CWI

Computable Semantics for CTL* on Discrete-Time and Continuous-Space Dynamic Systems

P.J. Collins, I.S. Zapreev

MAS-1001

Centrum Wiskunde & Informatica

Centrum Wiskunde & Informatica (CWI) is the national research institute for Mathematics and Computer Science. It is sponsored by the Netherlands Organisation for Scientific Research (NWO).
CWI is a founding member of ERCIM, the European Research Consortium for Informatics and Mathematics.

CWI's research has a theme-oriented structure and is grouped into four clusters. Listed below are the names of the clusters and in parentheses their acronyms.

Probability, Networks and Algorithms (PNA)

Software Engineering (SEN)

Modelling, Analysis and Simulation (MAS)

Information Systems (INS)

P.O. Box 94079, 1090 GB Amsterdam (NL)
Science Park 123, 1098 XG Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-3703

International Journal of Foundations of Computer Science

Computable Semantics for CTL^* on Discrete-Time and Continuous-Space Dynamic Systems *

Pieter Collins and Ivan S. Zapreev

Centrum Wiskunde & Informatica,
Science Park 123, 1098 XG Amsterdam
Pieter.Collins@cwi.nl, I.Zapreev@cwl.nl

Received (Day Month Year)
Accepted (Day Month Year)
Communicated by (xxxxxxxxxx)

In this work, we consider Discrete-Time Continuous-Space Dynamic Systems for which we study the computability of the standard semantics of CTL^* (CTL, LTL) and provide a variant thereof computable in the sense of Type-2 Theory of Effectivity. In particular, we show how the computable model checking of existentially and universally quantified path formulae of LTL can be reduced to solving, respectively, repeated reachability and persistence problems on a model obtained as a parallel composition of the original one and a non-deterministic Büchi automaton corresponding to the verified LTL formula.

Keywords: Computability; Model Checking; CTL; LTL; CTL^*; Computable semantics; Dynamic Systems

1. Introduction

This paper summarises the computable semantics of CTL^* on Discrete-Time Continuous-Space Dynamic Systems (DTCSDSs). CTL^* is a popular modal logic and DTCSDSs are used for modelling and analysis in biology, chemistry and engineering. The importance of this work lays in the fact that the computable semantics naturally induces computable model-checking algorithms. Remember that, for a DTCSDS model $\mathcal{M}$ and $\phi \in CTL^*$, model checking uses an exhaustive state-space exploration to answer: "Does $\mathcal{M}$ satisfy the system property ϕ?" This question is typically put as a formula: $\mathcal{M} \models \phi$, where $\models$ is a satisfiability relation.

The model-checking algorithms for CTL^* are induced by its semantics as well as by the considered system model. In order to be implementable on digital computers, these algorithms should be in some sense computable (semi-decidable). Yet, the state space of a DTCSDS is continuous and the ordinary computability and complexity theory is not powerful enough to express the computability of real-valued

*This research was supported by the Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO) Vidi grant 639.032.408.

functions and sets of any continuous domain. Therefore, we choose to use Type-2 Theory of Effectivity (TTE) [17] which defines computability based on Turing machines with finite and infinite input/output sequences. In this work, we analyse the standard semantics of CTL^* on DTCSDS with respect to computability and then provide a computable (semi-decidable) semantics thereof in the sense of TTE.

In [9] we devised a computable semantics for CTL on discrete-time continuous-space dynamic systems (DTCSDSs). Employing path spaces, in [10] this semantics was extended to a computable semantics of CTL^*. Our motivation was based on the fact that CTL^* is a strict super set of LTL and CTL, i. e. it allows for a wider range of reachability properties by combining linear- and branching-time semantics. The resulting computable semantics for CTL^* was not optimal due to a conservative approximation for the set of path of the henceforth operator. In particular, it could lead to inconclusive model-checking results even on finite-state models. The novelty of this work is that we give a new computable semantics for CTL^* such that it is exact on finite-state models. The latter is done by reducing the problem of verifying $\forall\phi/\exists\phi$ on a DTCSDS M to a simpler problem of verifying $\forall\Diamond\Box F/\exists\Box\Diamond F$ on a model obtained using a parallel composition of M and the non-deterministic Büchi automaton $\mathcal{L}_{\mathrm{A}_{\phi_n}}/\mathcal{L}_{\mathrm{A}_{\phi}}$. Here, $Sat(F)$ is open; ϕ and $\phi_n := \neg\phi$ are in NNF.

The computable semantics provided in this paper are topological, see also [12, 1]. Due to (necessary) choices, our computable semantics does not preserve the Law of Excluded Middle. Also, if the verified formula contains negation, henceforth or release operators then it can be true but not computably verifiable on a given model.

This paper is organised as follows: Section 2 contains preliminary material. Section 3 outlines the computable semantics of CTL. Section 4 states how to propagate quantifiers inside CTL^*. Section 5 provides the path-space-based computable semantics of CTL^*. Section 6 shows that this semantics can result in inconclusive mode checking on finite-state models. Section 7 provides a new computable semantics for CTL^* that is exact on finite-state models. Section 7.3 contains examples illustrating the superiority of the new approach and Section 8 concludes.

2. Preliminaries

Section 2.1 recalls the basics of the topology theory (heavily used in TTE). Section 2.2 discusses multivalued maps (used to represent DTCSDS models), their provides and continuity. Section 2.3 talks about TTE and computability of various sets and operations. Section 2.4 recalls the standard semantics of CTL, LTL and CTL^*. Section 2.5 outlines CTL^* model checking and introduces Büchi automata.

2.1. *Topological Spaces*

A *topological space* is a pair $T = (X, \tau)$ where X is an arbitrary set and $\tau \subseteq 2^X$ is such that: $\emptyset, X \in \tau$, $\forall U_1, U_2 \in \tau \Rightarrow U_1 \cap U_2 \in \tau$, and $\forall \mathbb{U} \subseteq \tau \Rightarrow \bigcup_{U\in\mathbb{U}} U \in \tau$. For a topological space T, elements of τ are called *open* and their complements in X are

called *closed*. Let $x \in X$ and $B \subseteq X$ then B is a *neighborhood* of point x if there exists an open set $U \in \tau$ such that $x \in U \subseteq B$. Let $B \subseteq X$ and $\mathbb{U} \subseteq \tau$ then $\mathbb{U}$ is an *open cover* of B if $B \subseteq \bigcup_{U \in \mathbb{U}} U$. Let $S \subseteq X$, then the set $\mathrm{Int}(S) = \bigcup \{U | U \subseteq S \wedge U \in \tau\}$ is called the *interior* of S and $\mathrm{Cl}(S) = \bigcap \{A | S \subseteq A \wedge A \text{ is closed}\}$ is called the *closure* of S. *Overtness* is an equivalence relation on sets defined by the following property. Let $S_1, S_2 \subseteq X$ then $S_1 \sim_o S_2$ iff $\{U \in \tau | S_1 \cap U \neq \emptyset\} = \{U \in \tau | S_2 \cap U \neq \emptyset\}$. The latter is equivalent to saying that $S_1 \sim_o S_2$ iff $\mathrm{Cl}(S_1) = \mathrm{Cl}(S_2)$. Therefore, for any $S \subseteq X$ the equivalence class $\{S\}_{\sim_o}$ is uniquely identified by its representative: $\mathrm{Cl}(S)$. Further, when we call the set S *overt*, we imply that it is only known up to its equivalence class. In other words, the only available information about S is provided by the set $\{U \in \tau | S \cap U \neq \emptyset\}$. Similar to overtness, *compactness* can be also seen as an equivalence relation on sets but for simplicity, we provide the classical definition: A set $C \subseteq X$ is *compact* iff every open cover of C has a finite sub cover. A subset of X is *pre-compact* iff its closure is compact. For T we define: $\mathcal{O}$ – a set of open, $\mathcal{A}$ – a set of closed, $\mathcal{K}$ – a set of compact and $\mathcal{V}$ – a set of overt sets.

Let $T = (X, \tau)$ be a topological space. Then $\beta \subseteq \tau$ is a *base* of the topology τ if every element of τ can be represented as a union of elements from β. A topological space is called *second countable* if its topology has a countable base. A *Hausdorff space (T_2 space)* is a topological space such that $\forall x, y \in X$ where $x \neq y$ there exist $U_x, U_y \in \tau$ such that $x \in U_x$, $y \in U_y$ and $U_x \cap U_y = \emptyset$.

A *path space* is a topological space (X^ω, τ^ω) where X^ω is the countable Cartesian product of X. Let $\sigma \in X^\omega$ and $\sigma = s_0 s_1 s_2 \ldots$ then $\forall i \in \mathbb{N}$ we define the *canonical projection* $p_i : X^\omega \to X$ such that $p_i(\sigma) = s_i$. Let τ be a topology on X, and any $U^\omega \in \tau^\omega$ be a countable (or finite) union of finite intersection of sets from $B^\omega := \{B_U^\omega \subseteq X^\omega | \exists n \in \mathbb{N} : ((\forall i < n : p_i(B_U^\omega) \in \tau) \wedge (\forall i \geq n : p_i(B_U^\omega) = X))\}$. Then, τ^ω is called a *product topology* on X^ω (induced by τ). The product topology is the coarsest topology for which all the projections p_i are continuous, and in addition every p_i is an open-valued map. If (X, τ) is second-countable Hausdorff space then (X^ω, τ^ω) with the product topology τ^ω is also second-countable and Hausdorff.

2.2. *DTCSDSs and Multivalued Maps*

We consider discrete-time continuous-space dynamic systems (DTCSDSs) for which the state-space is continuous and the time domain is discrete (the system state changes at discrete time points). In system theory, dynamic systems are given by functions $f : X \times U \to X$, where X is the state space, and U can either represent control or system noise. These functions are typically converted into multivalued maps $F : X \rightrightarrows X$ such that $F(x) = f(x, U)$.

A *multivalued map* $F : X \rightrightarrows Y$, also known as multivalued function or multifunction, is a total relation on $X \times Y$. If we define $F(S) = \{F(x) | x \in S\}$ for $S \subseteq X$ then F can be seen as a function $F : X \to 2^Y$. This last definition is more convenient when we want to talk about function composition. For example, for two multivalued maps $F : X \rightrightarrows Y$, $G : Y \rightrightarrows Z$ and their composition $G \circ F$ we have

$G \circ F : X \rightrightarrows Z$ and thus for any $x \in X$ we can simply write $G \circ F\,(x) = G\,(F\,(x))$. A *weak preimage* of F on $B \subseteq Y$ is $F^{-1}(B) = \{x \in X \,:\, F(x) \cap B \neq \emptyset\}$ and a *strong preimage* is $F^{\Leftarrow}(B) = \{x \in X \,:\, F(x) \subseteq B\}$. F is continuous iff it is both upper semicontinuous (USC) and lower semicontinuous (LSC). F is USC *iff* $F^{\Leftarrow}(U)$ is open if $U \subseteq Y$ is open, and F is LSC *iff* $F^{-1}(U)$ is open if $U \subseteq Y$ is open.

2.3. *Type-2 Theory of Effectivity (TTE)*

TTE [17] is based on Turing machines that allow for infinite inputs and outputs. Let M be a type-2 Turing machine with a fixed finite alphabet Σ, $k \geq 0$ input tapes, one output tape and $Y_i \in \{\Sigma^*, \Sigma^\omega\}$ where $i \in 0, \ldots, k$. Then, a (partial) string function $f_M : Y_1 \times \ldots \times Y_k \to Y_0$ is *computable* iff it is realised by a type-2 machine M. The latter means that for $y_i \in Y_i$ we have $f_M\,(y_1, \ldots, y_k) = y_0 \in \Sigma^*$ iff M halts on input $(y_1, \ldots, y_k)$ with y_0 on the output tape and $f_M\,(y_1, \ldots, y_k) = y_0 \in \Sigma^\omega$ iff M computes forever on input $(y_1, \ldots, y_k)$ and writes y_0 to the output.

The computability on Σ^* and Σ^ω is generalised by means of notations and representations. A *notation* of set X is a partial surjective function $\nu : \Sigma^* \to X$ and a *representation* is a partial surjective function $\delta : \Sigma^\omega \to X$. These functions encode elements of the domain X into strings and sequences which are called *names*.

A *computable Hausdorff space* is a tuple $T := (X, \tau, \beta, \nu)$ such that (X, τ) is a second-countable Hausdorff (T_2) space; β is a countable base of τ consisting of pre-compact open sets; ν is a notation of β; we take effectivity properties in [4] (Lemma 2.3) as axioms; and assume that $\mathrm{Cl} : \beta \to \mathcal{K}$ is computable.

In computability theory, cf. [8], the sets $\mathcal{O}$, $\mathcal{A}$, $\mathcal{K}$ and $\mathcal{V}$ can be seen as types. Every type defines a particular way of identifying its elements. For example, for a computable Hausdorff space T, any open set $U \in \mathcal{O}$ is identified by the (countable) sequence of names of the base elements $\{\beta_i\}_{i \in I}$, defined by ν, such that $U = \bigcup_{i \in I} \beta_i$. Then, any closed set $A \in \mathcal{A}$ can be identified by the list of names of all $U \in \mathcal{O}$ such that $U \cap A \neq \emptyset$. The latter forms a name of the set A. Further, if we write, e. g., A is *effectively* $\mathcal{A}$ it will mean that A has a computable name as defined by type $\mathcal{A}$.

For a computable Hausdorff space and the Sierpinski space $\mathcal{S}$, the following operations are *(effectively) computable* in a sense that given the names of the arguments we can compute the name of the result: countable union as $\mathcal{O} \times \mathcal{O} \to \mathcal{O}$, complement as $\mathcal{O} \to \mathcal{A}$, subset operation as $\mathcal{K} \times \mathcal{O} \to \mathcal{S}$, intersection operation as $\mathcal{V} \times \mathcal{O} \to \mathcal{S}$. Moreover, if $F : X \to X$ is effectively USC/LSC then $F^{\Leftarrow}(.)/F^{-1}(.)$ is (effectively) computable as $\mathcal{O} \to \mathcal{O}$. The following operations are known to be *uncomputable*: closure as $\mathcal{O} \to \mathcal{A}$, interior as $\mathcal{A} \to \mathcal{O}$.

2.4. *Standard Semantics of CTL and CTL**

CTL and CTL^* are typically interpreted over Kripke structures. A Kripke structure M is a tuple $(S,\ I,\ R,\ L)$ where S is a countable set of states; $I \subseteq S$ is a set of initial states; $R \subseteq S \times S$ is a transition relation such that $\forall s \in S, \exists s' \in S : (s, s') \in R$; AP is a finite set of atomic propositions; and $L : S \to 2^{AP}$ is an labelling function. A path

in M is an infinite sequence of states $s_0 s_1 s_2 \ldots$ such that $\forall i \geq 0 : (s_i, s_{i+1}) \in R$. A set of paths starting in state s is denoted as $Paths_M(s)$. For a path $\sigma \in Paths_M(s)$, where $\sigma = s_0 s_1 s_2 \ldots$, for any $j \geq 0$ we denote $\sigma_j := s_j s_{j+1} s_{j+2} \ldots$, and $\sigma[j] := s_j$.

Computational Tree Logic (CTL) [5] is divided into *state formulae*: $\Phi ::= p \mid \neg\Phi \mid \Phi \wedge \Phi \mid \forall\phi \mid \exists\phi$, and *path formulae*: $\phi ::= \mathcal{X}\Phi \mid \Phi\ \mathcal{U}\ \Phi \mid \Phi\ \mathcal{R}\ \Phi$. The state formulae have the following semantics: $s \models p$ iff $p \in L(s)$; $s \models \neg\Phi$ iff $\neg(s \models \Phi)$; $s \models \Phi \wedge \Psi$ iff $(s \models \Phi) \wedge (s \models \Psi)$; $s \models \exists\phi$ iff $\exists\sigma \in Paths_M(s) : \sigma \models \phi$; $s \models \forall\phi$ iff $\forall\sigma \in Paths_M(s) : \sigma \models \phi$. The semantics of path formulae is as follows: $\sigma \models \mathcal{X}\Phi$ iff $\sigma[1] \models \Phi$; $\sigma \models \Phi\ \mathcal{U}\ \Psi$ iff $\exists j \geq 0 : (\sigma[j] \models \Psi \wedge \forall 0 \leq i < j : \sigma[i] \models \Phi)$; $\sigma \models \Psi\ \mathcal{R}\ \Phi$ iff $(\forall i \geq 0 : \sigma[i] \models \Phi) \vee (\exists j \geq 0 : (\sigma[j] \models \Psi \wedge \forall 0 \leq i \leq j : \sigma[i] \models \Phi))$.

Linear Temporal Logic (LTL) [14] consists of *state formulae*: $\Phi ::= \forall\phi$ and *path formulae*: $\phi ::= p \mid \neg\phi \mid \phi \wedge \phi \mid \mathcal{X}\phi \mid \phi\ \mathcal{U}\ \phi \mid \psi\ \mathcal{R}\ \phi$ LTL reasons about computation sequences and therefore allows for a recursive use of path formulae. For example the semantics of until operator is: $\sigma \models \phi\ \mathcal{U}\ \psi$ iff $\exists j \geq 0 : (\sigma_j \models \psi \wedge \forall 0 \leq i < j : \sigma_i \models \phi)$; The state formulae have the following semantics: $s \models \forall\phi$ iff $\forall\sigma \in Paths_M(s) : \sigma \models \phi$. For the path $\sigma \in Paths_M(s)$, where $\sigma = s_0 s_1 s_2 \ldots$, for any $j \geq 0$ we have $\sigma_j = s_j s_{j+1} s_{j+2} \ldots$, and $\sigma[j] = s_j$, the semantics of path formulae is as follows: $\sigma \models p$ iff $p \in L(\sigma[0])$; $\sigma \models \neg\phi$ iff $\neg(\sigma \models \phi)$; $\sigma \models \phi \wedge \psi$ iff $(\sigma \models \phi) \wedge (\sigma \models \psi)$; $\sigma \models \mathcal{X}\phi$ iff $\sigma_1 \models \phi$; $\sigma \models \phi\ \mathcal{U}\ \psi$ iff $\exists j \geq 0 : (\sigma_j \models \psi \wedge \forall 0 \leq i < j : \sigma_i \models \phi)$; $\sigma \models \psi\ \mathcal{R}\ \phi$ iff $(\forall i \geq 0 : \sigma_i \models \phi) \vee (\exists j \geq 0 : (\sigma_j \models \psi \wedge \forall 0 \leq i \leq j : \sigma_i \models \phi))$.

Branching Temporal Logic (CTL)* [11] is a combination of LTL [14] and CTL, it's syntax is defined by *state formulae*: $\Phi ::= p \mid \neg\Phi \mid \Phi \wedge \Phi \mid \forall\phi \mid \exists\phi$ and *path formulae*: $\phi ::= \Phi \mid \neg\phi \mid \phi \wedge \phi \mid \mathcal{X}\phi \mid \phi\ \mathcal{U}\ \phi \mid \psi\ \mathcal{R}\ \phi$. The semantics of the state formulae is the same as for CTL, the semantics of path formulae is the same as for CTL, but instead of $\sigma \models p$ iff $p \in L(\sigma[0])$ we have $\sigma \models \Phi$ iff $\sigma[0] \models \Phi$.

Remarks: In CTL and CTL^*, path formulae can only be used as sub formulae. For a state formula Φ, we denote $Sat(\Phi) := \{s \in S | s \models \Phi\}$. In the following, we often identify Φ with the set $U_\Phi := Sat(\Phi)$; assume a standard atomic proposition *true* defined by $Sat(true) = S$; define *false* $:= \neg true$; and use the following (standard) abbreviations: *(i)* $\forall i \in \mathbb{N}$ we define $\mathcal{X}^i\phi := \underbrace{\mathcal{X}\ldots\mathcal{X}}_{\text{i times}}\phi$ and $\mathcal{X}^0\phi := \phi$, *(ii)* $\Diamond\psi :=$ *true* $\mathcal{U}\ \psi$, *(ii)* $\Box\phi :=$ *false* $\mathcal{R}\ \phi$. The temporal operators have the following names: $\mathcal{X}$ – *next*, $\Diamond$ – *eventually*, $\mathcal{U}$ – *until*, $\Box$ – *henceforth*, and $\mathcal{R}$ – *release*.

2.5. *Model checking CTL* (LTL) and Büchi automata*

This section is based on Sections 5.2 and 6.8.2 of [2]. For $\Phi \in CTL^*$, $M \models \Phi$ can be solved by using a combination of CTL and LTL model checking, and the bottom-up traversal of the syntax tree of Φ. While the traversal, one should verify encountered CTL sub-formulae and substitute them with new atomic propositions.

A *non-deterministic Büchi automaton (NBA)* is a tuple $\mathrm{A} := (Q, \Sigma, \delta, Q_0, Q_F)$ where Q is a finite set of states, Σ is an *alphabet*, $\delta : Q \times \Sigma \to 2^Q$ is a *transition function*, $Q_0 \subseteq Q$ is a set of *initial states*, and Q_F is a set of *accept states*. A *run*

$\sigma = A_0A_1A_2A_3\ldots \in \Sigma^\omega$ denotes an infinite sequence $q_0q_1q_2q_3\ldots$ of states in A such that $q_0 \in Q_0$ and $\forall i \geq 0 : q_i \xrightarrow{A_i} q_{i+1}$. The run $q_0q_1q_2q_3\ldots$ is *accepting* if $q_i \in Q_F$ for infinitely many indices $i \in \mathbb{N}$. The accepted language of A is $\mathcal{L}(\mathrm{A}) := \{\sigma \in \Sigma^\omega | \exists$ an accepting run for σ in A$\}$.

Let $\psi \in LTL$ is over AP then define $Words(\psi) := \left\{\sigma \in \left(2^{AP}\right)^\omega | \sigma \models \psi\right\}$ and $Paths(\mathrm{A}_\psi) := \bigcup \{Sat(\sigma[0]) \times Sat(\sigma[1]) \times \ldots | \sigma \in Words(\psi)\}$ where $Sat(\sigma[i]) := \bigwedge_{a\in\sigma[i]} Sat(a)$. The main result of LTL model checking can be interpreted as follows: $\forall \psi \in LTL$, such that ψ contains no negations, there exists an NBA A_ψ with $\Sigma := 2^{AP}$ such that it can be constructed in finite time and $\mathcal{L}(\mathrm{A}_\psi) = Words(\psi)$. We call the NBA A_ψ *non-blocking* iff for all $q \in Q$ we have that $\bigcup_{A\in\Sigma} Sat(\delta(q, A)) = S$.

3. Computable Semantics for *CTL*

In this section, we briefly outline and motivate the computable semantics of CTL, cf. [9], for the (extended) DTCSDS model given below.

Definition 1. *A discrete-time continuous-space control system (DTCSDS) is a tuple* $M = (T, F, L)$ *where:* $T = (X, \tau, \beta, \nu)$ *is a computable Hausdorff space;* $F \in \mathrm{C}(X, X)$ *is a multivalued map which defines the system's evolution; and* $L : X \to 2^{\mathrm{AP}}$ *is a labelling function where* AP *is a finite set of* atomic propositions. *For any* $p \in AP$ *and* $x \in X$ *we have that (respectively)* $Sat(p) \in \tau$ *and* $L(x) = \{p \in \mathrm{AP} | x \in Sat(p)\}$.

Here, elements of AP identify trivial system properties, which are given by open sets for reflecting the topological aspects of the: *(i)* computability theory, cf. Section 2.3; *(ii)* hybrid systems, cf. Section 5 of [13]; and *(iii)* logics for hybrid systems, cf. [12, 1].

For a DTCSDS model M, a set of initial states $I \subseteq X$, and $\Phi \in CTL$, proving $M, I \models \Phi$ is equivalent to showing that $I \subseteq Sat(\Phi)$, cf. Section 2.4. If $\Phi := p \in AP$ then we need to verify $I \subseteq Sat(p)$, where $Sat(p) \in \tau$. The latter, cf. Section 2.3, is computably verifiable only if I is compact. Thus, to make requirements on I uniform, and $M, I \models \Phi$ computable for any $\Phi \in CTL$: *(i)* we should only consider sets I that are compact; and *(ii)* we expect $Sat(\Phi)$ to be open.

Let $Sat(\Phi)$ be open then $Sat(\neg\Phi)$ is closed [a] and thus $I \subseteq Sat(\neg\Phi)$ is uncomputable. Defining $Sat(\neg\Phi) := \mathrm{Int}(X \setminus Sat(\Phi))$, as in [1], results in $Sat(\neg\Phi)$ being open, but, cf. Section 2.3, uncomputable. Thus, prior to model-checking, we suggest transforming a given CTL formula into its *negation normal form* (NNF), cf. [15]. Then, negations only prefix atomic propositions and we require that the representations of their open sets allow for computing the interior of the set complement.

To summarize, for $M, I \models \Phi$ to be computably verifiable we require that:

1. I is compact; 2. Φ is in NNF; 3. $\forall U \in \tau$, $F^{-1}(U)$, $F^{\Leftarrow}(U)$ are computable;
4. $\forall p \in AP$, such that $\neg p$ occurs in Φ, has a representation of $Sat(p)$ such that $\mathrm{Int}(X \setminus Sat(p))$ is computable.

[a] Note that, if $\Phi \in CTL$ then $\neg\Phi$ is a valid CTL formula too.

Under these conditions, Eq. 1 to 9 provide the computable semantics for the universal fragment of CTL. To account for the existential quantifier one should put the weak preimage F^{-1} in place of the strong preimage $F^{\Leftarrow}$. Below, $Sat'(\Phi)$ either equals to $Sat(\Phi)$ or is an open under approximation thereof.

$$Sat'(p) := U_p \text{ then } I \subseteq Sat'(p) \Leftrightarrow I \models p \tag{1}$$

$$Sat'(\neg p) := \mathrm{Int}(X \setminus U_p),\ I \subseteq Sat'(\neg p) \Rightarrow I \models \neg p \tag{2}$$

$$Sat'(\Phi \vee \Psi) := U_\Phi \cup U_\psi,\ I \subseteq Sat'(\Phi \vee \Psi) \Leftrightarrow I \models \Phi \vee \Psi \tag{3}$$

$$Sat'(\Phi \wedge \Psi) := U_\Phi \cap U_\psi,\ I \subseteq Sat'(\Phi \wedge \Psi) \Leftrightarrow I \models \Phi \wedge \Psi \tag{4}$$

$$Sat'(\forall(\mathcal{X}\Phi)) := F^{\Leftarrow}(U_\Phi),\ I \subseteq Sat'(\forall(\mathcal{X}\Phi)) \Leftrightarrow I \models \forall(\mathcal{X}\Phi) \tag{5}$$

$$Sat'(\forall(\Diamond\Psi)) := \bigcup_{n=0}^{\infty} S_n,\ S_0 = U_\Psi,$$

$$\forall n \geq 1:\ S_n = F^{\Leftarrow}(\bigcup_{i=0}^{n-1} S_i),\ I \subseteq Sat'(\forall(\Diamond\Psi)) \Leftrightarrow I \models \forall(\Diamond\Psi) \tag{6}$$

$$Sat'(\forall(\Phi\ \mathcal{U}\ \Psi)) := \bigcup_{n=0}^{\infty} S'_n,\ S'_0 = U_\Psi,$$

$$\forall n \geq 1:\ S'_n = F^{\Leftarrow}(\bigcup_{i=0}^{n-1} S'_i) \cap U_\Phi,\ I \subseteq Sat'(\forall(\Phi\ \mathcal{U}\ \Psi)) \Leftrightarrow I \models \forall(\Phi\ \mathcal{U}\ \Psi) \tag{7}$$

$$Sat'(\forall(\Box\Phi)) := \bigcup \{B_r \in \tau | \mathrm{Cl}(B_r) \subseteq U_\Phi \wedge \mathrm{Cl}(B_r) \subseteq F^{\Leftarrow}(B_r)\},\ I \subseteq Sat'(\forall(\Box\Phi)) \Rightarrow I \models \forall(\Box\Phi) \tag{8}$$

$$Sat'(\forall(\Psi\ \mathcal{R}\ \Phi)) := \bigcup \{B_r \in \tau | \mathrm{Cl}(B_r) \subseteq U_\Phi \wedge (\mathrm{Cl}(B_r) \subseteq U_\Psi \vee \mathrm{Cl}(B_r) \subseteq F^{\Leftarrow}(B_r \cup (U_\Psi \cap U_\Phi)))\},\ I \subseteq Sat'(\forall(\Psi\ \mathcal{R}\ \Phi)) \Rightarrow I \models \forall(\Psi\ \mathcal{R}\ \Phi) \tag{9}$$

In Eq. 8 to 9, each $B_r \in \tau$ is a finite union of open rational boxes in X and so $\mathrm{Cl}(B_r)$ is compact and computable. In case of the *negation* (Eq. 2), *henceforth* (Eq. 8), and *release* (Eq. 9) operators, the sets of states satisfying the formula are closed. Therefore, we use open and computable under approximations thereof. As a consequence, if Φ contains one of these operators, the fact that Φ does not hold (on M, I) does not imply that the negation of Φ holds. I. e. the Law of Excluded Middle breaks. Moreover, such a Φ can be true but not computably verifiable. Yet, the given computable semantics is optimal: the case of the negation operator was discussed earlier, the optimality for the henceforth (release) operators is shown in [7].

4. Equivalences and Implications for *CTL**

Since, in the traditional setting, the complexity of CTL and CTL^* (LTL) model checking are (respectively) *P-complete* and *PSPACE-complete*, further we provide a set of implication that allow to propagate quantifies inside the CTL^* formulae. Here, we assume the *standard* semantics of CTL^* on Kripke structures, cf. Section 2.4.

Definition 2. *Let* Φ *and* Ψ *be two state formulae, then* Φ implies Ψ *($\Phi \Rightarrow \Psi$) iff for any* M *and* I*:* $M, I \models \Phi \Rightarrow M, I \models \Psi$*;* Φ *and* Ψ *are* equivalent *($\Phi \equiv \Psi$) iff*

$\Phi \Rightarrow \Psi$ *and* $\Psi \Rightarrow \Phi$.

Theorem 3. *Let* $\phi, \psi \in CTL^*$ *be path formulae then the diagrams take place:*

$$\begin{array}{ccc} \forall(\phi \vee \psi) & \overset{\not\Leftarrow}{\Leftarrow} & \forall\phi \vee \forall\psi \\ \Downarrow\not\Downarrow & & \Downarrow\not\Downarrow \\ \exists(\phi \vee \psi) & \equiv & \exists\phi \vee \exists\psi \end{array} \quad (10) \qquad \begin{array}{ccc} \forall(\phi \wedge \psi) & \equiv & \forall\phi \wedge \forall\psi \\ \Downarrow\not\Downarrow & & \Downarrow\not\Downarrow \\ \exists(\phi \wedge \psi) & \overset{\not\Rightarrow}{\Rightarrow} & \exists\phi \wedge \exists\psi \end{array} \quad (11)$$

$$\begin{array}{ccc} \forall\mathcal{X}\phi & \equiv & \forall\mathcal{X}\forall\phi \\ \Downarrow\not\Downarrow & & \Downarrow\not\Downarrow \\ \exists\mathcal{X}\phi & \equiv & \exists\mathcal{X}\exists\phi \end{array} \quad (12) \qquad \begin{array}{ccc} \forall\Diamond\phi & \overset{\not\Leftarrow}{\Leftarrow} & \forall\Diamond\forall\phi \\ \Downarrow\not\Downarrow & & \Downarrow\not\Downarrow \\ \exists\Diamond\phi & \equiv & \exists\Diamond\exists\phi \end{array} \quad (13) \qquad \begin{array}{ccc} \forall\Box\phi & \equiv & \forall\Box\forall\phi \\ \Downarrow\not\Downarrow & & \Downarrow\not\Downarrow \\ \exists\Box\phi & \overset{\not\Rightarrow}{\Rightarrow} & \exists\Box\exists\phi \end{array} \quad (14)$$

$$\begin{array}{ccc} \forall(\phi\,\mathcal{U}\,\psi) & \overset{\not\Leftarrow}{\Leftarrow} & \forall(\forall\phi\,\mathcal{U}\,\forall\psi) \\ \Downarrow\not\Downarrow & & \Downarrow\not\Downarrow \\ \exists(\phi\,\mathcal{U}\,\psi) & \overset{\not\Leftarrow}{\Rightarrow} & \exists(\exists\phi\,\mathcal{U}\,\exists\psi) \end{array} \quad (15) \qquad \begin{array}{ccc} \forall(\psi\,\mathcal{R}\,\phi) & \overset{\not\Leftarrow}{\Leftarrow} & \forall(\forall\psi\,\mathcal{R}\,\forall\phi) \\ \Downarrow\not\Downarrow & & \Downarrow\not\Downarrow \\ \exists(\psi\,\mathcal{R}\,\phi) & \overset{\not\Leftarrow}{\Rightarrow} & \exists(\exists\psi\,\mathcal{R}\,\exists\phi) \end{array} \quad (16)$$

Remarks: (i) In some cases implications of Th. 3 can turn into equivalences. E. g., in Eq. 10 we get $\forall(\phi \vee \psi) \equiv \forall\phi \vee \forall\psi$ in case ϕ or ψ are state formulae[b]. *(ii)* Th. 3 assumes the standard semantics of CTL^* on Kripke structures which are defined for countable state spaces. In our case, the state space (X) can be uncountable, but this does not restrict the applicability of Th. 3, because the standard semantics, cf. Section 2.4, *does not* account for the cardinality of X.

5. Computable semantics for CTL^*, the first attempt

Below, we briefly outline and motivate the first version of a computable semantics for CTL^*, cf. [10]. Here, we shall work under the same assumptions as in Section 3, cf. conditions 1. to 4. Notice that, any CTL^* formula (in NNF) that we might need to verify is given by the syntax $\Phi ::= p \mid \neg p \mid \Phi \wedge \Phi \mid \Phi \vee \Phi \mid \forall\phi \mid \exists\phi$, where ϕ is an arbitrary path formula. Here, p, $\neg p$, $\Phi \wedge \Phi$, and $\Phi \vee \Phi$ inherit the computable semantics of CTL, but *for* $\forall\phi$ *and* $\exists\phi$ *we need another approach.* Since ϕ is an arbitrary path formula, it is natural to work with the set of paths satisfying it.

Let $Paths_M : X \to X^\omega$ be the multi-valued map that maps the set of initial states I into the set of system paths starting in I. Then from [6, 16], it follows that if F is USC/LSC and $F^{\Leftarrow}(U)/F^{-1}(U)$ are computable then $Paths_M$ is USC/LSC and $Paths_M^{\Leftarrow}(U')/Paths_M^{-1}(U')$ is computable for any $U' \in \tau^\omega$. Also, by the definition of DTCSDS and the results of Section 2.1, (X^ω, τ^ω) is a computable Hausdorff space and thus we can use the computability results of Section 2.3.

Let $Paths(\phi) \subseteq X^\omega$ be the set of all paths satisfying the path formula ϕ (regardless to the system-evolution function F). Then, if $Paths(\phi)$ is an open set of paths in X^ω equipped with the product topology τ^ω, we can define the computable

[b] In CTL^*, a state formula can be also seen as a path formula, cf. Section 2.4.

semantics for $\forall\phi$ and $\exists\phi$ as follows:

$$Sat'(\exists\phi) := Paths_M^{-1}(Paths(\phi)),\ I \subseteq Sat'(\exists\phi) \Leftrightarrow M, I \models \exists\phi, \tag{17}$$

$$Sat'(\forall\phi) := Paths_M^{\Leftarrow}(Paths(\phi)),\ I \subseteq Sat'(\forall\phi) \Leftrightarrow M, I \models \forall\phi. \tag{18}$$

To complete the computable semantics, it suffices to consider each path formula ϕ and to show that it either results in an open and computable set of paths $Paths(\phi)$, that we can provide its open and computable approximation $Paths'(\phi)$ such that:

$$\begin{aligned} Paths_M^{-1}(Paths'(\phi)) \subset Paths_M^{-1}(Paths(\phi)) \text{ and} \\ Paths_M^{\Leftarrow}(Paths'(\phi)) \subset Paths_M^{\Leftarrow}(Paths(\phi)). \end{aligned} \tag{19}$$

In the latter case, Eq. 17 to 18 turn into implications (from left to right).

The proof of computability and openness of the sets of paths is inductive. Consider path formulae ψ, ψ_1, and ψ_2 and a state formula Φ. Let $Paths(\psi)$, $Paths(\psi_1)$, $Paths(\psi_2)$, and $Sat'(\Phi) \subseteq Sat(\Phi)$ be open and computable. Then the sets of paths given by Eq. 20 to 27 are open and computable:

$$Paths'(\Phi) := Sat'(\Phi) \times X \times \ldots \times X \times \ldots \tag{20}$$

$$Paths(\psi_1 \vee \psi_2) := Paths(\psi_1) \cup Paths(\psi_2) \tag{21}$$

$$Paths(\psi_1 \wedge \psi_2) := Paths(\psi_1) \cap Paths(\psi_2) \tag{22}$$

$$Paths(\mathcal{X}\psi) := X \times Paths(\psi) \tag{23}$$

$$Paths(\Diamond\psi) := \bigcup_{i\in\mathbb{N}} U_i^{\psi},\ \text{where } U_i^{\psi} := \underbrace{X \times \ldots \times X}_{\text{i times}} \times Paths(\psi) \tag{24}$$

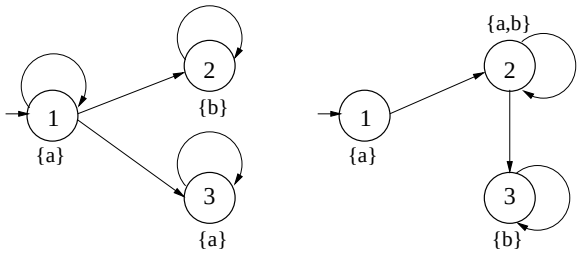
$$Paths(\psi_1\ \mathcal{U}\ \psi_2) := \bigcup_{i\in\mathbb{N}, i>0} \left(\bigcap_{j\in\mathbb{N}, j<i} U_j^{\psi_1} \cap U_i^{\psi_2} \right) \cup Paths(\psi_2) \tag{25}$$

$$\begin{aligned} Paths'(\Box\psi) := \bigcup \{ B_r^{\omega} \in B^{\omega} | \exists i \in \mathbb{N} : \mathrm{Cl}\left(p_i\left(B_r^{\omega}\right)\right) \subseteq Sat(\forall\Box\forall\psi) \wedge \\ \forall j < i : \mathrm{Cl}\left(Sh_j\left(B_r^{\omega}\right)\right) \subseteq Paths(\psi) \}. \end{aligned} \tag{26}$$

$$Paths'(\psi_2\ \mathcal{R}\ \psi_1) := Paths'(\Box\psi_1) \cup Paths(\psi_1\ \mathcal{U}\ (\psi_1 \wedge \psi_2)) \tag{27}$$

In the above, $Sh_j : X^{\omega} \to X^{\omega}$ is a *shift map* that removes the first j components of its argument. Sh_j is computable and open-valued. B_r^{ω} is a finite union of open rational balls in $(X^{\omega}, \tau^{\omega})$. Note that, $\mathrm{Cl}(Sh_j(B_r^{\omega}))$ is computed componentwise, and since $Sh_j(B_r^{\omega})$ is open in X^{ω}, we only need to compute the closure of finitely many components. Moreover, there are only finitely many i such that $p_i(Sh_j(B_r^{\omega})) \neq X$. For such i we have that $p_i(\mathrm{Cl}(Sh_j(B_r^{\omega})))$ is compact. The latter ensures computability of $\mathrm{Cl}(Sh_j(B_r^{\omega})) \subseteq Paths(\psi)$.

In [10] we noticed that $Paths'(\Box\psi)$, cf. Eq. 26, is open and computable, but it is rather restrictive. Yet, this approximation is optimal when propagating sets of paths through the formula because making a meaningful (non-empty) open *under* approximation for $Paths(\Box\psi)$ is generally *impossible* and for getting an open approximation, we can only put conditions on finite path prefixes. Eq. 26, is a good match for verifying $\forall\Box\psi$ because it is equivalent to $\forall\Box\forall\psi$, cf. Eq. 14 of Th. 3. When



(a) Model M_A (b) Model M_B

Fig. 1. Example LTS models

computing $Sat\,(\exists\Box\psi)$, instead of Eq. 26, we suggest using a better approximation:

$$Paths'\,(\Box\psi) := \bigcup\{B^\omega_r \in B^\omega | \exists n \in \mathbb{N} : \forall i \leq n : \mathrm{Cl}\,(Sh_i\,(B^\omega_r)) \subseteq Paths\,(\psi) \wedge \mathrm{Cl}\,(p_i\,(B^\omega_r)) \subseteq F^{-1}(\textstyle\bigcup_{j\leq n} p_j\,(B^\omega_r))\Big\}, \tag{28}$$

Here $\forall\sigma \in Paths'\,(\Box\psi)$ there exists $n \in \mathbb{N}$ for which $\forall i \leq n : \sigma_i \models \psi$ and $\exists\sigma' \in Paths\,(\sigma\,[n]) : \sigma' \models \Box\psi$. Since $Paths'\,(\Box\psi)$ contains paths with suffixes violating $\Box\psi$, it can *only* be used for verifying $\exists\Box\psi$. Note that, the set given by Eq. 26 satisfies Eq. 19 but the set given by Eq. 28 only satisfies its first part.

6. Analysing the first variant of the computable CTL^*

As it was mentioned in the previous section, when computing the set of paths for $\phi := \Box\psi$ (and subsequently $\phi := \psi_2\ \mathcal{R}\ \psi_1$) we generally have to use a crude approximation provided by Eq. 26. The latter is restrictive because, even on a finite-state system, can result in inconclusive model-checking results. We show this by two examples, where we verify CTL^* formulae using the semantics given in Section 5.

Example 4 (A universally quantified formula) *For the model M_A, cf. Fig. 1(a), we want to verify $\forall\,((a\,\mathcal{U}\,b) \vee \Box a)$. This formula is chosen because:* (i) *according to Eq. 10, cf. Th. 3, the universal quantifier can not be seamlessly propagated through the disjunction;* (ii) *it is clear that:*

$$M_A, 1 \models \forall\,((a\,\mathcal{U}\,b) \vee \Box a) \tag{29}$$

Using the computable semantics given in Section 5, Eq. 29 can be verified by showing that the left-hand side of Eq. 18 holds with $I := \{1\}$. For applying this equation we need to compute $Paths^{\Leftarrow}_M\,(Paths'\,((a\,\mathcal{U}\,b) \vee \Box a))$:

(1) Notice that: $Sat\,(a) = \{1,3\}$, $Sat\,(b) = \{2\}$, and $Sat\,(\forall\Box\forall a) = Sat\,(\forall\Box a) = \{3\}$.
(2) By Eq. 25: $Paths\,(a\,\mathcal{U}\,b) = \bigcup_{i\in\mathbb{N}} \underbrace{\{1,3\} \times \ldots \times \{1,3\}}_{i\ times} \times \{2\}^\omega$
(3) By Eq. 26: $Paths'\,(\Box a) = \bigcup_{i\in\mathbb{N}} \underbrace{\{1,3\} \times \ldots \times \{1,3\}}_{i\ times} \times \{3\}^\omega$

(4) By Eq. 21: $Paths'((a\,\mathcal{U}\,b)\vee\Box a) = Paths(a\,\mathcal{U}\,b)\bigcup Paths'(\Box a)$

(5) $Paths^{\Leftarrow}_{M}(Paths'((a\,\mathcal{U}\,b)\vee\Box a)) = \{2,3\}$ *because the computed set of paths contains all the model paths but the path* $\{1\}^{\omega}$.

Now, it is clear that the left-hand side of Eq. 18 is falsified. Also notice that, in this equation we only have an (left-to-right) implication since we had to use the approximation given by Eq. 26. Thus, our model checking result for Eq. 29 is: inconclusive.

Example 5 (An existentially quantified formula) *For the model* M_B*, cf. Fig. 1(b), we want to verify* $\exists((\mathcal{X}\Box b)\wedge\Box a)$*. This formula is chosen because:* (i) *according to Eq. 11, cf. Th. 3, the existential quantifier can not be propagated through the conjunction of path formulae;* (ii) *it is clear that the following holds:*

$$M_B, 1 \models \exists((\mathcal{X}\Box b)\wedge\Box a) \tag{30}$$

Similar to how it was done in the previous example, in order to verify Eq. 30 we compute: $Paths'(\mathcal{X}\Box b) = \{1,2,3\}\times\{2,3\}^{\omega}$ *using Eq. 23 and Eq. 26; and* $Paths'(\Box a) = \emptyset$*. The latter is because* $Sat(\forall\Box\forall a) = Sat(\forall\Box a) = \emptyset$*, cf. Eq. 26. Now, by Eq. 22 we have that* $Paths'((\mathcal{X}\Box b)\wedge\Box a) = \emptyset$*. Thus, the left hand side of Eq. 17 is falsified, yielding that the model checking result for Eq. 30 is:* inconclusive.

The reason for having inconclusive results in Example 4 to 5 is that the approach described in Section 5 fails to capture non-open sets of paths satisfying the formula.

7. Computable semantics for *CTL**, using Büchi automaton

In Section 6 we showed that the path-space-based computable semantics of CTL^* is too conservative in case of verifying formulae containing henceforth (release) operators. I. e. it can result in inconclusive model checking even on finite-state models. Below with the help of Büchi automata, we provide a new semantics for CTL^*, that is not only computable but always yields *conclusive (exact)* model-checking results on finite state models. Further, we keep working under the same assumptions as in Section 3, cf. conditions 1. to 4. The idea behind the new approach is as follows:

- CTL^* model checking can be split into model checking of CTL and LTL
- Any $\psi\in LTL$ can be represented by a non-blocking NBA A_ψ
- Running a DTCSDS M and A_ψ in parallel "marks the paths" satisfying ψ
- $M, I\models\forall\psi$ can be reduced to computing $Sat(\forall\Diamond\Box(X\times(Q\setminus Q_F)))$, where Q_F is a set of accept states of A_{ψ_n}, and $\psi_n := \neg\psi$ is in NNF.
- $M, I\models\exists\psi$ can be reduced to computing $Sat(\exists\Box\Diamond(X\times Q_F))$, where Q_F is a set of accept states of A_ψ, and ψ is in NNF.
- If $Sat(a)$ is open then $Sat(\forall\Diamond\Box a)$ and $Sat(\exists\Box\Diamond a)$ can be provided with computable under-approximations that are exact on finite-state models.

The last bullet is very important because it means that we can not obtain better model-checking results by simply discretizing the (original) continuous state space.

Further, Section 7.1 describes how to construct a parallel composition of $M||\mathrm{A}$ for an NBA A; build A_{ψ_n} and A_{ψ}; and reduce the problem of verifying $\forall\psi/\exists\psi$ to computing $Sat\left(\forall\Diamond\Box\left(X\times(Q\setminus Q_F)\right)\right)/\ Sat\left(\exists\Box\Diamond\left(X\times Q_F\right)\right)$. Section 7.2 provides computable under-approximations of $Sat\left(\forall\Diamond\Box a\right)$ and $Sat\left(\exists\Box\Diamond a\right)$. Section 7.3 gives several examples illustrating the superiority of the new approach.

7.1. *The idea of using Büchi automaton*

Let M be a DTCSDS model, cf. Def. 1, and $M' := (T',\ F',\ L')$ with $T' = (X',\ \tau',\ \beta',\ \nu')$ be a parallel composition of M and an NBA $\mathrm{A} := (Q, \Sigma, \delta, Q_0, Q_F)$. Below, we first show how to construct M', prove that T' is a computable Hausdorff space, and reveal under which conditions F' is USC/LSC and $F'^{\Leftarrow}(U)/F'^{-1}(U)$ are computable. Further, we show how the model-checking problem for an arbitrary formula $\forall\psi/\exists\psi$ can be reduced to an equivalent problem of computing $Sat\left(\forall\Diamond\Box\left(X\times(Q\setminus Q_F)\right)\right)/Sat\left(\exists\Box\Diamond\left(X\times Q_F\right)\right)$ on M' build using M and a specially-constructed non-blocking NBA $\mathrm{A}_{\psi_n}/\mathrm{A}_{\psi}$.

7.1.1. *Creating a parallel composition*

Consider M with a compact set of initial states $I \subseteq X$. Before constructing M' we extend M with a new initial state $x_i \notin X$ and get a transitional model $\tilde{M} := \left(\tilde{T}, \tilde{F}, \tilde{L}\right)$ with $\tilde{T} := \left(\tilde{X},\ \tilde{\tau},\ \tilde{\beta},\ \tilde{\nu}\right)$ such that $\tilde{X} := X \cup \{x_i\}$, $\tilde{F} := F \cup \{(\{x_i\}, I)\}$, $\tilde{L} := L$, and $\tilde{\beta} := \beta \cup \{x_i\}$ (this defines $\tilde{\tau}$). Extending ν to $\tilde{\nu}$ is natural and trivial. Note that, a compact set in $\tilde{T}$ is $\tilde{K} := K \cup D$ such that K is a compact set in T and $D \in \{\emptyset, \{x_i\}\}$. Let us prove that $\tilde{M}$ is a DTCSDS in the sense of Def. 1.

Theorem 6. *$\tilde{T}$ is a computable Hausdorff space, i. e. $\tilde{T}$ is: second countable, Hausdorff, with the base consisting of pre-compact open sets,* $\mathrm{Cl} : \tilde{\beta} \to \mathcal{K}$ *is computable, and the effectivity properties of Lemma* 2.3 *in* [4] *hold.*

The following theorem completes the proof of the fact that $\tilde{M}$ is a DTCSDS and also shows that it satisfies condition 3. of Section 3 (is required for computability).

Theorem 7. $\tilde{F} \in \mathrm{C}\left(\tilde{X}, \tilde{X}\right)$*. If I is the compact set of initial states of the model M then $\forall U \in \tilde{\tau}$ we have that $\tilde{F}^{\Leftarrow}(U)$ is computable. If I is given an overt-type name V then $\tilde{F}^{-1}(U)$ is also computable $\forall U \in \tilde{\tau}$.*

By the above theorem, in order to have $\tilde{F}^{-1}(U)$ computable $\forall U \in \tilde{\tau}$, we need the compact set I to be provided with its overt-type name V. This is not a serious limitation because: *(i)* I defines V in a unique way, cf. Section 2.1; *(ii)* I is user-defined and thus one can require V to be provided.

Let us consider $\tilde{M}$, an NBA A and their parallel composition $M' := (T',\ F',\ L')$ with $T' = (X',\ \tau',\ \beta',\ \nu')$ and $L'\left((x,q)\right) := \{p \in AP | x \in Sat\left(p\right)\}$. If we take a discrete topology τ_d on Q then, cf. [8], T' is a computable Hausdorff space, where

$X' = \tilde{X} \times Q$ and τ' is a product (box) topology induced by $\tilde{T}$ and T_Q, here $T_Q := (Q, \tau_d)$. Moreover, the evolution function $F' := \tilde{F}||\mathrm{A}$ defined as

$$F'((x,q)) := \bigcup_{a \in \Sigma} \left(\left(\tilde{F}(x) \cap Sat(a) \right) \times \delta(q,a) \right) \tag{31}$$

is USC/LSC if $Sat(a)$ is a closed/open and $F'^{\Leftarrow}(U)/F'^{-1}(U)$ is computable. This is required for computing under-approximations of $Sat(\forall\Diamond\Box U)/Sat(\exists\Box\Diamond U')$.

7.1.2. *Creating Büchi automata for $\forall\psi$ and $\exists\psi$*

Let us assume that we have a non-blocking NBA $\mathrm{A}_{\psi_n}/\mathrm{A}_\psi$ corresponding to $\forall\psi/\exists\psi$ such that $F' := \tilde{F}||\mathrm{A}_{\psi_n}/:= \tilde{F}||\mathrm{A}_\psi$ is effectively USC/LSC. Since the automata are non-blocking, F' preserves the behavior defined by the evolution function $\tilde{F}$ (and also F). Clearly, F' acts on $X' := (X \cup \{x_i\}) \times Q$ where every reachable state of the original model M gets attributed with a corresponding state of the automaton, run in parallel. The accepted paths of the automaton are the ones going through the set of accept states Q_F infinitely often. Therefore, the paths in M' that always eventually go through states $X \times Q_F$ correspond to the paths possible in M and satisfying the LTL formula represented by the automaton. The initial states of the model M' are $\{x_i\} \times Q_0$ but the initial states of M are I, and $\tilde{F}(x_i) := I$. This implies the following computable semantics:

$$M, I \models \forall\psi \Leftrightarrow \{x_i\} \times Q_0 \subseteq Sat(\forall\Diamond\Box(X \times (Q \setminus Q_F))) \tag{32}$$

$$M, I \models \exists\psi \Leftrightarrow I \subseteq p_0(Sat(\exists\Box\Diamond(X \times Q_F))) \tag{33}$$

Note that, $\{x_i\} \times Q_0$ is compact, $X \times (Q \setminus Q_F)$ and $X \times Q_F$ are open in T' (are also computable). The projection $p_0(.)$ is an effectively-computable open-valued map.

Further, we assume that $\forall\psi$ and $\exists\psi$ are such that ψ is an LTL path formula that contains no negations. This is valid. Consider an arbitrary $\phi \in CTL^*$ in NNF. Take ϕ and substitute all of its negated atomic propositions with new labels. This results in a negation-free formula ϕ'. In particular, for each $\neg a$, we should assign a new label a' such that $Sat(a') := \mathrm{Int}(X \setminus Sat(a))$. Then, ϕ' is such that $Paths(\mathrm{A}_{\phi'})$ is an under-approximation of $Paths(\phi)$. Using ϕ' in place of ϕ, turns Eq. 33 into a right-to-left implication, which is sufficient. Also notice that, cf. Section 2.5, the CTL^* model checking can be split into model checking of LTL and CTL, the computable semantics for CTL was given in Section 3.

For computability reasons, we require F', cf. Eq. 31, to be USC/LSC in case of $\forall\psi/\exists\psi$. The latter implies that, we need to build an automaton $\mathrm{A}_{\psi_n}/\mathrm{A}_\psi$ with transition labels resulting in closed/open sets. From Section 2.5 we know that, for every LTL path formula ψ, such that ψ contains no negations, there exists an automaton A_ψ where $\mathcal{L}(\mathrm{A}_\psi) = Words(\psi)$. This automaton can be constructed in finite time and its transition labels result in open sets. Clearly, for $\psi_n := \neg\psi$ (in NNF), we can also construct an automaton A_{ψ_n} where all transitions labels contain only negated atomic propositions, or *true*. For any $a \in AP$, to make $Sat(\neg a)$ a

closed set we should use non-topological semantics of negation (it is computable). Then generally speaking, $\mathit{Paths}\,(\mathrm{A}_{\psi_n})$ is an over-approximation of $\mathit{Paths}\,(\psi_n)$. This turns Eq. 32 into a right-to-left implication, which suffices. Both A_{ψ_n} and A_{ψ} can be blocking, to make them non-blocking, preserving the language, it suffices to extend them with an absorbing non-accept state $\perp$ and for each blocking state of the original automaton to add a transition to $\perp$, labelled with *true*. The automata A_{ψ_n} and A_{ψ} then have the properties we need, i. e. their transition labels result in, respectively, closed and open sets and both of the automata are non-blocking.

7.2. *Under-approximating* $\mathit{Sat}\,(\forall\Diamond\Box a)$ *and* $\mathit{Sat}\,(\exists\Box\Diamond a)$

Let $\mathit{Sat}\,(a) \in \tau'$. Then $\mathit{Sat}\,(\exists\Box\Diamond a)$ has a computable under approximation $\mathit{Paths}_M^{-1}\,(\mathit{Paths}\,(\Box\Diamond a))$ induced by the computable semantics of $\Diamond a$ and $\exists\Box\phi$, cf. Eq. 24 and Eq. 28 in Section 5. This approximation is not conservative and is exact on finite-state models because it accounts for the model's finite cycles that yield paths satisfying $\Box\phi$. To deal with $\mathit{Sat}\,(\forall\Diamond\Box a)$, we shall not employ the conservative approximation $\mathit{Paths}'\,(\Box\psi)$, cf. Eq. 26, but rather under- approximate $\mathit{Sat}\,(\forall\Diamond\Box a)$.

Further, we first devise a convenient fixed point characterisation of $\forall\Diamond\Box a$. Then, we give a recursive procedure for computing an under approximation of $\mathit{Sat}\,(\forall\Diamond\Box a)$ using CTL model checking only. In the end, we show that, on finite-state models, this procedure terminates and results in $\mathit{Sat}\,(\forall\Diamond\Box a)$.

Let us first provide a fixed-point characterisation for $\Diamond\Box a$. Below, without loss of generality, we assume that every state of the model has an outgoing transition.

Theorem 8. *For any state formula* Φ*:*

$$\Diamond\Box\Phi \equiv \Box\,(\Phi \vee \mathcal{X}\Diamond\Box\Phi) \tag{34}$$

Now, based on Th. 8 we can provide a fixed-point characterisation for $\forall\Diamond\Box a$.

Theorem 9. *For any state formula* Φ*:*

$$\forall\Diamond\Box\Phi \equiv \forall\Box\,(\Phi \vee \forall\mathcal{X}\forall\Diamond\Box\Phi) \tag{35}$$

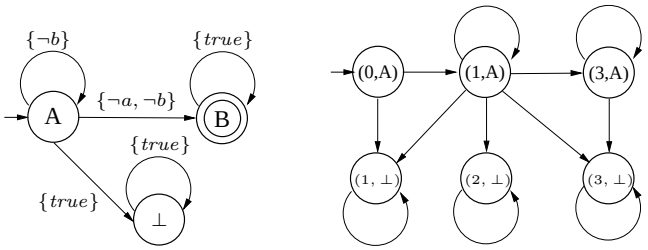
The result of Th. 9 allows to characterise $\mathit{Sat}\,(\forall\Diamond\Box\Phi)$ in terms of CTL only.

Theorem 10. *Let* M *be a model with a, possibly uncountable, state space* S*. The states of* M *are labelled with atomic proposition from a finite set* AP*. If for some* $a \in AP$ *and* $s_i \in S$ *we have that* $M, s_i \models \forall\Diamond\Box a$ *then* (i) $\forall\sigma \in \mathit{Paths}\,(s_i)$ *and* $\forall i \in \mathbb{N}$ *we have that* $M, \sigma[i] \models \forall\Diamond\Box a$*;* (ii) $M, s_i \models \exists\Diamond\forall\Box a$*.*

Clearly, if $M \models \forall\Diamond\Box a$ then $\mathit{Sat}\,(\forall\Diamond\forall\Box a) \neq \emptyset$. This follows from the point *(ii)* of Th. 10 and the fact that $\mathit{Sat}\,(\forall\Box a) \subseteq \mathit{Sat}\,(\forall\Diamond\forall\Box a)$. The following two theorems give an open and computable under-approximation for $\mathit{Sat}\,(\forall\Diamond\Box a)$.

Theorem 11. *For any* $n > 0$*, let us define*

$$G_0 := \mathit{Sat}\,(\forall\Diamond\forall\Box a)\,,\ G_n := \mathit{Sat}\,(\forall\Box\,(a \vee \forall\mathcal{X}G_{n-1}))\,. \tag{36}$$



(a) A_{ψ_n} (b) $M'_A := \tilde{M}_A || \mathrm{A}_{\psi_n}$

Fig. 2. Example: A universally quantified formula

Then $\forall i \in \mathbb{N} : (G_i \subseteq Sat\,(\forall\Diamond\Box a)) \wedge (G_i \subseteq G_{i+1})$*, and* $\exists i \in \mathbb{N} : G_i = G_{i+1}$ *implies* $\forall j > i : G_i = G_j$

Theorem 12. *For a finite model* M*, if* $\exists k : G_k = G_{k+1}$ *then* $G_k = Sat\,(\forall\Diamond\Box a)$.

Th. 11 proves that Eq. 36 provides an iterative procedure for obtaining an open under approximation of $Sat\,(\forall\Diamond\Box a)$. From Th. 12 it follows that for finite-state models, in finite time, the procedure converges to $Sat\,(\forall\Diamond\Box a)$.

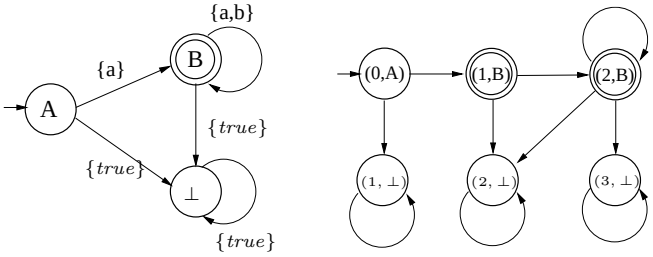
Let us summarize: The computable under-approximations for $Sat\,(\exists\Box\Diamond a)$ and $Sat\,(\forall\Diamond\Box a)$ are: *(i)* $Sat'\,(\exists\Box\Diamond a) := Paths_M^{-1}\,(Paths\,(\Box\Diamond a))$, induced by Eq. 24 and Eq. 28. *(ii)* $Sat'\,(\forall\Diamond\Box a) := G_i$, induced by Eq. 36; *Remark:* The sequence $\{G_i\}_{i\in\mathbb{N}}$ is strictly monotone until it reaches the minimal fixed point, i. e. some $k \in \mathbb{N} : G_k = G_{k+1}$, cf. Th. 11. On finite state models, $Sat'\,(\exists\Box\Diamond a)$ and $Sat'\,(\forall\Diamond\Box a)$ are exact. The latter is in case of taking G_i such that $G_i = G_{i+1}$, cf. Th. 12.

7.3. *Is the Büchi-based approach better?*

Let us take the examples from Section 6, and show that, unlike the path-space based approach of Section 5, the new approach of Section 7 solves them without a hitch.

Example 13 (A universally quantified formula) *Let us consider Example 4 from Section 6, and check if* $M_A, 1 \models \forall\,((a\,\mathcal{U}\,b) \vee \Box a)$ *holds. Here,* $\psi := (a\,\mathcal{U}\,b) \vee \Box a$ *and* $\psi_n := (\neg a\,\mathcal{R}\,\neg b) \wedge \Diamond\neg a$*. Fig. 2 contains* A_{ψ_n}*, and* $M'_A := \tilde{M}_A || \mathrm{A}_{\psi_n}$*, constructed by Eq. 31, where* $\tilde{M}_A$ *is obtained from the original model* M_A *by adding a new initial state* $x_i := 0$*. Notice that,* $Q_F := \{B\}$ *and thus* $X \times (Q \setminus Q_F) := \{1,2,3\} \times \{A, \bot\}$*. By Eq. 36 of Th. 11, cf. Section 7.2, we get* $G_0 = \{(0,A), (1,A), (3,A), (1,\bot), (2,\bot), (3,\bot)\} = G_1$*. Since,* $\{x_i\} \times Q_0 = \{(0,A)\}$*, by Eq. 32 of Section 7.1.2, we get that* $M_A, 1 \models \forall\,((a\,\mathcal{U}\,b) \vee \Box a)$.

Example 14 (An existentially quantified formula) *Let us consider Example 5 from Section 6 and check if* $M_B, 1 \models \exists\,((\mathcal{X}\Box b) \wedge \Box a)$ *holds. Here,* $\psi := (a\,\mathcal{U}\,b) \vee \Box a$ *and Fig. 3 contains* A_ψ*, and* $M'_B := \tilde{M}_B || \mathrm{A}_\psi$*, constructed by Eq. 31,*



(a) A_ψ (b) $M'_B := \tilde{M}_B || \mathrm{A}_\psi$

Fig. 3. Example: An existentially quantified formula

where $\tilde{M}_B$ *is obtained from the original model* M_B *by adding a new initial state* $x_i := 0$. *Notice that,* $Q_F := \{B\}$ *and thus* $X \times Q_F := \{1,2,3\} \times \{B\}$. *By Eq. 24, cf. Section 5, it is easy to see that* $Paths\left(\Diamond\left(X \times Q_F\right)\right)$ *contains open sets of paths with prefixes* $(1,B) \times (2,B) \times (2,B)$ *and* $(2,B) \times (2,B)$. *These paths also belong to* $Paths'\left(\Box\Diamond\left(X \times Q_F\right)\right)$, *when the latter is computed by Eq. 28. Thus,* $\{(1,B),(2,B)\} \subseteq Paths_M^{-1}\left(Paths'\left(\Box\Diamond\left(X \times Q_F\right)\right)\right)$, *and* $I := \{1\} \subseteq \{1,2\} \subseteq p_0\left(Paths_M^{-1}\left(Paths'\left(\Box\Diamond\left(X \times Q_F\right)\right)\right)\right)$. *Hence by Eq. 33:* $M_B, 1 \models \exists\left((\mathcal{X}\Box b) \wedge \Box a\right)$.

8. Concluding remarks

This article completes and summarises our work on computable, in the sense of Type Two Effectivity theory (TTE), semantics for CTL^* on Discrete-Time Continuous-Space Dynamic Systems. Here, we review the computable semantics of CTL and the first version of the computable semantics of CTL^*, based on path spaces. We show that the latter one is conservative for the henceforth and release operators, i. e. can be inconclusive on finite-state models. The computability requirements for these semantics are: *(i)* the set of initial states I is compact; *(ii)* $\Phi \in CTL^*$ is in NNF; *(iii)* the DTCSDS $M = (T,\ F,\ L)$ is such that T is a computable Hausdorff space, and F is a continuous map where $F^{-1}(U)$ and $F^{\Leftarrow}(U)$ are computable for any open U; *(iv)* the negation-prefixed atomic propositions of Φ have representations that allow for computing interiors of their complements. The novelty of this paper is that we provide a new computable semantics for CTL^* that is exact on finite-state models. It is based on reducing the problem of verifying $\forall\phi/\exists\phi$ on M to a problem of verifying $\forall\Diamond\Box F/\exists\Box\Diamond F$ on a model obtained using a parallel composition of M and the non-blocking NBA $\mathcal{L}_{\mathrm{A}_{\phi_n}}/\mathcal{L}_{\mathrm{A}_\phi}$. Here, $Sat\left(F\right)$ is open; ϕ and $\phi_n := \neg\phi$ are in NNF; for verifying $\exists\phi$, we also need to know an overt-type name of I.

In the provided semantics, if the verified state formula Φ contains negation, henceforth, or release operators then Φ can be true (on M, I) but not computably verifiable. Note that, if the formula holds in the computable semantics, then it also holds in the original one, but the Law of Excluded Middle does not hold. Since, in the traditional setting, the complexity of CTL and CTL^* (LTL) model

checking are (respectively) *P-complete* and *PSPACE-complete*, we also provide a set of implication that allow to propagate quantifies inside the CTL^* formulae.

Further, we plan to extend our approach towards hybrid systems and to implement the induced computable model-checking algorithms in the framework for reachability analysis of hybrid systems called Ariadne [3].

References

[1] S. N. Artemov, J.M. Davoren, and A. Nerode. Logic, Topological Semantics and Hybrid Systems. In *International Conference on Decision and Control, CDC'97*, volume 1, pages 698–701, CA, USA, 1997. IEEE Press.

[2] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. MIT Press, Cambridge, MA, USA, 2008.

[3] Andrea Balluchi, Alberto Casagrande, Pieter Collins, Alberto Ferrari, Tiziano Villa, and Alberto L. Sangiovanni-Vincentelli. Ariadne: A Framework for Reachability Analysis of Hybrid Automata. In *Symposium on Mathematical Theory of Networks and Systems (MTNS 2006)*, Kyoto, Japan, July 2006. To appear.

[4] Vasco Brattka and Gero Presser. Computability on subsets of metric spaces. *Theoretical Computer Science*, 305(1-3):43–76, 2003.

[5] E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *AMC Transactions On Programming Languages And Systems*, 8(2):244–263, 1986.

[6] Pieter Collins. Continuity and computability of reachable sets. *Theoretical Computer Science*, 341(1):162–195, 2005.

[7] Pieter Collins. Optimal Semicomputable Approximations to Reachable and Invariant Sets. *Theory of Computing Systems*, 41(1):33–48, 2007.

[8] Pieter J. Collins. Computable Types for Dynamic Systems. In *Computability in Europe (CiE)*, 2009. Local pre-conference proceedings volume.

[9] Pieter J. Collins and Ivan S. Zapreev. Computable CTL for Discrete-Time and Continuous-Space Dynamic Systems. In *Computability in Europe (CiE)*, pages 110–119, 2009. Local pre-conference proceedings volume.

[10] Pieter J. Collins and Ivan S. Zapreev. Computable CTL* for Discrete-Time and Continuous-Space Dynamic Systems. In *Proceedings of the 3rd International Workshop on Reachability Problems (RP)*, pages 107–119. Springer-Verlag, 2009.

[11] E. A. Emerson and J. Y. Halpern. "sometimes" and "Not Never" Revisited: On Branching versus Linear Time Temporal Logic. *Journal of the Association for Computing Machinery (ACM)*, 33(1):151–178, 1986.

[12] Philip Kremer and Grigori Mints. Dynamic topological logic. *Annals of Pure and Applied Logic*, 131(1–3):133–158, 2005.

[13] Anil Nerode and Wolf Kohn. Models for Hybrid Systems: Automata, Topologies, Controllability, Observability. In *Hybrid Systems*, volume 736 of *LNCS*, pages 317–356. Springer, 1992.

[14] Amir Pnueli. The Temporal Semantics of Concurrent Programs. In *Semantics of Concurrent Computation*, pages 1–20. Springer, 1979.

[15] Klaus Schneider. *Verification of Reactive Systems: Formal Methods and Algorithms*. TTCSS. Springer-Verlag, Berlin, Heidelberg, 2004.

[16] Dieter Spreen. On the Continuity of Effective Multifunctions. *Electron. Notes Theor. Comput. Sci.*, 221:271–286, 2008.

[17] Klaus Weihrauch. *Computable Analysis: An Introduction*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2000.

Appendix A. Proofs

Theorem 1 (Th. 3 of Section 4) *Let $\phi, \psi \in CTL^*$ be path formulae then the diagrams commute:*

$$\begin{array}{ccc} \forall(\phi \vee \psi) & \overset{\not\Leftarrow}{\Leftarrow} & \forall\phi \vee \forall\psi \\ \Downarrow\not\Downarrow & & \Downarrow\not\Downarrow \\ \exists(\phi \vee \psi) & \equiv & \exists\phi \vee \exists\psi \end{array} \quad \text{(A.1)} \qquad \begin{array}{ccc} \forall(\phi \wedge \psi) & \equiv & \forall\phi \wedge \forall\psi \\ \Downarrow\not\Downarrow & & \Downarrow\not\Downarrow \\ \exists(\phi \wedge \psi) & \overset{\not\Rightarrow}{\Rightarrow} & \exists\phi \wedge \exists\psi \end{array} \quad \text{(A.2)}$$

$$\begin{array}{ccc} \forall\mathcal{X}\phi & \equiv & \forall\mathcal{X}\forall\phi \\ \Downarrow\not\Downarrow & & \Downarrow\not\Downarrow \\ \exists\mathcal{X}\phi & \equiv & \exists\mathcal{X}\exists\phi \end{array} \quad \text{(A.3)} \qquad \begin{array}{ccc} \forall\Diamond\phi & \overset{\not\Leftarrow}{\Leftarrow} & \forall\Diamond\forall\phi \\ \Downarrow\not\Downarrow & & \Downarrow\not\Downarrow \\ \exists\Diamond\phi & \equiv & \exists\Diamond\exists\phi \end{array} \quad \text{(A.4)} \qquad \begin{array}{ccc} \forall\Box\phi & \equiv & \forall\Box\forall\phi \\ \Downarrow\not\Downarrow & & \Downarrow\not\Downarrow \\ \exists\Box\phi & \overset{\not\Rightarrow}{\Rightarrow} & \exists\Box\exists\phi \end{array} \quad \text{(A.5)}$$

$$\begin{array}{ccc} \forall(\phi\,\mathcal{U}\,\psi) & \overset{\not\Leftarrow}{\Leftarrow} & \forall(\forall\phi\,\mathcal{U}\,\forall\psi) \\ \Downarrow\not\Downarrow & & \Downarrow\not\Downarrow \\ \exists(\phi\,\mathcal{U}\,\psi) & \overset{\not\Rightarrow}{\Rightarrow} & \exists(\exists\phi\,\mathcal{U}\,\exists\psi) \end{array} \quad \text{(A.6)} \qquad \begin{array}{ccc} \forall(\psi\,\mathcal{R}\,\phi) & \overset{\not\Leftarrow}{\Leftarrow} & \forall(\forall\psi\,\mathcal{R}\,\forall\phi) \\ \Downarrow\not\Downarrow & & \Downarrow\not\Downarrow \\ \exists(\psi\,\mathcal{R}\,\phi) & \overset{\not\Rightarrow}{\Rightarrow} & \exists(\exists\psi\,\mathcal{R}\,\exists\phi) \end{array} \quad \text{(A.7)}$$

Sketch. First we prove Eq. A.1 to A.3, and then Eq. A.4 to A.7 follow as simple consequences. Note that, for each equation, implications from the row with the universal quantifiers to the row with the existential ones are trivial. Without loss of generality, we will assume that the set of initial states $I := \{s\}$.

- **Eq. A.1:**
  - $\forall(\phi \vee \psi) \not\Leftarrow \forall\phi \vee \forall\psi$: Clearly, there exists a model M and an initial state s such that $Paths_M(s) = \{\sigma', \sigma''\}$ where $\sigma' \models \phi$ and $\sigma'' \models \psi$. Then it is easy to see that $M, s \models \forall(\phi \vee \psi)$ but $M, s \not\models \forall\phi \vee \forall\psi$.
  - $\forall(\phi \vee \psi) \Leftarrow \forall\phi \vee \forall\psi$: $M, s \models \forall\phi \vee \forall\psi \Leftrightarrow (\exists\sigma' \in Paths_M(s) : \sigma' \models \psi) \vee (\exists\sigma'' \in Paths_M(s) : \sigma'' \models \phi) \Longrightarrow (\exists\sigma' \in Paths_M(s) : \sigma' \models \phi \vee \psi) \vee (\exists\sigma'' \in Paths_M(s) : \sigma'' \models \phi \vee \psi) \Leftrightarrow M, s \models \forall(\phi \vee \psi)$.
  - $\exists(\phi \vee \psi) \equiv \exists\phi \vee \exists\psi$: $M, s \models \exists(\phi \vee \psi) \Leftrightarrow \exists\sigma \in Paths_M(s) : \sigma \models \phi \vee \psi \Leftrightarrow \exists\sigma \in Paths_M(s) : \sigma \models \phi \vee \sigma \models \psi \Leftrightarrow (\exists\sigma' \in Paths_M(s) : \sigma' \models \phi \vee \sigma' \models \psi) \vee (\exists\sigma'' \in Paths_M(s) : \sigma'' \models \phi \vee \sigma'' \models \psi) \Leftrightarrow (\exists\sigma' \in Paths_M(s) : \sigma' \models \phi) \vee (\exists\sigma'' \in Paths_M(s) : \sigma'' \models \psi) \Leftrightarrow M, s \models \exists\phi \vee \exists\psi$.
- **Eq. A.2:** Follows from Eq. A.1 by negating the diagram and taking into account that: $\neg(\forall(\phi \vee \psi)) \equiv \exists(\neg\phi \wedge \neg\psi)$, $\neg(\forall\phi \vee \forall\psi) \equiv \exists\neg\phi \wedge \exists\neg\psi$, $\neg(\exists(\phi \vee \psi)) \equiv \forall(\neg\phi \wedge \neg\psi)$, $\neg(\exists\phi \vee \exists\psi) \equiv \forall\neg\phi \wedge \forall\neg\psi$.
- **Eq. A.3:** For an arbitrary model M and $I := \{s\}$:
  - $\forall\mathcal{X}\phi \equiv \forall\mathcal{X}\forall\phi$: Notice that $M, s \models \forall\mathcal{X}\forall\phi \Leftrightarrow \forall\sigma \in Paths_M(s) : \sigma[1] \models \forall\phi \Leftrightarrow \forall\sigma \in Paths_M(s)\, \forall\sigma' \in Paths_M(\sigma[1]) : \sigma' \models \phi$ and $M, s \models \forall\mathcal{X}\phi \Leftrightarrow \forall\sigma'' \in Paths_M(s) : \sigma'' \models \mathcal{X}\phi \Leftrightarrow \forall\sigma'' \in Paths_M(s) : \sigma''_1 \models \phi$.
    Clearly, $\forall\mathcal{X}\forall\phi \Rightarrow \forall\mathcal{X}\phi$ because if $M, s \models \forall\mathcal{X}\forall\phi$ then for any $\sigma'' \in Paths_M(s)$ we have that $\sigma''_1 \in Paths_M(\sigma[1])$ for some $\sigma \in Paths_M(s)$.

At the same time $\forall\sigma \in Paths_M(s)\,\forall\sigma' \in Paths_M(\sigma[1]) : \sigma' \models \phi$ and thus $\sigma_1'' \models \phi$.
Clearly, $\forall\mathcal{X}\phi \Rightarrow \forall\mathcal{X}\forall\phi$. Let $M, s \models \forall\mathcal{X}\phi$ and we chose any $\sigma \in Paths_M(s)$ and consider the set $\{\sigma_1''|\sigma''[1] = \sigma[1], \text{ and } \sigma'' \in Paths_M(s)\}$. Notice that this set equals to $Paths_M(\sigma[1])$ and for any σ' from the set we have that $\sigma' \models \phi$, because for any $\sigma'' \in Paths_M(s)$ we have that $\sigma_1'' \models \phi$.

– $\exists\mathcal{X}\phi \equiv \exists\mathcal{X}\exists\phi$: Follows from $\forall\mathcal{X}\phi \equiv \forall\mathcal{X}\forall\phi$ by the fact that $\neg\forall\mathcal{X}\phi \equiv \exists\mathcal{X}\neg\phi$ and $\neg\forall\mathcal{X}\forall\phi \equiv \exists\mathcal{X}\exists\neg\phi$.

Before we proceed, let us notice that Eq. A.1 (Eq. A.2) holds for any countable disjunction (conjunction) of formulae.

- **Eq. A.4:** Follows from Eq. A.1 and Eq. A.3 by the fact that for any model path σ we have $\sigma \models \Diamond\phi$ iff $\sigma \models \bigvee_{i\in\mathbb{N}} \mathcal{X}^i\phi$.
- **Eq. A.5:** Follows from Eq. A.2 and Eq. A.3 by the fact that for any model path σ we have $\sigma \models \Box\phi$ iff $\sigma \models \bigwedge_{i\in\mathbb{N}} \mathcal{X}^i\phi$.
- **Eq. A.6:** Follows from Eq. A.1 to A.3 by the fact that for any model path σ we have $\sigma \models \phi\,\mathcal{U}\,\psi$ iff $\sigma \models \bigvee_{i\in\mathbb{N}} \left(\bigwedge_{j\in\mathbb{N},j<i} \mathcal{X}^i\phi \wedge \mathcal{X}^i\psi\right)$.
- **Eq. A.7:** Follows from Eq. A.1 to A.3 by the fact that for any model path σ we have $\sigma \models \psi\,\mathcal{R}\,\phi$ iff $\sigma \models \left(\bigwedge_{i\in\mathbb{N}} \mathcal{X}^i\phi\right) \vee \bigvee_{j\in\mathbb{N}} \left(\bigwedge_{k\in\mathbb{N},k\leq j} \mathcal{X}^k\phi \wedge \mathcal{X}^j\psi\right)$. □

Theorem 2 (Th. 6 of Section 7.1) *$\tilde{T}$ is a computable Hausdorff space, i. e. $\tilde{T}$ is: second countable, Hausdorff, with the base consisting of pre-compact open sets, $\mathrm{Cl} : \tilde{\beta} \to \mathcal{K}$ is computable, and the effectivity properties of Lemma* 2.3 *in* [4] *hold.*

Proof. $\tilde{T}$ is second countable, i. e. has a countable base, because T is second countable and we defined $\tilde{\beta} := \beta \cup \{x_i\}$.

$\tilde{T}$ is Hausdorff iff $\forall x, y \in \tilde{X} : x \neq y$ exist neighbourhoods thereof $U_x, U_y \in \tilde{\tau}$ such that $U_x \cap U_y = \emptyset$. Since T is Hausdorff, then if $x, y \in X$, it is trivial. If $x = x_i$ and $y \in X$ then we can take $U_x := \{x_i\}$ and $U_y \in \tau$.

$\tilde{\beta}$ consists of pre-compact open sets because β consists of pre-compact open sets and $\mathrm{Cl}(\{x_i\}) = \{x_i\}$ – is open, closed, and compact in $\tilde{T}$.

$\mathrm{Cl} : \tilde{\beta} \to \mathcal{K}$ is computable because $\mathrm{Cl}(\{x_i\}) = \{x_i\}$ can be trivially added to the machine computing the closure. The effectivity properties of Lemma 2.3 in [4] hold because X is extended with a distinct x_i disjoint from every element of X. □

Theorem 3 (Th. 7 of Section 7.1) *$\tilde{F} \in \mathrm{C}\left(\tilde{X}, \tilde{X}\right)$. If I is the compact set of initial states of the model M then $\forall U \in \tilde{\tau}$ we have that $\tilde{F}^{\Leftarrow}(U)$ is computable. If I is given an overt-type name V then $\tilde{F}^{-1}(U)$ is also computable $\forall U \in \tilde{\tau}$.*

Proof. $\tilde{F}$ is continuous *iff* it is upper and lower semicontinuous.

$\tilde{F}$ is upper semicontinuous *iff* $\forall U \in \tilde{\tau} : \tilde{F}^{\Leftarrow}(U) \in \tilde{\tau}$. Clearly, $\tilde{F}^{\Leftarrow}(U) := F^{\Leftarrow}(U) \cup \{x_i\}$ if $I \subseteq U$ and otherwise $\tilde{F}^{\Leftarrow}(U) := F^{\Leftarrow}(U)$. Notice that $F^{\Leftarrow}(U)$, $\{x_i\}$, $F^{\Leftarrow}(U) \cup$

$\{x_i\}$ are open and computable, and $I \subseteq U$ is also computable, cf. Section 2.3. Therefore $\tilde{F}^{\Leftarrow}(U)$ is open and computable.

$\tilde{F}$ is lower semicontinuous *iff* $\forall U \in \tilde{\tau} : \tilde{F}^{-1}(U) \in \tilde{\tau}$. Clearly, $\tilde{F}^{-1}(U) := F^{-1}(U) \cup \{x_i\}$ if $I \cap U \neq \emptyset$ and otherwise $\tilde{F}^{-1}(U) := F^{-1}(U)$. Notice that $F^{-1}(U)$, $\{x_i\}$, $F^{-1}(U) \cup \{x_i\}$ are open and computable. If I is given an overt-type name V, cf. Sections 2.1 and 2.3, then $I \cap U \neq \emptyset$, is computable as $V \cap U \neq \emptyset$ $(\mathcal{V} \times \mathcal{O} \to \mathcal{S})$. □

Theorem 4 (Th. 8 of Section 7.2) *For any state formula* Φ*:*

$$\Diamond\Box\Phi \equiv \Box\,(\Phi \vee \mathcal{X}\Diamond\Box\Phi) \tag{A.8}$$

Proof. Notice that:

$$\sigma \models \Diamond\Box\Phi \Leftrightarrow \exists j \in \mathbb{N} : \forall i \geq j : \sigma[i] \models \Phi \tag{A.9}$$

$$\sigma \models \Box\,(\Phi \vee \mathcal{X}\Diamond\Box\Phi) \Leftrightarrow \forall p \in \mathbb{N} : \sigma_p \models \Phi \vee \mathcal{X}\Diamond\Box\Phi \tag{A.10}$$

To prove that Eq. A.8 holds it suffices to show that for any path σ:

$\boldsymbol{\sigma \models \Diamond\Box\Phi \Rightarrow \sigma \models \Box\,(\Phi \vee \mathcal{X}\Diamond\Box\Phi)}$**:** Let Eq. A.9 holds then $\forall p \in \mathbb{N}$: *(i)* if $p < j$ then $\sigma_p \models \mathcal{X}\Diamond\Box\Phi \Rightarrow \sigma_p \models \Phi \vee \mathcal{X}\Diamond\Box\Phi$; *(ii)* if $p \geq j$ then $\sigma_p \models \Phi \Rightarrow \sigma_p \models \Phi \vee \mathcal{X}\Diamond\Box\Phi$.

$\boldsymbol{\sigma \models \Box\,(\Phi \vee \mathcal{X}\Diamond\Box\Phi) \Rightarrow \sigma \models \Diamond\Box\Phi}$**:** Let Eq. A.10 holds then *(i)* if $\forall p \in \mathbb{N} : \sigma_p \models \Phi$ then $\sigma \models \Box\Phi \Rightarrow \Diamond\Box\Phi$; *(ii)* if $\exists p \in \mathbb{N} : \sigma_p \models \mathcal{X}\Diamond\Box\Phi$ then $\sigma_{p+1} \models \Diamond\Box\Phi \Rightarrow \sigma \models \Diamond\Box\Phi$.

□

Theorem 5 (Th. 9 of Section 7.2) *For any state formula* Φ*:*

$$\forall\Diamond\Box\Phi \equiv \forall\Box\,(\Phi \vee \forall\mathcal{X}\forall\Diamond\Box\Phi) \tag{A.11}$$

Proof. It suffices to prove that $\forall\Box\,(\Phi \vee \mathcal{X}\Diamond\Box\Phi) \equiv \forall\Box\,(\Phi \vee \forall\mathcal{X}\forall\Diamond\Box\Phi)$. Using Th. 3, by Eq. 14, we have that $\forall\Box\,(\Phi \vee \mathcal{X}\Diamond\Box\Phi) \equiv \forall\Box\forall\,(\Phi \vee \mathcal{X}\Diamond\Box\Phi)$. Since Φ is a state formula, cf. the last paragraph of Section 4, $\forall\Box\forall\,(\Phi \vee \mathcal{X}\Diamond\Box\Phi) \equiv \forall\Box\,(\Phi \vee \forall\mathcal{X}\Diamond\Box\Phi)$. Then by Eq. 12, $\forall\Box\,(\Phi \vee \forall\mathcal{X}\Diamond\Box\Phi) \equiv \forall\Box\,(\Phi \vee \forall\mathcal{X}\forall\Diamond\Box\Phi)$. □

Theorem 6 (Th. 10 of Section 7.2) *Let* M *be a model with a, possibly uncountable, state space* S*. The states of* M *are labelled with atomic proposition from a finite set* AP*. If for some* $a \in AP$ *and* $s_i \in S$ *we have that* $M, s_i \models \forall\Diamond\Box a$ *then* (i) $\forall\sigma \in Paths\,(s_i)$ *and* $\forall i \in \mathbb{N}$ *we have that* $M, \sigma[i] \models \forall\Diamond\Box a$*;* (ii) $M, s_i \models \exists\Diamond\forall\Box a$*.*

Proof.

(i) By contradiction. Let $\exists\sigma \in Paths\,(s_i)$, $\exists i \in \mathbb{N}$ and $\exists\sigma' \in Paths\,(\sigma[i])$ such that $\sigma \not\models \Diamond\Box a$ then the path $\sigma[0, i-1] \oplus \sigma' \not\models \Diamond\Box a$. Here, $\oplus$ is a trivial concatenation.

(ii) By contradiction. Let $s_i \models \forall\Box\exists\Diamond\neg a$ then we can construct an infinite path $\sigma_b \in Paths\,(s_i)$ such that $\sigma_b \models \Box\Diamond\neg a$. Consider any path $\sigma \in Paths\,(s_i)$ then $\sigma \models \Diamond\Box a$ and therefore we can choose some $i > 0$ such that $\sigma[i] \models a$. Define $\sigma_b^0 := \sigma[0, i]$ and notice that $\sigma_b^0[i] \models a$ and $\sigma_b^0[i] \models \exists\Diamond\neg a$. Thus we can choose $\sigma' \in Paths\left(\sigma_b^0[i]\right)$ such that for some $j > 0$ we have $\sigma'[j] \models \neg a$. Define $\sigma_b^1 := \sigma_b^0 \oplus \sigma'[1, j]$. Clearly, $\sigma_b^1[i] \models a$ and $\sigma_b^1[i + j] \models \neg a$. Any path with prefix σ_b^1 satisfies $\Diamond\Box a$, thus there exists a path $\sigma'' \in Paths\left(\sigma_b^1[i + j]\right)$ such that for some $l > 0$ we have $\sigma''[l] \models a$. Define $\sigma_b^2 := \sigma_b^1 \oplus \sigma''[1, l]$ and notice that for $i, j, l > 0$ we have that $\sigma_b^2[i] \models a$, $\sigma_b^2[i+j] \models \neg a$, $\sigma_b^2[i+j+l] \models a$ and $\sigma_b^2[i+j+l] \models \exists\Diamond\neg a$. Now it is easy to see that for any $n > 0$ we can construct a path prefix σ_b^n from σ_b^{n-1} in such a way that we always get an alternation of a and $\neg a$ states on this path prefix. Continuing this process we inductively construct and infinite path $\sigma_b \in Paths\,(s_i)$ that satisfies $\Box\Diamond\neg a$ and thus violate $M, s_i \models \forall\Diamond\Box a$. □

Theorem 7 (Th. 11 of Section 7.2) *For any $n > 0$, let us define*

$$G_0 := Sat\,(\forall\Diamond\forall\Box a)\,,\ G_n := Sat\,(\forall\Box\,(a \vee \forall\mathcal{X} G_{n-1}))\,. \tag{A.12}$$

Then $\forall i \in \mathbb{N} : (G_i \subseteq Sat\,(\forall\Diamond\Box a)) \wedge (G_i \subseteq G_{i+1})$, and $\exists i \in \mathbb{N} : G_i = G_{i+1}$ implies $\forall j > i : G_i = G_j$

Proof.

$\forall i \in \mathbb{N} : G_i \subseteq Sat\,(\forall\Diamond\Box a)$: Let us use simple induction. Due to Eq. 13 of Th. 3 we have that $\forall\Diamond\forall\Box a \Rightarrow \forall\Diamond\Box a$ and thus $G_0 \subseteq Sat\,(\forall\Diamond\Box a)$. Let $G_k \subseteq Sat\,(\forall\Diamond\Box a)$ then we want to prove that $G_{k+1} \subseteq Sat\,(\forall\Diamond\Box a)$. By definition $G_{k+1} = Sat\,(\forall\Box\,(a \vee \forall\mathcal{X} G_k))$, and since $G_k \subseteq Sat\,(\forall\Diamond\Box a)$ it follows that $G_{k+1} \subseteq Sat\,(\forall\Box\,(a \vee \forall\mathcal{X}\forall\Diamond\Box a))$. The latter, cf. Eq. 35, implies $G_{k+1} \subseteq Sat\,(\forall\Diamond\Box a)$.

$\forall i \in \mathbb{N} : G_i \subseteq G_{i+1}$: Consider induction by i.

Let us first prove that $G_0 \subseteq G_1$, notice that because $\forall\Box a$ is a state formula, due to Eq. 12 of Th. 3, and $Sat\,(\forall\Box a) \subseteq Sat\,(a)$ the following holds:

$$\begin{aligned} G_0 = Sat\,(\forall\Diamond\forall\Box a) = Sat\,(\forall\,(\forall\Box a \vee \mathcal{X}\Diamond\forall\Box a)) = Sat\,(\forall\Box a \vee \forall\mathcal{X}\forall\Diamond\forall\Box a) = \\ = Sat\,(\forall\Box a) \cup Sat\,(\forall\mathcal{X}\forall\Diamond\forall\Box a) = Sat\,(\forall\Box a) \cup Sat\,(\forall\mathcal{X} G_0) \subseteq \\ \subseteq Sat\,(a) \cup Sat\,(\forall\mathcal{X} G_0) = Sat\,(a \vee \forall\mathcal{X} G_0)\,. \end{aligned}$$

Let us define $A := a \vee \forall\mathcal{X} G_0$ then we have that $G_0 \subseteq Sat\,(A)$ and $G_1 = Sat\,(\forall\Box A)$ which implies that $Sat\,(\forall\Box G_0) \subseteq G_1$. Now, it suffices to prove that $G_0 \subseteq Sat\,(\forall\Box G_0)$, i. e. that $\forall\Diamond\forall\Box a \Rightarrow \forall\Box\forall\Diamond\forall\Box a$. We prove this by contradiction. Assume $s \models \forall\Diamond\forall\Box a$, i. e. $\forall\sigma \in Paths\,(s) : \sigma \models \Diamond\forall\Box a$, and $s \models \exists\Diamond\Box\exists\Diamond\neg a$ (by Eq. 14 of Th. 3 $\forall\Box\forall\Diamond\forall\Box a \equiv \forall\Box\Diamond\forall\Box a$), i. e. $\exists\sigma' \in Paths\,(s) : \sigma' \models \Diamond\Box\exists\Diamond\neg a$. Notice that, $(\sigma' \models \Diamond\Box\exists\Diamond\neg a) \Rightarrow (\sigma' \models \Box\exists\Diamond\neg a)$ since $\exists j \in \mathbb{N} : \forall k \geq j : \sigma'[k] \models \exists\Diamond\neg a$ and $\forall k < j : \sigma'[k] \models \exists\Diamond\neg a$ because σ goes to $\sigma'[j]$ and $\sigma'[j] \models \exists\Diamond\neg a$. Therefore $\sigma' \models \neg\Diamond\forall\Box a (\equiv \Box\exists\Diamond\neg a)$ i. e. we have a contradiction, and thus $G_0 \subseteq G_1$.

If $G_{k-1} \subseteq G_k$ then clearly $G_k \subseteq G_{k+1}$ because $G_k = Sat\,(\forall\Box\,(a \vee \forall\mathcal{X} G_{k-1})) \subseteq Sat\,(\forall\Box\,(a \vee \forall\mathcal{X} G_k)) = G_{k+1}$.

$\exists i \in \mathbb{N} : G_i = G_{i+1}$ implies $\forall j > i : G_i = G_j$: The proof is trivial by induction. It suffices to notice that if $G_i = G_{i+1}$ then $G_{i+2} = Sat\,(\forall\Box\,(a \vee \forall\mathcal{X} G_{i+1})) = Sat\,(\forall\Box\,(a \vee \forall\mathcal{X} G_i)) = G_{i+1}$. □

Theorem 8 (Th. 12 of Section 7.2) *For a finite model M, if $\exists k : G_k = G_{k+1}$ then $G_k = Sat\,(\forall\Diamond\Box a)$.*

Proof. The fact that $G_k \subseteq Sat\,(\forall\Diamond\Box a)$ follows from Th. 11. let us prove that $Sat\,(\forall\Diamond\Box a) \subseteq G_k$.

For any $i \in \mathbb{N}$ define $G_0^n := Sat\,(\exists\Box\exists\Diamond\neg a)$ and $G_{i+1}^n := Sat\,(\exists\Box\,(\neg a \wedge \exists\mathcal{X} G_i^n))$. Since $\forall i \in \mathbb{N}$ we have that $G_i^n = \neg G_i$ it follows that $G_{i+1}^n \subseteq G_i^n$, and $Sat\,(\exists\Box\Diamond\neg a) \subseteq G_i^n$. Moreover, if $\exists k : G_k = G_{k+1}$ then $\forall j \geq k : G_k^n = G_j^n$, cf. Th. 11. If follows from the definition of G_j^n that it contains states from which there are paths with at least j states on the path satisfying $\neg a$.

A proof by contradiction: Assume that $\exists s \in Sat\,(\forall\Diamond\Box a)$ and $s \notin G_k$. The latter implies that $\forall i : s \notin G_i$ because $\forall i : G_i \subseteq G_{i+1}$ and $\forall j \geq k : G_k = G_j$. Notice that, $\forall j : s \notin G_j$ implies $\forall j : s \in G_j^n$. This means that for any j there is a path starting in s such that it contains j states that satisfy $\neg a$. The model M is finite, let it have $N \in \mathbb{N}$ states. Because $s \in G_{N+1}^n$ there exists a path $\sigma \in Paths\,(s)$ such that σ contains $N+1$ states that satisfy $\neg a$. Clearly, there must be a $\neg a$ state that occurs (at least) twice on the chosen path. Thus, M contains a cycle that is reachable from s and contains a $\neg a$ state, i. e. $s \models \exists\Box\Diamond\neg a$. This contradicts to $s \in Sat\,(\forall\Diamond\Box a)$. □

Centrum Wiskunde & Informatica (CWI) is the national research institute for mathematics and computer science in the Netherlands. The institute's strategy is to concentrate research on four broad, societally relevant themes: earth and life sciences, the data explosion, societal logistics and software as service.

Centrum Wiskunde & Informatica (CWI) is het nationale onderzoeksinstituut op het gebied van wiskunde en informatica. De strategie van het instituut concentreert zich op vier maatschappelijk relevante onderzoeksthema's: aard- en levenswetenschappen, de data-explosie, maatschappelijke logistiek en software als service.

Bezoekadres:
Science Park 123
Amsterdam

Postadres:
Postbus 94079, 1090 GB Amsterdam
Telefoon 020 592 93 33
Fax 020 592 41 99
info@cwi.nl
www.cwi.nl

CWI

Centrum Wiskunde & Informatica