Proceedings of the
1st Workshop on Artificial Intelligence and Formal Verification, Logics, Automata and Synthesis (OVERLAY),
Rende, Italy, November 19–20, 2019

Model Checking BDI Logics over Finite-state Worlds*

Salvatore La Torre[1] and Gennaro Parlato[2]

[1]University of Salerno, Italy
[2]University of Molise, Italy

Abstract

Logics with *belief*, *desire* and *intention* attitudes (BDI logics) are among the most widely studied formal languages for modelling rational agents. We consider the logic $\text{CTL}^*_{\text{BDI}}$ that augments the branching-time logic CTL^* with the BDI modalities and adopt the possible-world semantics by Rao and Georgeff. In this paper, we introduce the model-checking problem of $\text{CTL}^*_{\text{BDI}}$ over finite-state models that are described by tuples of Kripke structures (one for each world) and where the BDI relations are captured by finite-state relations. We then solve the problem by giving an exponential time decision algorithm that is obtained by adapting the standard decision algorithm for CTL^*.

1 Introduction

The use of rational agents for modeling real world systems has been heavily investigated and is now well accepted. An architecture that has emerged sees the systems as rational agents having certain mental attitudes of *belief*, *desire*, and *intention* (BDI *agents*). The beliefs express what the system knows about the state of the environment, the desires capture the information about the objectives to be accomplished and the intentions represent a high-level plan coming with the agent's commitment to achieve it (intentions force the agent to pursue certain desires) [6].

BDI agents, and the related specification languages denoted as BDI logics, have received different formulations (see [12]). Their increasing use in the design and implementation of safety-critical applications has also motivated several approaches to the verification of such models (see [4, 5, 3]).

Here, we adopt the approach by Rao and Georgeff [14, 15] based on a possible worlds semantics where each possible world is not an instantaneous state but a transition system: possible worlds share (and are synchronized over) a branching-time structure whose time points represent the instantaneous states. Belief, desire and intentions are expressed through accessibility relations that relate the possible worlds at each time point. These relations can possibly vary over time, which constitutes an important feature for modeling systems (see [9] and references therein).

In this setting, system properties are expressed by using extensions of CTL and CTL^* [8] with the belief, desire, and intention modal operators. We denote these logics respectively as CTL_{BDI} and $\text{CTL}^*_{\text{BDI}}$. For CTL_{BDI}, the model checking question, i.e., determining whether a given model satisfies a given specification, and the satisfiability question, i.e., whether a formula admits a model where it is fulfilled, have been studied respectively in [15] and in [16]. In particular, in [15] a polynomial time decision algorithm is given by restricting the worlds of the system models to have only a finite number of time points.

*This work was partially supported by GNCS 2019 and MIUR-FARB 2017-2018 grants.

Although using only a finite number of time points suffices to model some realistic scenarios such as those obtained from scenarios modeled as decision trees (see [16]), in general this seems to be a serious restriction when dealing with reactive systems that exhibit non-terminating behaviors. In this paper, we thus investigate the model-checking question for BDI branching-time logics and study the model-checking problem for CTL_{BDI} and $\text{CTL}^*_{\text{BDI}}$ over *finite-state* structures that can exhibit infinitely many time points. We assume that models are described as tuples of Kripke structures (one for each world) and the BDI relations are captured by finite-state automata.

Besides the introduction of this new model, as further contributions we give a decision algorithm for the CTL_{BDI} and the $\text{CTL}^*_{\text{BDI}}$ model-checking problems. The algorithms are based on the construction of a finite graph obtained as the synchronous cross product of the Kripke structures that define the possible worlds, and the finite automata that express the BDI-accessibility relations. For CTL_{BDI}, along the same line as for the standard CTL model-checking algorithm [7], our algorithm iteratively labels each node u of such graph with the sub-formulas of the input formula that hold true at u. The formulas are processed starting from the atomic propositions and then by increasing number of operators. The only additions to the algorithm from [7] are the rules that handle the BDI operators. The resulting algorithm takes time linear in the sizes of the graph and the input formula, and thus takes *exponential* time in the size of the input. For $\text{CTL}^*_{\text{BDI}}$, we extend the algorithm just given for $\text{CTL}^*_{\text{BDI}}$ similarly to how the standard decision algorithm for CTL^* is obtained from that of CTL [7]. The overall time taken by this algorithm is still *exponential* in the input size.

2 Branching-time logic with BDI modalities

In this section, we briefly recall the syntax and the semantics of $\text{CTL}^*_{\text{BDI}}$ [15].

In the rest of the paper, for an integer $k > 0$, $[k]$ will denote the set $\{1, \ldots, k\}$.

Syntax. A $\text{CTL}^*_{\text{BDI}}$ formula can be a *state* or a *path* formula. State formulas are inductively defined starting from atomic propositions by applying the logical connectives, the path quantifiers (to path formulas) and the *belief* (BEL), *desire* (DES), and *intention* (INT) operators. Path formulas are either state formulas or obtained by applying temporal operators such as *next* ($\bigcirc$) and *until* ($\mathcal{U}$). The syntax of CTL_{BDI} can be obtained from that of $\text{CTL}^*_{\text{BDI}}$ by disallowing the nesting of Boolean and temporal operators in path formulas. A formal definition for both logics can be found in [15].

Semantics. The meaning of the formulas from $\text{CTL}^*_{\text{BDI}}$, and thus CTL_{BDI}, is defined according to a possible world semantics where each possible world is not an instantaneous state but a transition system. All the worlds are synchronized over a shared branching-time structure whose time points (nodes) represent the instantaneous states. The meaning of the belief-desire-intention (BDI) operators is given through accessibility relations that relate the possible worlds at each time point and thus can possibly vary over time. The meaning of temporal operators is instead related to the temporal accessibility relation defined by the the branching-time structure.

We start recalling the notion of *tree-structure*.

For $k > 0$, a k-ary *tree-structure* is a pair $(\mathcal{T}, \mathcal{R})$ where $\mathcal{T} \subseteq [k]^*$ is a prefix-closed set and $\mathcal{R} = \{(t, t') \mid t, t' \in \mathcal{T} \text{ and } t' = t.i \text{ for } i \in [k]\}$. The empty word ε denotes the root of the tree. We assume $\mathcal{T}$ to be infinite, and refer to the elements of $\mathcal{T}$ as *time points*.

Definition 1. *A* structure *for* $\text{CTL}^*_{\text{BDI}}$ *formulas is a tuple* $\mathcal{M} = (AP, \mathcal{T}, \mathcal{R}, \mathcal{W}, \mathcal{B}, \mathcal{D}, \mathcal{I})$ *where:*

- AP *is a set of atomic propositions;* $(\mathcal{T}, \mathcal{R})$ *is a tree-structure;*
- $\mathcal{W}$ *is a set of possible worlds where each world* $w \in \mathcal{W}$ *is a tuple* $(\mathcal{T}_w, \mathcal{R}_w, \mathcal{L}_w)$ *where* $\mathcal{T}_w \subseteq \mathcal{T}$, $\mathcal{R}_w$ *is the restriction of* $\mathcal{R}$ *to* $\mathcal{T}_w$, $(\mathcal{T}_w, \mathcal{R}_w)$ *is a tree-structure and* $\mathcal{L}_w : \mathcal{T}_w \to 2^{AP}$ *assigns a set of atomic propositions to each time point of* w*;*
- *for* $\mathcal{K} \in \{\mathcal{B}, \mathcal{D}, \mathcal{I}\}$, $\mathcal{K} \subseteq \mathcal{W} \times \mathcal{T} \times \mathcal{W}$ *is such that for* $(w, t, v) \in \mathcal{K}$, $t \in \mathcal{T}_w \cap \mathcal{T}_v$ *must hold (i.e.,* BDI *accessibility relations are consistently defined with respect to the world time points).*

$\mathcal{R}$ *(resp.,* $\mathcal{B}$, $\mathcal{D}$, $\mathcal{I}$*) is called the temporal (resp., belief, desire, intention) accessibility relation.* □

A *path* π in a world $w = (\mathcal{T}_w, \mathcal{R}_w, \mathcal{L}_w)$ is a sequence of time points $t_0 t_1 \dots$ such that $(t_i, t_{i+1}) \in \mathcal{R}_w$ for $i \geq 0$. The meaning of formulas is given by the satisfaction relation that is defined starting from a time point for state formulas and along a path for path formulas. We omit here a formal definition which can be obtained along the same lines as for CTL* and just recall the case of BDI operators. For a structure $\mathcal{M} = (AP, \mathcal{T}, \mathcal{R}, \mathcal{W}, \mathcal{B}, \mathcal{D}, \mathcal{I})$, $w \in \mathcal{W}$ and a time point t:

- $\mathcal{M}, w, t \models \text{BEL}\varphi$ iff for all $v \in \mathcal{W}$ such that $(w, t, v) \in \mathcal{B}$, it must hold $\mathcal{M}, v, t \models \varphi$ (similarly for $\text{DES}\varphi$ and $\text{INT}\varphi$);

We say $\mathcal{M}$ *satisfies* a $\text{CTL}^*_{\text{BDI}}$ formula φ at a world w, written $\mathcal{M}, w \models \varphi$, iff $\mathcal{M}, w, \textit{root} \models \varphi$.

Model-checking. We assume that the reader is familiar with the notions of Kripke structure (a finite-state transition system whose states are labeled with atomic propositions) and its tree-unrolling (see [7]), and finite automaton (see [10]). For a finite automaton A and a state s, we denote with $L(A, s)$ the language accepted by A assuming s as the *sole* accepting state (i.e., the language accepted by A is $L(A) = \bigcup_{s \in F} L(A, s)$ where F is the accepting set of A).

For the model-checking problem we consider (1) a finite number of possible worlds each corresponding to the unrolling of a Kripke structure, and (2) each BDI relation defined by a finite automaton over the paths of the corresponding tree-structures. We refer to such structures as *finite-state structures*.

We omit a formal definition of above notion and just observe that a finite-state structure $\mathcal{M} = (AP, \mathcal{T}, \mathcal{R}, \mathcal{W}, \mathcal{B}, \mathcal{D}, \mathcal{I})$ has a finite representation of the form $(AP, k, \mathcal{W}, \bar{K}, A_{\mathcal{B}}, A_{\mathcal{D}}, A_{\mathcal{I}}, \mu_{\mathcal{B}}, \mu_{\mathcal{D}}, \mu_{\mathcal{I}})$ where:

- $\bar{K} = \{(K_w, s_w) \mid w \in \mathcal{W}\}$ and s_w is a state for a Kripke structure K_w of arity k for $w \in \mathcal{W}$, and
- for $\mathcal{K} \in \{\mathcal{B}, \mathcal{D}, \mathcal{I}\}$, $A_{\mathcal{K}}$ is finite automaton over time points, $\mu_{\mathcal{K}}$ maps each state of $A_{\mathcal{K}}$ to a subset of $\mathcal{W} \times \mathcal{W}$, and denoting $Q_{\mathcal{K}}$ the set of states of $A_{\mathcal{K}}$, $\mathcal{K} = \bigcup_{s \in Q_{\mathcal{K}}} \{(w, t, w') \mid (w, w') \in \mu_{\mathcal{K}}(s)$ and $t \in L(A_{\mathcal{K}}, s) \cap \mathcal{T}\}$.

In the following, we will denote finite-state structures through their finite representation.

Definition 2. *Given a finite-state structure $\mathcal{M} = (AP, k, \mathcal{W}, \bar{K}, A_{\mathcal{B}}, A_{\mathcal{D}}, A_{\mathcal{I}}, \mu_{\mathcal{B}}, \mu_{\mathcal{D}}, \mu_{\mathcal{I}})$, a world w and a $\text{CTL}^*_{\text{BDI}}$ formula φ, the* $\text{CTL}^*_{\text{BDI}}$ model-checking problem *asks whether $\mathcal{M}, w \models \varphi$.*

3 Decision algorithms

In this section, we discuss our solution to the introduced decision problems.

We start giving the construction of a finite graph that captures the entire input model. This graph essentially consists of the synchronous cross product of the Kripke structures defining the possible worlds along with the finite automata capturing the BDI-accessibility relations. For an input formula φ, the decision procedure for CTL_{BDI} is then obtained by labeling each node of this graph with the φ sub-formulas that hold true at it. For this, we adapt the decision algorithm given for the CTL model-checking (see [8]) which iteratively label the states of a Kripke structure by considering sub-formulas of increasing sizes. We conclude the section by arguing how to extend the CTL_{BDI} model-checking algorithm to decide the $\text{CTL}^*_{\text{BDI}}$ model-checking problem and then discussing the computational complexity of this approach.

Construction of the graph $\mathcal{G}_{\mathcal{M}}$. For a set of worlds $\mathcal{W} = \{w_1, \dots, w_n\}$ for $n > 0$, and a finite-state structure $\mathcal{M} = (AP, k, \mathcal{W}, \bar{K}, A_{\mathcal{B}}, A_{\mathcal{D}}, A_{\mathcal{I}}, \mu_{\mathcal{B}}, \mu_{\mathcal{D}}, \mu_{\mathcal{I}})$ with $\bar{K} = \{(K_w, s_w) \mid w \in \mathcal{W}\}$ and $K_w = (S_w, \nu_w, \lambda_w)$, the graph $G_{\mathcal{M}}$ is defined as follows (i, j range over the set $\{1, \dots, n\}$, and $\mathcal{K} \in \{\mathcal{B}, \mathcal{D}, \mathcal{I}\}$ unless otherwise specified).

The *vertices of $G_{\mathcal{M}}$* are of the form $(w_i, s_1, \dots, s_n, q_{\mathcal{B}}, q_{\mathcal{D}}, q_{\mathcal{I}})$ where: (1) w_i denotes the current world, (2) s_j either belongs to K_{w_j} and is the current state of world w_j or is a dummy state $\perp$ denoting that the current one is not a time point of world w_j, and (3) $q_{\mathcal{K}}$ is the current state of $A_{\mathcal{K}}$.

The *edges of $G_{\mathcal{M}}$* come from the accessibility relations of $\mathcal{M}$ and are labeled consistently: edges derived from the temporal accessibility relation are labeled with a corresponding index from $[k]$ (recall that k is the arity of the Kripke structures) while those derived from the accessibility relation $\mathcal{K}$ with a fresh symbol $\sigma_{\mathcal{K}}$ for $\mathcal{K} \in \{\mathcal{B}, \mathcal{D}, \mathcal{I}\}$. Formally, denote $\bar{\nu}_w$ the total function obtained by completing ν_w by assigning $\perp$

whenever it is not defined, i.e., $\bar{\nu}_w(s,j) = \nu(s,j)$ if $\nu_w(s,j)$ is defined and $\bar{\nu}_w(s,j) = \bot$ otherwise . For vertices $u = (w_i, s_1, \ldots, s_n, q_\mathcal{B}, q_\mathcal{D}, q_\mathcal{I})$ and $u' = (w_{i'}, s'_1, \ldots, s'_n, q'_\mathcal{B}, q'_\mathcal{D}, q'_\mathcal{I})$ of $G_\mathcal{M}$, (u, γ, u') is an edge of $G_\mathcal{M}$ iff either one of the following cases holds (we assume that components of u' equals the corresponding ones from u unless differently specified):

- $\gamma \in [k]$, $i' = i$, $s'_j = \bar{\nu}_{w_i}(s_j, \gamma)$ for $j \in [n]$, and $(q_\mathcal{K}, \gamma, q'_\mathcal{K})$ is a transition of $A_\mathcal{K}$ for $\mathcal{K} \in \{\mathcal{B}, \mathcal{D}, \mathcal{I}\}$ (we say that u' is a γ-*temporal successor* of u);
- $\gamma \subseteq \{\sigma_\mathcal{B}, \sigma_\mathcal{D}, \sigma_\mathcal{I}\}$, $\gamma \neq \emptyset$, and for $\sigma_\mathcal{K} \in \gamma$, $\mu_\mathcal{K}(q_\mathcal{K}) = (w_i, w_{i'})$ [u' is a $\mathcal{K}$ successor of u for each $\sigma_\mathcal{K} \in \gamma$.

Note that each vertex of $G_\mathcal{M}$ has at most $k + n$ successors.

Define $\lambda_\mathcal{M}(u) = \lambda_{w_i}(s_i)$, for each vertex $u = (w_i, s_1, \ldots, s_n, q_\mathcal{B}, q_\mathcal{D}, q_\mathcal{I})$ of $G_\mathcal{M}$. The labeled graph $\mathcal{G}_\mathcal{M}$ is defined as $G_\mathcal{M}$ with the labeling function $\lambda_\mathcal{M}$. Note that $\mathcal{G}_\mathcal{M}$ differs from Kripke structures only for the distinction of the transitions into temporal and BDI ones.

Properties of $\mathcal{G}_\mathcal{M}$. We denote $\tau(\mathcal{G}_\mathcal{M}, u)$ the tree-structure obtained from $\mathcal{G}_\mathcal{M}$ starting from u and by unrolling the loops of $\mathcal{G}_\mathcal{M}$.

By $\tau(\mathcal{G}_\mathcal{M}, u)$, we can define the satisfiability of $\text{CTL}^*_{\text{BDI}}$ (and hence of CTL_{BDI}) formulas w.r.t. $\mathcal{G}_\mathcal{M}$ by treating a formula of the form $\mathcal{K}\varphi$ as the corresponding temporal logic formula $\forall \bigcirc \varphi$ where the universal quantification is restricted to only the $\mathcal{K}$ successors of the current vertex. Analogously, standard path quantifiers are restricted to only the temporal successors. The formal definition can be easily obtained from semantics of $\text{CTL}^*_{\text{BDI}}$ and the observations above. Thus, we omit it here, and again use $\mathcal{G}_\mathcal{M}, t \models \varphi$ (resp., $\mathcal{G}_\mathcal{M}, \pi \models \varphi$) meaning that φ holds in $\mathcal{G}_\mathcal{M}$ starting from time point t (resp., along path π).

Directly from the given semantics, we have that the $\text{CTL}^*_{\text{BDI}}$ (resp., CTL_{BDI}) model-checking problem reduces to the corresponding question on the labeled graph $\mathcal{G}_\mathcal{M}$. For a world w, we define the *initial* vertex of $\mathcal{G}_\mathcal{M}$ corresponding to w the only vertex of the form $(w, s_{w_1}, \ldots, s_{w_n}, q^0_\mathcal{B}, q^0_\mathcal{D}, q^0_\mathcal{I})$ where $q^0_\mathcal{K}$ is the initial state of $A_\mathcal{K}$ for $\mathcal{K} \in \{\mathcal{B}, \mathcal{D}, \mathcal{I}\}$ (recall that each s_{w_i} is the state coupled with the Kripke structure K_{w_i} in the finite-state structure we have fixed earlier in this section).

Lemma 1. *For a world w and a $\text{CTL}^*_{\text{BDI}}$ formula φ, we get that $\mathcal{M}, w \models \varphi$ iff $\mathcal{G}_\mathcal{M}, u \models \varphi$, where u is the initial state of of $\mathcal{G}_\mathcal{M}$ corresponding to w.*

A crucial property of $\mathcal{G}_\mathcal{M}$ is that as for standard Kripke structures the truth of branching-time state formulas depends only on the state, i.e., a state formula φ is true at a time point t of $\tau(\mathcal{G}_\mathcal{M}, u)$ if and only if it is true at any other time point t' such that t and t' correspond to the same vertex of $\mathcal{G}_\mathcal{M}$. To see this, for a k-ary tree-structure $(\mathcal{T}, \mathcal{R})$, we define the *abstract subtree* rooted at $t \in \mathcal{T}$ as the k-ary tree-structure $(\mathcal{T}', \mathcal{R}')$ where $\mathcal{T}' = \{t' \mid t.t'\}$ and $\mathcal{R}' = \{(t', t'.i) \in \mathcal{R} \mid t' \in \mathcal{T}'\}$. Thus, directly from the definition of $\tau(\mathcal{G}_\mathcal{M}, u)$, we get that the abstract sub-trees rooted at t and t' coincide for all time points t, t' of $\tau(\mathcal{G}_\mathcal{M}, u)$ that correspond to the same vertex of $\mathcal{G}_\mathcal{M}$ and hence the property stated above holds.

Lemma 2. *Given a $\text{CTL}^*_{\text{BDI}}$ state formula φ, for all time points t, t' of $\tau(\mathcal{G}_\mathcal{M}, u)$ that correspond to the same vertex of $\mathcal{G}_\mathcal{M}$ we get: $\tau(\mathcal{G}_\mathcal{M}, u), t \models \varphi$ iff $\tau(\mathcal{G}_\mathcal{M}, u), t' \models \varphi$.*

Decision algorithms. Lemma 2 allows us to give for our model-checking questions two fixed-point decision algorithms in the style of those given for CTL and CTL*. Such algorithms proceed bottom-up on the syntactic structure of φ and starting from the labeling given by the truth of the atomic propositions, progressively label each vertex u of the graph with the sub-formulas that holds true there. The rules of the algorithm for CTL_{BDI}, denoted `Alg`-CTL_{BDI}, are given in Figure 1.

To get a decision algorithm for $\text{CTL}^*_{\text{BDI}}$ we can reason similarly to how a decision algorithm CTL* is obtained from that for CTL (see [7] for details). In particular, for a path formula φ denote with φ' the formula obtained by replacing in φ its state sub-formulas with new atomic propositions. Thus, the truth of φ at a vertex u of $\mathcal{G}_\mathcal{M}$ is determined by a query to an LTL model-checking algorithm on φ' by taking for the added atomic proposition the evaluation given by *lab* to the corresponding state formulas. We denote with `Alg`-$\text{CTL}^*_{\text{BDI}}$ the resulting algorithm. The correctness of algorithms `Alg`-CTL_{BDI} and `Alg`-$\text{CTL}^*_{\text{BDI}}$ is a consequence of Lemmas 1 and 2, and the above observations. Thus we have:

Let $\mathcal{M} = (AP, k, \mathcal{W}, \bar{K}, A_{\mathcal{B}}, A_{\mathcal{D}}, A_{\mathcal{I}}, \mu_{\mathcal{B}}, \mu_{\mathcal{D}}, \mu_{\mathcal{I}})$ where:
$\bar{K} = \{(K_w, s_w) \mid w \in \mathcal{W}\}$ and $K_w = (S_w, \nu_w, \lambda_w)$.

Initialization.
For each vertex u of $\mathcal{G}_{\mathcal{M}}$, set $lab(u) = \lambda_{\mathcal{M}}(u)$.

Update rules.
For each vertex u of $\mathcal{G}_{\mathcal{M}}$:

1. if $\varphi = \neg\psi$, then $\varphi \in lab(u)$ iff $\psi \notin lab(u)$;
2. if $\varphi = \varphi_1 \vee \varphi_2$ then $\varphi \in lab(u)$ iff either $\varphi_1 \in lab(u)$ or $\varphi_2 \in lab(u)$;
3. if $\varphi = \exists \bigcirc \psi$, then $\varphi \in lab(u)$ iff there is a temporal successor u' of u such that $\psi \in lab(u')$;
4. if $\varphi = \forall \bigcirc \psi$, then $\varphi \in lab(u)$ iff for all temporal successors u' of u it holds that $\psi \in lab(u')$;
5. if $\varphi = \exists(\varphi_1 \mathcal{U} \varphi_2)$, then $\varphi \in lab(u)$ iff either $\varphi_2 \in lab(u)$, or $\varphi_1 \in lab(u)$ and there is a temporal successor u' of u such that $\varphi \in lab(u')$;
6. if $\varphi = \forall(\varphi_1 \mathcal{U} \varphi_2)$, then $\varphi \in lab(u)$ iff either $\varphi_2 \in lab(u)$, or $\varphi_1 \in lab(u)$ and for all temporal successors u' of u it holds that $\varphi \in lab(u')$;
7. if $\varphi = \text{BEL}\psi$, then $\varphi \in lab(u)$ iff for all $\mathcal{B}$ successors u' of u it holds that $\psi \in lab(u')$ (similarly for $\text{DES}\varphi$ and $\text{INT}\varphi$).

Figure 1: Fixed-point decision algorithm `Alg-`CTL_{BDI} for CTL_{BDI} model-checking.

Lemma 3. *Given a* CTL_{BDI} *(resp.,* $\text{CTL}^*_{\text{BDI}}$*) state formula* φ*, a finite-state structure* $\mathcal{M}$ *and a world* w*,* $\mathcal{M}, w \models \varphi$ *iff* $\varphi \in lab(u)$ *where lab is the labeling computed by* `Alg-`CTL_{BDI} *(resp.,* `Alg-`$\text{CTL}^*_{\text{BDI}}$*) and* u *is the initial state of of* $\mathcal{G}_{\mathcal{M}}$ *corresponding to* w*.*

Computational complexity. We observe that the construction of $\mathcal{G}_{\mathcal{M}}$ causes an exponential blow-up in the size of $\mathcal{M}$. In fact, the number of vertices of $\mathcal{G}_{\mathcal{M}}$ is $O(n \cdot \chi^n \cdot \eta^3)$ where χ is the maximum number of states over the n Kripke structures denoting the possible worlds of $\mathcal{M}$ and η is the maximum number of states over the finite-state automata denoting the BDI accessibility relations of $\mathcal{M}$. Moreover, for each vertex of $\mathcal{G}_{\mathcal{M}}$ there are at most $k + n$ outgoing edges where k is the arity of the Kripke structures. Thus, the overall number of $\mathcal{G}_{\mathcal{M}}$ edges is $O(k \cdot n^2 \cdot \chi^n \cdot \eta^3)$. For a formula φ the number of its sub-formulas is linear in the size of φ (denoted $|\varphi|$). Thus the fixed-point algorithm `Alg-`CTL_{BDI} will converge in at most $O(|\varphi| \cdot k \cdot n^2 \cdot \chi^n \cdot \eta^3)$ steps, and since each step require at most $O(n)$ time, we get the following result.

Theorem 1. *The* CTL_{BDI} *model-checking problem can be solved in time exponential in the number of the worlds, and polynomial in the size of the formula and the size of the automata capturing the* BDI *relations.*

Since LTL model-checking can be solved in time exponential in the size of the formula and linear in the size of the model [13], `Alg-`$\text{CTL}^*_{\text{BDI}}$ requires exponential time also in the size of the formula.

Theorem 2. *The* $\text{CTL}^*_{\text{BDI}}$ *model-checking problem can be solved in time exponential in the number of worlds and the size of the formula, and polynomial in the size of the automata capturing the* BDI *relations.*

4 Conclusions

In this paper, we have presented some preliminary results on the model-checking problem of CTL_{BDI} and $\text{CTL}^*_{\text{BDI}}$ over finite-state models. We have shown that these decision problems are decidable in time exponential in both the size of the input model and the size of the input formula. Our results extend the decidability of the considered logics to systems that exhibit infinitely many time points resulting from the unrolling of the finite-state models.

As future research, we wish to investigate further the considered decision problems. First, we will look deeper into their computational complexity and possibly show tight complexity bounds for the problems stated in this paper. Second, we will consider modular descriptions of systems where each world is composed of modules that can call each other possibly recursively, similarly to what is done for standard temporal logics (see [1, 11, 2]). This will give a more faithful representation for many real systems and will yield more succinct models (modules can be shared among worlds). Moreover, the presence of recursive calls will extend further the class of models considered for the logics CTL_{BDI} and $\text{CTL}^*_{\text{BDI}}$.

References

[1] R. Alur, M. Benedikt, K. Etessami, P. Godefroid, T. W. Reps, and M. Yannakakis. Analysis of recursive state machines. *ACM Trans. Program. Lang. Syst.*, 27(4):786–818, 2005.

[2] B. Aminof, O. Kupferman, and A. Murano. Improved model checking of hierarchical systems. *Inf. Comput.*, 210:68–86, 2012.

[3] M. Benerecetti, F. Giunchiglia, and L. Serafini. Model checking multiagent systems. *J. Log. Comput.*, 8(3):401–423, 1998.

[4] J. Blee, D. Billington, G. Governatori, and A. Sattar. Levels of modality for BDI logic. *J. Applied Logic*, 9(4):250–273, 2011.

[5] R. H. Bordini, M. Fisher, W. Visser, and M. J. Wooldridge. Verifying multi-agent programs by model checking. *Autonomous Agents and Multi-Agent Systems*, 12(2):239–256, 2006.

[6] M. E. Bratman. Intentions, plans, and practical reason. *Harvard University Press*, 1987.

[7] E. A. Emerson. Temporal and modal logic. In *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)*, pages 995–1072. The MIT Press/Elsevier, 1990.

[8] E. A. Emerson and E. M. Clarke. Using branching time temporal logic to synthesize synchronization skeletons. *Sci. Comput. Program.*, 2(3):241–266, 1982.

[9] J. Y. Halpern and M. Y. Vardi. The complexity of reasoning about knowledge and time. i. lower bounds. *J. Comput. Syst. Sci.*, 38(1):195–237, 1989.

[10] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to automata theory, languages, and computation - (2. ed.).* Addison-Wesley series in computer science. Addison-Wesley-Longman, 2001.

[11] S. La Torre, M. Napoli, M. Parente, and G. Parlato. Verification of scope-dependent hierarchical state machines. *Inf. Comput.*, 206(9-10):1161–1177, 2008.

[12] J.-J. C. Meyer, J. Broersen, and A. Herzig. Bdi logics. In *Handbook of Epistemic Logic*, pages 453–498. College Publications, 2015.

[13] A. Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977*, pages 46–57, 1977.

[14] A. S. Rao and M. P. Georgeff. Modeling rational agents within a bdi-architecture. In J. F. Allen, R. Fikes, and E. Sandewall, editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91). Cambridge, MA, USA, April 22-25, 1991.*, pages 473–484. Morgan Kaufmann, 1991.

[15] A. S. Rao and M. P. Georgeff. A model-theoretic approach to the verification of situated reasoning systems. In R. Bajcsy, editor, *Proceedings of the 13th International Joint Conference on Artificial Intelligence. Chambéry, France, August 28 - September 3, 1993*, pages 318–324. Morgan Kaufmann, 1993.

[16] A. S. Rao and M. P. Georgeff. Decision procedures for BDI logics. *J. Log. Comput.*, 8(3):293–342, 1998.