

Bachelors Thesis:  
Model Checking Metric First Order  
Temporal Logic over Dynamic Graphs

Jana Melani Klitzsch

Submitted on the 12th of May 2025

First Examiner: Prof. Dr. Martin Lange

Second Examiner: Dr. Florian Bruse

**U N I K A S S E L** | E L E C T R I C A L  
**V E R S I T Ä T** | E N G I N E E R I N G  
C O M P U T E R  
S C I E N C E

## **Declaration**

I hereby declare that this thesis was written by myself and presents my own work. All sources and resources used were properly cited. This thesis has not been submitted, wholly or in part, in any previous submissions for a degree. The submitted electronic version of this thesis matches the submitted printed version. The code handed in electronically together with this thesis was written by myself.

---

Klitzsch, Jana Melani  
Kassel, 12th of May 2025

## **Acknowledgments**

Thank you to Prof. Dr. Martin Lange for giving me the opportunity to work on this thesis, and thank you to Dr. Florian Bruse for agreeing to be the second examiner on this thesis.

I want to especially thank my supervisor Marco Sälzer, whose feedback and support in the writing process has been invaluable.

Thank you to all the people in my life who have supported me during the time of writing this thesis. Without you this thesis would not exist today as it does. And while my life has been turbulent during the writing process, your support helped me get through those troubles way better than I would have alone.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Section Summary . . . . .	1
1.2	Related Work . . . . .	2
<b>2</b>	<b>Preliminaries</b>	<b>3</b>
2.1	Static Graphs . . . . .	3
2.2	Dynamic Graphs . . . . .	3
2.3	Metric First Order Temporal Logic (MFOTL) . . . . .	5
2.4	Representing Dynamic Graphs in MFOTL . . . . .	9
2.5	Propositional Logic and Boolean SAT . . . . .	10
<b>3</b>	<b>Model Checking for Dynamic Graphs</b>	<b>12</b>
<b>4</b>	<b>Reducing Dynamic Graph problems to Boolean SAT</b>	<b>14</b>
4.1	Boolean Formula for Graph Equivalence . . . . .	15
4.2	Boolean Formula for Events . . . . .	18
4.3	Translation of a MFOTL Formula . . . . .	22
4.4	Reductions to SAT . . . . .	26
<b>5</b>	<b>Implementation</b>	<b>29</b>
<b>6</b>	<b>Discussion and Outlook</b>	<b>31</b>
<b>7</b>	<b>References</b>	<b>33</b>
<b>A</b>	<b>Implementation Test Results</b>	<b>34</b>

# 1 Introduction

Model checking [2] is one of the multiple different forms of formal verification. We can use it to check if a structure fulfills properties specified in different logics, like linear temporal logic (LTL) and computation tree logic (CTL). Different logics lend themselves for describing different properties and for different kinds of structures.

The structures we focus on in this thesis are dynamic graphs. Dynamic graphs can be an important way to structure different kinds of data. From social networks analysis [9] to epidemiology [12], there are many different fields in which they are used. With the rising use of machine learning in many different fields, it has also found more use with dynamic graphs, even if their static counterparts are more common [11].

However, with the rising use of machine learning, we also have to take into account how to use and create these machine learning models ethically. The aim of having AI be trustworthy is also one of the goal in the EU's ethics guidelines for trustworthy AI [4]. Formal verification can be one important part of achieving trustworthiness for different forms of AI [10].

Thus in this thesis, we lay the foundations for an approach for formal verification of dynamic graph neural-networks. We do this by using an extension of LTL called metric first-order temporal logic (MFOTL) [3]. It extends it by adding existential- and all-quantifiers, as well as adding intervals to the temporal operators of LTL. The temporal structures that MFOTL uses for its models lend themselves well for modeling dynamic graphs.

While there are different ways to model dynamic graphs, we define them using static graphs with potentially multiple edge and attribute relations. Thus different kinds of graphs can be modeled. For example, graphs with edge attributes can be modeled by having an additional edge relation represent that attribute. We then model these dynamic graphs directly within temporal structures of MFOTL using the same edge and attribute relations. We also add an additional unary relation for nodes, indicating whether or not a node exists at that current timestamp.

We also define two different model checking problems based on these models of dynamic graphs, a basic one that simply checks if the dynamic graph fulfills the property specified in the formula, and an extended one that checks if the dynamic graph could fulfill the property if more events were applied to it. Furthermore we also define a satisfiability problem for dynamic graphs and MFOTL, where the question is whether a dynamic graph with the specified number of nodes, timestamps, and edge and attribute relations exists, that fulfills a specific formula.

We then reduce all three problems to the satisfiability problem of propositional logic and prove the correctness of all these reductions. Additionally we also discuss the feasibility of model checking dynamic graphs this way. For this we also look at test results from a proof-of-concept implementation of the reduction, where we decide the resulting Boolean formulas using a SAT-Solver. Specifically we look at how different factors influence the runtime of deciding the decision problems. Finally, we also discuss which directions future research in this topic can focus on.

## 1.1 Section Summary

In Section 2 we define the mathematical foundations we need for this thesis. We define two variants of dynamic graphs, as well as the syntax and semantics for both MFOTL

and propositional logic. We also define the way in which we model dynamic graphs as temporal structures, which are the structures for which MFOTL is defined. Here we use one of the two dynamic graph models, the discrete-time dynamic graphs (DTDGs). However, the other dynamic graph model we present can be converted to DTDGs. We also define events, both for the other type of dynamic graphs, and for use in the decision problems we define.

We define our newly introduced model checking problems in Section 3. Additionally, we go over some restrictions we put in place for our MFOTL formulas. We make these restrictions so that the reduction of the model checking problems to the Boolean SAT problem are possible. Furthermore, we also define a SAT problem for dynamic graphs and MFOTL.

In Section 4 we reduce all three decision problems to Boolean SAT. We do this by creating multiple propositional formulas, which then in various combinations make up the reductions for the different problems. When introducing a new formula we also prove Lemmas for each formula, to then prove the correctness of each reduction.

We present the results of a proof-of-concepts implementation of these reductions in Section 5. Here we present how different parameters, such as the number of nodes of the graph, or how many timestamps it has, can affect the runtime. We also present the data from the runtime tests in Appendix A.

Finally in Section 6, we give an outlook how future research could expand on the foundations laid in this work.

## 1.2 Related Work

MFOTL was first introduced in Basin et al. [3]. There the logic was used in the context of runtime monitoring, where they used a safety fragment. Similarly to them, in this thesis we also only allow bounded future operators to remain decidable.

There has also been work on giving explanations for decisions during runtime monitoring in Lima, Huerta y Munive, and Traytel [8]. Within our implementation we base our encoding of MFOTL formulas on how they are used within the WHYMON tool used and introduced within that paper.

There have been many works on dynamic graphs, and machine learning models based on them. We base our definitions for dynamic graphs on Kazemi et al. [6], which surveys representation learning. They focus on three different types of representation learning, being node classification, edge prediction, and graph classification. Skarding, Gabrys, and Musial [11] focus on giving a more general view of different machine learning models for dynamic graphs. Among other things, they do also reach the conclusion that since real world datasets are often huge, scalability of models can be an issue.

## 2 Preliminaries

For the purpose of this thesis, we consider 0 to be part of  $\mathbb{N}$ . We base the definitions for static and dynamic graphs on Kazemi et al. [6]. However, we do also extend them.

### 2.1 Static Graphs

Before we consider the dynamic graphs which are the focus of this thesis, we must first define static graphs.

**Definition 2.1.** Let  $V$  be a set of vertices, let  $E_0, \dots, E_{n-1}$  be  $n$  binary relations, and  $A_0, \dots, A_{m-1}$  be  $m$  unary relations, with  $E_i \subseteq V \times V$  and with  $A_i \subseteq V$ . We call  $E_i$  edge relations and  $A_i$  attribute relations, or edges and attributes.

We call  $G = (V, E_0, \dots, E_{n-1}, A_0, \dots, A_{m-1})$  a static graph. We say a vertex  $v \in V$  has an attribute  $A_i$  if  $v \in A_i$ . We say there is an  $E_i$ -edge between  $v_0, v_1 \in V$  if  $(v_0, v_1) \in e_i$ . Furthermore, we also refer to vertices as nodes.

Note that we can also model unlabeled graphs without node attributes, by including  $m = 0$  attribute relations.

**Example 2.2.** Figure 1 shows a static graph  $G = (\{v_0, v_1, v_2\}, E_0, E_1, A_0, A_1)$  with  $E_0 = \{(v_0, v_1), (v_1, v_1)\}$ ,  $E_1 = \{(v_0, v_2), (v_1, v_2)\}$ ,  $A_0 = \{v_1\}$ ,  $A_1 = \{v_0, v_1\}$ . Here we note  $E_0$  with fully drawn lines, and  $E_1$  with dotted lines.  $A_0$  and  $A_1$  are depicted as vectors of 0 and 1, where  $A_0$  is at the top of the vector.

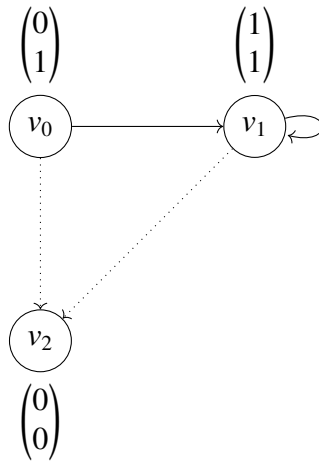


Figure 1: A static graph

### 2.2 Dynamic Graphs

There are two main types of dynamic graphs. Firstly there are continuous-time dynamic graphs (CTDGs). To properly define them, we first need to define events.

**Definition 2.3.** Let  $\text{Types} = \{\text{AddNode}, \text{RemoveNode}, \text{AddEdge}, \text{RemoveEdge}, \text{AddAttr}, \text{RemoveAttr}, \text{NoneEvent}\}$  be a set of event types, let  $G = (V, E_0, \dots, E_{n-1}, A_0, \dots, A_{m-1})$  be a static graph, and let  $V'$  be a set of nodes disjunctive from  $V$ . We call a triple  $(x, e, t)$  an event, where  $x \in \text{Types}$ ,  $t \in \mathbb{R}$ , and  $e$  depends on  $x$  as follows:

- $e \in V'$  if  $x = \text{AddNode}$
- $e \in V$  if  $x = \text{RemoveNode}$ .
- $e \in \{E_0, \dots, E_{n-1}\} \times V \times V$  if  $x \in \{\text{AddEdge}, \text{RemoveEdge}\}$ .
- $e \in \{A_0, \dots, A_{m-1}\} \times V$  if  $x \in \{\text{AddAttr}, \text{RemoveAttr}\}$
- $e = \emptyset$  if  $x = \text{NoneEvent}$

For an event  $o = (x, e, t)$  we also define the function type as  $\text{type}(o) = x$ . We also write define the function timestamp as  $\text{timestamp}(o) = t$ .

We can now apply these events to static graphs as follows, to get a new static graph.

**Definition 2.4.** Let  $G = (V, E_0, \dots, E_{n-1}, A_0, \dots, A_{m-1})$  be a static graph, and let  $o = (x, e, t)$  be an event. We can apply that event to a graph, resulting in a new graph  $G' = (V', E'_0, \dots, E'_{n-1}, A'_0, \dots, A'_{m-1})$ . Which is defined as follows:

- If  $x = \text{AddNode}$ , then  $V' = V \cup \{e\}$ ,  $E'_i = E_i$ ,  $A'_j = A_j$ , for  $i \in \{0, \dots, n-1\}$ ,  $j \in \{0, \dots, m-1\}$ ,
- If  $x = \text{RemoveNode}$ , then  $V' = V \setminus \{e\}$ ,  $E'_i = E_i \cap V' \times V'$ ,  $A'_j = A_j \cap V'$  for  $i \in \{0, \dots, n-1\}$ ,  $j \in \{0, \dots, m-1\}$ ,
- If  $x = \text{AddEdge}$  and  $e = (E_i, v_0, v_1)$ , then  $V' = V$ ,  $E'_i = E_i \cup \{(v_0, v_1)\}$ ,  $E'_k = E_k$ ,  $A'_j = A_j$  for  $k \in \{0, \dots, n-1\} \setminus \{i\}$ ,  $j \in \{0, \dots, m-1\}$ ,
- If  $x = \text{RemoveEdge}$  and  $e = (E_i, v_0, v_1)$ , then  $V' = V$ ,  $E'_i = E_i \setminus \{(v_0, v_1)\}$ ,  $E'_k = E_k$ ,  $A'_j = A_j$  for  $k \in \{0, \dots, n-1\} \setminus \{i\}$ ,  $j \in \{0, \dots, m-1\}$ ,
- If  $x = \text{AddAttr}$  and  $e = (A_j, v_0)$ , then  $V' = V$ ,  $E'_i = E_i$ ,  $A'_j = A_j \cup \{v_0\}$ ,  $A'_k = A_k$  for  $k \in \{0, \dots, m-1\} \setminus \{j\}$ ,  $i \in \{0, \dots, n-1\}$ ,
- If  $x = \text{RemoveAttr}$  and  $e = (A_j, v_0)$ , then  $V' = V$ ,  $E'_i = E_i$ ,  $A'_j = A_j \setminus \{v_0\}$ ,  $A'_k = A_k$  for  $k \in \{0, \dots, m-1\} \setminus \{j\}$ ,  $i \in \{0, \dots, n-1\}$
- If  $x = \text{NoneEvent}$ ,  $G = G'$ .

We also define  $o(G) = G'$  for a given event  $o$  and a graph  $G$  as a function from the set of static graphs to the set of static graphs, where  $G'$  is the graph you get by applying  $o$  to  $G$ .

For the purposes of this thesis, we assume that an event when applied to a static graph changes that graph, unless the event is (`NoneEvent`).

Using these events and their application to static graphs, we can define one type of dynamic graphs, being CTDGs, as follows:

**Definition 2.5.** Let  $G$  be a static graph and let  $O$  be a set of events with timestamps bigger than 0. We call a pair  $\mathcal{G} = (G, O)$  a continuous-time dynamic graph (CTDG). We call  $\text{snapshot}(\mathcal{G}, t)$  the snapshot of  $\mathcal{G}$  at timestamp  $t$ . We get  $\text{snapshot}(\mathcal{G}, t)$  by applying all events  $o \in O$  with  $\text{timestamp}(o) \in ]0, t]$  in ascending order. To avoid ambiguity, we restrict  $O$  so that any 2 events have different timestamps.

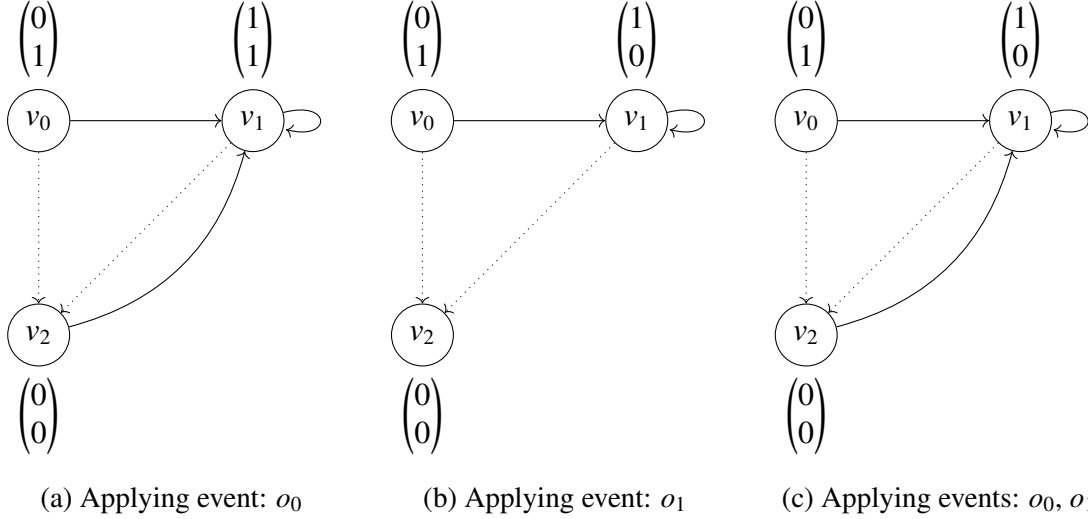


Figure 2: Examples of applying the events from example 2.6 to the graph from example 2.2. We can also see a snapshot of the CTDG  $\mathcal{G}$  from 2.6 in (c).

**Example 2.6.** *The following are examples of events for the static graph  $G$  from example 2.2.*

- $o_0 = (\text{AddEdge}, (E_0, v_2, v_1), 2)$
- $o_1 = (\text{RemoveAttr}, (A_1, v_1), 9.1)$

We can see both resulting graphs in Figure 2.

Furthermore, we can use both of these events to create a CTDG  $\mathcal{G} = (G, O)$  with  $O = \{o_0, o_1\}$ . We can also see a snapshot of  $\mathcal{G}$  at timestamp 10 in 2.

The second relevant type of dynamic graphs are *discrete-time dynamic graphs* (DT-DGs). These will be the graphs that we directly use for model checking later.

**Definition 2.7.** *Let  $G_i = (V_i, E_{0,i}, \dots, E_{n-1,i}, A_{0,i}, \dots, A_{m-1,i})$  for  $i \in 0, \dots, T - 1$  for some  $T \in \mathbb{N}$  be static graphs. We call a List  $\mathcal{G} = (G_0, \dots, G_{T-1})$  a discrete-time dynamic graph.*

**Example 2.8.** *The DTDG  $\mathcal{G}' = (\mathcal{G}^0, \mathcal{G}^2, \mathcal{G}^{10})$ , where  $\mathcal{G}^i = \text{snapshot}(\mathcal{G}, i)$  are snapshots of the CTDG from Example 2.6, is a DTDG consisting of 3 static graphs. We can obtain DTDGs from CTDGs in different ways, for example by taking snapshots at each event's timestamp, taking them at a regular intervals, or arbitrarily. Which approach is used should depend on the use case.*

### 2.3 Metric First Order Temporal Logic (MFOTL)

Metric First Order Temporal Logic (MFOTL) was first introduced in Basin et al. [3]. We will follow the definitions given in that paper. It uses standard LTL [2] as a basis, and extends it in multiple ways.

**Definition 2.9.** *Let AP be a set of atomic propositions. The following are LTL formulas:*

- (i) true,

- (ii)  $a$  where  $a \in \text{AP}$ ,
- (iii)  $\varphi_0 \wedge \varphi_1$  where  $\varphi_0$  and  $\varphi_1$  are LTL formulas,
- (iv)  $\neg\varphi$  where  $\varphi$  is a LTL formula,
- (v)  $\bigcirc\varphi$  where  $\varphi$  is a LTL formula,
- (vi)  $\varphi_0 \cup \varphi_1$  where  $\varphi_0$  and  $\varphi_1$  are LTL formulas.

The semantics of LTL are commonly defined over *infinite words*.

**Definition 2.10.** Let  $\Sigma$  be a finite set of symbols called an *alphabet*.  $\Sigma^\omega$  is the set of all infinite sequences of elements of  $\Sigma$ . We also call such  $w \in \Sigma^\omega$  *infinite words*.

Using these infinite words, we can define semantics for LTL.

**Definition 2.11.** Given a set of atomic propositions  $\text{AP}$ , let  $\sigma \in (2^{\text{AP}})^\omega$ , with  $\sigma = X_0X_1X_2\dots$ , be infinite words over the alphabet  $2^{\text{AP}}$ . We write  $\sigma[i\dots] = X_iX_{i+1}X_{i+2}\dots$  to mean the word  $\sigma$  without the sets of atomic propositions before  $i$ . Given a LTL formula  $\varphi$ , we write  $\sigma \models \varphi$  to mean that  $\sigma$  fulfills  $\varphi$ , if:

- (i)  $\sigma \models \text{true}$
- (ii)  $\sigma \models a$  if and only if  $a \in X_0$ , with  $a \in \text{AP}$
- (iii)  $\sigma \models \varphi_1 \wedge \varphi_2$  if and only if  $\sigma \models \varphi_1$  and  $\sigma \models \varphi_2$
- (iv)  $\sigma \models \neg\varphi$  if and only if  $\sigma \not\models \varphi$
- (v)  $\sigma \models \bigcirc\varphi$  if and only if  $\sigma[1\dots] \models \varphi$
- (vi)  $\sigma \models \varphi_0 \cup \varphi_1$  if and only if  $\exists j \geq 0 : \sigma[j\dots] \models \varphi_1$  and  $\sigma[i] \models \varphi_0$  for all  $0 \leq i < j$

While LTL formulas are defined based on a set of atomic propositions, MFOTL formulas are defined over a given signature.

**Definition 2.12.** Let  $C$  be a finite set of symbols, let  $\mathcal{R} = \{R_0, \dots, R_i\}$  for some  $i \in \mathbb{N}$  be a disjunctive set of symbols, and let  $\iota : \mathcal{R} \rightarrow \mathbb{N}$  be a function. We call  $S = (C, \mathcal{R}, \iota)$  a *Signature for MFOTL formulas*. We call  $C$  the *constant symbols or constants*,  $\mathcal{R}$  are the *predicate symbols*, and  $\iota(R)$  for  $R \in \mathcal{R}$  gives the *arity of each predicate*.

Using such a signature, we can now define the syntax of MFOTL formulas.

**Definition 2.13.** Let  $S = (C, \mathcal{R}, \iota)$  be a MFOTL-signature, and  $V$  be a countably infinite set of variable symbols. We always assume that with  $C \cap V = \emptyset$  and  $R \notin V$  for all  $R \in \mathcal{R}$ . The following are MFOTL formulas:

- (i) **true**,
- (ii)  $t \approx t'$  is a formula, where  $t, t' \in C \cup V$ ,
- (iii)  $R(t_0, \dots, t_{\iota(R)-1})$  is a formula, where  $t_0, \dots, t_{\iota(R)-1} \in C \cup V, R \in \mathcal{R}$ ,
- (iv)  $(\neg\phi)$ ,  $(\phi \vee \psi)$  and  $\exists x.\phi$  are all formulas, where  $x \in V$ , and  $\phi, \psi$  are MFOTL formulas over  $S$ , and

( $v$ )  $\phi U_I \psi$ ,  $\phi S_I \psi$ ,  $\bigcirc_I \phi$  and  $\bullet_I \phi$  where  $I \subseteq \mathbb{N} \times (\mathbb{N} \cup \{\infty\})$ , and  $\phi, \psi$  are MFOTL formulas over  $S$ .

At the end of this subsection, we also look at a bigger example (Example 2.19) of MFOTL formulas, and explain them in the context of the semantics of MFOTL when we define it.

For our use in this thesis, we restrict these formulas in certain ways. Firstly, we do not allow *free variables* as defined in the following. The reasoning for this restriction will be explained later when it becomes relevant.

**Definition 2.14.** *Given a MFOTL formula  $\phi$  over  $(C, \mathcal{R}, \iota)$ , we call a variable  $v \in V$  a free variable, if it is not contained within a subformula  $\psi$ , such that  $\exists v. \psi$  is a subformula of  $\phi$ . If  $v$  is contained in such a subformula, we call  $v$  a bound variable, and say  $v$  is bound to  $\exists v$ .*

To define the semantics of MFOTL formulas we further need *temporal structures* over the formulas signature, as well as *valuations*. We start with the definition of static structures.

**Definition 2.15.** *Let  $S = (C, \mathcal{R}, \iota)$  be a MFOTL-Signature. We define a static structure  $D = (U, d)$  over  $S$  as consisting of both a finite set  $U$  called a domain, as well as a function  $d$  with  $d(c) \in U$  for  $c \in C \cup V$ , and  $d(R) \in U^{(R)}$  for  $R \in \mathcal{R}$ . The domain  $U$  is a non empty set. We call  $d$  the interpretation-function.*

Effectively, the interpretation-function  $d$  assigns each symbol within the signature a meaning within the universe  $U$ .

We can now define temporal structures using static structures.

**Definition 2.16.** *Let  $S = (C, \mathcal{R}, \iota)$  be a MFOTL-Signature, let  $V$  be an infinite set of variable symbols, and let  $D_i = (U_i, d_i)$  for  $i \in \mathbb{N}$  be static structures over the signature  $S$ . We call  $(\overline{D}, \overline{\tau})$  a temporal structure over  $S$ , where  $\overline{D} = (D_0, D_1, \dots)$  is an infinite sequence of static structures over  $S$ ,  $\overline{\tau} = (\tau_0, \tau_1, \dots)$  is an infinite sequence of increasing positive timestamps, and the following criteria apply:*

1.  $\overline{\tau}$  is monotonically increasing, and for every  $\tau \in \mathbb{N}$  there exists an  $i \in \mathbb{N}$  such that  $\tau_i > \tau$ .
2.  $U_i = U_{i+1}$  for  $i \in \mathbb{N}$ .
3.  $d_i(c) = d_{i+1}(c)$  for  $i \in \mathbb{N}, c \in C$ .

Since all static structures  $D_i$  have the same domains  $U_i$ , we refer to them within the temporal structure as just  $U$ .

While we restrict formulas to have no free variables, it is still important to define valuations of MFOTL formulas to define their semantics.

**Definition 2.17.** *Let  $V$  be set of variables, and  $(\overline{D}, \overline{\tau})$  be a temporal structure with the domain  $U$  and constant symbols  $C$ . We call a mapping  $\text{val} : V \rightarrow U$  a valuation. Given a variable vector  $\bar{x} = (x_0, \dots, x_{n-1})$  and  $\bar{u} = (u_0, \dots, u_{n-1}) \in U^n$ , we write  $\text{val}[\bar{x} \mapsto \bar{u}]$  to mean the mapping, where  $x_i$  maps to  $u_i$  for  $i \in \{0, \dots, n-1\}$ . We also*

write  $\Phi[\bar{x} \rightarrow \bar{c}]$  with  $\bar{c} = (c_0, \dots, c_{n-1}) \in C^n$  to mean the formula  $\Phi$ , where the variable  $x_i$  have been replaced with the constant  $c_i$  for all  $i \in \{0, \dots, n-1\}$ . Furthermore, we abuse notation and apply the valuation to constant symbols  $c \in C$ , with the meaning  $\text{val}(c) = d_i(c)$  for all  $i \in \mathbb{N}$ .

Given these definitions, we can now define the semantics of MFOTL.

**Definition 2.18.** Let  $S = (C, R, \iota)$  be a MFOTL-signature, let  $(\bar{D}, \bar{\tau})$  be a temporal structure with domain  $U$ , let  $\text{val}$  be a valuation, let  $\varphi$  be a MFOTL formula over  $S$ , and  $i \in \mathbb{N}$ . We define  $(\bar{D}, \bar{\tau}, \text{val}, i) \models \varphi$  (read as  $(\bar{D}, \bar{\tau}, v, i)$  models  $\varphi$ ) inductively:

1.  $(\bar{D}, \bar{\tau}, \text{val}, i) \models \text{true}$  is always true
2.  $(\bar{D}, \bar{\tau}, \text{val}, i) \models t \approx t'$  if and only if  $\text{val}(t) = \text{val}(t')$
3.  $(\bar{D}, \bar{\tau}, \text{val}, i) \models R(t_0, \dots, t_{\iota(R)-1})$  if and only if  $(\text{val}(t_0), \dots, \text{val}(t_{\iota(R)-1})) \in d_i(R)$
4.  $(\bar{D}, \bar{\tau}, \text{val}, i) \models (\neg\varphi)$  if and only if  $(\bar{D}, \bar{\tau}, \text{val}, i) \not\models \varphi$
5.  $(\bar{D}, \bar{\tau}, \text{val}, i) \models (\varphi \vee \psi)$  if and only if  $(\bar{D}, \bar{\tau}, \text{val}, i) \models \varphi$  or  $(\bar{D}, \bar{\tau}, \text{val}, i) \models \psi$
6.  $(\bar{D}, \bar{\tau}, \text{val}, i) \models \exists x.\varphi$  if and only if  $(\bar{D}, \bar{\tau}, \text{val}[x \mapsto u], i) \models \varphi$  for some  $u \in U$
7.  $(\bar{D}, \bar{\tau}, \text{val}, i) \models \bullet_I\varphi$  if and only if  $i > 0$  and  $\tau_i - \tau_{i-1} \in I$ , and  $(\bar{D}, \bar{\tau}, \text{val}, i-1) \models \varphi$
8.  $(\bar{D}, \bar{\tau}, \text{val}, i) \models \circ_I\varphi$  if and only if  $\tau_{i+1} - \tau_i \in I$ , and  $(\bar{D}, \bar{\tau}, \text{val}, i+1) \models \varphi$
9.  $(\bar{D}, \bar{\tau}, \text{val}, i) \models (\varphi \mathbf{S}_I\psi)$  if and only if for some  $j \leq i$ ,  $\tau_i - \tau_j \in I$ ,  $(\bar{D}, \bar{\tau}, v, j) \models \psi$  and  $(\bar{D}, \bar{\tau}, \text{val}, k) \models \varphi$  for all  $k \in \mathbb{N}$  with  $j < k \leq i$
10.  $(\bar{D}, \bar{\tau}, \text{val}, i) \models (\varphi \mathbf{U}_I\psi)$  if and only if for some  $j \geq i$ ,  $\tau_j - \tau_i \in I$ ,  $(\bar{D}, \bar{\tau}, \text{val}, j) \models \psi$  and  $(\bar{D}, \bar{\tau}, \text{val}, k) \models \varphi$  for all  $k \in \mathbb{N}$  with  $i \leq k < j$

We also define other standard operators as abbreviations of the above defined logical symbols:  $\text{false} \equiv \neg\text{true}$ ,  $\phi \wedge \psi \equiv \neg((\neg\phi) \vee (\neg\psi))$ ,  $\forall x.\phi \equiv \neg\exists x.\neg\phi$ ,  $\mathbf{F}_I\phi \equiv \text{true}\mathbf{U}_I\phi$  ("finally"/"eventually"),  $\mathbf{G}_I\phi \equiv \neg(\mathbf{F}_I(\neg\phi))$  ("generally"/"always"),  $\mathbf{O}_I\phi \equiv \text{true}\mathbf{S}_I\phi$  ("once"), and  $\mathbf{H}_I\phi \equiv \neg(\mathbf{O}_I(\neg\phi))$  ("historically").

Now we can give an example of MFOTL formulas, with some of these abbreviations:

**Example 2.19.** As an example we look at a case where we use MFOTL to model the relationships between social media accounts. There we could use the following signature  $S = (C, \mathcal{R}, \iota)$ , with the relation symbols:  $\mathcal{R} = \{\text{follows, blocked, private, verified}\}$  and the arities  $\iota(\text{follows}) = \iota(\text{blocked}) = 2$  and  $\iota(\text{private}) = \iota(\text{verified}) = 1$ . For this example we use three constants for users, being  $C = \{u_0, u_1, u_2\}$ .

We can now use MFOTL formulas to make statements about the users. For example, the formula  $\Phi_0 = \mathbf{G}_{[0, \infty]} \neg \exists x.\text{blocked}(x, x)$  is satisfied if there never exists a user in the future who has blocked themselves. The formula  $\Phi_1 = \text{follows}(u_0, u_1) \mathbf{U}_{[0, 5]} \text{blocked}(u_1, u_0)$  is satisfied, if within the interval of  $[0, 5]$   $u_0$  follows  $u_1$  until  $u_1$  blocks  $u_0$ . The formula  $\Phi_2 = \mathbf{F}_{[0, \infty]} \mathbf{G}_{[0, \infty]} \text{private}(u_2)$  is satisfied if user  $u_2$  at some point sets their account to private, and never switches off that again.

As it is relevant for later discussion, we also define a measurement for how many nested  $\exists$  quantifiers are contained within a MFOTL formula.

**Definition 2.20.** Let  $\Phi$  be a MFOTL formula. We define the  $\exists$ -depth of  $\Phi$   $d(\Phi)$  recursively as follows:

$d(\Phi) = 0$  if  $\Phi$  is an atomic formula. If  $\Phi$  has the form  $\exists x.\Phi'$ , then  $d(\Phi) = d(\Phi') + 1$ . In all other cases,  $d(\Phi) = \max(d(\Phi'))$  for  $\Phi'$  is a subformula of  $\Phi$  with  $\Phi \neq \Phi'$ .

## 2.4 Representing Dynamic Graphs in MFOTL

With the goal of using MFOTL formulas to describe properties of DTDGs, we need to express them in the form of a temporal structure. We also require a fitting signature for both the structure and the formulas. For that signature we introduce signatures which are implied by a DTDG.

**Definition 2.21.** Let  $\mathcal{G} = (G_0, \dots, G_{T-1})$  be a DTDG with  $T \in \mathbb{N}$ , where  $G_i = (V_i, E_{0,i}, \dots, E_{n-1,i}, A_{0,i}, \dots, A_{m-1,i})$  with finite  $V_i, E_{j,i}$  and  $A_{k,i}$  for  $i, j, k, n, m \in \mathbb{N}$ . We call  $S_{\mathcal{G}} = (C_{\mathcal{G}}, \mathcal{R}_{\mathcal{G}}, \iota_{\mathcal{G}})$  the signature implied by  $\mathcal{G}$ , where  $C_{\mathcal{G}}$  and  $\mathcal{R}_{\mathcal{G}}$  are the following list of symbols, and  $\iota_{\mathcal{G}}$  is defined as follows:

- $C_{\mathcal{G}} = \bigcup_{i \in \{0, \dots, T-1\}} V_i$
- $\mathcal{R}_{\mathcal{G}} = \{R_{E_i} | i \in \{0, \dots, n-1\}\} \cup \{R_{A_j} | j \in \{0, \dots, m-1\}\} \cup \{L\}$ , and
- $\iota_{\mathcal{G}}(R_{E_i}) = 2$  for  $i \in \{0, \dots, n-1\}$  and  $\iota_{\mathcal{G}}(R_{A_j}) = 1$  for  $j \in \{0, \dots, m-1\}$  and  $\iota_{\mathcal{G}}(L) = 1$ .

The restrictions here on  $V_i, E_{j,i}$  and  $A_{k,i}$  being finite result from MFOTL requiring both the set of constant and relation symbols to be finite. We have relation symbols for each edge and attribute, as well as the new relation symbol  $L$ . MFOTL requires the domains of each static structure within one temporal structure to be the same. However, DTDGs with node additions/deletions don't necessarily have the same set of vertices for each of their static graphs. The unary relation  $L$  is thus used to indicate which nodes currently exist within the graph at a certain point in time.

**Definition 2.22.** Let  $\mathcal{G} = (G_0, \dots, G_{T-1})$  be a DTDG, where  $G_i = (V_i, E_{0,i}, \dots, E_{n-1,i}, A_{0,i}, \dots, A_{m-1,i})$ , and let  $S_{\mathcal{G}} = (C_{\mathcal{G}}, \mathcal{R}_{\mathcal{G}}, \iota_{\mathcal{G}})$  be the signature implied by  $\mathcal{G}$ . We call  $(\overline{D}_{\mathcal{G}}, \overline{\tau}_{\mathcal{G}})$  the temporal structure implied by  $\mathcal{G}$ , with  $\overline{D}_{\mathcal{G}} = (D_0, D_1, \dots)$ ,  $\overline{\tau} = (0, 1, \dots)$  and  $D_i = (U, d_i)$  for  $i \in \mathbb{N}$ , where:

- $U_i = \bigcup_{j \in \{0, \dots, T-1\}} V_j$  for all  $i \in \mathbb{N}$ ,
- $d_i(c) = c$  for all  $c \in C_{\mathcal{G}}, i \in \mathbb{N}$ ,
- $d_i(R_{A_k}) = A_{k,i}$  for all  $k \in \{0, \dots, m-1\}, i \in \{0, \dots, T-1\}$ ,
- $d_i(R_{E_j}) = E_{j,i}$  for all  $j \in \{0, \dots, n-1\}, i \in \{0, \dots, T-1\}$ ,
- $d_i(L) = V_i, i \in \{0, \dots, T-1\}$ ,
- $d_i(x) = d_{i-1}(x)$  for all  $i > T-1, x \in \mathcal{R}_{\mathcal{G}}$ .

We note that here we define  $U_i$  the same as  $C_{\mathcal{G}}$ . Thus for the constants of the signature,  $d_i$  is the identity function. We note however, that within the signature these elements are understood as constant symbols, while within  $U$  they are understood as elements of the domain. Furthermore  $d$  is also well defined for the symbols  $R_{A_k}$  and  $R_{E_j}$ , since  $U$  is a superset of each  $V_i$ .

We note that these temporal structures consist of an infinite sequence of static structures, while the DTDG only consists of a finite amount of static graphs. This is due to the original definition of MFOTL requiring these kind of structures. For our formulas, we will however only look at finite intervals.

For simplicity, we define the following:

**Definition 2.23.** Let  $\mathcal{G} = (G_0, \dots, G_{T-1})$  be a DTDG, and let  $\Phi$  be a MFOTL formula without free variables over the signature implied by  $\mathcal{G}$ . Let  $(\bar{D}, \bar{\tau})$  be the temporal structure implied by  $\mathcal{G}$ .

We define  $(\mathcal{G}, i) \models \Phi$  to be true if and only if  $(\bar{D}, \bar{\tau}, \text{val}, i) \models \Phi$  for all valuations  $\text{val}$ . Furthermore, whenever we write  $(\mathcal{G}, i) \models \Phi$ , we assume that  $\Phi$  is a valid formula over the signature implied by  $\mathcal{G}$ . We also note that because there are no free variables,  $\text{val}$  is effectively irrelevant.

Instead of  $(\mathcal{G}, i) \models \Phi$  we also say  $\mathcal{G}$  is model of  $\Phi$ .

## 2.5 Propositional Logic and Boolean SAT

In the next section we reduce the extended dynamic graph model checking problem to the satisfiability problem for propositional Logic (Boolean SAT).

**Definition 2.24.** Let  $P$  be a countably finite set of propositions. We define Boolean formulas inductively as follows:

- $p \in P$  is a formula.
- $\text{true}$  is a formula.
- $\neg\Psi$ , where  $\Psi$  is a formula, is a formula.
- $\Psi_0 \vee \Psi_1$ , where  $\Psi_0$  and  $\Psi_1$  are formulas, is a formula.

We also use the following abbreviations:  $\text{false} = \neg\text{true}$ ,  $\phi \wedge \psi = \neg(\neg\phi \vee \neg\psi)$ ,  $\phi \rightarrow \psi = \neg\phi \vee \psi$ ,  $\phi \leftrightarrow \psi = (\phi \wedge \psi) \vee (\neg\phi \wedge \neg\psi)$ ,  $\bigwedge_{i=0}^n \phi_i = \phi_0 \wedge \phi_1 \wedge \dots \wedge \phi_n$ , and  $\bigvee_{i=0}^n \phi_i = \phi_0 \vee \phi_1 \vee \dots \vee \phi_n$ .

We can now define interpretations over a set of propositions, to afterwards define valuations of Boolean formulas.

**Definition 2.25.** Let  $P$  be a countable finite set of propositions. We call a function  $\mathcal{I} : P \rightarrow \{\top, \perp\}$  an interpretation function, or just an interpretation.

Given an interpretation, we can now define valuations of Boolean formulas.

**Definition 2.26.** Let  $P$  be a countable finite set of propositions, let  $\mathcal{I}$  be an interpretation of  $P$ , and let  $\Phi$  be a Boolean formula. We call  $\text{val}_{\mathcal{I}}(\Phi)$  the valuation of  $\Phi$  given  $\mathcal{I}$ , with it being defined as follows:

- $\text{val}_{\mathcal{I}}(p) = \mathcal{I}(p)$  for all  $p \in P$
- $\text{val}_{\mathcal{I}}(\text{true}) = \top$
- $\text{val}_{\mathcal{I}}(\neg\Phi) = \top$  iff  $\text{val}_{\mathcal{I}}(\Phi) = \perp$
- $\text{val}_{\mathcal{I}}(\Phi_0 \vee \Phi_1) = \top$  iff  $\text{val}_{\mathcal{I}}(\Phi_0) = \top$  or  $\text{val}_{\mathcal{I}}(\Phi_1) = \top$

When  $\text{val}_{\mathcal{I}}(\Phi) = \top$ , we also say  $\mathcal{I}$  satisfies  $\Phi$ .

We can now define the decision problem, that we want to reduce the decision problems from the next section to.

**Definition 2.27.** We define the Boolean SAT problem as follows:

**Given:** A Boolean formula  $\Psi$ .

**Question:** Does there exist a interpretation  $\mathcal{I}$  with  $\text{val}_{\mathcal{I}}(\Psi) = \top$ ?

For decision problems, we also call an instance of what was given an *instance of the problem*. So for example, a Boolean formula  $\Psi$  would be an instance of the Boolean SAT problem. An instance for which the answer to the question is yes, we call a *positive instance*, while any other instances we call a *negative instance*.

We also define the size of a Boolean formula as the number of atomic formulas it contains. We don't count how many unique ones there are here, just the number in total:

**Definition 2.28.** Let  $\Psi$  and  $\Psi'$  be Boolean formulas, and let  $P$  be its set of propositions.

We call  $|\Psi|$  the size of  $\Psi$ , with  $|\Psi|$  being defined as follows:

$|\text{true}| = 1$ ,  $|p| = 1$  for all  $p \in P$ ,  $|\neg\Psi| = |\Psi|$ , and  $|\Psi \vee \Psi'| = |\Psi| + |\Psi'|$ .

### 3 Model Checking for Dynamic Graphs

The goal of this thesis is to explore ways of describing and checking properties of dynamic graphs. We use MFOTL to describe those properties, and model the graphs using temporal structures as described in Section 2.4.

However, we need to restrict the MFOTL formulas we use in a few ways. We will define these formulas as a subset of all MFOTL formulas.

**Definition 3.1.** *We call a MFOTL formula  $\Phi$  over the signature  $(C_{\mathcal{G}}, \mathcal{R}_{\mathcal{G}}, \iota_{\mathcal{G}})$  implied by a DTDG  $\mathcal{G}'$  a DG-MFOTL formula, if it fulfills the following criteria:*

1.  $\Phi$  has no free variables.
2. For each of  $\Phi$ 's subformulas of the form  $\exists x.\Phi'$ ,  $\Phi'$  has the form  $L(x) \wedge \Phi''$ , where  $\Phi''$  is another subformula. Here  $L(x)$  is the relation defined in Section 2.4.
3. All intervals  $I$  in  $\Phi$  are finite, meaning for  $I = [t_0, t_1]$ ,  $t_0, t_1 \in \mathbb{N}$ .

These restrictions are needed to ensure that all the reductions later are correct. The first two are needed, because for two of the decision problems they would require more restrictions on the problem instance. The last restriction is needed, because we will unroll the  $U_I$  and  $S_I$  operators within the reduction, and they would not necessarily be finite otherwise.

Using these restrictions on MFOTL, we define the basic model checking problem for dynamic graphs as follows.

**Definition 3.2.** *We call the following problem the Basic Dynamic Graph Model Checking Problem (BDGMC):*

**Given:** A DTDG  $\mathcal{G}$ , and a DG-MFOTL formula  $\Phi$ .

**Question:** Is  $(\mathcal{G}, 0) \models \Phi$  true?

We also consider another decision problem, motivated by the fact that with our definition of DTDGs, consisting of a finite amount of static graphs, we might be interested if a graph can fulfill a property in the near future.

**Example 3.3.** *Let  $\mathcal{G} = (G_0, \dots, G_4)$  be a DTDG consisting of 5 static graphs. Let  $\Phi$  be a DG-MFOTL formula.  $\mathcal{G} \models \Phi$  may not be true, but another DTDG with more timestamps might satisfy  $\Phi$ . We can thus ask if there exists a graph  $G' = (G_0, \dots, G_4, G_5, G_6)$  which might be a model of  $\Phi$ . Here it then makes sense that we limit the added static graphs  $G_5$  and  $G_6$ , as for most use cases a complete change of the graph would not be a reasonable assumption. Therefore, we can ask if a graph  $G'$  exists which fulfills  $\Phi$ , and where there exists events  $o_4, o_5$  such that  $o_4(G_4) = G_5$  and  $o_5(G_5) = G_6$ .*

**Definition 3.4.** *We call the following problem the Extended Dynamic Graph Model checking Problem (EDGMC):*

**Given:** A DTDG  $\mathcal{G} = (G_0, \dots, G_{T-1})$ , and a DG-MFOTL formula  $\Phi$ , a set of allowed event types  $\mathcal{E} \subseteq \text{Types}$ , and a timestamp  $T' \in \mathbb{N}$  with  $T \leq T'$ .

**Question:** Does there exist a DTDG  $\mathcal{G}' = (G_0, \dots, G_{T-1}, \dots, G_{T'-1})$  such that  $(\mathcal{G}', 0) \models \Phi$ , where for all  $G_i \in \{G_{T-1}, \dots, G_{T'-2}\}$  there exists an event  $o_i$  with  $\text{type}(o_i) \in \mathcal{E}$ , such that  $o_i(G_i) = G_{i+1}$ ?

We note that the DTDG  $\mathcal{G}'$  may imply a different signature, specifically one with a superset of constants, compared to  $\mathcal{G}$ . However, the formula  $\Phi$  is obviously still a valid formula over that signature.

Furthermore, while  $timestamp(o)$  of an event  $o$  is relevant for the definition CTDGs, we can ignore them here, as they are not relevant when applying an event to a static graph as part of the definition of  $\mathcal{G}'$ .

As a third problem, we define a bounded satisfiability problem for DTDGs over DG-MFOTL formulas.

**Definition 3.5.** *We call the following problem the Dynamic Graph Bounded SAT Problem (DG-BSAT):*

**Given:** *A DG-MFOTL formula  $\Phi$ , a timestamp  $T' \in \mathbb{N}$ , a number of nodes  $y \in \mathbb{N}$ , a number of edge relations  $n \in \mathbb{N}$ , and a number of attribute relations  $m \in \mathbb{N}$ .*

**Question:** *Do there exist a DTDG  $\mathcal{G} = (G_0, \dots, G_{T'-1})$  and a list of events  $O = (o_0, \dots, o_{T'-2})$  with  $(\mathcal{G}, O) \models \Phi$ , where  $G_0 = (V_0, E_{0,0}, \dots, E_{n-1,0}, A_{0,0}, \dots, A_{m-1,0})$  with  $|V_0| = y$ , and for every  $i \in \{0, \dots, T' - 2\}$ ,  $o_i(G_i) = G_{i+1}$ ?*

## 4 Reducing Dynamic Graph problems to Boolean SAT

In this section we reduce the problems from section 3 to Boolean SAT. We will achieve this by proving lemmas about Boolean formulas, over a partially shared set of propositions. Using these formulas, we can then build reductions from BDGMC, EDGMC and DG-BSAT to Boolean SAT, where different conjunctions of the Boolean formulas form the Boolean SAT instances.

We first define the set of propositions that is shared by all formulas, based on a DTDG  $\mathcal{G}$  and a timestamp  $T'$ . For each of the decision problems, we either have this graph be given, or the necessary parameters, being the amount of nodes, attribute relations and edge relations.

**Definition 4.1.** *Let  $\mathcal{G} = (G_0, \dots, G_{T-1})$  be a DTDG, with  $G_i = (V_i, E_{0,1}, \dots, E_{n-1,i}, A_{0,i}, \dots, A_{m-1,i})$ , let  $T' \in \mathbb{N}$  be a timestamp with  $T' \geq T$ , and let  $V_{\mathcal{G}}^{T'} = (\bigcup_{i=0}^{T'-1} V_i) \cup X$  with  $X = \{x_i \mid i \in \mathbb{N} \text{ and } T \leq i < T'\}$  be a set of nodes, where  $X$  here is a set of nodes that are not part of any  $V_i$ . We call the set  $P_{\mathcal{G}}^{T'}$  the propositions implied by  $\mathcal{G}$  and  $T'$ . It includes the following propositions:*

$$\begin{aligned} l_i(v) & \text{ for all } i \in \{0, \dots, T' - 1\}, v \in V_{\mathcal{G}}^{T'} \\ a_{A_{j,i}}(v) & \text{ for all } i \in \{0, \dots, T' - 1\}, v \in V_{\mathcal{G}}^{T'}, j \in \{0, \dots, m - 1\} \\ e_{E_{j,i}}(v_0, v_1) & \text{ for all } i \in \{0, \dots, T' - 1\}, (v_0, v_1) \in V_{\mathcal{G}}^{T'} \times V_{\mathcal{G}}^{T'}, j \in \{0, \dots, n - 1\} \end{aligned}$$

We now define an interpretation  $\mathcal{I}_{\mathcal{G}'}$ , for  $P_{\mathcal{G}'}^{T'}$ , implied by a DTDG  $\mathcal{G}'$ , as well as a DTDG  $\mathcal{G}^I$  implied by an interpretation  $\mathcal{I}$ . We will use those to show that if a graph with a certain property exists, there also exists an interpretation that satisfies the formula and the other way around. It is not sufficient that we prove that if a graph exists, the Boolean formula is satisfiable, as the conjunction of formulas might not be. Conversely, proving that if a formula is satisfiable, there then exists a graph with a certain property, does not mean that if their conjunction is satisfiable, a graph exists which fulfills both properties.

**Definition 4.2.** *Let  $\mathcal{G} = (G_0, \dots, G_{T-1})$  be a DTDG, let  $T' \geq T$  be a timestamp, and let  $P_{\mathcal{G}}^{T'}$  be the propositions implied by  $\mathcal{G}$  and  $T'$ . Furthermore, let  $\mathcal{G}' = (G'_0, \dots, G'_{T'-1})$  be a DTDG, with  $G'_i = (V'_i, E'_{0,1}, \dots, E'_{n-1,i}, A'_{0,i}, \dots, A'_{m-1,i})$ . We call  $\mathcal{I}_{\mathcal{G}'}$  the interpretation implied by  $\mathcal{G}'$  of  $P_{\mathcal{G}}^{T'}$ , with  $\mathcal{I}_{\mathcal{G}'}$  being defined as follows:*

$$\begin{aligned} \mathcal{I}_{\mathcal{G}'}(l_i(v)) &= \top \text{ if and only if } v \in V'_i \\ \text{for all } j \in \{0, \dots, m - 1\} : \mathcal{I}_{\mathcal{G}'}(a_{A_{j,i}}(v)) &= \top \text{ if and only if } v \in A'_{j,i} \\ \text{for all } j \in \{0, \dots, n - 1\} : \mathcal{I}_{\mathcal{G}'}(e_{E_{j,i}}(v_0, v_1)) &= \top \text{ if and only if } (v_0, v_1) \in E'_{j,i} \end{aligned}$$

We also call  $\text{val}_{\mathcal{G}'} = \text{val}_{\mathcal{I}_{\mathcal{G}'}}$  the valuation implied by  $\mathcal{G}'$ .

We note that in the above definition the set of propositions and the interpretation, are based on different DTDGs.

We need to take more caution with using the DTDG implied by an interpretation, as not every interpretation corresponds to a valid graph. For example, due to the relation  $L$  used in the DG-MFOTL formulas, and the equivalent  $l_i(v)$  propositions, we could have an interpretation where no  $l_i(v)$  is true, however we can have  $a_{A_{k,i}}(v)$  be true for

some  $k, i$ . Thus, the DTDG defined in the following definition is not always a well-defined DTDG for every possible interpretation  $\mathcal{I}$ . We prove that every satisfying interpretation for the formulas used in the reductions will always have a well defined implied DTDG.

**Definition 4.3.** Let  $\mathcal{G} = (G_0, \dots, G_{T-1})$  be a DTDG, with  $G_i = (V_i, E_{0,1}, \dots, E_{n-1,i}, A_{0,i}, \dots, A_{m-1,i})$ , let  $T' \in \mathbb{N}$  be a timestamp with  $T' \geq T$ , let  $P_{\mathcal{G}}^{T'}$  be the set of propositions implied by  $\mathcal{G}$  and  $T'$ , and let  $\mathcal{I}$  be an interpretation of  $P_{\mathcal{G}}^{T'}$ . We then define the following sets:

$$\begin{aligned} V_i^{\mathcal{I}} &= \{v \mid \mathcal{I}(l_i(v)) = \top\} && \text{for all } i \in \{0, \dots, T' - 1\} \\ A_{j,i}^{\mathcal{I}} &= \{v \mid \mathcal{I}(a_{A_{j,i}}(v)) = \top\} && \text{for all } j \in \{0, \dots, m - 1\}, i \in \{0, \dots, T' - 1\} \\ E_{j,i}^{\mathcal{I}} &= \{(v_0, v_1) \mid \mathcal{I}(e_{E_{j,i}}(v_0, v_1)) = \top\} && \text{for all } j \in \{0, \dots, n - 1\}, i \in \{0, \dots, T' - 1\} \end{aligned}$$

We can now define  $\mathcal{G}^{\mathcal{I}}$  as follows:  $\mathcal{G}^{\mathcal{I}} = (G_0^{\mathcal{I}}, \dots, G_{T'-1}^{\mathcal{I}})$  with  $G_i^{\mathcal{I}} = (V_i^{\mathcal{I}}, E_{0,1}^{\mathcal{I}}, \dots, E_{n-1,i}^{\mathcal{I}}, A_{0,i}^{\mathcal{I}}, \dots, A_{m-1,i}^{\mathcal{I}})$ . We call  $\mathcal{G}^{\mathcal{I}}$  the DTDG implied by  $\mathcal{I}$ , if and only if it fulfills the following two properties for each  $G_i^{\mathcal{I}}$ :

$$\begin{aligned} A_{j,i}^{\mathcal{I}} &\subseteq V_i^{\mathcal{I}} && \text{for all } j \in \{0, \dots, m - 1\} \\ E_{j,i}^{\mathcal{I}} &\subseteq V_i^{\mathcal{I}} \times V_i^{\mathcal{I}} && \text{for all } j \in \{0, \dots, n - 1\} \end{aligned}$$

Note that if these properties are not fulfilled, the given  $G_i^{\mathcal{I}}$  is not a static graph according to Definition 2.1, and thus  $\mathcal{G}^{\mathcal{I}}$  is not a DTDG according to Definition 2.7 either.

For the final formulas used in the reductions, we can thus prove that  $\mathcal{G}^{\mathcal{I}}$  is a DTDG, by proving that all  $G_i^{\mathcal{I}}$  are static graphs. We achieve this by proving that different formulas ensure this for different  $i \in \{0, \dots, T' - 1\}$ .

We also note that by construction, given a DTDG  $\mathcal{G}'$ ,  $\mathcal{G}^{\mathcal{I}_{\mathcal{G}'}} = \mathcal{G}'$ .

In the following subsections we will define multiple formulas, which in conjunction form the Boolean formula we reduce to for the different problems. Most of these formulas will be part of multiple reductions.

The first one ensures that given DTDGs  $\mathcal{G}$  and  $\mathcal{G}'$ , the interpretation  $\mathcal{I}_{\mathcal{G}'}$  implied by  $\mathcal{G}'$  satisfies the formula if and only if for the first  $T$  timestamps the static graphs of the DTDGs are identical. We also define a different formula for DG-BSAT, which ensures that  $\mathcal{I}_{\mathcal{G}'}$  is satisfied if and only if  $G_0$  is a well defined static graph.

Secondly, we define a formula which ensures that the formula is satisfied by  $\mathcal{I}_{\mathcal{G}'}$  if and only if for all  $i \in \{T - 1, \dots, T' - 2\}$  there exists an event  $o_i$  such that  $o_i(G'_i) = G'_{i+1}$ .

Thirdly, we define a formula that is satisfied by  $\mathcal{I}_{\mathcal{G}'}$  if and only if  $\mathcal{G}' \models \Phi$  for some DG-MFOTL formula  $\Phi$ .

## 4.1 Boolean Formula for Graph Equivalence

The first major formula we construct ensures that the first  $T$  layers of a given DTDG  $\mathcal{G}$  will be equal to the first  $T$  layers of  $\mathcal{G}^{\mathcal{I}}$  for all satisfying interpretations  $\mathcal{I}$ .

**Definition 4.4.** Let  $\mathcal{G} = (G_0, \dots, G_{T-1})$  be a DTDG, with  $G_i = (V_i, E_{0,1}, \dots, E_{n-1,i}, A_{0,i}, \dots, A_{m-1,i})$ , let  $T' \in \mathbb{N}$  be some timestamp with  $T' \geq T$ , and let  $V_{\mathcal{G}}^{T'}$  be defined as in Definition 4.1. Given those we define the following formulas:

$$\Psi_{\mathcal{G}}^T = \Psi_{\mathcal{G},V}^T \wedge \Psi_{\mathcal{G},A}^T \wedge \Psi_{\mathcal{G},E}^T$$

$$\begin{aligned}
\Psi_{\mathcal{G},V}^T &= \bigwedge_{i=0}^{T-1} \left( \bigwedge_{v \in V_i} l_i(v) \wedge \bigwedge_{v \in V_{\mathcal{G}}^{T'} \setminus V_i} \neg l_i(v) \right) \\
\Psi_{\mathcal{G},A}^T &= \bigwedge_{k=0}^{m-1} \Psi_{\mathcal{G},A_k}^T \\
\Psi_{\mathcal{G},A_k}^T &= \bigwedge_{i=0}^{T-1} \left( \bigwedge_{v \in A_{k,i}} a_{A_{k,i}}(v) \wedge \bigwedge_{v \in V_{\mathcal{G}}^{T'} \setminus A_{k,i}} \neg a_{A_{k,i}}(v) \right) \\
\Psi_{\mathcal{G},E}^T &= \bigwedge_{j=0}^{n-1} \Psi_{\mathcal{G},E_j}^T \\
\Psi_{\mathcal{G},E_j}^T &= \bigwedge_{i=0}^{T-1} \left( \bigwedge_{(v_0, v_1) \in E_{j,i}} e_{E_{j,i}}(v_0, v_1) \wedge \bigwedge_{(v_0, v_1) \in (V_{\mathcal{G}}^{T'} \times V_{\mathcal{G}}^{T'}) \setminus E_{j,i}} \neg e_{E_{j,i}}(v_0, v_1) \right)
\end{aligned}$$

We first prove that an interpretation  $\mathcal{I}$  satisfies  $\Psi_{\mathcal{G}}^T$  if and only if the first  $T$  static graphs in  $\mathcal{G}^{\mathcal{I}}$  match  $\mathcal{G}$ . To achieve this, we first prove for every  $M \in \{V, E_0, \dots, E_{n-1}, A_0, \dots, A_{m-1}\}$  that an interpretation  $\mathcal{I}$  satisfies  $\Psi_{\mathcal{G},M}^T$  if and only if  $M_i = M_i^{\mathcal{I}}$  for all  $i \in \{0, \dots, T-1\}$ . Here  $M_i$  and  $M_i^{\mathcal{I}}$  are the sets that make up the static graphs  $G_i$  and  $G_i^{\mathcal{I}}$ , which are in turn part of  $\mathcal{G}$  and  $\mathcal{G}^{\mathcal{I}}$ . For example, for the case  $M = A_k$ ,  $M_i$  would equal  $A_{k,i}$ .

**Lemma 4.5.** *Let  $\mathcal{G} = (G_0, \dots, G_{T-1})$  be a DTDG, with  $G_i = (V_i, E_{0,i}, \dots, E_{n-1,i}, A_{0,i}, \dots, A_{m-1,i})$ , let  $T' \in \mathbb{N}$  be some timestamp with  $T' \geq T$ , let  $P_{\mathcal{G}}^{T'}$  be the set of propositions implied by  $\mathcal{G}$  and  $T'$ , let  $V_{\mathcal{G}}^{T'}$  be defined as in Definition 4.1, and let  $\mathcal{I}$  be an interpretation of  $P_{\mathcal{G}}^{T'}$ .*

*For all  $M \in \{V, E_0, \dots, E_{n-1}, A_0, \dots, A_{m-1}\}$ , we have that  $\mathcal{I}$  satisfies  $\Psi_{\mathcal{G},M}^T$  if and only if  $M_i = M_i^{\mathcal{I}}$  for all  $i \in \{0, \dots, T-1\}$ .*

**Proof.** *We begin with the case  $M = V$ .*

*We assume that  $\mathcal{I}$  satisfies  $\Psi_{\mathcal{G},V}^T$ . According to Definition 2.18, this is equivalent to for all  $v \in V_{\mathcal{G}}^{T'}$ ,  $i \in \{0, \dots, T-1\}$ :  $\mathcal{I}(l_i(v)) = \top$  if and only if  $v \in V_i$ . As  $V_i \subseteq V_{\mathcal{G}}^{T'}$ , we know that for all  $v \in V_{\mathcal{G}}^{T'}$ ,  $i \in \{0, \dots, T-1\}$  that  $v \in V_i$  is equivalent to  $v \in V_i^{\mathcal{I}}$ . This again is equivalent to  $V_i = V_i^{\mathcal{I}}$ .*

*Now it is easy to see that the cases  $M = A_k$  and  $M = E_j$  are analogous, with slight differences. Specifically for  $M = E_j$  we have  $E_j$  be a subset of  $V_{\mathcal{G}}^{T'} \times V_{\mathcal{G}}^{T'}$ , not  $V_{\mathcal{G}}^{T'}$ .*

*Thus, the lemma is proven for all cases.  $\square$*

We can now prove that an interpretation  $\mathcal{I}$  that satisfies  $\Psi_{\mathcal{G}}^T$  if and only if the first  $T$  layers of  $\mathcal{G}^{\mathcal{I}}$  are equal to the first  $T$  layers of  $\mathcal{G}$ .

**Lemma 4.6.** *Let  $\mathcal{G} = (G_0, \dots, G_{T-1})$  be a DTDG, let  $T' \in \mathbb{N}$  be some timestamp with  $T' \geq T$ , let  $P_{\mathcal{G}}^{T'}$  be the set of propositions implied by  $\mathcal{G}$  and  $T'$ , and let  $\mathcal{I}$  be an interpretation of  $P_{\mathcal{G}}^{T'}$ .*

*We have that  $\mathcal{I}$  satisfies  $\Psi_{\mathcal{G}}^T$  if and only if  $G_i = G_i^{\mathcal{I}}$  for all  $i \in \{0, \dots, T-1\}$ .*

**Proof.** *Let everything be given as in Lemma 4.6, and let  $G_i = (V_i, E_{0,i}, \dots, E_{n-1,i}, A_{0,i}, \dots, A_{m-1,i})$ . We assume that  $\mathcal{I}$  satisfies  $\Psi_{\mathcal{G}}^T$ . Following Definition 2.26, this is equivalent to*

$\mathcal{I}$  satisfying  $\Psi_M^T$  for all  $M \in \{V, E_0, \dots, E_n, A_0, \dots, A_m\}$ . In the following we use  $M_i$  and  $M_i^I$  to refer to elements of  $\{V_i, E_{0,i}, \dots, E_{n-1,i}, A_{0,i}, \dots, A_{m-1,i}\}$  and  $\{V_i^I, E_{0,i}^I, \dots, E_{n-1,i}^I, A_{0,i}^I, \dots, A_{m-1,i}^I\}$  respectively. For example, for  $M = V$ ,  $M_i = V_i$ . From Lemma 4.5 we know that  $\mathcal{I}$  satisfying  $\Psi_{\mathcal{G}}^T$  is equivalent to  $M_i = M_i^I$  for all  $i \in \{0, \dots, T-1\}$  and  $M \in \{V, E_0, \dots, E_n, A_0, \dots, A_m\}$ . This is equivalent to  $G_i = G_i^I$  for all  $i \in \{0, \dots, T-1\}$ .  $\square$

As the above lemma is an equivalence, and we know that for any DTDG  $\mathcal{G}'$  that  $\mathcal{G}' = \mathcal{G}^{\mathcal{I}_{\mathcal{G}'}}$ , it trivially follows that if the first  $T$  layers of  $\mathcal{G}'$  and  $\mathcal{G}$  are identical,  $\mathcal{I}_{\mathcal{G}'}$  satisfies  $\Psi_{\mathcal{G}}^T$ .

**Lemma 4.7.** *Let  $\mathcal{G} = (G_0, \dots, G_{T-1})$  be a DTDG and let  $\mathcal{G}' = (G'_0, \dots, G'_{T'-1})$  be a DTDG with  $T' \geq T$ . We have that  $G_i = G'_i$  for all  $i \in \{0, \dots, T-1\}$  if and only if  $\mathcal{I}_{\mathcal{G}'}$  satisfies  $\Psi_{\mathcal{G}}^T$ .*

**Proof.** *Let everything be given as in Lemma 4.7. We assume that  $G_i = G'_i$  for all  $i \in \{0, \dots, T-1\}$ . Let  $\mathcal{I}_{\mathcal{G}'}$  be the interpretation implied by  $\mathcal{G}'$ , and let  $\mathcal{G}^{\mathcal{I}_{\mathcal{G}'}}$  be the DTDG implied by that interpretation. We know from their construction that  $\mathcal{G}' = \mathcal{G}^{\mathcal{I}_{\mathcal{G}'}}$ . From Lemma 4.6 the equivalence to  $\mathcal{I}_{\mathcal{G}'}$  satisfying  $\Psi_{\mathcal{G}}$  follows directly.  $\square$*

We also prove that in combination with a different formula, that formula and  $\Psi_{\mathcal{G}}^T$  together ensure that  $\mathcal{G}^I$  is a DTDG for all satisfying interpretations. We prove this when proving the three reduction in Section 4.4.

However for the DG-BSAT problem, the question is whether a DTDG exists with certain properties, but no requirement that the first  $T$  layers are the same as those of a different DTDG. Thus when creating the reduction to Boolean SAT, we need to ensure that the first static graph of the DTDG  $\mathcal{G}^I$  implied by an interpretation  $\mathcal{I}$  is a valid static graph. We do this with the following formula:

**Definition 4.8.** *Let  $T' \in \mathbb{N}$  be a timestamp, let  $y \in \mathbb{N}$  be a number of nodes, let  $n \in \mathbb{N}$  be a number of edge relations, let  $m \in \mathbb{N}$  be a number of attribute relations, and let  $V = \{v_0, \dots, v_{y-1}, x_1, \dots, x_{T'-1}\}$ . For the purposes of this definition, we have  $V$  replace  $V_{\mathcal{G}}^{T'}$  for the definition of the set of propositions  $P_{\mathcal{G}}^{T'}$ . We define  $\Psi_{y,n,m}^{T'}$  as follows:*

$$\begin{aligned} \Psi_{y,n,m}^{T'} &= \bigwedge_{i=0}^{y-1} I_0(v_i) \wedge \bigwedge_{i=1}^{T'} \neg I_0(x_i) \\ &\wedge \bigwedge_{v \in V} \left( \bigvee_{j=0}^{m-1} a_{A_{j,0}}(v) \Rightarrow I_0(v) \right) \\ &\wedge \bigwedge_{(v_0, v_1) \in V \times V} \left( \bigvee_{k=0}^{n-1} e_{E_{k,0}}(v_0, v_1) \Rightarrow (I_0(v_0) \wedge I_0(v_1)) \right) \end{aligned}$$

We can easily prove that for any satisfying interpretation of  $\Psi_{y,n,m}^{T'}$ , the first static graph  $G_0^I$  of the implied DTDG  $\mathcal{G}^I$  is a well defined.

**Lemma 4.9.** *Let  $\mathcal{I}$  be an interpretation of  $\Psi_{y,n,m}^{T'}$  for some  $T', y, n, m \in \mathbb{N}$ , and let  $\mathcal{G}^I = (G_0^I, \dots, G_{T'-1}^I)$  be the DTDG implied by  $\mathcal{I}$ , with  $G_0^I = (V_0^I, E_0^I, \dots, E_{n-1}^I, A_0^I, \dots, A_{m-1}^I)$ . We have that  $\mathcal{I}$  satisfies  $\Psi_{y,n,m}^{T'}$  if and only if  $G_0^I$  is well defined, and if  $V_0^I = \{v_0, \dots, v_{y-1}\}$ .*

**Proof.** Let everything be given as in Lemma 4.9. Through Definition 2.26, we know that  $\mathcal{I}$  satisfying  $\Psi_{y,n,m}^{T'}$  is equivalent to all the following statements being true: For all  $v \in \{v_0, \dots, v_{y-1}\}$   $\mathcal{I}(l_0(v)) = \top$ , and for all  $v \in \{x_1, \dots, x_{T'-1}\}$   $\mathcal{I}(l_0(v)) = \perp$ . For all  $v \in V$  and  $A \in \{A_{0,0}, \dots, A_{m-1,0}\}$  we have that  $\mathcal{I}(l_A(v)) = \top$  implies that  $\mathcal{I}(l_0(v)) = \top$ . For all  $(v_0, v_1) \in V \times V$  and  $E \in \{E_{0,0}, \dots, E_{n-1,0}\}$  we have that  $\mathcal{I}(e_E(v_0, v_1)) = \top$  implies that  $\mathcal{I}(l_0(v_0)) = \top$  and  $\mathcal{I}(l_0(v_1)) = \top$ . By Definition 4.3, these statements are again equivalent to  $V_0^{\mathcal{I}} = \{v_0, \dots, v_{T'-1}\}$ ,  $E_0^{\mathcal{I}} \subseteq V_0^{\mathcal{I}} \times V_0^{\mathcal{I}}$  for all  $E_0^{\mathcal{I}} \in \{E_{0,0}^{\mathcal{I}}, \dots, E_{n-1,0}^{\mathcal{I}}\}$ , and  $A_0^{\mathcal{I}} \subseteq V_0^{\mathcal{I}}$  for all  $A_0^{\mathcal{I}} \in \{A_{0,0}^{\mathcal{I}}, \dots, A_{m-1,0}^{\mathcal{I}}\}$ . The last two of these statements are equivalent to  $G_0^{\mathcal{I}}$  being a well defined static graph. We have thus shown the equivalence.  $\square$

## 4.2 Boolean Formula for Events

The second formula we construct ensures that an interpretation  $\mathcal{I}_{\mathcal{G}'}$  implied by a DTDG  $\mathcal{G}' = (G_0, \dots, G_{T'-1})$  only satisfies it if the following holds for some timestamp  $T \in \mathbb{N}$  with  $T \leq T'$ . If for all the graphs  $G_{T-1}, \dots, G_{T'-2}$  there exists an event  $o_i$  such that  $o_i(G_i) = G_{i+1}$ , then  $\mathcal{I}_{\mathcal{G}'}$  fulfills the formula. Conversely, for any satisfying interpretation  $\mathcal{I}$ , for its implied DTDG  $\mathcal{G}^{\mathcal{I}} = (G_0^{\mathcal{I}}, \dots, G_{T'-1}^{\mathcal{I}})$  there exists such an event. We note that there will be cases where satisfying interpretations may imply a structure that is not a graph. However, we prove in Section 4.4 that combined with  $\Psi_{\mathcal{G}}^T$  or  $\Psi_{y,n,m}^{T'}$  all satisfying interpretations imply a DTDG.

For the formula discussed in this subsection, we need an additional set of propositions. First, we define the following set:

**Definition 4.10.** Let  $\mathcal{G} = (G_0, \dots, G_{T-1})$  be a DTDG with  $G_i = (V_i, E_{0,i}, \dots, E_{n-1,i}, A_{0,i}, \dots, A_{m-1,i})$ , let  $T' \in \mathbb{N}$  be a timestamp with  $T' \geq T$ , let  $V_{\mathcal{G}}^{T'}$  be defined as in Definition 4.1, and let  $\mathcal{E}$  be a subset of the set of event types **Types**.

We call  $M_{\mathcal{G},\mathcal{E}}^{T'}$  the set of possible events implied by  $\mathcal{G}$ ,  $\mathcal{E}$  and  $T'$ , with  $M_{\mathcal{G},\mathcal{E}}^{T'}$  being defined as follows:

$$\begin{aligned}
M_{\mathcal{G},\mathcal{E}}^{T'} &= \bigcup_{t \in \mathcal{E}} M_{\mathcal{G},t}^{T'} \\
M_{\mathcal{G},t}^{T'} &= \bigcup_{v \in V_{\mathcal{G}}^{T'}} \{(t, v)\} \\
&\text{for } t \in \{\text{AddNode}, \text{RemoveNode}\} \\
M_{\mathcal{G},t}^{T'} &= \bigcup_{v \in V_{\mathcal{G}}^{T'}} \bigcup_{j=0}^{m-1} \{(t, (A_j, v))\} \\
&\text{for } t \in \{\text{AddAttr}, \text{RemoveAttr}\} \\
M_{\mathcal{G},t}^{T'} &= \bigcup_{(v_0, v_1) \in V_{\mathcal{G}}^{T'} \times V_{\mathcal{G}}^{T'}} \bigcup_{j=0}^{n-1} \{(t, (E_j, v_0, v_1))\} \\
&\text{for } t \in \{\text{AddEdge}, \text{RemoveEdge}\} \\
M_{\mathcal{G},\text{NoneEvent}}^{T'} &= (\text{NoneEvent})
\end{aligned}$$

While events as defined in Definition 2.3 contain timestamps, here we omit them for the purposes of this set. We do this because we only care about all possible in-

stances of each event type, regardless of timestamp. We can now define the set of propositions we need for the following formulas:

**Definition 4.11.** Let  $\mathcal{G} = (G_0, \dots, G_{T-1})$  be a DTDG, let  $T'$  be a timestamp with  $T' \geq T$ , let  $\mathcal{E}$  be a subset of the set of event types **Types**, and let  $M_{\mathcal{G},\mathcal{E}}^{T'}$  be the set of possible events implied by  $\mathcal{G}, \mathcal{E}$ , and  $T'$ .

We call  $P_{\mathcal{G},\mathcal{E}}^{T'}$  the event propositions implied by  $\mathcal{G}, \mathcal{E}$  and  $T'$ , with  $P_{\mathcal{G},\mathcal{E}}^{T'}$  being defined as follows:

$$P_{\mathcal{G},\mathcal{E}}^{T'} = \bigcup_{i=T-1}^{T'-2} \bigcup_{o \in M_{\mathcal{G},\mathcal{E}}^{T'}} \{d_o^i, b_o^i\}$$

We are now able to define the formula  $\Psi_{choice}^i$ . This formula is constructed in such a way that for every satisfying interpretation  $\mathcal{I}$ , there exists exactly one  $o \in M_{\mathcal{G},\mathcal{E}}^{T'}$ , such that  $\mathcal{I}(d_o^i) = \top$ . Conversely, for every  $o \in M_{\mathcal{G},\mathcal{E}}^{T'}$ , there exists a satisfying interpretation  $\mathcal{I}$  such that  $\mathcal{I}(d_o^i) = \top$ .

**Definition 4.12.** Let  $\mathcal{G} = (G_0, \dots, G_{T-1})$  be a DTDG, let  $T' \in \mathbb{N}$  be a timestamp with  $T' \geq T$ , let  $\mathcal{E} \subseteq \mathbf{Types}$  be a set of event types, and let  $M_{\mathcal{G},\mathcal{E}}^{T'} = \{m_0, \dots, m_{p-1}\}$  be the set of possible events implied by  $\mathcal{G}, \mathcal{E}$  and  $T'$ .

We define  $\Psi_{choice}^i$  over  $P_{\mathcal{G},\mathcal{E}}^{T'}$  for  $i \in \{T-1, \dots, T'-2\}$  as follows:

$$\begin{aligned} \Psi_{choice}^i = & \bigvee_{o \in M_{\mathcal{G},\mathcal{E}}^{T'}} d_o^i \wedge \bigwedge_{q=0}^{p-2} ((b_{m_q}^i \rightarrow b_{m_{q+1}}^i) \wedge d_{m_q}^i \rightarrow (\neg b_{m_q}^i \wedge b_{m_{q+1}}^i)) \\ & \wedge (d_{m_{p-1}}^i \rightarrow \neg b_{m_{p-1}}^i) \end{aligned}$$

Now we can prove the property that every satisfying interpretation  $\mathcal{I}$  has exactly one  $o \in M_{\mathcal{G},\mathcal{E}}^{T'}$  for every  $i \in \{T-1, \dots, T'-2\}$  with  $\mathcal{I}(d_o^i) = \top$ .

**Lemma 4.13.** Let  $\mathcal{G} = (G_0, \dots, G_{T-1})$  be a DTDG, let  $T' \in \mathbb{N}$  be a timestamp with  $T' \geq T$ , let  $i \in \{T-1, \dots, T'-2\}$  be another timestamp, let  $\mathcal{E} \subseteq \mathbf{Types}$  be a set of event types, let  $M_{\mathcal{G},\mathcal{E}}^{T'}$  be the set of possible events implied by  $\mathcal{G}, \mathcal{E}$  and  $T'$ , let  $P_{\mathcal{G},\mathcal{E}}^{T'}$  be the set of propositions implied by  $\mathcal{G}, \mathcal{E}$  and  $T'$ , and let  $\mathcal{I}$  be an interpretation of  $P_{\mathcal{G},\mathcal{E}}^{T'}$  that satisfies  $\Psi_{choice}^i$ . We have that there exists exactly one  $m \in M_{\mathcal{G},\mathcal{E}}^{T'}$  with  $\mathcal{I}(d_m^i) = \top$ .

**Proof.** Let everything be given as in Lemma 4.13 and let  $M_{\mathcal{G},\mathcal{E}}^{T'} = \{m_0, \dots, m_{p-1}\}$ .

Assuming  $\mathcal{I}$  satisfies  $\Psi_{choice}^i$ , then there must exist a  $m \in M_{\mathcal{G},\mathcal{E}}^{T'}$  such that  $\mathcal{I}(d_m^i) = \top$ , as otherwise the disjunction over all elements of  $M_{\mathcal{G},\mathcal{E}}^{T'}$  would be not satisfied.

We prove that there exists at most one by contradiction. We assume  $\mathcal{I}$  satisfies  $\Psi_{choice}^i$  and there exist  $m_j, m_k \in M_{\mathcal{G},\mathcal{E}}^{T'}$  with  $m_j \neq m_k$ ,  $\mathcal{I}(d_{m_j}^i) = \top$  and  $\mathcal{I}(d_{m_k}^i) = \top$ . We also assume  $j < k$ . Thus, from  $\text{val}_{\mathcal{I}}(d_{m_q}^i \rightarrow (\neg b_{m_q}^i \wedge b_{m_{q+1}}^i)) = \top$  for all  $q \in \{0, \dots, p-2\}$  and  $\text{val}_{\mathcal{I}}(d_{m_{p-1}}^i \rightarrow \neg b_{m_{p-1}}^i)$ , it follows that  $\mathcal{I}(b_{m_j}^i) = \mathcal{I}(b_{m_k}^i) = \perp$ . However, it is also true that  $\mathcal{I}(d_{m_{j+1}}^i) = \mathcal{I}(d_{m_{k+1}}^i) = \top$ . From  $\text{val}_{\mathcal{I}}(b_{m_q}^i \rightarrow b_{m_{q+1}}^i) = \top$  for all  $q \in \{0, \dots, p-2\}$ , we know that  $\mathcal{I}(b_{m_{j+1}}^i) = \top$  implies  $\mathcal{I}(b_{m_x}^i) = \top$  for all  $x \geq j+1$ . As  $k > j$ , we know that  $k \geq j+1$ . Thus it follows that  $\mathcal{I}(b_{m_k}^i) = \top$ . This is a contradiction.

Thus we know there exist exactly one  $m \in M_{\mathcal{G},\mathcal{E}}^{T'}$  with  $\mathcal{I}(d_m^i) = \top$ .  $\square$

Finally, we define a formula that ensures the property described at the start of this subsection.

**Definition 4.14.** Let  $\mathcal{G} = (G_0, \dots, G_{T-1})$  be a DTDG, with  $G_i = (V_i, E_{0,i}, \dots, E_{n-1,i}, A_{0,i}, \dots, A_{m-1,i})$ , let  $T' \in \mathbb{N}$  be a timestamp with  $T' \geq T$ , let  $\mathcal{E} \subseteq \text{Types}$  be a set of event types, and let  $V_{\mathcal{G}}^{T'}$  be defined as in Definition 4.1. We define the following formulas:

$$\begin{aligned}
\Psi_{events} &= \bigwedge_{i=T-1}^{T'-2} \Psi_{events}^i \\
\Psi_{events}^i &= \Psi_{\text{AddNode}}^i \wedge \Psi_{\text{RemoveNode}}^i \wedge \Psi_{\text{AddAttr}}^i \wedge \Psi_{\text{RemoveAttr}}^i \wedge \Psi_{\text{AddEdge}}^i \wedge \Psi_{\text{RemoveEdge}}^i \\
&\quad \wedge \Psi_{\text{choice}}^i \\
\Psi_{\text{AddNode}}^i &= \bigwedge_{v \in V_{\mathcal{G}}^{T'}} \left( \neg l_i(v) \rightarrow (l_{i+1}(v) \leftrightarrow d_{(\text{AddNode},(v))}^i) \right) \\
&\quad \wedge d_{(\text{AddNode},(v))}^i \rightarrow \neg l_i(v) \\
\Psi_{\text{RemoveNode}}^i &= \bigwedge_{v \in V_{\mathcal{G}}^{T'}} \left( l_i(v) \rightarrow (\neg l_{i+1}(v) \leftrightarrow d_{(\text{RemoveNode},(v))}^i) \right) \\
&\quad \wedge d_{(\text{RemoveNode},(v))}^i \rightarrow l_i(v) \\
\Psi_{\text{AddAttr}}^i &= \bigwedge_{k=0}^{m-1} \bigwedge_{v \in V_{\mathcal{G}}^{T'}} \left( \neg a_{A_{k,i}}(v) \rightarrow (a_{A_{k,i+1}}(v) \leftrightarrow d_{(\text{AddAttr},(A_k,v))}^i) \right) \\
&\quad \wedge d_{(\text{AddAttr},(A_k,v))}^i \rightarrow (\neg a_{A_{k,i}}(v) \wedge l_i(v)) \\
\Psi_{\text{RemoveAttr}}^i &= \bigwedge_{k=0}^{m-1} \bigwedge_{v \in V_{\mathcal{G}}^{T'}} \left( a_{A_{k,i}}(v) \rightarrow (\neg a_{A_{k,i+1}}(v) \leftrightarrow (d_{(\text{RemoveAttr},(A_k,v))}^i \vee d_{(\text{RemoveNode},(v))}^i)) \right) \\
&\quad \wedge d_{(\text{RemoveAttr},(A_k,v))}^i \rightarrow (a_{A_{k,i}}(v)) \\
\Psi_{\text{AddEdge}}^i &= \bigwedge_{j=0}^{n-1} \bigwedge_{(v_0,v_1) \in V_{\mathcal{G}}^{T'} \times V_{\mathcal{G}}^{T'}} \left( \neg e_{E_{j,i}}(v_0, v_1) \rightarrow (e_{E_{j,i+1}}(v_0, v_1) \leftrightarrow d_{(\text{AddEdge},(E_j,v_0,v_1))}^i) \right) \\
&\quad \wedge d_{(\text{AddEdge},(E_j,v_0,v_1))}^i \rightarrow (\neg e_{E_{j,i}}(v_0, v_1) \wedge l_i(v_0) \wedge l_i(v_1)) \\
\Psi_{\text{RemoveEdge}}^i &= \bigwedge_{j=0}^{n-1} \bigwedge_{(v_0,v_1) \in V_{\mathcal{G}}^{T'} \times V_{\mathcal{G}}^{T'}} \left( e_{E_{j,i}}(v_0, v_1) \rightarrow (\neg e_{E_{j,i+1}}(v_0, v_1) \leftrightarrow (d_{(\text{RemoveEdge},(E_j,v_0,v_1))}^i \vee d_{(\text{RemoveNode},(v_0))}^i \vee d_{(\text{RemoveNode},(v_1))}^i)) \right) \\
&\quad \wedge d_{(\text{RemoveEdge},(E_j,v_0,v_1))}^i \rightarrow (e_{E_{j,i}}(v_0, v_1))
\end{aligned}$$

We note that depending on  $\mathcal{E}$ , some  $d_o^i$  might not be defined if  $o \notin M_{\mathcal{G},\mathcal{E}}^{T'}$ . In these cases, we replace  $d_o^i$  in these formulas with  $\perp$ .

We can now prove properties of  $\Psi_{events}$ . For the two properties in the following lemma we also need a list of events  $\mathcal{O} = (o_{T-1}, \dots, o_{T'-2})$ . Given an interpretation  $\mathcal{I}$ , the event  $o_i$  for some  $i \in \{T-1, \dots, T'-2\}$  is the event for which  $\mathcal{I}(d_{o_i}^i) = \top$ . Ordinarily, this would not be well defined. However for the following properties we know that  $\Psi_{\text{choice}}^i$  is satisfied by the interpretation, either through the premise of an implication, or an

assumption we make. The properties are statements about the relation between the interpretation  $\mathcal{I}$ , and the DTDG  $\mathcal{G}^{\mathcal{I}} = (G_0, \dots, G_{T'-1})$  implied by  $\mathcal{I}$ . The main property we state in this lemma is, that under the assumption that  $G_{T-1}$  is a well defined graph,  $\mathcal{I}$  satisfying  $\Psi_{events}$  implies for all  $i \in \{T-1, \dots, T'-2\}$  that  $o_i(G_i) = G_{i+1}$ . We can also state another property, being that under the assumption that for all  $i \in \{T-1, \dots, T'-2\}$   $\mathcal{I}$  satisfies  $\Psi_{choice}^i$ , this implication is actually an equivalence. We need this second statement to prove a separate lemma.

**Lemma 4.15.** *Let  $\mathcal{G} = (G_0, \dots, G_{T-1})$  be a DTDG, let  $T' \in \mathbb{N}$  be a timestamp with  $T' \geq T$ , let  $\mathcal{E} \subseteq \text{Types}$  be a set of event types, let  $\mathcal{I}$  be an interpretation of  $\Psi_{events}$ , and let  $\mathcal{G}^{\mathcal{I}} = (G_0^{\mathcal{I}}, \dots, G_{T'-1}^{\mathcal{I}})$  be the DTDG implied by  $\mathcal{I}$ .*

*Furthermore, let  $O = (o_{T-1}, \dots, o_{T'-2})$  be a list of events, where for each  $o_i$ ,  $\mathcal{I}(d_{o_i}^i) = \top$ . Note that this is not a well defined list for all  $\mathcal{I}$ . However they are trivially well defined for the statements of this lemma.*

*We assume that  $G_{T-1}^{\mathcal{I}}$  is a well defined static graph. We also assume that every event that is not (NoneEvent) changes a graph when it is applied to it.*

*Then, we have that  $\mathcal{I}$  satisfying  $\Psi_{events}$  implies  $o_i(G_i^{\mathcal{I}}) = G_{i+1}^{\mathcal{I}}$  for all  $i \in \{T-1, \dots, T'-2\}$ . Furthermore, under the assumption that  $\mathcal{I}$  satisfies  $\Psi_{choice}^i$  for every  $i \in \{T-1, \dots, T'-2\}$ , we have that  $\mathcal{I}$  satisfying  $\Psi_{events}$  is equivalent to  $o_i(G_i^{\mathcal{I}}) = G_{i+1}^{\mathcal{I}}$  for all  $i \in \{T-1, \dots, T'-2\}$ .*

**Proof.** *Let everything be given as in Lemma 4.15 and let  $V_{\mathcal{G}}^{T'}$  be defined as in Definition 4.1.  $\mathcal{I}$  satisfying  $\Psi_{events}$  is equivalent to  $\mathcal{I}$  for all  $i \in \{T-1, \dots, T'-2\}$  satisfying  $\Psi_{events}^i$ . The following proof is for every  $i \in \{T-1, \dots, T'-2\}$ . Note that we assume that  $G_i^{\mathcal{I}}$  is a well defined static graph for all  $i$ , not just  $i = T-1$ . We can make this assumption for all  $i' \in \{T, \dots, T'-2\}$  inductively, as applying an event to a well defined static graph returns a well defined static graph.*

*$\Psi_{events}^i$  consists of a conjunction of multiple formulas, one for each event type, barring NoneEvent, as well as  $\Psi_{choice}^i$ . We know that all these formulas are satisfied by  $\mathcal{I}$ . In the following, under the assumption that  $\mathcal{I}$  satisfies  $\Psi_{choice}^i$ , everything is equivalent. We will argue directly what statements  $\Psi_{AddNode}^i$  being satisfied is equivalent to, as the formulas for the other event types are similar.*

*Following Definition 2.26,  $\mathcal{I}$  satisfying  $\Psi_{AddNode}^i$  is equivalent to the following two statements being true for all  $v \in V_{\mathcal{G}}^{T'}$ :  $\mathcal{I}(l_i(v)) = \perp$  implies  $\mathcal{I}(l_{i+1}(v)) = \top$  if and only if  $\mathcal{I}(d_{(AddNode,(v))}^i) = \top$ . And  $\mathcal{I}(d_{(AddNode,(v))}^i) = \top$  implies  $\mathcal{I}(l_i(v)) = \perp$ . Following Definition 4.3 and how we defined  $o_i$ , these statements are equivalent to  $v \notin V_i^{\mathcal{I}}$  implies  $v \in V_{i+1}^{\mathcal{I}}$  if and only if  $o_i = (AddNode, (v))$ , and  $o_i = (AddNode, (v))$  implying  $v \notin V_i^{\mathcal{I}}$ . The other event types have analogous arguments, with a few differences worth pointing out: Firstly, any event that removes a node/attribute/edge from the graph,  $o_i$  being that event implies that the node/attribute/edge was part of the graph  $G_i^{\mathcal{I}}$ . This ensures the graph is actually changed by the event. Without this, the assumption that the graph changes for all events that are not (NoneEvent) would not hold. Secondly, edges and attributes also get removed, meaning for example  $v \in A_{k,i}^{\mathcal{I}}$  and  $v \notin A_{k,i+1}^{\mathcal{I}}$ , if the corresponding node  $v$  gets removed, meaning  $o_i = (RemoveNode, (v))$ . Thirdly, adding an attribute or edge implies that the node(s) they are added to are part of the implied DTDG.*

*Using these implications, we can form equivalent statements about the sets included in  $G_i^{\mathcal{I}}$  and  $G_{i+1}^{\mathcal{I}}$ . For  $o_i = (AddNode, (v))$  we get the following set relations:  $V_i^{\mathcal{I}} \cup \{v\} = V_{i+1}^{\mathcal{I}}$ ,  $E_{j,i}^{\mathcal{I}} = E_{j,i+1}^{\mathcal{I}}$  for all  $j \in \{0, \dots, n-1\}$ , and  $A_{k,i}^{\mathcal{I}} = A_{k,i+1}^{\mathcal{I}}$  for all  $k \in \{0, \dots, m-1\}$ . This*

by Definition 2.4 is equivalent to  $o_i(G_i^I) = G_{i+1}^I$ . For all other cases, this equivalence to  $o_i(G_i^I) = G_{i+1}^I$  still holds.

Then, we have that  $\mathcal{I}$  satisfying  $\Psi_{events}$  implies  $o_i(G_i^I) = G_{i+1}^I$  for all  $i \in \{T-1, \dots, T'-2\}$ . We also have that, under the assumption that  $\mathcal{I}$  satisfies  $\Psi_{choice}^i$  for all  $i \in \{T-1, \dots, T'-2\}$ ,  $\mathcal{I}$  satisfying  $\Psi_{events}$  is equivalent to  $o_i(G_i^I) = G_{i+1}^I$  for all  $i \in \{T-1, \dots, T'-2\}$ .  $\square$

We can informally understand the following lemma as the reverse of the implication within the previous lemma. To prove it we will prove the assumption for the second statement of that lemma, and then use the equivalence. Instead of looking at the relation between an interpretation and the DTDG implied by it, we look at the relation between a DTDG  $\mathcal{G}' = (G'_0, \dots, G'_{T'-1})$ , and an interpretation  $\mathcal{I}_{\mathcal{G}'}$  implied by it. Here we assume that there exists a list of events  $O = \{o_{T-1}, \dots, o_{T'-2}\}$  with  $o_i(G'_i) = G'_{i+1}$  for all  $i \in \{T-1, \dots, T'-2\}$ . We also note that  $\mathcal{G}'$  only implies part of  $\mathcal{I}_{\mathcal{G}'}$  here, so within the lemma we further specify a specific interpretation, that fits the formula  $\Psi_{events}$ . Then we can show that that interpretation  $\mathcal{I}_{\mathcal{G}'}$  satisfies  $\Psi_{events}$ .

**Lemma 4.16.** *Let  $\mathcal{G} = (G_0, \dots, G_{T-1})$  be a DTDG, let  $\mathcal{G}' = (G'_0, \dots, G'_{T'-1})$  be a DTDG with  $T' \geq T$ , let  $\mathcal{E} \subseteq \text{Types}$  be a set of event types, let  $M_{\mathcal{G}, \mathcal{E}}^{T'} = \{m_0, \dots, m_{p-1}\}$  be the set of possible events implied by  $\mathcal{G}, \mathcal{E}$  and  $T'$ , let  $O = \{o_{T-1}, \dots, o_{T'-2}\}$  be a list of events with  $o_i \in M_{\mathcal{G}, \mathcal{E}}^{T'}$ , and let  $\mathcal{I}_{\mathcal{G}'}$  be an interpretation implied by  $\mathcal{G}'$ . Furthermore, we let the parts of  $\mathcal{I}_{\mathcal{G}'}$  not given by  $\mathcal{G}'$  be defined as follows for all  $i \in \{T-1, \dots, T'-2\}$ :*

$$\begin{aligned} \mathcal{I}_{\mathcal{G}'}(d_o^i) &= \top \text{ if and only if } o = o_i \\ \mathcal{I}_{\mathcal{G}'}(b_{m_j}^i) &= \top \text{ if and only if } j > k \text{ with } k \text{ being given through } o_i = m_k \end{aligned}$$

We then have  $\mathcal{I}_{\mathcal{G}'}$  satisfies  $\Psi_{events}$ .

**Proof.** *Let everything be given as in Lemma 4.16.*

*First we can prove that  $\mathcal{I}_{\mathcal{G}'}$  satisfies  $\Psi_{choice}^i$  for all  $i \in \{T-1, \dots, T'-2\}$ . With this we can then use the second statement in Lemma 4.15. We will argue that the different subformula of  $\Psi_{choice}^i$  are satisfied individually. With how we have constructed  $\mathcal{I}_{\mathcal{G}'}$  within the lemma, we know that the disjunction is trivially true, as there exists an  $o \in M_{\mathcal{G}, \mathcal{E}}^{T'}$  such that  $\mathcal{I}_{\mathcal{G}'}(d_o^i) = \top$ . We also know that for all  $m_q \in M_{\mathcal{G}, \mathcal{E}}^{T'} \setminus \{m_{p-1}\}$  the implication  $b_{m_q}^i \rightarrow b_{m_{q+1}}^i$  is satisfied. This is because for some  $k$  the premise is not fulfilled for all  $q \leq k$ , and for all other  $q > k$  the conclusion of the implication is fulfilled. For the remaining implications, we know that for only exactly one of them the premise is fulfilled, and thus all others are trivially true. The remaining one, where  $\mathcal{I}_{\mathcal{G}'}(d_o^i) = \top$ , is also trivially satisfied, per construction.*

*We know per construction that the DTDG  $\mathcal{G}'$  is equal to the DTDG  $\mathcal{G}^{\mathcal{I}_{\mathcal{G}'}}$  implied by the interpretation  $\mathcal{I}_{\mathcal{G}'}$  implied by  $\mathcal{G}'$ . With this, we know from the second statement of Lemma 4.15 that  $\mathcal{I}_{\mathcal{G}'}$  satisfies  $\Psi_{events}$ .  $\square$*

### 4.3 Translation of a MFOTL Formula

While the previous formulas focus on the graph and its restrictions, the formula in this section focuses on a Boolean formula based on a MFOTL formula  $\Phi$ . We use  $\Phi$  for MFOTL formula here, and  $\Psi$  for Boolean formulas. Given a DTDG  $\mathcal{G}$ , the formula defined in the following can only be satisfied by its implied interpretation  $\mathcal{I}_{\mathcal{G}}$  if  $(\mathcal{G}, 0) \models \Phi$ . Conversely, if an interpretation  $\mathcal{I}$  satisfies the formula, then  $(\mathcal{G}^{\mathcal{I}}, 0) \models \Phi$ ,

where  $\mathcal{G}^I$  is the DTDG implied by  $\mathcal{I}$ . We define this new formula inductively over the structure of a MFOTL formula  $\Phi$ , in the form of a function  $tr$ , as follows:

**Definition 4.17.** Let  $\mathcal{G} = (G_0, \dots, G_{T-1})$  be a DTDG, with  $G_i = (V_i, E_{0,i}, \dots, E_{n-1,i}, A_{0,i}, \dots, A_{m-1,i})$ , let  $T' \in \mathbb{N}$  be a timestamp with  $T' \geq T$ , let  $\Phi_0$  and  $\Phi_1$  be MFOTL formulas over the signature implied by  $\mathcal{G}$ , let  $V_{\mathcal{G}}^{T'}$  be defined as in Definition 4.1, let  $v_0, v_1 \in V_{\mathcal{G}}^{T'}$ , and let  $i \in \mathbb{N}$  be some integer. We call  $tr$  the translation function, with  $tr$  being defined as follows:

$$\begin{aligned}
tr(\text{true}, i) &= \top \\
tr(v_0 \approx v_1, i) &= \begin{cases} \top, & \text{if } v_0 = v_1 \\ \perp, & \text{else} \end{cases} \\
tr(R_{E_j}(v_0, v_1), i) &= e_{E_{j, \min(i, T'-1)}}(v_0, v_1) \\
tr(R_{A_k}(v_0), i) &= a_{A_{k, \min(i, T'-1)}}(v_0) \\
tr(L(v_0), i) &= l_{\min(i, T'-1)}(v_0) \\
tr(\neg \Phi_0, i) &= \neg tr(\Phi_0, i) \\
tr(\Phi_0 \vee \Phi_1, i) &= tr(\Phi_0, i) \vee tr(\Phi_1, i) \\
tr(\exists x. \Phi_0, i) &= \bigvee_{u \in V_{\mathcal{G}}^{T'}} tr(\Phi_0[x \rightarrow u], i) \\
tr(\bigcirc_I \Phi_0, i) &= \begin{cases} tr(\Phi_0, i+1) & \text{if } 1 \in I \\ \perp & \text{else} \end{cases} \\
tr(\bigodot_I \Phi_0, i) &= \begin{cases} tr(\Phi_0, i-1) & \text{if } i \geq 1, 1 \in I \\ \perp & \text{else} \end{cases} \\
tr(\Phi_0 \mathbf{U}_{[t_0, t_1]} \Phi_1, i) &= \begin{cases} tr(\Phi_1, i) & \text{if } t_0 = t_1 = 0 \\ tr(\Phi_1, i) \vee tr(\Phi_0, i) \wedge tr(\Phi_0 \mathbf{U}_{[t_0, t_1-1]} \Phi_1, i+1) & \text{else if } t_0 = 0, t_1 \geq 1 \\ tr(\Phi_0, i) \wedge tr(\Phi_0 \mathbf{U}_{[t_0-1, t_1-1]} \Phi_1, i+1) & \text{else} \end{cases} \\
tr(\Phi_0 \mathbf{S}_{[t_0, t_1]} \Phi_1, i) &= \begin{cases} tr(\Phi_1, i) & \text{if } t_0 = t_1 = 0 \\ & \text{or } t_0 = i = 0 \\ tr(\Phi_1, i) \vee tr(\Phi_0, i) \wedge tr(\Phi_0 \mathbf{S}_{[t_0, t_1-1]} \Phi_1, i-1) & \text{else if } t_0 = 0, \\ & t_1 \geq 1, i > 0 \\ tr(\Phi_0, i) \wedge tr(\Phi_0 \mathbf{S}_{[t_0-1, t_1-1]} \Phi_1, i-1) & \text{else if } t_0 \geq 0, \\ & t_1 \geq 1, i > 0 \\ \perp & \text{else} \end{cases}
\end{aligned}$$

Now we prove the one of the above mentioned properties for  $tr$  inductively. We only need to prove one of them, as the other one follows trivially.

**Lemma 4.18.** Let  $\Phi$  be a DG-MFOTL formula, let  $\mathcal{G} = (G_0, \dots, G_{T-1})$  be a DTDG, let  $T' \in \mathbb{N}$  be another timestamp with  $T' \geq T$ , and let  $P_{\mathcal{G}}^{T'}$  be the propositions implied by  $\mathcal{G}$  and  $T'$ , let  $\mathcal{I}$  be some interpretation of  $P_{\mathcal{G}}^{T'}$ , and let  $\mathcal{G}^I$  be the DTDG implied by  $\mathcal{I}$ . We then have that  $\text{val}_{\mathcal{I}}(tr(\Phi, 0)) = \top$  if and only if  $(\mathcal{G}^I, 0) \models \Phi$ .

**Proof.** With everything given as in Lemma 4.18, let  $\Phi'$  be a MFOTL formula without free variables, where all intervals are finite, and where the same restrictions on  $\exists$  apply

as for DG-MFOTL in Definition 3.1. Let  $i \in \mathbb{N}$ , and let  $V_{\mathcal{G}}^{T'}$  be defined as in Definition 4.1.

For this proof via structural induction, we need to prove a more general statement than that in Lemma 4.18.

We will prove this new statement by induction over the structure of  $\Phi'$ :

Either  $\text{val}_{\mathcal{I}}(\text{tr}(\Phi', i)) = \top$  if and only if  $(\mathcal{G}^{\mathcal{I}}, i) \models \Phi'$ , or  $\Phi'$  is not a formula over the signature of  $\Phi$ .

We first prove the base cases where  $\Phi'$  does not contain any subformulas except for itself. We also assume the signature of  $\Phi$  is  $(C, \mathcal{R}, \iota)$ . Multiple of the following formulas are defined for some  $v_0, v_1 \in V_{\mathcal{G}}^{T'}$ . Trivially, the statement is true if  $v_0 \notin C$  or  $v_1 \notin C$ , as then  $(C, \mathcal{R}, \iota)$  is not a fitting signature for  $\Phi'$ . Thus we only need to prove the other cases, and can assume in the following that  $v_0, v_1 \in C$ .

- Case  $\Phi' = \text{true}$ . By Definition 2.18  $(\mathcal{G}^{\mathcal{I}}, i) \models \text{true}$  is trivially true. By Definition 4.17,  $\text{val}_{\mathcal{I}}(\text{tr}(\Phi', i)) = \text{val}_{\mathcal{I}}(\text{tr}(\text{true}, i)) = \top$ .
- Case  $\Phi' = v_0 \approx v_1$ . By Definition 2.18  $(\mathcal{G}^{\mathcal{I}}, i) \models v_0 \approx v_1$  if and only if  $d_i(v_0) = d_i(v_1)$ . By definition 2.22  $d_i(v_0) = v_0$  and  $d_i(v_1) = v_1$ , thus by definition  $(\mathcal{G}^{\mathcal{I}}, i) \models v_0 \approx v_1$  if and only if  $v_0 = v_1$ . By Definition 4.17  $v_0 = v_1$  is equivalent to  $\text{val}_{\mathcal{I}}(\text{tr}(v_0 \approx v_1, i)) = \text{val}_{\mathcal{I}}(\top) = \top$ .
- Case  $\Phi' = L(v_0)$ ,  $\Phi' = A_k(v_0)$ ,  $\Phi' = E_j(v_0, v_1)$ . We here prove the case  $\Phi' = L(v_0)$  explicitly, as the other two cases are analogous. By Definition 2.18  $(\mathcal{G}^{\mathcal{I}}, i) \models L(v_0)$  if and only if  $v_0 \in d_i(L)$ , which by Definition 2.22 is equivalent to  $v_0 \in V_i^{\mathcal{I}}$  for  $i \in \{0, \dots, T' - 1\}$  or  $v_0 \in V_{T'-1}^{\mathcal{I}}$  if  $i \geq T'$ . This can be summarized to  $v \in V_{\min(i, T'-1)}^{\mathcal{I}}$ , which by Definition 4.3 is equivalent to  $\mathcal{I}(l_{\min(i, T'-1)}(v)) = \top$ . By Definition 4.17 this is equivalent to  $\text{val}_{\mathcal{I}}(\text{tr}(L(v), i)) = \top$ .

We can now prove the remaining cases inductively. As an induction hypothesis, we assume that for the formulas  $\Phi_0$  and  $\Phi_1$  the statement is true. If they are not formulas over the signature of  $\Phi$ , the statement is trivially true for the following cases. Thus in the following, we will assume that they are over the the signature of  $\Phi$ .

- Case  $\Phi' = \neg\Phi_0$ .  
By Definition 2.18,  $(\mathcal{G}^{\mathcal{I}}, i) \models \neg\Phi_0$  if and only if  $(\mathcal{G}^{\mathcal{I}}, i) \not\models \Phi_0$ , which per induction hypothesis is equivalent to  $\text{val}_{\mathcal{I}}(\text{tr}(\Phi_0, i)) = \perp$ . This by Definition 2.26 is equivalent to  $\text{val}_{\mathcal{I}}(\text{tr}(\neg\Phi_0, i)) = \text{val}_{\mathcal{I}}(\text{tr}(\neg\Phi_0, i)) = \top$ .
- Case  $\Phi' = \Phi_0 \vee \Phi_1$ .  
By Definition 2.18,  $(\mathcal{G}^{\mathcal{I}}, i) \models \Phi_0 \vee \Phi_1$  if and only if  $(\mathcal{G}^{\mathcal{I}}, i) \models \Phi_0$  or  $(\mathcal{G}^{\mathcal{I}}, i) \models \Phi_1$ . Per induction hypothesis, this is equivalent to  $\text{val}_{\mathcal{I}}(\text{tr}(\Phi_0, i)) = \top$  or  $\text{val}_{\mathcal{I}}(\text{tr}(\Phi_1, i)) = \top$ , which by Definition 2.26 is equivalent to  $\text{val}_{\mathcal{I}}(\text{tr}(\Phi_0, i) \vee \text{tr}(\Phi_1, i)) = \text{val}_{\mathcal{I}}(\text{tr}(\Phi_0 \vee \Phi_1, i)) = \top$ .
- Case  $\Phi' = \exists x.\Phi'_0$ .  
Due to the restrictions on  $\Phi'$ ,  $\Phi'_0$  must have the form  $\Phi'_0 = L(x) \wedge \Phi'_1$  for some  $\Phi'_1$ . As  $x$  is a variable bound to  $\exists x$ , we do not assume the induction hypothesis for  $\Phi'_0$  and  $\Phi'_1$ , as without  $\exists x$ , they contain the free variable  $x$ .  
By Definition 2.23  $(\mathcal{G}^{\mathcal{I}}, i) \models \exists x.\Phi'_0$  is equivalent to  $(\overline{D}, \overline{\tau}, \text{val}, i) \models \exists x.\Phi'_0$ , where

$(\bar{D}, \bar{\tau})$  is the temporal structure implied by  $\mathcal{G}^I$ , and  $\text{val}$  is any valuation of the free variables in  $\Phi'_0$ . As there exist no free variables in  $\Phi'_0$ ,  $\text{val}$  is thus exactly defined. We know by Definition 2.18 that this is equivalent to there existing a  $u \in U_i$  such that  $(\bar{D}, \bar{\tau}, \text{val}[x \rightarrow u], i) \models \Phi'_0$ . By Definition 2.22 the set of constants  $C_{\mathcal{G}^I}$  implied by  $\mathcal{G}^I$  and the universe  $U_i$  implied by  $\mathcal{G}^I$  are identical. Furthermore, the interpretation function  $d_i$  is the identity for these constants. As by Definition 2.17  $\text{val}(u) = d_i(u)$  for all constants  $u \in C_{\mathcal{G}^I}$ , we can trivially see that  $(\bar{D}, \bar{\tau}, \text{val}[x \rightarrow u], i) \models \Phi'_0$  and  $(\bar{D}, \bar{\tau}, \text{val}, i) \models \Phi'_0[x \rightarrow u]$  are equivalent here. Thus by Definition 2.23,  $(\mathcal{G}^I, i) \models \exists x.\Phi'_0$  is equivalent to there existing a  $u \in U_i$  such that  $(\mathcal{G}^I, i) \models \Phi'_0[x \rightarrow u]$ .  $(\mathcal{G}^I, i) \models \Phi'_0[x \rightarrow u]$  is equivalent to  $(\mathcal{G}^I, i) \models L(u)$  and  $(\mathcal{G}^I, i) \models \Phi'_1[x \rightarrow u]$ . The former is equivalent to  $u \in V_i$ . As  $V_i \subseteq U_i$  by Definition 2.22, the statement where we only consider  $u \in V_i$  is equivalent, as for  $u \in U \setminus V_i$  the statement cannot be true. Conversely, we can extend the statement to there existing a  $u \in V_{\mathcal{G}^I}^{T'}$ , as  $V_i \subseteq V_{\mathcal{G}^I}^{T'}$ . As  $\Phi'$  has no free variables, we know that  $\Phi'_0$  has no free variables besides  $x$ , and thus  $\Phi'_0[x \rightarrow u]$  also has no free variables. We can thus assume the induction hypothesis for the formula for all  $u \in V_{\mathcal{G}^I}^{T'}$ , and thus the statement is equivalent to there existing a  $u \in V_{\mathcal{G}^I}^{T'}$  such that  $\text{val}_{\mathcal{I}}(\text{tr}(\Phi'_0[x \rightarrow u], i)) = \top$ . This then is equivalent to  $\text{val}_{\mathcal{I}}(\bigvee_{u \in V_{\mathcal{G}^I}^{T'}} \text{tr}(\Phi'_0[x \rightarrow u], i)) = \text{val}_{\mathcal{I}}(\text{tr}(\exists x.\Phi'_0, i)) = \top$  following Definition 2.26.

- Case  $\Phi' = \bullet_I \Phi_0$ .

Following Definition 2.18  $(\mathcal{G}, i) \models (\bullet_I \Phi_0)$  is true if and only if  $i > 0$ ,  $\tau_i - \tau_{i-1} \in I$  and  $(\mathcal{G}, i-1) \models \Phi_0$ . Note that if  $i = 0$ , then  $\tau_{i-1}$  is not defined. Since  $i > 0$  and  $\tau_i = i$ ,  $\tau_i - \tau_{i-1} = 1$  and thus  $1 \in I$ .

$\text{val}_{\mathcal{I}}(\text{tr}(\bullet_I \Phi_0, i)) = \top$  is equivalent to  $i > 0$ ,  $\text{val}_{\mathcal{I}}(\text{tr}(\Phi_0, i-1)) = \top$  and  $1 \in I$ , as  $\text{tr}(\perp) \neq \top$ .  $\text{val}_{\mathcal{I}}(\text{tr}(\Phi_0, i-1)) = \top$  is equivalent to  $(\mathcal{G}^I, i-1) \models \Phi_0$  due to the induction hypothesis.

- Case  $\Phi' = \Phi_0 \mathbf{S}_I \Phi_1$  with  $I = [t_0, t_1]$  for some  $t_0, t_1 \in \mathbb{N}$ .

We can unroll the definition of  $\text{tr}(\Phi_0 \mathbf{S}_I \Phi_1)$  as follows:

$$\begin{aligned} \text{tr}(\Phi_0 \mathbf{S}_I \Phi_1, i) &= \bigwedge_{p=0}^{t_0-1} (\text{tr}(\Phi_0, i-p)) \\ &\quad \wedge \left( \text{tr}(\Phi_1, i-t_0) \vee \text{tr}(\Phi_0, i-t_0) \right) \\ &\quad \wedge \left( \text{tr}(\Phi_1, i-t_0-1) \vee \text{tr}(\Phi_0, i-t_0-1) \right) \\ &\quad \wedge (\dots) \\ &\quad \wedge \left( \text{tr}(\Phi_1, \max(i-t_1, 0)) \vee \dots \right) \end{aligned}$$

An interpretation  $\mathcal{I}$  satisfies this formula if and only if the following two properties are true. Firstly, if  $\text{val}_{\mathcal{I}}(\text{tr}(\Phi_1, j)) = \top$  for some  $j \in \mathbb{N}$  with  $i-t_0 \geq j \geq \max(i-t_1, 0)$ . And secondly, if  $\text{val}_{\mathcal{I}}(\text{tr}(\Phi_0, k)) = \top$  for all  $k \in \mathbb{N}$  with  $j < k \leq i$ . Following the induction hypothesis,  $\text{val}_{\mathcal{I}}(\text{tr}(\Phi_1, j)) = \top$  is equivalent to  $(\mathcal{G}^I, j) \models \Phi_1$ , and  $\text{val}_{\mathcal{I}}(\text{tr}(\Phi_0, k)) = \top$  is equivalent to  $(\mathcal{G}^I, k) \models \Phi_0$ .

All that is left for us to show is that  $i-t_0 \geq j \geq \max(i-t_1, 0)$  is equivalent to  $\tau_i - \tau_j \in I$ .  $j \geq \max(i-t_1, 0)$  enforces that  $j \geq 0$ , which means that for all these  $j$ ,  $\tau_j$  is defined, as for  $j < 0$   $\tau_j$  is not defined. For  $\tau_i - \tau_j \in [t_0, t_1]$  to be true, both  $t_0 \leq \tau_i - \tau_j$  and  $\tau_i - \tau_j \leq t_1$  need to be true. As  $\tau_q = q$  for all  $q \in \mathbb{N}$ ,

$i - t_0 \geq j$  is trivially equivalent to  $t_0 \leq \tau_i - \tau_j$ .  $j \geq \max(i - t_1, 0)$  is equivalent to  $j \geq 0$  and  $j \geq i - t_1$ , which is equivalent to  $\tau_i - \tau_j \leq t_1$ . Then by Definition 2.18,  $\text{val}_{\mathcal{I}}(\text{tr}(\Phi_0 \mathcal{S}_I \Phi_1, i)) = \top$  is equivalent to  $(\mathcal{G}^I, i) \models \Phi_0 \mathcal{S}_I \Phi_1$ . Thus we have shown the equivalence in this case.

- The cases  $\Phi' = \bigcirc_I \Phi_0$  and  $\Phi' = \Phi_0 \bigcup_I \Phi_1$  are analogous to the cases  $\Phi' = \bigodot_I \Phi_0$  and  $\Phi' = \Phi_0 \mathcal{S}_I \Phi_1$ .

Thus we have proven the statement for all cases. We have thus also proven the specific case that  $\Phi' = \Phi$ . As  $\Phi$  is a formula over the signature of  $\Phi$ ,  $\text{val}_{\mathcal{I}}(\text{tr}(\Phi, 0)) = \top$  if and only if  $(\mathcal{G}_I, i) \models \Phi$ . Thus we have proven the first part of the lemma. As per construction  $\mathcal{G}^{I_{\mathcal{G}'}} = \mathcal{G}'$ , it follows that  $(\mathcal{G}', 0) \models \Phi$  if and only if  $\text{val}_{\mathcal{G}'}(\text{tr}(\Phi, 0)) = \top$ .  $\square$

#### 4.4 Reductions to SAT

We can now define and prove reductions from BDGMC (Definition 3.2), EDGMC (Definition 3.4), and DG-BSAT (Definition 3.5) to Boolean SAT. All of these problems can be reduced to a conjunction over multiple of the previously defined formula.

We start with the reduction from BDGMC to Boolean SAT.

**Theorem 4.19.** *Let  $\mathcal{G}, \Phi$  be given as in Definition 3.2. Let  $\Psi_{\mathcal{G}}^T$  and  $\text{tr}(\Phi, 0)$  be given as in Definition 4.4 and Definition 4.17. We have that  $(\mathcal{G}, \Phi)$  is a positive instance of BDGMC if and only if  $\Psi_{\mathcal{G}}^T \wedge \text{tr}(\Phi, 0)$  is satisfiable.*

**Proof.** *Let everything be given as in Theorem 4.19. We first prove that if  $(\mathcal{G}, 0) \models \Phi$ , it follows that  $\Psi_{\mathcal{G}}^T \wedge \text{tr}(\Phi, 0)$  is satisfiable. Specifically, we show that the interpretation  $\mathcal{I}_{\mathcal{G}}$  implied by  $\mathcal{G}$  satisfies  $\Psi_{\mathcal{G}}^T \wedge \text{tr}(\Phi, 0)$ . From Lemma 4.7 we know that an interpretation  $\mathcal{I}_{\mathcal{G}'}$  implied by a DTDG  $\mathcal{G}' = (G'_0, \dots, G'_{T-1})$  satisfies  $\Psi_{\mathcal{G}'}^T$  if and only if for all  $i \in \{0, \dots, T-1\}$   $G_i = G'_i$ . Thus  $\mathcal{I}_{\mathcal{G}}$  trivially satisfies  $\Psi_{\mathcal{G}}^T$ . From Lemma 4.18 we know that  $\mathcal{I}_{\mathcal{G}}$  satisfies  $\text{tr}(\Phi, 0)$  if and only if for  $\mathcal{G}^{I_{\mathcal{G}}}$ , the DTDG implied by  $\mathcal{I}_{\mathcal{G}}$ ,  $(\mathcal{G}^{I_{\mathcal{G}}}, 0) \models \Phi$  is true. As per construction  $\mathcal{G} = \mathcal{G}^{I_{\mathcal{G}}}$ , it follows that  $\mathcal{I}_{\mathcal{G}}$  satisfies  $\text{tr}(\Phi, 0)$ . Thus if  $(\mathcal{G}, 0) \models \Phi$ , then  $\mathcal{I}_{\mathcal{G}}$  satisfies  $\Psi_{\mathcal{G}}^T \wedge \text{tr}(\Phi, 0)$ , and it is thus satisfiable.*

*We now prove that if  $\Psi_{\mathcal{G}}^T \wedge \text{tr}(\Phi, 0)$  is satisfiable, it follows that  $(\mathcal{G}, 0) \models \Phi$ . Assuming that  $\Psi_{\mathcal{G}}^T \wedge \text{tr}(\Phi, 0)$  is satisfiable, let  $\mathcal{I}$  be an interpretation that satisfies it.  $\mathcal{I}$  thus satisfies both  $\Psi_{\mathcal{G}}^T$  and  $\text{tr}(\Phi, 0)$ . We can now make statements about the DTDG*

*$\mathcal{G}^{\mathcal{I}} = (G_0^{\mathcal{I}}, \dots, G_{T-1}^{\mathcal{I}})$  implied by  $\mathcal{I}$ , and also show that  $\mathcal{G} = \mathcal{G}^{\mathcal{I}}$ . From Lemma 4.6 we know that  $\mathcal{I}$  satisfying  $\Psi_{\mathcal{G}}^T$  implies for all  $i \in \{0, \dots, T-1\}$  that  $G_i = G_i^{\mathcal{I}}$ . As both  $\mathcal{G}$  and  $\mathcal{G}^{\mathcal{I}}$  consist of  $T$  static graphs, we know that  $\mathcal{G} = \mathcal{G}^{\mathcal{I}}$ . By Lemma 4.18  $\mathcal{I}$  satisfying  $\text{tr}(\Phi, 0)$  is equivalent to  $(\mathcal{G}^{\mathcal{I}}, 0) \models \Phi$ . As  $\mathcal{G} = \mathcal{G}^{\mathcal{I}}$ , it follows that  $(\mathcal{G}, 0) \models \Phi$ .*

*Thus we have proven the equivalence.  $\square$*

We have thus reduced BDGMC to Boolean SAT. For EDGMC we reduce to a different formula, we note however that it includes the formula for the BDGMC reduction. Thus, some of the following proof will be analogous to the previous proof, even if some adjustments and additions need to be made.

**Theorem 4.20.** *Let  $\mathcal{G}, \Phi, \mathcal{E}, T'$  be given as in Definition 3.4. Let  $\Psi_{\mathcal{G}}^T$ ,  $\Psi_{\text{events}}$  and  $\text{tr}(\Phi, 0)$  be given as in Definition 4.4, Definition 4.14 and Definition 4.17. We have that  $(\mathcal{G}, \Phi, \mathcal{E}, T')$  is a positive instance of EDGMC if and only if  $\Psi_{\mathcal{G}}^T \wedge \Psi_{\text{events}} \wedge \text{tr}(\Phi, 0)$  is satisfiable.*

**Proof.** Let everything be given as in Theorem 4.20, and let  $M_{\mathcal{G},\varepsilon}^{T'}$  be given as in Definition 4.10. We prove both directions of the equivalence individually.

First we assume that such  $\mathcal{G}' = (G'_0, \dots, G'_{T'-1})$  and  $O$  as in Definition 3.4 exist. By the definition of  $\mathcal{G}'$  we know for all  $i \in \{0, \dots, T' - 1\}$  that  $G_i = G'_i$ . Thus we know that the interpretation  $\mathcal{I}_{\mathcal{G}'}$  implied by  $\mathcal{G}'$  satisfies  $\Psi_{\mathcal{G}}^T$  due to Lemma 4.7. Now we let the parts of  $\mathcal{I}_{\mathcal{G}'}$  that are not given by  $\mathcal{G}'$  be defined as in Lemma 4.16. Then due to Lemma 4.16, we have that  $\mathcal{I}_{\mathcal{G}'}$  satisfies  $\Psi_{events}$ . The proof that  $\mathcal{G}'$  satisfies  $tr(\Phi, 0)$  is analogous to how we proved it for Theorem 4.19. Thus we have that  $\mathcal{I}_{\mathcal{G}'}$  satisfies  $\Psi_{\mathcal{G}}^T$ ,  $\Psi_{events}$  and  $tr(\Phi, 0)$ , and thus it also satisfies  $\Psi_{\mathcal{G}}^T \wedge \Psi_{events} \wedge tr(\Phi, 0)$ . Therefore,  $\Psi_{\mathcal{G}}^T \wedge \Psi_{events} \wedge tr(\Phi, 0)$  is satisfiable.

We now prove that assuming that  $\Psi_{\mathcal{G}}^T \wedge \Psi_{events} \wedge tr(\Phi, 0)$  is satisfiable, there exist a  $\mathcal{G}$  and  $O$  that meet the requirements of Definition 3.4. Let  $\mathcal{I}$  be a satisfying interpretation.  $\mathcal{I}$  thus satisfies  $\Psi_{\mathcal{G}}^T$ ,  $\Psi_{events}$ , and  $tr(\Phi, 0)$ . We can now make statements about the DTDG  $\mathcal{G}^{\mathcal{I}} = (G_0^{\mathcal{I}}, \dots, G_{T'-1}^{\mathcal{I}})$  implied by  $\mathcal{I}$ . From Lemma 4.6 we know that  $\mathcal{I}$  satisfying  $\Psi_{\mathcal{G}}^T$  implies for all  $i \in \{0, \dots, T' - 1\}$  that  $G_i = G_i^{\mathcal{I}}$ . Since  $G_{T'-1}^{\mathcal{I}}$  is thus a well defined static graph, we know from Lemma 4.15 that for all  $i \in \{T' - 1, \dots, T' - 2\}$   $o_i(G_i^{\mathcal{I}}) = G_{i+1}^{\mathcal{I}}$  for some  $o_i \in M_{\mathcal{G},\varepsilon}^{T'}$ . We thus also know that  $\mathcal{G}^{\mathcal{I}}$  is a well defined DTDG, as all its static graphs are well defined. From Lemma 4.18 we know that  $\mathcal{I}$  satisfying  $tr(\Phi, 0)$  is equivalent to  $(\mathcal{G}^{\mathcal{I}}, 0) \models \Phi$ . Thus  $\mathcal{G}^{\mathcal{I}}$  and  $O = (o_{T'-1}, \dots, o_{T'-2})$  meet the requirements in Definition 3.4.

We have thus proven the equivalence.  $\square$

Thus we have reduced both BDGMC and EDGMC to Boolean SAT. The reduction from DG-BSAT to Boolean SAT is more similar to that of EDGMC. However, we replace the Boolean formula  $\Psi_{\mathcal{G}}^{T'}$  with  $\Psi_{y,n,m}^{T'}$ . We also note, that many definitions are in some form based on a specific DTDG  $\mathcal{G}$  being given. Here we don't have such a DTDG given. However, in all the definitions needed for the following theorem and proof, not all parts of that DTDG are relevant. The relevant parts of that DTDG are the number of nodes within the DTDG, which here is the same as the initial static graph, and the number of edge and attribute relations. As these are given for the DG-BSAT problem, these definitions and the related proofs still hold. Furthermore, for these values the set  $V_{\mathcal{G}}^{T'}$  is identical to  $V$  as defined in Definition 4.8. Thus the relevant proofs based on it also still hold. With this, we can now define the reduction from DG-BSAT to Boolean SAT:

**Theorem 4.21.** Let  $\Phi, T', y, n, m$  be given as in Definition 3.5. Let  $\Psi_{y,n,m}^{T'}$  be defined as in Definition 4.8, and let  $\Psi_{events}$  and  $tr(\Phi, 0)$  be defined as in their Definition 4.14 and Definition 4.17, with  $\mathcal{G}$  in those definitions being based on  $y, n, m$  and  $T = 1$ . We have that  $(\Phi, T', y, n, m)$  is a positive instance of DG-BSAT if and only if  $\Psi_{y,n,m}^{T'} \wedge \Psi_{events} \wedge tr(\Phi, 0)$  is satisfiable.

**Proof.** Let everything be given as in Theorem 4.21. We prove both directions of the equivalence individually.

First we assume that  $\mathcal{G}$  and  $O$  as in Definition 3.5 exist. From there we show that  $\mathcal{I}_{\mathcal{G}}$ , specifically the version defined in Definition 4.16 where it is given as  $\mathcal{I}_{\mathcal{G}'}$ , satisfies  $\Psi_{y,n,m}^{T'} \wedge \Psi_{events} \wedge tr(\Phi, 0)$ . With how  $\mathcal{G}$  is defined, we know from Lemma 4.9 that  $\Psi_{y,n,m}^{T'}$  is trivially satisfied by  $\mathcal{I}_{\mathcal{G}}$ , due to  $\mathcal{G} = \mathcal{G}^{\mathcal{I}_{\mathcal{G}}}$ . Similarly, with how  $\mathcal{G}$  and  $O$  are defined, it trivially follows from Lemma 4.16 that  $\Psi_{events}$  is satisfied. The proof that  $\mathcal{G}$  satisfies  $tr(\Phi, 0)$  is analogous to how we proved Theorem 4.19. Thus we have that  $\mathcal{I}_{\mathcal{G}}$  satisfies

$\Psi_{y,n,m}^T$ ,  $\Psi_{events}$  and  $tr(\Phi, 0)$ , and thus it also satisfies  $\Psi_{y,n,m}^T \wedge \Psi_{events} \wedge tr(\Phi, 0)$ . Therefore,  $\Psi_{y,n,m}^T \wedge \Psi_{events} \wedge tr(\Phi, 0)$  is satisfiable.

We now prove that assuming that  $\Psi_{y,n,m}^T \wedge \Psi_{events} \wedge tr(\Phi, 0)$  is satisfiable, there exist a  $\mathcal{G}$  and  $O$  that meet the requirements of Definition 3.5. Let  $\mathcal{I}$  be a satisfying interpretation.  $\mathcal{I}$  thus satisfies  $\Psi_{y,n,m}^T$ ,  $\Psi_{events}$  and  $tr(\Phi, 0)$ . We can make statements about the DTDG  $\mathcal{G}^{\mathcal{I}} = (G_0^{\mathcal{I}}, \dots, G_{T'-1}^{\mathcal{I}})$  implied by  $\mathcal{I}$ . From Lemma 4.9 it follows that  $G_0^{\mathcal{I}}$  is a well defined static graph, and that  $|V| = y$ . We then know analogously to the proof of Theorem 4.20, that  $\mathcal{G} = \mathcal{G}^{\mathcal{I}}$  meets the other requirements, and that there exists an  $O$  that meets them as well.

We have thus proven the equivalence.  $\square$

## 5 Implementation

To explore the feasibility of using this reduction for model checking dynamic graphs, we developed a proof-of-concept implementation of the reduction. Here we look at the runtime performance of deciding the Boolean SAT problem for the resulting formula, as well as the runtime of generating the formula. The reduction was implemented using Python, using Lark [7] to parse the MFOTL formula, and PySAT [5] to build the formulas and check their satisfiability. We specifically test with the Glucose 4.2.1 SAT-Solver [1]. We encode the MFOTL formula using a slightly modified version of the .mfotl files used in Lima, Huerta y Munive, and Traytel [8]. In Section 6 we further discuss the implication of the test results.

All the following tests were performed on a laptop using a AMD Ryzen 7 8840H processor, 32 GB DDR5-5600 MHz RAM, and a NVIDIA GeForce RTX 4070 Laptop-GPU.

For the tests we will use different DG-MFOTL formulas. The formulas  $\Phi_0$ ,  $\Phi_1$  and  $\Phi_2$  are formulas that are satisfied if and only if there exist 3/4/5 different nodes at timestamp 0. We use these formulas for testing due to them having different  $\exists$ -depths of 3, 4 and 5 respectively.

For a slightly more complex formula we test with  $\Phi'_3$ , where  $t \in \mathbb{N}$  is the upper bound for two of the intervals within the formula. The formula is satisfied, if the following three statements are true: For some timestamp between 0 and  $t$  all nodes within the DTDG have the attribute  $A_0$ . At timestamp 0 there exist at most two nodes with the attribute  $A_0$ . For all timestamps between 1 and  $t$  a node can only have the attribute  $A_0$  if in there exists a  $E_0$  edge from a node that during the previous timestamp had the attribute  $A_0$ .

$$\Phi_0 = \exists x_0. \left( L(x_0) \wedge \exists x_1. \left( L(x_1) \wedge \exists x_2. \left( L(x_2) \wedge \neg(x_0 \approx x_1 \vee x_0 \approx x_2 \vee x_1 \approx x_2) \right) \right) \right)$$

$$\Phi_1 = \exists x_0. \left( L(x_0) \wedge \exists x_1. \left( L(x_1) \wedge \exists x_2. \left( L(x_2) \wedge \exists x_3. \left( L(x_3) \wedge \neg(x_0 \approx x_1 \vee x_0 \approx x_2 \vee x_0 \approx x_3 \vee x_1 \approx x_2 \vee x_1 \approx x_3 \vee x_2 \approx x_3) \right) \right) \right) \right)$$

$$\Phi_2 = \exists x_0. \left( L(x_0) \wedge \exists x_1. \left( L(x_1) \wedge \exists x_2. \left( L(x_2) \wedge \exists x_3. \left( L(x_3) \wedge \exists x_4. \left( L(x_4) \wedge \neg(x_0 \approx x_1 \vee x_0 \approx x_2 \vee x_0 \approx x_3 \vee x_0 \approx x_4 \vee x_1 \approx x_2 \vee x_1 \approx x_3 \vee x_1 \approx x_4 \vee x_2 \approx x_3 \vee x_2 \approx x_4 \vee x_3 \approx x_4) \right) \right) \right) \right) \right)$$

$$\begin{aligned} \Phi'_3 = & F_{[0,t]} \left( \forall x. (\neg L(x) \vee R_{A_0}(x)) \right) \\ & \wedge \exists x. \left( L(x) \wedge \exists y. \left( L(y) \wedge \forall z. (\neg L(z) \vee z \approx x \vee z \approx y \vee \neg R_{A_0}(z)) \right) \right) \\ & \wedge G_{[1,t]} \left( \forall x. \left( \neg L(x) \vee \left( R_{A_0}(x) \rightarrow \exists y. \left( L(y) \wedge R_{E_0}(y, x) \wedge \bullet_{[1,1]}(R_{A_0}(y)) \right) \right) \right) \right) \end{aligned}$$

In addition to testing with these formulas, we also performed tests for some of their negations.

We performed several tests for DG-BSAT, EDGMC and BDGMC with different formulas, graphs, and parameters. Here we varied the number of nodes, number of timestamps, the number of edge and attribute relations, and which events were allowed. A list of tests run and the corresponding results can be found in Appendix A.

Increasing the number of nodes had a significant impact on the runtime of the reduction across the different tests. In most cases this increase in runtime roughly follows the increase in the size of  $|tr(\Phi)|$  for the respective MFOTL formula  $\Phi$ . We can estimate this size asymptotically using the  $\exists$ -depth  $d(\Phi)$  of the formula as  $O(|V|^{d(\Phi)})$ , where  $|V|$  is the total number of nodes accounted for by the reduction. This includes both the nodes included in the initial DTDG, as well as the ones which are part of the reduction for the cases where a node is added.

The increase in nodes affected the runtime significantly more for formulas with a higher  $\exists$ -depth. Comparing results of deciding DG-BSAT for  $\Phi_0$ ,  $\Phi_1$  and  $\Phi_2$  with 15 nodes and  $T = 5$  timestamps, deciding the problem took 3.36s, 47.46s and 1561.0s respectively. For  $\Phi_3^4$  the time to create the formula was similar if a bit higher compared to  $\Phi_0$ . However the time to solve the formula at 100 nodes was significantly higher at 24.04s compared to 0.41s. Increasing the number of edge relations and attribute relations from 1 to 3 also significantly increased the solving time further, even though the additional attributes and edges were not represented in  $\Phi_3^4$ . Here the solving time increased by a factor of more than 4. Lowering the allowed event types also significantly improved solving time for solving DG-BSAT for  $\Phi_3^4$ . While for most tests the time to generate the formula was longer than the time to solve the formula, this is not true for all cases. Specifically when testing a EDGMC reduction for  $\Phi_3^{19}$  with increasing numbers of timestamps, the time to solve significantly outweighed the time it took to generate the formulas. This was specifically the case when the formula was not satisfiable. In general increasing the number of timestamps also significantly increased the solving time for EDGMC and DG-BSAT. For BDGMC the solving time remained relatively low, even when the time to generate the formula increased significantly.

## 6 Discussion and Outlook

In Section 4 we have constructed a reduction to Boolean SAT, for each of the problems we defined in Section 3. Thus we have reductions for a satisfiability problem, as well as two different model checking problems. We defined these reductions for our general definition of dynamic graphs. Thereby we have worked out a solid foundation for model checking dynamic graphs, that future works could expand upon in different ways. These expansions could include optimizing the reductions for less allowed event types, expanding upon the event definitions we have given, or adjusting the logic used in different ways, among others. Different use cases might benefit from one or multiple of these expansions. With the test results from Section 5 in mind, we can discuss some of these avenues for future work:

- For use cases where node additions do not occur, we do not need to account for additional nodes that can be added within our reduction. Specifically for the EDGMC and DG-BSAT problem this also means that the number of timestamps  $T'$  does not influence the effective number of nodes we account for in the reduction. While depending on the problem the effect here might not be too significant, for simpler formulas it could enable us to check for significantly higher amounts of timestamps. If additionally node removal is not allowed, this might lead to an even simpler reduction, as we do not need to account for whether or not a node exists for a specific timestamp.
- In Section 5 we mostly focused on formulas with an  $\exists$ -depth of  $d = 3$ , as for  $d > 3$  the runtime and size of the formulas grows significantly faster. Thus any property that would require a formula with a higher  $\exists$ -depth would not be feasible for any actual use cases. Such examples would include any property that includes statements like "does there exist a node with four different neighbors, such that...", as to check that we would require a similar formula to  $\Phi_2$  from Section 5. Thus exploring options to either more efficiently reduce such specific cases, or to even expand the logic we use could be beneficial. This could potentially be achieved by using a similar Boolean formula as in Definition 4.12. With the option of expanding the logic, an existence quantifier like  $\exists^n x$  that requires the existence of  $n$  unique variables might be one way to formalize this. Specifically for DG-BSAT, allowing some constants for nodes known to be part of the DTDG might also be an option, as we can thus ensure that there exist different nodes, and they are not part of a preexisting DTDG.
- As the amount of nodes is a big contributing factor to the size of a reduction, limiting it by only checking properties for subgraphs of larger graphs is an option that could be further explored. This could enable model checking for higher timestamps to be feasible.
- The dynamic graph and event definitions in this work allow more complex events, such as removing all attributes from one node, only through applying multiple events in succession. As increases in timestamps were shown to generally increase the runtime significantly, the option to define more complex events should be explored further.
- In Section 2.4 we restrict the timestamps to specific discrete values for the temporal structure implied by a DTDG. One avenue of further exploration could be

the expansion of this definition, to allow different timestamps within the temporal structure implied by the DTDG.

## 7 References

- [1] audemard. *Glucose SAT solver*. <https://github.com/audemard/glucose>. [Accessed 10-05-2025].
- [2] Christel Baier and Joost-Pieter Katoen. “Principles of model checking”. In: 2008. URL: <https://api.semanticscholar.org/CorpusID:5302889>.
- [3] David Basin et al. “Monitoring Metric First-Order Temporal Properties”. In: *J. ACM* 62.2 (May 2015). ISSN: 0004-5411. DOI: 10.1145/2699444. URL: <https://doi.org/10.1145/2699444>.
- [4] European Commission et al. *Ethics guidelines for trustworthy AI*. Publications Office, 2019. DOI: doi/10.2759/346720.
- [5] Alexey Ignatiev, Antonio Morgado, and Joao Marques-Silva. “PySAT: A Python Toolkit for Prototyping with SAT Oracles”. In: *SAT*. 2018, pp. 428–437. DOI: 10.1007/978-3-319-94144-8\_26. URL: [https://doi.org/10.1007/978-3-319-94144-8\\_26](https://doi.org/10.1007/978-3-319-94144-8_26).
- [6] Seyed Mehran Kazemi et al. *Representation Learning for Dynamic Graphs: A Survey*. 2020. arXiv: 1905.11485 [cs.LG].
- [7] *Lark-parser*. <https://github.com/lark-parser/lark>. [Accessed 10-05-2025].
- [8] Leonardo Lima, Jonathan Julián Huerta y Munive, and Dmitriy Traytel. “Explainable Online Monitoring of Metric First-Order Temporal Logic”. In: *Tools and Algorithms for the Construction and Analysis of Systems*. Ed. by Bernd Finkbeiner and Laura Kovács. Cham: Springer Nature Switzerland, 2024, pp. 288–307. ISBN: 978-3-031-57246-3.
- [9] Shengjie Min et al. “STGSN — A Spatial–Temporal Graph Neural Network framework for time-evolving social networks”. In: *Knowledge-Based Systems* 214 (2021), p. 106746. ISSN: 0950-7051. DOI: <https://doi.org/10.1016/j.knosys.2021.106746>. URL: <https://www.sciencedirect.com/science/article/pii/S0950705121000095>.
- [10] Sanjit A. Seshia, Dorsa Sadigh, and S. Shankar Sastry. “Toward verified artificial intelligence”. In: *Commun. ACM* 65.7 (June 2022), pp. 46–55. ISSN: 0001-0782. DOI: 10.1145/3503914. URL: <https://doi.org/10.1145/3503914>.
- [11] Joakim Skarding, Bogdan Gabrys, and Katarzyna Musial. “Foundations and Modeling of Dynamic Networks Using Dynamic Graph Neural Networks: A Survey”. In: *IEEE Access* 9 (2021), pp. 79143–79168. DOI: 10.1109/ACCESS.2021.3082932.
- [12] Yufan Zheng et al. *Epidemiology-informed Graph Neural Network for Heterogeneity-aware Epidemic Forecasting*. 2024. arXiv: 2411.17372 [cs.LG]. URL: <https://arxiv.org/abs/2411.17372>.

## A Implementation Test Results

In this appendix we show a selection of test results for various settings and graphs that had some sort of significance. In the following tables  $y$  refers to the number of nodes,  $r$  indicates whether or not the solver returned a True (satisfiable) or False (not satisfiable). TTS refers to the time it took to solve the Boolean formula, and TT refers to the total time it took to decide the problem, both given in seconds. We use the formulas introduced in Section 5. First we look at test results for DG-BSAT in Table 1 and Table 2. Here  $T$  is the number of timestamps,  $n$  is the number of edge relations, and  $m$  is the number of attribute relations.

y	r	TTS	TT
0	1	2.89e-04	0.163
1	1	3.81e-04	0.238
2	1	8.34e-04	0.325
3	1	8.22e-04	0.424
5	1	1.31e-03	0.740
10	1	2.82e-03	1.84
20	1	0.0205	6.75
30	1	0.0350	16.23
50	1	0.427	55.39
100	1	3.93	343.70

(a)  $\neg\Phi_3^4$ :  $T=5, n=1, m=1$   
Events: All

y	r	TTS	TT
0	0	1.12e-05	0.0778
1	0	1.45e-05	0.109
2	0	1.10e-05	0.153
3	1	2.77e-04	0.213
5	1	3.70e-04	0.375
10	1	9.77e-04	1.04
20	1	3.54e-03	4.27
30	1	8.31e-03	10.84
50	1	0.031	37.98
100	1	0.17	239.01

(b)  $\Phi_0$ :  $T=5, n=1, m=1$   
Events: NoneEvent

y	r	TTS	TT
0	0	2.15e-05	0.115
1	0	1.41e-05	0.153
2	0	1.12e-05	0.219
3	1	8.43e-04	0.307
5	1	1.21e-03	0.546
10	1	5.61e-03	1.55
15	1	0.0169	3.36
20	1	0.0122	5.67
30	1	0.0253	13.71
50	1	0.0748	45.52
100	1	0.408	268.02

(c)  $\Phi_0$ :  $T=5, n=1, m=1$   
Events: All

y	r	TTS	TT
0	0	1.22e-05	0.158
1	0	1.34e-05	0.233
2	0	1.10e-05	0.330
3	1	5.51e-04	0.493
5	1	1.42e-03	0.839
10	1	4.30e-03	2.42
20	1	0.0181	8.62
30	1	0.0337	20.10
50	1	0.0917	60.07
100	1	0.397	324.74

(d)  $\Phi_0$ :  $T=5, n=3, m=3$   
Events: NoneEvent

y	r	TTS	TT
0	0	1.50e-05	0.260
1	0	1.24e-05	0.362
2	0	1.14e-05	0.534
3	1	2.66e-03	0.814
5	1	4.33e-03	1.36
10	1	9.17e-03	3.76
20	1	0.075	12.47
30	1	0.079	28.31
50	1	0.566	81.22
100	1	1.13	397.10

(e)  $\Phi_0$ :  $T=5, n=3, m=3$   
Events: All

y	r	TTS	TT
0	0	1.69e-05	1.06
1	0	2.60e-05	1.33
2	0	1.05e-05	1.62
3	1	7.46e-03	2.21
5	1	0.0178	3.15
10	1	0.129	6.17
15	1	0.0671	10.18
20	1	0.0557	16.05
30	1	0.177	31.70
50	1	0.736	86.07
100	1	24.88	428.46

(f)  $\Phi_0$ :  $T=10, n=1, m=1$   
Events: All

y	r	TTS	TT
0	1	2.63e-04	0.117
1	1	3.39e-04	0.158
2	1	5.20e-04	0.230
3	0	1.03e-05	0.295
5	0	9.30e-06	0.475
10	0	1.00e-05	1.29
20	0	1.03e-05	5.03
30	0	1.17e-05	11.96
50	0	1.17e-05	40.74
100	0	1.65e-05	248.14

(g)  $\neg\Phi_0$ :  $T=5, n=1, m=1$   
Events: All

y	r	TTS	TT
0	0	1.41e-05	0.207
1	0	1.22e-05	0.368
2	0	1.67e-05	0.669
3	0	1.31e-05	1.10
5	1	1.55e-03	2.80
10	1	6.59e-03	14.70
15	1	0.0151	47.46
20	1	0.0311	118.33
30	1	0.0881	467.02

(h)  $\Phi_1$ :  $T=5, n=1, m=1$   
Events: All

y	r	TTS	TT
0	0	2.74e-05	0.801
1	0	2.67e-05	2.12
2	0	1.07e-05	5.12
3	0	1.14e-05	10.68
5	1	2.66e-03	38.28
10	1	0.0245	340.66
15	1	0.108	1561.0

(i)  $\Phi_2$ :  $T=5, n=1, m=1$   
Events: All

Table 1: Test results for DG-BSAT for the formulas. The parameters with which the tests were run are shown below the respective tables.

y	r	TTS	TT	y	r	TTS	TT	y	r	TTS	TT
0	0	1.22e-05	0.149	0	0	2.72e-05	0.305	0	0	1.00e-05	0.126
1	1	5.17e-04	0.230	1	1	8.94e-04	0.449	1	1	2.17e-04	0.176
2	1	1.09e-03	0.325	2	1	1.30e-03	0.703	2	1	6.05e-04	0.250
3	1	1.08e-03	0.425	3	1	1.86e-03	0.971	3	1	4.6e-04	0.335
5	1	1.50e-03	0.722	5	1	0.0103	1.63	5	1	8.77e-04	0.553
10	0	0.219	2.10	10	0	0.709	4.89	10	0	0.0161	1.50
20	0	0.597	7.60	20	0	1.93	15.55	20	0	0.0395	5.91
30	0	1.80	18.53	30	0	5.85	37.15	30	0	0.0986	14.14
50	0	6.84	63.19	50	0	22.24	114.80	50	0	0.328	48.99
100	0	24.04	365.48	100	0	106.22	579.18	100	0	1.70	316.14

(a)  $\Phi_3^4$ : T=5, n=1, m=1 Events: All  
(b)  $\Phi_3^4$ : T=5, n=3, m=3 Events: All  
(c)  $\Phi_3^4$ : T=5, n=1, m=1 Events: NoneEvent, RemoveNode

Table 2: More test results for DG-BSAT for the formulas. The parameters with which the tests were run are shown below the respective tables.

We also look at the results of tests for both BDGMC and EDGMC in Table 3. We define an edge relation  $E = \{(0, 0), (0, 1), (1, 2), (2, 3), (1, 4)\}$ , and a set of nodes  $V^i = \{0, \dots, i - 1\}$ , as both this set and edge relation are used in the definitions of multiple DTDG used in the following tests.

Formula	r	TTS	TT	Formula	r	TTS	TT
$\Phi_0$	1	2.61	243.91	$\Phi_0$	1	0.0417	162.63
$\neg\Phi_0$	0	1.91e-05	240.62	$\neg\Phi_0$	0	1.84e-05	168.12
$\Phi_3^4$	0	1.98	313.82	$\Phi_3^4$	0	1.69e-05	226.25
$\neg\Phi_4^3$	1	0.87	329.17	$\neg\Phi_3^4$	1	0.126	244.78

(a) EDGMC tests run on  $\mathcal{G}^0 = ((V^{100}, E, \{0\}))$  with  $T = 5$  and all event types allowed.  
(b) BDGMC tests run on  $\mathcal{G}^1 = (G_0^1, \dots, G_4^1)$ ,  $G_i^1 = (V^{100}, E, \{0, \dots, i\})$ , with  $T = 5$  and all event types allowed.

T	r	TTS	TT	Formula	T	r	TTS	TT
16	0	428.33	444.55	$\Phi_3^{29}$	30	1	7.71e-03	34.98
17	0	4977.24	4995.33	$\Phi_3^{49}$	50	1	0.0499	273.88
18	1	0.144	22.61	$\Phi_3^{99}$	100	1	0.382	7638.58

(c) EDGMC tests run on  $\mathcal{G}^2 = (G_0^1, \dots, G_4^1)$ ,  $G_i^2 = (V^{12}, E, \{0, \dots, i\})$  and  $\Phi_3^{19}$ , with only AddEdge and AddAttr events being allowed.  
(d) BDGMC tests run on  $\mathcal{G}^3 = (G_0^1, \dots, G_{T-1}^1)$ ,  $G_i^3 = (V^T, V^T \times V^T, V^T)$ , with AddEdge and AddAttr events being allowed.

Table 3: Multiple tests of EDGMC and BDGMC being run with different formulas and graphs.