsilon(z)}}_{\varepsilon_{\varepsilon(z)}} \wedge \bigwedge_{z \in \mathrm{dom}(v_K) \setminus Z} \neg [\![\varepsilon(z)]\!]^{v_{\varepsilon(z)}}_{\varepsilon_{\varepsilon(z)}} \wedge \mathsf{EX}[\![\Phi]\!]^{v'}_\varepsilon\Big) \quad (6.11)$$

for some non-empty $Z \subseteq \mathrm{dom}(v_K)$, and with $v'(z) = v(z) + 1$ if $z \in Z$ and $v(z) \leq K$ and $v'(z) = v(z)$ otherwise, or

$$\Psi = \mathsf{E}\Big([\![\varphi]\!]^{v_\varphi}_{\varepsilon_\varphi} \wedge \Theta^v_\varepsilon\Big)\mathsf{U}[\![\psi]\!]^{v_\psi}_{\varepsilon_\psi}. \quad (6.12)$$

In the former case (Eqn. (6.11)), there must exist a run $\rho = q_0 q_1 \ldots$ and some $k \geq 0$ such that $q_i \not\models_{\mathcal{S}} [\![\varepsilon(z)]\!]^{v_{\varepsilon(z)}}_{\varepsilon_{\varepsilon(z)}}$ for all $i < k$ and $z \in \mathrm{dom}(v_K)$, $q_i \models_{\mathcal{S}} [\![\varphi]\!]^{v_\varphi}_{\varepsilon_\varphi}$ for all $i \leq k$, $q_k \models_{\mathcal{S}} [\![\varepsilon(z)]\!]^{v_{\varepsilon(z)}}_{\varepsilon_{\varepsilon(z)}}$ for all $z \in Z$, $q_k \not\models_{\mathcal{S}} [\![\varepsilon(z)]\!]^{v_{\varepsilon(z)}}_{\varepsilon_{\varepsilon(z)}}$ for all $z \in \mathrm{dom}(v_K) \setminus Z$, and $q_{k+1} \models_{\mathcal{S}} [\![\Phi]\!]^{v'}_\varepsilon$.

Since $Z \neq \varnothing$, we have $v' \triangleleft v$, hence by our induction hypotheses over the structure of Φ and the ordering of valuations, we obtain that $(q_i, v, \varepsilon) \not\models_{\mathcal{S}} \varepsilon(z)$ for all $i < k$ and $z \in \mathrm{dom}(v_K)$, $(q_i, v, \varepsilon) \models_{\mathcal{S}} \varphi$ for all $i \leq k$, $(q_k, v, \varepsilon) \models_{\mathcal{S}} \varepsilon(z)$ for all $z \in Z$, $(q_k, v, \varepsilon) \not\models_{\mathcal{S}} \varepsilon(z)$ for all $z \in \mathrm{dom}(v_K) \setminus Z$, and $(q_{k+1}, v', \varepsilon) \models_{\mathcal{S}} \Phi$.

Since the truth value of any subformula is independent of the variables which are irrelevant for that subformula or whose value is already greater than K at the beginning of the run, and given the truth values of formulas $\varepsilon(z)$ along ρ, this implies that $(q_{k+1}, v +_\varepsilon \rho|_k, \varepsilon) \models_{\mathcal{S}} \mathsf{E}\varphi\mathsf{U}\psi$ and $\forall i \leq k, (q_i, v +_\varepsilon \rho|_{i-1}, \varepsilon) \models_{\mathcal{S}} \varphi$, hence $(q_0, v, \varepsilon) \models_{\mathcal{S}} \Phi$, and this remains true for any valuation v'' and environment ε'' such that $v''_\Phi = v$ and $\varepsilon''_\Phi = \varepsilon$.

The latter case (Eqn. (6.12)) is easier and is solved similarly. As previously, the A quantifier is also treated in the same fashion. □

Example 6.12. For the $\mathsf{CCTL}^\vee$ formula $\Phi = z[P].z'[z > 0].\mathsf{EF}(z' > 0 \wedge P')$, we obtain (after simplification) the following translation:

$$[\![\Phi]\!] \stackrel{\text{def}}{=} \mathsf{E}\,(\neg P)\,\mathsf{U}\Big(P \wedge \mathsf{EX}\Big(\mathsf{EX}\,\mathsf{EF}P'\Big)\Big)$$

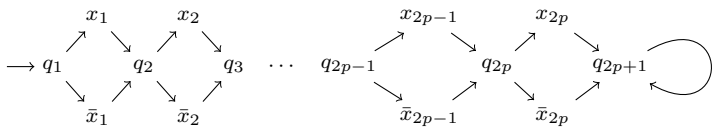


FIGURE 2. Kripke structure associated to a QBF instance over $\{x_1, \ldots, x_{2p}\}$.

The two nested EX modalities are necessary because one must distinguish the first state r where P holds true from its successor r', which is the first to satisfy $z > 0$, and from the successor r'' of r' which is the first state satisfying $z' > 0$.

6.1.2. *Model checking.*

Theorem 6.13. *Model checking closed $\mathsf{CCTL}^\vee$ formulas is PSPACE-complete.*

Proof. • PSPACE-hardness can be proved by a reduction from the quantified Boolean formula problem (QBF)[7]. Consider a QBF instance $\mathcal{I} = \exists x_1 \forall x_2 \ldots \exists x_{2p-1} \forall x_{2p} \cdot \Phi$ where Φ is a propositional formula in 3-conjunctive normal form (3-CNF) $\bigwedge_{j=1\ldots m}(\ell_1^j \vee \ell_2^j \vee \ell_3^j)$ over $\{x_1, \ldots, x_{2p}\}$. Now consider the KS $\mathcal{S}_\mathcal{I} = \langle Q, R, \ell \rangle$ in Figure 2. We assume that every state q_i is labeled with its name, and every state x_i (resp. $\bar{x}_i$) is labeled by the atomic proposition C_j iff x_i (resp. $\neg x_i$) is one of the literals in $\{\ell_1^j, \ell_2^j, \ell_3^j\}$. Then $\mathcal{I}$ is positive iff q_1 satisfies the following formula:

$$z_1[C_1] \cdots z_m[C_m] \cdot \mathsf{EF}\Big(q_2 \wedge \mathsf{AF}\Big(q_3 \wedge \mathsf{EF} \ldots \big(q_{2p} \wedge \mathsf{AF}(q_{2p+1} \wedge \bigwedge_{i=1\ldots m} (z_i \geq 1))\big)\Big)\Big)$$

• PSPACE-membership is obtained by considering a non-deterministic algorithm working in polynomial space to decide whether a closed $\mathsf{CCTL}^\vee$ formula Φ holds for a state q within a KS $\mathcal{S}$. This provides an NSPACE procedure which, by Savitch's theorem, implies the existence of a PSPACE algorithm.

We assume that Φ contains n variables $z_1, z_2, \ldots z_n$. Let us call *configuration* any triple (q, v, ε) where q is a state, v a valuation and ε an environment. First note that valuations can be encoded in space polynomial in $|\Phi|$ since it is sufficient to store the value for each variable z as a $K+1$-bounded counter, where K is the maximal constant occurring in Φ, which requires at most $|\Phi|$ bits per variable. Hence configurations can be encoded in space polynomial in $|\Phi|$ and linear in $|\mathcal{S}|$.

For any consistent closure (Ψ, ε) with $\Psi \in \mathrm{cl}(\Phi)$, we define an NSPACE procedure $\mathit{Check}(q, v, \varepsilon, \Psi)$ to decide whether Ψ holds over (q, v, ε). We consider several cases according to the structure of Ψ, of which we omit the simplest.

- $\Psi = z_i[\psi_i].\varphi_i$: the returned value is $\mathit{Check}(q, v[z_i \leftarrow 0], \varepsilon[z_i \leftarrow \psi_i], \varphi_i)$.
- $\Psi = \sum_{i=1}^{\ell} \alpha_i \cdot z_i \sim c$: the returned value is the boolean evaluation of the constraint $\sum_{i=1}^{\ell} \alpha_i \cdot v(z_i) \sim c$.
- $\Psi = \mathsf{E}\varphi_1 \mathsf{U} \varphi_2$: if $\mathit{Check}(q, v, \varepsilon, \varphi_2)$ is evaluated to $\top$, then the returned value is $\top$. Else if $\mathit{Check}(q, v, \varepsilon, \varphi_1)$ is $\bot$, then the result is $\bot$. Otherwise we proceed as follows:
  (1) for every $z_i \in RV(\Psi, \varepsilon)$, call $\mathit{Check}(q, v_{\varepsilon(z_i)}, \varepsilon_{\varepsilon(z_i)}, \varepsilon(z_i))$ and assign 1 to an integer variable δ_i if the result is $\top$, and 0 otherwise;

[7] This is a simplification of the reduction used for TCTL_c over KS [LST03].

(2) guess a transition $q \to q'$ in $\mathcal{S}$;
(3) replace the current configuration (q, v, ε) by (q', v', ε) with $v'(z_i) = \min(K + 1, v(z_i) + \delta_i)$ for all $z_i \in RV(\Psi, \varepsilon)$, and check whether φ_2 holds for it, and so on.

- $\Psi = \mathsf{EG}\varphi$: since there are finitely many $K+1$-bounded configurations, if there is a run for Ψ starting in q then there must also exist one whose corresponding sequence of bounded configurations is ultimately periodic, i.e. consists of a finite sequence of configurations followed by an infinite repetition of a finite configuration cycle (up to valuation equivalence). The procedure $\mathit{Check}(q, v, \varepsilon, \Psi)$ can thus consist of the following steps:
  (1) start guessing a sequence of transitions as in the previous case, updating the current state and valuation accordingly;
  (2) in each new configuration (q, v, ε), verify that $\mathit{Check}(q, v, \varepsilon, \varphi)$ is *top*;
  (3) at some point, non-deterministically assume the current (bounded) configuration to occur infinitely often in some ultimately periodic run satisfying Ψ, and store the corresponding state q_r and valuation v_r;
  (4) resume guessing transitions, checking at each step that $\mathit{Check}(q, v, \varepsilon, \varphi)$ is $\top$;
  (5) return $\top$ if the previously stored recurring configuration is ever encountered again.

Deciding $q \models \Phi$ is then achieved by calling $\mathit{Check}(q, v_0, \varepsilon, \Phi)$.

The space used by $\mathit{Check}(q, v_0, \varepsilon, \Phi)$ is evaluated as follows: for $\mathsf{E}\varphi_1\mathsf{U}\varphi_2$ or $\mathsf{EG}\varphi_1$, we need to store at most three configurations (q, v, ε) and k boolean values. We also need space for the recursive calls over subformulas. The maximal number of such nested calls is bounded[8] by $th(\Phi) + \sum_{i=1}^{n} th(\varepsilon(z_i))$: indeed the first term comes from the recursive calls for $\mathit{Check}(q, v, \varphi_i)$ and the second from the calls $\mathit{Check}(q, v, \varepsilon(z_i))$. Thus the maximal number of nested calls is bounded by $|\Phi|$. □

Remark 6.14. As soon as subtractions are allowed in $\mathsf{CCTL}^\vee$, model checking becomes undecidable as a simple consequence of Thm. 4.12 and Rem. 6.2.

6.1.3. *Satisfiability.* As in the case of CCTL, the translation of $\mathsf{CCTL}^\vee$ formulas into CTL provides an optimal decision procedure for satisfiability:

Theorem 6.15. *The satisfiability problem for $\mathsf{CCTL}^\vee$ is 2-EXPTIME-complete.*

Proof. A closed $\mathsf{CCTL}^\vee$ formula Φ is satisfiable (*i.e.* it holds for a state q in a finite KS $\mathcal{S}$) iff the CTL formula $[\![\Phi]\!]$ is satisfiable. The (DAG) size of $[\![\Phi]\!]$ is in $2^{O(|\Phi|^2)}$, which yields a 2EXPTIME procedure to decide satisfiability of Φ. Hardness is a consequence of Thm. 5.2 and Rem. 6.2. □

[8] where $th(\varphi)$ is the temporal height of φ defined as usual except for the reset operator for which we have: $th(z[\psi].\varphi) = th(\varphi)$.

6.2. **Cumulative semantics for CCTL.** We now define a variant of CCTL based on an alternative semantics for E_U_ and A_U_ modalities. In this semantics, nesting two temporal modalities no longer resets the counting process for the evaluation of the innermost modality: its constraints are then interpreted over the whole run. In order to relax this semantics, we add the modality N (for 'now', or rather 'from now on') which specifies that the counters have to be reset in the current state and start counting again from the current position. Let us fix the syntax of $\mathsf{CCTL}^{\mathsf{c}}$:

Definition 6.16. Given a set of atomic propositions AP, we define:

$$\mathsf{CCTL}^{\mathsf{c}} \ni \varphi, \psi \ ::= \ P \mid \varphi \wedge \psi \mid \neg\varphi \mid \mathsf{N}\varphi \mid \mathsf{E}\varphi\mathsf{U}^{\mathsf{c}}_{[C]}\psi \mid \mathsf{A}\varphi\mathsf{U}^{\mathsf{c}}_{[C]}\psi$$

with $P \in \mathsf{AP}$. As in the case of CCTL, we use shorthands $\mathsf{F}^{\mathsf{c}}_{[C]}\varphi$ and $\mathsf{G}^{\mathsf{c}}_{[C]}\varphi$ to denote $\top\mathsf{U}^{\mathsf{c}}_{[C]}\varphi$ and $\neg\mathsf{F}^{\mathsf{c}}_{[C]}\neg\varphi$ respectively.

$\mathsf{CCTL}^{\mathsf{c}}$ formulas are interpreted over pairs (π, q) where q is a state of some Kripke structure $\mathcal{S}$ and π is a history (*i.e.* a finite prefix) such that $\pi \cdot q \in \mathrm{Prefs}(\mathcal{S})$. The following clauses[9] define when a $\mathsf{CCTL}^{\mathsf{c}}$ formula Φ holds for (π, q):

$$\begin{array}{lll}
(\pi,q) \models_{\mathcal{S}} \mathsf{E}\varphi\mathsf{U}^{\mathsf{c}}_{[C]}\psi & \text{iff } \exists \rho \in \mathrm{Runs}(q),\ \exists i \geq 0, & (\pi \cdot \rho_{|i-1}, \rho(i)) \models_{\mathcal{S}} \psi,\ \pi \cdot \rho_{|i-1} \models_{\mathcal{S}} C, \\
 & & \text{and } \forall 0 \leq j < i,\ (\pi \cdot \rho_{|j-1}, \rho(j)) \models_{\mathcal{S}} \varphi \\
(\pi,q) \models_{\mathcal{S}} \mathsf{A}\varphi\mathsf{U}^{\mathsf{c}}_{[C]}\psi & \text{iff } \forall \rho \in \mathrm{Runs}(q),\ \exists i \geq 0, & (\pi \cdot \rho_{|i-1}, \rho(i)) \models_{\mathcal{S}} \psi,\ \pi \cdot \rho_{|i-1} \models_{\mathcal{S}} C, \\
 & & \text{and } \forall 0 \leq j < i,\ (\pi \cdot \rho_{|j-1}, \rho(j)) \models_{\mathcal{S}} \varphi \\
(\pi,q) \models_{\mathcal{S}} \mathsf{N}\varphi & \text{iff } (\epsilon, q) \models_{\mathcal{S}} \varphi &
\end{array}$$

The addition of the N modality allows us to easily express CCTL properties. Indeed each CCTL formula Φ can be easily translated into a $\mathsf{CCTL}^{\mathsf{c}}$ formula Ψ by guarding each of its temporal modalities with N. Both formulas are equivalent, in the sense that for any state q and history π, we have $q \models \Phi \iff (\pi, q) \models \Psi$. We also have the following useful property:

$$(\pi, q) \models \mathsf{E}\bot\mathsf{U}^{\mathsf{c}}_{[C]}\top \iff \pi \models C \tag{6.13}$$

For simplicity, in the following we will thus allow ourselves to directly write constraints in the formula and not only as subscripts of temporal modalities.

Example 6.17. The $\mathsf{CCTL}^{\mathsf{c}}$ formula $\mathsf{EF}^{\mathsf{c}}_{[\sharp\top \leq k_1]}(P_1 \wedge \mathsf{EF}^{\mathsf{c}}_{[\sharp\top \leq k_2]}P_2)$ with $k_1 \leq k_2$ holds for a state q if and only if there exists a run with less than k_2 transitions leading to some state satisfying P_2 and along this run there is a state satisfying P_1 located at less than k_1 transitions from q.

Example 6.18. The $\mathsf{CCTL}^{\mathsf{c}}$ formula $\mathsf{EF}^{\mathsf{c}}_{[\sharp\varphi \geq k_1]}\mathsf{EF}^{\mathsf{c}}_{[\sharp\varphi \leq k_2]}\psi$ is semantically equivalent to the $\mathsf{CCTL}_{\wedge 1}$ formula $\mathsf{EF}_{[k_1 \leq \sharp\varphi \leq k_2]}\psi$.

Proposition 6.19. *Model checking $\mathsf{CCTL}^{\mathsf{c}}$ is PSPACE-hard.*

Proof. We reduce the QBF problem to a model-checking problem for $\mathsf{CCTL}^{\mathsf{c}}$ by using exactly the same reduction as for $\mathsf{CCTL}^{\mathsf{v}}$ (Thm. 6.13): given an instance $\mathcal{I}$ of QBF, we consider the same KS $\mathcal{S}_{\mathcal{I}}$ and the following formula:

$$\mathsf{EF}\Big(q_2 \wedge \mathsf{AF}\Big(q_3 \wedge \mathsf{EF} \ldots \mathsf{AF}\Big(q_{2p+1} \wedge \bigwedge_{i=1\ldots m} (\sharp C_i \geq 1)\Big)\Big)\Big)$$

[9] As previously, we only give the formal semantics of the main modalities. Boolean connectives are interpreted in a natural way.

Recall that we can use constraints directly inside the formula due to the equivalence (6.13) above. □

Note that we do not use N to prove PSPACE-hardness. To prove membership in PSPACE, we show that one can translate any $\mathsf{CCTL}^{\mathsf{c}}$ formula φ into an equivalent (and succinct) $\mathsf{CCTL}^{\mathsf{v}}$ formula $\overline{\varphi}$. First given $\varphi \in \mathsf{CCTL}^{\mathsf{c}}$, we use $\mathsf{S}^{\sharp}_{\varphi}$ to denote the set of subformulas ψ of φ such that $\sharp\psi$ occurs in a counting constraint inside φ. We now define $\overline{\Phi}$ as follows:

$$\overline{P} = P \qquad \overline{\varphi \wedge \psi} = \overline{\varphi} \wedge \overline{\psi} \qquad \overline{\neg\varphi} = \neg\overline{\varphi}$$

$$\overline{\mathsf{N}\varphi} = z_{\psi_1}[\overline{\psi_1}]. \ldots z_{\psi_k}[\overline{\psi_k}].\overline{\varphi} \text{ with } \mathsf{S}^{\sharp}_{\varphi} = \{\psi_1, \ldots, \psi_k\}$$

$$\overline{\mathsf{E}\varphi\mathsf{U}^{\mathsf{c}}_{[C]}\psi} = \mathsf{E}\overline{\varphi}\mathsf{U}(\overline{C} \wedge \overline{\psi}) \qquad \overline{\mathsf{A}\varphi\mathsf{U}^{\mathsf{c}}_{[C]}\psi} = \mathsf{A}\overline{\varphi}\mathsf{U}(\overline{C} \wedge \overline{\psi})$$

$$\overline{\textstyle\sum_i \alpha_i \cdot \sharp\varphi_i \sim c} = \textstyle\sum_i \alpha_i \cdot z_{\varphi_i} \sim c$$

Given a set of formulas S, a prefix π, a valuation v for a set of variables V and an environment ε, we say that (v, ε) is *compatible* with (S, π) (written $(v, \varepsilon) \trianglerighteq (S, \pi)$) if and only if for any $\psi \in S$, there is some $z_\psi \in \mathrm{dom}(v)$ such that $\varepsilon(z_\psi) = \psi$ and $v(z_\psi) = |\pi|_\psi$.

We have the following property:

Lemma 6.20. *Let Φ be a $\mathsf{CCTL}^{\mathsf{c}}$ formula, q a state in some KS $\mathcal{S}$, and $\pi \in \mathrm{Prefs}(\mathcal{S})$ be a finite run such that $\pi \cdot q \in \mathrm{Prefs}(\mathcal{S})$. Let $v : V \to \mathbb{N} \cup \{\bot\}$ be a valuation for a set of variables V and let ε be an environment such that (v, ε) is compatible with $(\mathsf{S}^{\sharp}_{\Phi}, \pi)$. Then:*

$$(\pi, q) \models_{\mathcal{S}} \Phi \iff (q, v, \varepsilon) \models_{\mathcal{S}} \overline{\Phi}$$

Proof. The proof is done by structural induction over Φ. The result is direct for atomic propositions and boolean connectives.

Let $\Phi = \mathsf{E}\varphi\mathsf{U}^{\mathsf{c}}_{[C]}\psi$, and assume $(\pi, q) \models \Phi$. Then there exist $\rho \in \mathrm{Runs}(q)$ and $i \geq 0$ such that (a) $(\pi \cdot \rho_{|i-1}, \rho(i)) \models \psi$, (b) $\pi \cdot \rho_{|i-1} \models C$ and (c) for all $0 \leq j < i$ we have: $(\pi \cdot \rho_{|j-1}, \rho(j)) \models \varphi$. Consider a valuation v and an environment ε such that (v, ε) is compatible with $(\mathsf{S}^{\sharp}_{\Phi}, \pi)$. Let v_k be the valuation $(v +_\varepsilon \rho_{|k-1})$ for $k \in \{0, 1, \ldots, i\}$ (where $v_0 = v$). Clearly $(v_k, \varepsilon) \trianglerighteq (\mathsf{S}^{\sharp}_{\Phi}, \pi \cdot \rho_{|k-1})$, and since $\mathsf{S}^{\sharp}_{\psi} \subseteq \mathsf{S}^{\sharp}_{\Phi}$ and $\mathsf{S}^{\sharp}_{\varphi} \subseteq \mathsf{S}^{\sharp}_{\Phi}$, (v_k, ε) is compatible with $(\mathsf{S}^{\sharp}_{\psi}, \pi \cdot \rho_{|k-1})$ and $(\mathsf{S}^{\sharp}_{\varphi}, \pi \cdot \rho_{|k-1})$. By induction hypothesis, we can deduce from (a) and (c) that (a') $(\rho(i), v_i, \varepsilon) \models \overline{\psi}$, and (c') $(\rho(j), v_j, \varepsilon) \models \overline{\varphi}$ for any $j = 0, \ldots, i-1$. Moreover from (b) we can deduce: (b') $v_i \models \overline{C}$. Thus $(q, v, \varepsilon) \models \mathsf{E}\overline{\varphi}\mathsf{U}(\overline{C} \wedge \overline{\psi})$.

Conversely, assume $(q, v, \varepsilon) \models \mathsf{E}\overline{\varphi}\mathsf{U}(\overline{C} \wedge \overline{\psi})$. Then there exists $\rho \in \mathrm{Runs}(q)$ and $i \geq 0$ such that (a) $(q, v +_\varepsilon \rho_{|i-1}, \varepsilon) \models \overline{\psi}$, (b) $v +_\varepsilon \rho_{|i-1} \models \overline{C}$ and (c) for all $0 \leq j < i$ we have: $(q, v +_\varepsilon \rho_{|j-1}, \varepsilon) \models \overline{\varphi}$. Now consider a prefix π such that $\pi \cdot q \in \mathrm{Prefs}(\mathcal{S})$ and $(v, \varepsilon) \trianglerighteq (\mathsf{S}^{\sharp}_{\Phi}, \pi)$. By induction hypothesis, we have: $(\pi \cdot \rho_{|i-1}, \rho(i)) \models \psi$ and $(\pi \cdot \rho_{|j-1}, \rho(j)) \models \varphi$ for any $j = 0, \ldots, i-1$. Hence $(\pi, q) \models \mathsf{E}\varphi\mathsf{U}^{\mathsf{c}}_{[C]}\psi$.

The case $\Phi = \mathsf{A}\varphi\mathsf{U}^{\mathsf{c}}_{[C]}\psi$ is treated similarly.

Let now $\Phi = \mathsf{N}\varphi$, and assume $\mathsf{S}^{\sharp}_{\varphi} = \{\psi_1, \ldots, \psi_k\}$. Let ε be an environment such that $\mathsf{S}^{\sharp}_{\Phi} \subseteq \mathrm{dom}(\varepsilon)$. Then for any valuation v_0 that assigns 0 to every ψ_i,

$$(\pi, q) \models \mathsf{N}\varphi \iff (\epsilon, q) \models \varphi \iff (q, v_0, \varepsilon) \models \overline{\varphi}.$$

This is equivalent to $(q, v, \varepsilon) \models z_{\psi_1}[\overline{\psi_1}]. \ldots z_{\psi_k}[\overline{\psi_k}].\overline{\varphi}$ for any valuation v such that $(v, \varepsilon) \trianglerighteq (\mathsf{S}^{\sharp}_{\Phi}, \pi)$. □

In fact, $\mathsf{CCTL}^{\mathsf{c}}$ can be seen as a variant of $\mathsf{CCTL}^{\mathsf{v}}$ where only a *global* reset operator is available, whose effect corresponds to the N modality in $\mathsf{CCTL}^{\mathsf{c}}$. A direct consequence is:

Proposition 6.21. *The model checking problem for $\mathsf{CCTL}^{\mathsf{c}}_{\wedge}$ is in PSPACE.*

This implies the following corollary:

Corollary 6.22. *The model checking problem for all $\mathsf{CCTL}^{\mathsf{c}}$ variants up to $\mathsf{CCTL}^{\mathsf{c}}_{\wedge}$ is PSPACE-complete.*

Again, as soon as diagonal constraints are allowed model checking becomes undecidable:

Theorem 6.23. *Model checking $\mathsf{CCTL}^{\mathsf{c}}_{\pm 1}$ is undecidable.*

Proof. The proof is based on the same technique as that of Theorem 4.12. Consider a two-counter machine $\mathcal{M}$ with counters C and D and n instructions. We define a Kripke structure $\mathcal{S}_{\mathcal{M}} = \langle Q, R, \ell \rangle$ where $Q = \{q_1, \ldots, q_n\} \cup \{r_i, s_i \mid \texttt{inst}_i = \langle \texttt{if } \ldots \rangle\}$. The transition relation is defined as follows:

- if $\texttt{inst}_i = \langle X\texttt{++}, j \rangle$, then $(q_i, q_j) \in R$; and
- if $\texttt{inst}_i = \langle \texttt{if} X\texttt{=0 then } j \texttt{ else } X\texttt{--}, k \rangle$, then (q_i, r_i), (r_i, q_k), (q_i, s_i), (s_i, q_j) in R.

The labeling ℓ is defined over the set $\{\mathsf{halt}\} \cup \bigcup_{X \in \{\mathtt{C},\mathtt{D}\}} \{X^+, X^-, X^0\}$ as $\ell(q_i) = \{X^+\}$ if $\texttt{inst}_i$ is an increment of X, $\ell(r_i) = \{X^-\}$ and $\ell(s_i) = \{X^0\}$ if $\texttt{inst}_i$ is a conditional decrement of X and $\ell(q_i) = \{\mathsf{halt}\}$ if $\texttt{inst}_i$ is the halting instruction. One can show that there exists a divergent run iff q_1 satisfies the formula $\Phi_{\mathcal{M}}$ defined as follows:

$$\mathsf{EG}\Big[\neg\mathsf{halt} \wedge \bigwedge_{X \in \{\mathtt{C},\mathtt{D}\}} \Big(\big(X^0 \Rightarrow (\sharp X^+ = \sharp X^-)\big) \wedge \big(X^- \Rightarrow (\sharp X^+ > \sharp X^-)\big)\Big)\Big]$$

Note that we do not use N to prove undecidability. □

Using the same techniques as previously, we obtain the following results for satisfiability:

Theorem 6.24. *The satisfiability problem for all variants of $\mathsf{CCTL}^{\mathsf{c}}$ from $\mathsf{CCTL}^{\mathsf{c}}_1$ up to $\mathsf{CCTL}^{\mathsf{c}}_{\wedge}$ is 2EXPTIME-complete, and becomes undecidable for $\mathsf{CCTL}^{\mathsf{c}}_{\pm 1}$.*

7. Conclusion

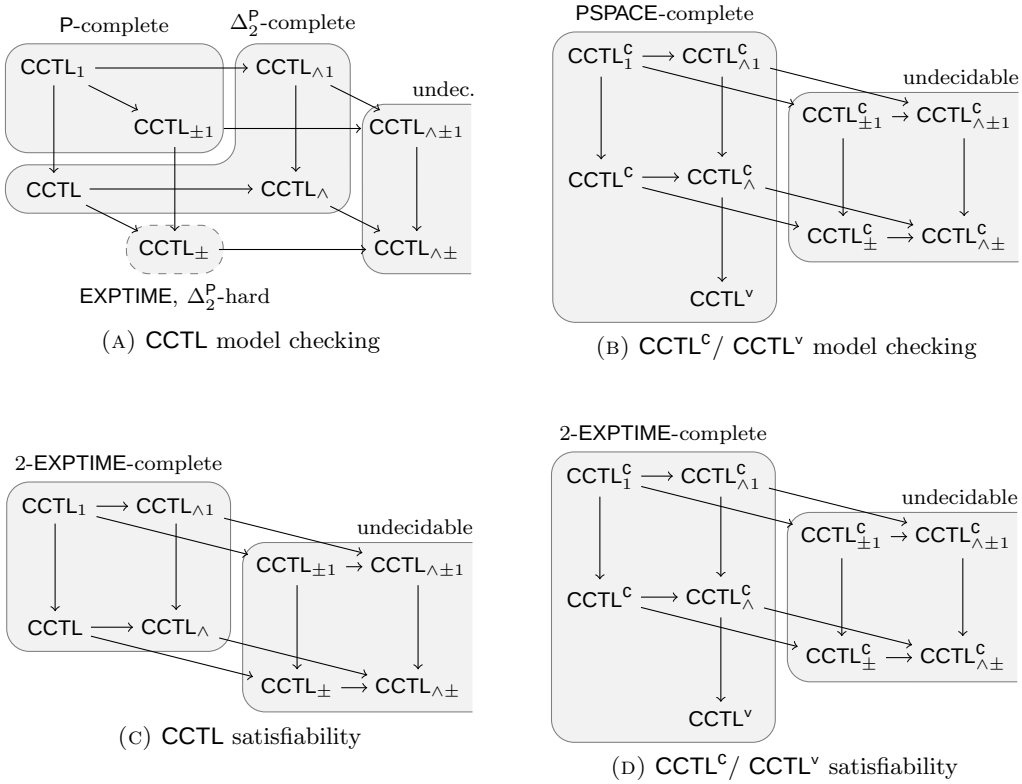
In several cases (particularly $\mathsf{CCTL}^{\mathsf{v}}$ and thus also $\mathsf{CCTL}_{\wedge}$ and $\mathsf{CCTL}^{\mathsf{c}}_{\wedge}$), the logics we introduce are not more expressive than CTL but can concisely express properties which would be difficult to write in that logic. In particular, even the fragment CCTL_1, as well as $\mathsf{CCTL}_{\wedge}$ with unary-encoded coefficients, can yield exponentially more succinct formulas than CTL.

In terms of algorithmic complexity, even though $\mathsf{CCTL}_{\pm 1}$ is strictly more expressive than CTL, its model-checking remains polynomial. The introduction of either coefficients or Boolean combinations increases the complexity to Δ_2^{P}, while the interplay between Boolean connectives and possibly negative coefficients yields undecidability. Similarly, satisfiability is 2-$\mathsf{EXPTIME}$-complete for all classes without negative coefficients (when it is simply $\mathsf{EXPTIME}$-complete for CTL [EH85]), and undecidable for all above classes. All complexity results are summarized in Figure 3.

Further work on CCTL will include completing the study of succinctness of its fragments with respect to each other and to other logics, looking for an upper complexity bound for the model-checking of $\mathsf{CCTL}_{\pm}$, as well as investigating new kinds of constraints. We also

wish to pursue the work described in this article and in [LMP10b] by investigating counting extensions of other temporal logics (for instance with past operators) as well as μ-calculus.

Acknowledgements. The authors would like to thank the anonymous referees for their very accurate and helpful comments.



(A) CCTL model checking

(B) $\mathsf{CCTL}^{\mathsf{c}}$/ $\mathsf{CCTL}^{\vee}$ model checking

(C) CCTL satisfiability

(D) $\mathsf{CCTL}^{\mathsf{c}}$/ $\mathsf{CCTL}^{\vee}$ satisfiability

FIGURE 3. Summary of model-checking and satisfiability complexity results. Arrows indicate syntactic inclusion or straight-forward linear translations (case of $\mathsf{CCTL}^{\vee}$).

REFERENCES

[ACD93] R. Alur, C. Courcoubetis, and D. L. Dill. Model-checking in dense real-time. *Inf. Comput.*, 104(1):2–34, 1993.

[AH93] R. Alur and T. A. Henzinger. Real-time logics: Complexity and expressiveness. *Inf. Comput.*, 104(1):35–77, 1993.

[AH94] R. Alur and T. A. Henzinger. A really temporal logic. *Journal of the ACM*, 41(1):181–203, 1994.

[AI03] M. Adler and N. Immerman. An $n!$ lower bound on formula size. *ACM Transactions on Computational Logic*, 4(3):296–314, 2003.

[BdA95] A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In *15th FSTTCS*, volume 1026 of *LNCS*, pages 499–513. Springer, 1995.

[BEH95a] A. Bouajjani, R. Echahed, and P. Habermehl. On the verification problem of nonregular properties for nonregular processes. In *Proc. 10th LICS*, pages 123–133. IEEE Comp. Soc. Press, 1995.

[BEH95b] A. Bouajjani, R. Echahed, and P. Habermehl. Verifying infinite state processes with sequential and parallel composition. In *Proc. 22nd POPL*, pages 95–106, 1995.

[BFH05] D. Bustan, D. Fisman, and J. Havlicek. Automata construction for PSL. Technical report, The Weizmann Institute of Science, 2005. Available as Tech. Report MCS05- 04.

[BVW94] O. Bernholtz, M. Vardi, and P. Wolper. An automata-theoretic approach to branching-time model-checking. In *Proc. 6th CAV*, volume 818 of *LNCS*, pages 142–155. Springer, 1994.

[CE81] E. M. Clarke and E. A. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In *Logics of Programs*, volume 131 of *LNCS*, pages 52–71. Springer, 1981.

[EH85] E. A. Emerson and J. Y. Halpern. Decision procedures and expressiveness in the temporal logic of branching time. *J. Comput. Syst. Sci.*, 30(1):1–24, 1985.

[Eme90] E. A. Emerson. Temporal and modal logic. In *Handbook of Theoretical Computer Science*, volume B, chapter 16, pages 995–1072. Elsevier Science, 1990.

[EMSS92] E. A. Emerson, A. K. Mok, A. P. Sistla, and J. Srinivasan. Quantitative temporal reasoning. *Real-Time Systems*, 4(4):331–352, 1992.

[ET97] E. A. Emerson and R. J. Trefler. Generalized quantitative temporal reasoning: An automata-theoretic approach. In *Proc. 7th TAPSOFT*, volume 1214 of *LNCS*, pages 189–200. Springer, 1997.

[ET99] E. A. Emerson and R. J. Trefler. Parametric quantitative temporal reasoning. In *Proc. 14th LICS*, pages 336–343. IEEE Comp. Soc. Press, 1999.

[KVW00] O. Kupferman, M. Y. Vardi, and P. Wolper. An automata-theoretic approach to branching-time model checking. *Journal of the ACM*, 47(2):312–360, 2000.

[LMP10a] F. Laroussinie, A. Meyer, and E. Petonnet. Counting ctl. In C.-H. Luke Ong, editor, *FOSSACS*, volume 6014, pages 206–220. Springer, 2010.

[LMP10b] F. Laroussinie, A. Meyer, and E. Petonnet. Counting LTL. *International Syposium on Temporal Representation and Reasoning (TIME'10)*, pages 51–58, 2010.

[LMS01] F. Laroussinie, N. Markey, and Ph. Schnoebelen. Model checking CTL^+ and FCTL is hard. In *Proc. 4th FoSSaCS*, volume 2030 of *LNCS*, pages 318–331. Springer, 2001.

[LMS06] F. Laroussinie, N. Markey, and Ph. Schnoebelen. Efficient timed model checking for discrete-time systems. *Theor. Comput. Sci.*, 353(1-3):249–271, 2006.

[LS00] F. Laroussinie and Ph. Schnoebelen. Specification in CTL+Past for verification in CTL. *Inf. Comput.*, 156(1/2):236–263, 2000.

[LST03] F. Laroussinie, Ph. Schnoebelen, and M. Turuani. On the expressivity and complexity of quantitative branching-time temporal logics. *Theor. Comput. Sci.*, 297(1–3):297–315, 2003.

[Pnu77] A. Pnueli. The temporal logic of programs. In *Proc. 18th FOCS*, pages 46–57. IEEE Comp. Soc. Press, 1977.

[psl03] *Property Specification Language Reference Manual, Version 1.1*, 2003. `http://www.eda-stds.org/vfv/docs/PSL-v1.1.pdf`.

[QS82] J.-P. Queille and J. Sifakis. Specification and verification of concurrent systems in CESAR. In *Proc. 5th Int. Symp. on Programming*, volume 137 of *LNCS*, pages 337–351. Springer, 1982.

[Sch03] Philippe Schnoebelen. The complexity of temporal logic model checking. In Philippe Balbiani, Nobu-Yuki Suzuki, Frank Wolter, and Michael Zakharyaschev, editors, *Selected Papers from the 4th Workshop on Advances in Modal Logics (AiML'02)*, pages 393–436, Toulouse, France, 2003. King's College Publication. Invited paper.

[Wil99] T. Wilke. CTL^+ is exponentially more succinct than CTL. In *Proc. 19th FSTTCS*, volume 1738 of *LNCS*, pages 110–121. Springer, 1999.

[Wol83] P. Wolper. Temporal logic can be more expressive. *Inf. and Control*, 56(1/2):72–99, 1983.

[YMW97] J. Yang, A. K. Mok, and F. Wang. Symbolic model checking for event-driven real-time systems. *ACM Transactions on Programming Languages and Systems*, 19(2):386–412, 1997.