

Article

On Syntactical Simplification of Temporal Operators in Negation-Free Metric Temporal Logic

Mathijs van Noort ^{1,*} , Femke Ongenae ¹  and Pieter Bonte ² ¹ IDLab, Ghent University—imec, 9052 Ghent, Belgium; femke.ongenae@ugent.be² Department of Computer Science, KU Leuven Campus Kulak, 8500 Kortrijk, Belgium; pieter.bonte@kuleuven.be

* Correspondence: mathijs.vannoort@ugent.be

Abstract

Temporal reasoning in dynamic, data-intensive environments increasingly demands expressive yet tractable logical frameworks. Traditional approaches often rely on negation to express absence or contradiction. In such contexts, negation-as-failure is commonly used to infer negative information from the lack of positive evidence. However, for open and distributed systems such as IoT networks and the Semantic Web, negation-as-failure semantics become unreliable due to incomplete and asynchronous data. This has led to growing interest in negation-free fragments of temporal rule-based systems, which preserve monotonicity and enable scalable reasoning. This paper investigates the expressive power of negation-free Metric Temporal Logic (MTL), a temporal logic framework designed for rule-based reasoning over time. We show that the “always” operators $\Box$ and $\Box$, often treated as syntactic sugar for combinations of other temporal constructs, can be eliminated using “once”, “since” and “until” operators. Remarkably, even the “once” operators can be removed, yielding a fragment based solely on “until” and “since”. These results challenge the assumption that negation is necessary for expressing universal temporal constraints and reveal a robust fragment capable of capturing both existential and invariant temporal patterns. Furthermore, the results induce a reduction in the syntax of MTL, which, in turn, can provide benefits for both theoretical study as well as for implementation efforts.

Keywords: negation; metric temporal logic; temporal logic; logic programming; model semantics**MSC:** 03B44Academic Editor: Asieh Salehi
Fathabadi

Received: 26 February 2026

Revised: 17 March 2026

Accepted: 23 March 2026

Published: 27 March 2026

Copyright: © 2026 by the authors.
Licensee MDPI, Basel, Switzerland.
This article is an open access article
distributed under the terms and
conditions of the [Creative Commons
Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Negation is a widely used construct in formal, rule-based reasoning frameworks. It allows for the expression of absence, exclusion, or contradiction, and is often relied upon across a variety of application domains. For instance, in financial systems [1], sensor monitoring [2], or E-health services [3], the ability to express that something did not occur, or is not present, is often necessary. Without some form of negation, it becomes difficult to formally capture notions such as a no-show guest, a missing data point, or the absence of a required condition.

In logical rule-based systems, a common way to support negation is through the Closed World Assumption (CWA) [4]. Under this assumption, if a fact is not explicitly known to be true, it is assumed to be false. This type of negation, referred to as Negation-as-Failure (NAF), is often a reasonable simplification in systems where the data is centrally managed and assumed to be complete [4].

However, recent developments in information systems have shifted attention toward open and distributed settings, where the CWA becomes harder to maintain [5,6]. Examples include the World Wide Web, where data is distributed across many sources, and Internet of Things (IoT) environments [7,8], where components may be numerous, heterogeneous, and subject to change. In such settings, assuming that all relevant information is available and up-to-date at all times is increasingly difficult to justify. This poses particular challenges for negation: the absence of a fact in one part of the system does not necessarily imply its global absence. For instance, a missing sensor reading may be due to transmission delay, device failure, or simply a temporary disconnection, rather than a true absence of the measured phenomenon. Similarly, in federated or decentralized systems, the lack of a record in one data source does not guarantee that the record does not exist elsewhere. These ambiguities undermine the reliability of NAF semantics, which depend on a complete and static view of the data. Moreover, the dynamic and asynchronous nature of distributed systems—where data sources may evolve independently—complicates the interpretation of negation and calls for more cautious reasoning strategies that take these dynamic and asynchronous aspects into account. As a result, frameworks for open systems require explicit mechanisms to handle partial knowledge and delayed information, further challenging traditional logical foundations of negation.

At the same time, the growing availability of real-time data has led to increased interest in temporal rule-based reasoning [9,10]. Many phenomena of interest are not characterized by isolated facts, but by patterns that unfold over time. For example, Santipantakis et al. [11] observes the spatio-temporal positions of vessels in order to detect various types of vessel activities, and Bellomarini et al. [12] manages and evaluates evolutions in financial markets, most notably company ownership. This has motivated the development of temporal extensions to classical logic-based systems, based on either Linear Temporal Logic (LTL) or MTL. Some recent developments in this field include Linear Temporal Public Announcement Logic (LTPAL) [13], Metric Spatio-Temporal Logic (MSTL) [14], Logic-based framework for Analytic Reasoning over Streams (LARS) [15], and DatalogMTL [16].

In addition to the considerations of more open and distributed systems, negation also presents specific challenges in streaming environments, where data arrives incrementally, and the system must reason over a continuously evolving state [17]. In such settings, the absence of a fact at a given moment does not necessarily imply its permanent absence—it may simply not have arrived yet. This makes it difficult to apply classical NAF semantics, which rely on a fixed and complete dataset. To address this, various stratified or windowed negation approaches have been proposed [18–20], where negation is restricted to certain layers of the program or applied only within bounded time intervals. These are, however, constructions that inherently treat negation differently, with good reason: entailment in First-Order Logic (FOL)—and thereby in many rule-based reasoning systems—is undecidable [21]. By introducing negation, either via NAF or stratified negation, a reasoning framework risks undecidability and may compromise its reliability, especially when the reasoning process must be both timely and robust in the face of incomplete information.

These challenges have led to a growing interest in identifying expressive yet tractable fragments of temporal rule-based systems that avoid negation altogether. In particular, researchers have explored “positive fragments”—systems that exclude negation to preserve monotonicity and decidability. For example, Urbani et al. [22] study positive and stratified LARS programs, while work on LTL and MTL often focuses on negation-free fragments to enable efficient model checking [4,23,24].

While such fragments are intuitively less expressive, i.e., negation allows us to express opposites and contradictions, they remain able to express the same temporal expressions of their larger counterparts that include negation: A recurring theme in temporal logic is the

equivalence between constructs like “always P ” and “not once not P .” For instance, in LTL, this is expressed as $G\phi \equiv \neg F(\neg\phi)$ [25], and in LARS, $\Box\alpha \equiv \neg\Diamond\neg\alpha$ [26]. These equivalences suggest that temporal “always” operators can be syntactic sugar for expressions involving negation and “once”.

This raises an important question: what happens to such constructs when negation is removed? Must we reintroduce “always” as a core operator of the syntax? And if we do not, does this limit the expressive power of the system?

In this work, we show that in MTL, negation is not indispensable in the syntax: the “always” operators $\boxplus$ and $\boxminus$ can be eliminated despite the lack of negation. This result indicates that a meaningful class of temporal expressions—specifically, those that describe properties that hold continuously over intervals—can be expressed within negation-free fragments of MTL. This shows that the expressivity of negation-free MTL is not as limited as intuition might suggest. They can capture not only existential or event-based patterns, but also universal temporal expressions. Furthermore, it shows that even in negation-free fragments, MTL over bounded intervals only requires temporal operators S and $\mathcal{U}$.

Classical results in temporal logic relate “always” and “once” operators through negation (e.g. $G\phi$ as $\neg F(\neg\phi)$ in LTL), and several works in the MTL literature discuss how temporal operators can be inter-expressed under the assumption that negation is available [26–28]. However, these existing results do not address whether such relationships continue to hold in negation-free MTL. In particular, the question of whether universal operators such as $\boxplus$ and $\boxminus$ can be eliminated without relying on negation has, to our knowledge, not been answered before. Likewise, it has not been shown that the “once” operators themselves can be removed in a negation-free setting. More generally, syntactical reduction of systems, as considered in the wider literature of model theory [29–31], has remained unexplored with regard to MTL-based systems as a whole.

The results presented in this paper demonstrate that negation-free MTL is more expressive than previously assumed: $\boxplus$ and $\boxminus$ can be rewritten using only S and $\mathcal{U}$, and even the “once” operators are unnecessary for capturing all bounded temporal behavior. Moreover, by providing punctuality-free variants of these transformations, we extend the operator-elimination results to Metric Interval Temporal Logic (MITL) and related fragments that forbid singleton intervals for reasons of robustness and decidability. These findings place negation-free MTL within a more precise landscape of temporal expressiveness and identify a minimal, yet fully adequate, set of operators for bounded reasoning.

From a practical perspective, the results imply that negation-free MTL frameworks only need to account for two temporal operators, compared to six. This significant reduction in syntactic complexity opens the door for a slimmer and more maintainable codebase. By showcasing how certain uses of negation can be eliminated through rewriting without sacrificing expressivity, we illustrate that a viable negation-free fragment of MTL exists, which adequately balances expressivity with achieving a scalable reasoning system.

The remainder of this paper is structured as follows; We first familiarize the reader with the syntax and semantics of DatalogMTL, as defined in the literature, in Section 2. The theoretical rewritings, supported by their formal proofs, are presented in Section 3. We discuss the implications of our results in Section 4, with a concluding note in Section 5.

2. Materials and Methods

MTL, introduced by Koymans [27], introduced a framework for quantitative temporal reasoning in real-time systems. It has since found its way into multiple reasoning systems that deal with temporal data or changing states, such as DatalogMTL [16], Predictive MTL (P-MTL) [32] and Metric Equilibrium Logic (MEL) [33]. We briefly recall the syntax

and semantics as they have been adopted throughout the literature [16,27,34]. Within the scope of this work, we opt for a model-theoretic semantics and rational timeline $(\mathbb{Q}, <)$.

Definition 1. Given a list of predicate symbols $\mathcal{P}$, a model M is a triple $(\mathbb{Q}, <, V)$ where V is a mapping from time points in $\mathbb{Q}$ to a subset of predicates in $\mathcal{P}$.

Example 1. By means of example, we consider a setting of healthcare monitoring using Internet-of-Things (IoT) data. Information is collected on a single patient's heart rate, blood pressure, and whether they received medication. When the medical circumstances require action, an alarm can be sent out. This data is captured using the set of predicates $P = \{hrHigh, bpLow, medGiven, alarm\}$ over a rational timeline $\mathbb{Q}$ with the usual order $<$.

A model M represents a particular course of events, where pieces of the above information are observed at various moments in time; Let $M = (\mathbb{Q}, <, V)$ where V assigns the following (event-time) valuations near $t = 10$ (minutes): $V(7) = \{hrHigh\}$, $V(8) = \{hrHigh, bpLow\}$, $V(9) = \{medGiven\}$, $V(10) = \{hrHigh\}$, and $V(t) = \emptyset$ otherwise.

Within the scope of this work, we consider Bounded MTL (BMTL), which only considers bounded intervals in the subscript of temporal operators:

Definition 2. The set of temporal formulae is defined recursively as follows:

$$A := p \mid \top \mid \neg A \mid A \wedge A \mid \boxplus_I A \mid \boxminus_I A \mid \boxplus_I A \mid \boxminus_I A \mid \mathcal{S}_I A \mid \mathcal{A}\mathcal{U}_I A$$

where I is a positive bounded interval over T , i.e. $I = [i_1, i_2]$ with $i_1, i_2 \in T$ and $0 \leq i_1 \leq i_2$.

Example 2. Using Definition 2, we are able to describe the medical evolution of our patient in greater detail. Predicates in P capture information of a single moment in time. Formulae allow us to capture temporal patterns, such as

$$\begin{array}{ll} \boxminus_{[0,2]} (hrHigh \wedge bpLow) & \text{“high heart rate and low blood pressure occurred} \\ & \text{together at least once in the last 2 min” ,} \\ (alarm)\mathcal{S}_{[0,5]}(hrHigh \wedge bpLow) & \text{“the alarm has been on since the last joint occurrence of} \\ & \text{high heart rate and low blood pressure within 5 min” .} \end{array}$$

BMTL can be considered a fragment of “full” MTL, which also considers intervals of the form $[0, +\infty[$ in the subscript. A summary of BMTL and several other fragments and respective theoretical properties can be found in Ouaknine and Worrell [35].

Following a model-theoretic approach, we define the semantics of temporal formulae A by Table 1, with M an interpretation; t , t' and t'' time points on timeline T ; and I a positive bounded interval within T .

What can occur is that two formulae appear to be different but have the same semantics. An example thereof is $\boxplus_{[0,i]} A$ and $\mathcal{A}\mathcal{U}_{[i,i]} \top$; the first says A must hold from t (now) up to $t + i$. The latter says A must hold from t up until when $\top$ holds in interval $[t + i, t + i]$. Since $\top$ holds for any time point, it also does so for $t + i$, meaning the latter states A holds from t up until $t + i$. We refer to such formulae with identical semantics as *equivalent*, formally defined as follows:

Table 1. Semantics of formulae.

Atom	Semantics
Predicate	$M, t \models p$ if $p \in V(t)$
True	$M, t \models \top$ for each t
Negation	$M, t \models \neg A$ iff $M, t \models A$ does not hold
Conjunction	$M, t \models A_1 \wedge A_2$ iff $M, t \models A_1$ and $M, t \models A_2$
Once (past)	$M, t \models \Diamond_I A$ iff $M, t' \models A$ for at least one t' with $t - t' \in I$
Once (future)	$M, t \models \Diamond_I A$ iff $M, t' \models A$ for at least one t' with $t' - t \in I$
Always (past)	$M, t \models \Box_I A$ iff $M, t' \models A$ for all t' with $t - t' \in I$
Always (future)	$M, t \models \Box_I A$ iff $M, t' \models A$ for all t' with $t' - t \in I$
Since	$M, t \models A_1 \mathcal{S}_I A_2$ iff $M, t' \models A_2$ for at least one t' with $t - t' \in I$ and $M, t'' \models A_1$ for all $t'' \in [t', t]$
Until	$M, t \models A_1 \mathcal{U}_I A_2$ iff $M, t' \models A_2$ for at least one t' with $t' - t \in I$ and $M, t'' \models A_1$ for all $t'' \in [t, t']$

Definition 3. Two formulae A_1 and A_2 are considered to be equivalent if (and only if) for every model M and for every time point t , it holds that A_1 is true at t in M if A_2 is true at t in M , and vice versa. We denote this by writing $A_1 \equiv A_2$.

Example 3. To illustrate the notion of equivalence from Definition 3, we continue the healthcare monitoring scenario introduced above.

Consider the following two formulae:

$$A := \Box_{[0,2]} \text{hrHigh} \quad \text{and} \quad B := (\text{hrHigh}) \mathcal{S}_{[0,2]} \top.$$

Intuitively, both A and B express the same medical condition: “the patient’s heart rate has been continuously high throughout the last two minutes”. Formula A states this directly using the past “always” operator. Formula B expresses the same requirement using only the “since” operator: It looks for some time point in the last two minutes where $\top$ held. Since $\top$ always holds, the required time point is early as exactly two minutes ago. B then expresses that, since exactly two minutes ago, hrHigh has held up until now, without interruption.

For the model M specified earlier, we indeed have that for each $t' \in [8, 10]$, $M, t' \models \text{hrHigh}$, meaning that both A and B hold at $t = 10$. More generally, by evaluating both formulae over all models and time points, one sees that A is true at (M, t) if and only if B is true at (M, t) . Therefore, $A \equiv B$.

Table 1 does not specify the exact interpretation for “ $M, t \models A$ does not hold”. Each system can decide what is understood as “it does not hold”. As considered in the Introduction, closed systems often employ a negation-as-failure approach; when there is no conclusive evidence in favor of $M, t \models A$, it assumes $M, t \models A$ does not hold. Imagine, for example, the reservation list of a restaurant; if there is no evidence that you have a reservation, then the waiter assumes that you do not have a reservation. NAF is not always a desired approach, since it may lead to premature conclusions. Perhaps you do have a reservation, but the waiter’s list is outdated. In more open-ended systems, e.g., IoT or the Web, it is preferable to only say $M, t \models A$ does not hold when there is explicit evidence that says so. For example, you may not have a reservation for eight people, since there is specific evidence against it: your reservation is for two people.

3. Results

As indicated in Section 1, rewriting operators in function of other operators is a known practice. For example, Gutiérrez-Basulto et al. [28] reiterates the syntax of $\text{LTL}_{ACC}^{\text{bin}}$, where operations $\Diamond_I C$ and $\Box_I C$ are introduced merely as a shorthand for $\top \mathcal{U}_I C$ and $\neg(\Diamond_I \neg C)$.

Similarly, Finger and Gabbay [25] define their respective temporal operators solely in function of $\mathcal{U}$ and $\mathcal{S}$, i.e., $\mathbf{F}(A) := \mathcal{U}(A, \top)$ and $\mathbf{P}(A) := \mathcal{S}(A, \top)$ for propositional temporal logics. Koymans [27] notes the same relations hold in MTL: $\top \mathcal{U}_I A \equiv \Diamond_I A$ and $\top \mathcal{S}_I A \equiv \Diamond_I A$. These equivalencies migrate to other frameworks as well, as seen in DatalogMTL [16]. These results can be summarized as follows:

Proposition 1. Suppose $t \in T$, I is a non-negative interval, and A is a formula. Then, $\Diamond_I A \equiv \top \mathcal{U}_I A$ and $\Diamond_I A \equiv \top \mathcal{S}_I A$.

In Gutiérrez-Basulto et al. [28], the MTL operator $\boxplus A$ is defined via the equivalence $\boxplus A \equiv \neg(\Diamond \neg A)$. This formulation relies on the presence of negation to express “always A ”. However, in the absence of negation, such an equivalence cannot be reproduced within MTL, which raises the natural question: if we exclude negation $\neg$ and the “always” operators $\boxplus$ and $\boxminus$ from the syntax of MTL, does this result in a strictly less expressive fragment compared to the full language?

The following theorems and proofs demonstrate that the answer is negative: even without these operators, the ability of MTL to express temporal relations remains intact. This finding challenges the intuition that negation is essential for expressing universal temporal properties and shows that such constructs can be captured through alternative means.

Theorem 1. Suppose t is a time point, $I = [i_1, i_2]$ is a non-negative interval and A is a formula. Then,

$$\boxplus_I A \equiv \Diamond_{[i_1, i_1]} \left(A \mathcal{U}_{[i_2 - i_1, i_2 - i_1]} \top \right) \equiv \top \mathcal{U}_{[i_1, i_1]} \left(A \mathcal{U}_{[i_2 - i_1, i_2 - i_1]} \top \right).$$

Proof. Consider an arbitrary model M and time point t :

$$\begin{aligned} & M, t \models \Diamond_{[i_1, i_1]} \left(A \mathcal{U}_{[i_2 - i_1, i_2 - i_1]} \top \right) \\ \Leftrightarrow & \exists t' \in [t + i_1, t + i_1] : M, t' \models A \mathcal{U}_{[i_2 - i_1, i_2 - i_1]} \top \\ \Leftrightarrow & \exists t' \in [t + i_1, t + i_1] : \left(\exists t^{(2)} \in [t' + (i_2 - i_1), t' + (i_2 - i_1)] : \right. \\ & \left. (M, t^{(2)} \models \top) \wedge (\forall t^{(3)} \in [t', t^{(2)}] : M, t^{(3)} \models A) \right) \\ \Leftrightarrow & \exists t^{(2)} \in [t + i_2, t + i_2] : \left((t^{(2)} \models \top) \wedge (\forall t^{(3)} \in [t + i_1, t^{(2)}] : t^{(3)} \models A) \right) \\ \Leftrightarrow & \forall t^{(3)} \in [t + i_1, t + i_2] : t^{(3)} \models A \\ \Leftrightarrow & t \models \boxplus_I A \end{aligned}$$

The proof works as follows. In order for the equivalence to hold, the two formulae must hold (and not hold) simultaneously for every model and time point. If this can be proven for an arbitrary pair of model and time point, it must hold for every pair. Hence, we consider an unspecified model M and time point t for which $M, t \models \Diamond_{[i_1, i_1]} \left(A \mathcal{U}_{[i_2 - i_1, i_2 - i_1]} \top \right)$. Following the semantics of Section 2, there exists a future time point t' , such that $t' - t \in [i_1, i_1]$; in other words, $t' = t + i_1$, for which $M, t' \models A \mathcal{U}_{[i_2 - i_1, i_2 - i_1]} \top$. By the semantics of the $\mathcal{U}$ operator, this is equivalent to first the existence of a time point $t^{(2)}$ for which $t^{(2)} - t' \in [i_2 - i_1, i_2 - i_1]$; in other words, $t^{(2)} = t' + (i_2 - i_1)$. By substituting the fact that $t' = t + i_1$, we get $t^{(2)} = t + i_2$. For this $t^{(2)}$ $M, t^{(2)} \models \top$ holds. Second, it entails that for every $t^{(3)}$ between t' and $t^{(2)}$, it holds that $M, t^{(3)} \models A$. Since $\top$ holds for any M and $t^{(2)}$ and $t^{(2)}$ can only be $t + i_2$, this is equivalent to $M, t^{(3)} \models A$ for every $t^{(3)}$ in interval $[t + i_1, t + i_2]$. In other words,

$M, t \models \boxplus_{[i_1, i_2]} A$. It is easily verified that each step likewise holds in the opposite direction. Lastly, the second equivalence follows from Proposition 1 by substituting the $\boxplus$ operator. $\square$

We obtain an analogous result for $\boxminus$:

Theorem 2. Suppose t is a time point, $I = [i_1, i_2]$ is a non-negative interval and A is a formula. Then,

$$\boxminus_I A \equiv \boxplus_{[i_1, i_1]} \left(A \mathcal{S}_{[i_2 - i_1, i_2 - i_1]} \top \right) \equiv \top \mathcal{S}_{[i_1, i_1]} \left(A \mathcal{S}_{[i_2 - i_1, i_2 - i_1]} \top \right).$$

Following Theorems 1 and 2, any negation-free formula in BMTL has an equivalent formula free of operators $\boxplus$ and $\boxminus$. Combined with Proposition 1, a negation-free BMTL fragment can be defined using only $\mathcal{S}$ and $\mathcal{U}$:

$$A := p | \top | A \wedge A | A \mathcal{S}_I A | A \mathcal{U}_I A$$

In this fragment, the same temporal relations as in full MTL can be expressed. The only compromise is situated within the non-temporal relations; $A_1 \vee A_2$, often achieved via $\neg(\neg A_1 \wedge \neg A_2)$, cannot be expressed in negation-free BMTL.

Theorems 1 and 2 express $\boxplus$ and $\boxminus$ using singleton intervals such as $[i_1, i_1]$ and $[i_2 - i_1, i_2 - i_1]$. These punctual constraints are admissible in BMTL but are disallowed in MITL and its subfragments, where the syntax explicitly forbids singleton intervals. Therefore, to obtain analogous eliminations of $\boxplus$ and $\boxminus$ within Bounded MITL, we must develop alternative equivalences that avoid punctual constraints entirely. The next theorems provide such constructions. However, the restriction increases the intricacy of the matter, resulting in equivalencies and proofs that stray further from intuition than the previous theorems.

The idea of Theorem 3 is based on the following observation: $\boxplus_{[i_1, i_2]} A$ at time t means A holds over interval $[t + i_1, t + i_2]$, meaning A started somewhere before, or at the latest at $t + i_1$, and end no earlier than $t + i_2$. We split this expression into two parts, one that states A starts holding at the latest at $t + i_1$ and another that states A stops holding at the earliest at $t + i_2$, with the condition that the two conditions must overlap to ensure that A holds continuously over the entire interval. For the first part, we postulate that A needs to hold for at least a length $i_2 - i_1$, starting somewhere between $t + i_1 - \frac{i_2 - i_1}{2}$ and $t + i_1$, meaning A will hold until at least $t + i_1 + \frac{i_2 - i_1}{2}$, the halfway point of the interval $[t + i_1, t + i_2]$. Likewise, the second part states that A should hold for $i_2 - i_1$ time points, since at least $t + i_1 + \frac{i_2 - i_1}{2}$. These two parts are illustrated in Figure 1.

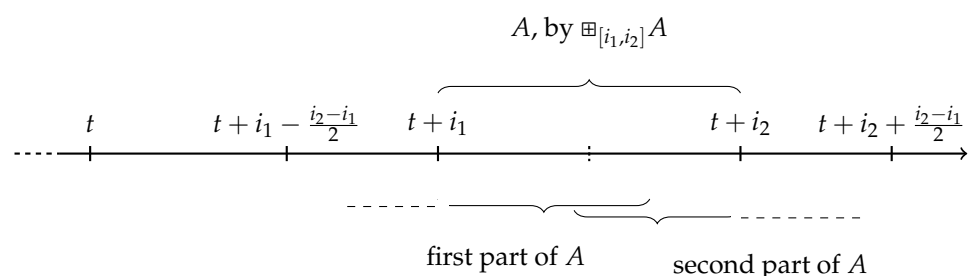


Figure 1. Illustration of Proof 1. The accolades of parts 1 and 2 have a set length of $i_2 - i_1$ but shift according to the exact location of t^- and t^+ . Regardless of the exact location of t^- and t^+ , the two accolades will always produce an area spanning at least $[t + i_1, t + i_2]$, in which A holds everywhere.

To clarify the construction further, we explicitly indicate the extremal endpoints enforced by each conjunct in Theorem 3. The first conjunct ensures that every witness t^- satisfies

$$t^- \leq t + i_1 \quad \text{and} \quad t^- + (i_2 - i_1) \geq t + i_1 + \frac{i_2 - i_1}{2},$$

so the minimal right endpoint covered is exactly $t + i_1 + \frac{i_2 - i_1}{2}$ and the maximal left endpoint is $t + i_1$. Symmetrically, the second conjunct enforces that every t^+ satisfies

$$t^+ \geq t + i_2 \quad \text{and} \quad t^+ - (i_2 - i_1) \leq t + i_1 + \frac{i_2 - i_1}{2}.$$

Thus, its maximal left endpoint is $t + i_1 + \frac{i_2 - i_1}{2}$ and its minimal right endpoint is $t + i_2$. Together, these yield exactly the interval $[t + i_1, t + i_2]$ as the portion necessarily covered by A .

In the following proof, we show that these two parts translate to expressions without box operators or singleton intervals.

Theorem 3. Suppose t is a time point, $I = [i_1, i_2]$ is a non-negative interval, κ and λ are positive time points and A is a formula. Then,

$$\boxplus_{[i_1, i_2]} A \equiv \left(\boxplus_{[\frac{3i_1 - i_2}{2}, i_1]} (A\mathcal{U}_{[i_2 - i_1, i_2 - i_1 + \kappa]} \top) \right) \wedge \left(\boxplus_{[i_2, \frac{3i_2 - i_1}{2}]} (A\mathcal{S}_{[i_2 - i_1, i_2 - i_1 + \lambda]} \top) \right)$$

Proof. For clarity, we structure the argument as two implications:

$$\boxplus_{[i_1, i_2]} A \Rightarrow \Phi \quad \text{and} \quad \Phi \Rightarrow \boxplus_{[i_1, i_2]} A,$$

where Φ denotes the conjunction of the two interval-indexed $\boxplus\mathcal{U}$ and $\boxplus\mathcal{S}$ components.

Let κ and λ be arbitrary but fixed positive time points. We consider the following two formulae:

$$\boxplus_{[\frac{3i_1 - i_2}{2}, i_1]} (A\mathcal{U}_{[i_2 - i_1, i_2 - i_1 + \kappa]} \top) \tag{1}$$

$$\boxplus_{[i_2, \frac{3i_2 - i_1}{2}]} (A\mathcal{S}_{[i_2 - i_1, i_2 - i_1 + \lambda]} \top) \tag{2}$$

If for a given model M , Formula (1) holds at time t in M , semantics assert that there exists a time point within the interval $[t + \frac{3i_1 - i_2}{2}, t + i_1]$, say t^- , at which the formula $A\mathcal{U}_{[i_2 - i_1, i_2 - i_1 + \kappa]} \top$ holds. This means that, starting from t^- , A must hold continuously for a duration of at least $i_2 - i_1$ time units, until a point is reached where the trivially true formula $\top$ holds in M .

Dually, Formula (2) captures a symmetric condition. It states that if the Formula (2) holds in M at time t , there exists a time point t^+ within the interval $[t + i_2, t + \frac{3i_2 - i_1}{2}]$ such that the formula $A\mathcal{S}_{[i_2 - i_1, i_2 - i_1 + \lambda]} \top$ holds in M . This implies that, looking backward from t^+ , the proposition A must have held continuously for a duration of at least $i_2 - i_1$, starting from a point where $\top$ is satisfied.

Observe that the lower bound of the interval $[t + \frac{3i_1 - i_2}{2}, t + i_1]$ can be rewritten as

$$\begin{aligned} t + \frac{3i_1 - i_2}{2} &= t + \frac{3i_1}{2} - \frac{i_2}{2} = t + i_1 + \frac{i_1}{2} - \frac{i_2}{2} \\ &= t + i_1 + \frac{i_1 - i_2}{2} = t + i_1 - \frac{i_2 - i_1}{2} \end{aligned}$$

This latter expression identifies the time point that lies exactly one interval length of $i_2 - i_1$ before the halfway point of the interval $[t + i_1, t + i_2]$.

From any time point within the range $[t + i_1 - \frac{i_2 - i_1}{2}, t + i_1]$, the formula in Equation (1) requires a forward temporal progression (a “jump”) of at least $i_2 - i_1$ time units. This jump begins no later than $t + i_1$ and, in the minimal case permitted by the interval constraints, lands at $t + i_1 - \frac{i_2 - i_1}{2} + (i_2 - i_1) = t + i_1 + \frac{i_2 - i_1}{2}$, i.e., the midpoint of the interval $[t + i_1, t + i_2]$. Since the proposition A is required to hold throughout the entire duration of this jump, it follows that A must hold continuously over at least the first half of the interval $[t + i_1, t + i_2]$, that is, from $t + i_1$ to $t + i_1 + \frac{i_2 - i_1}{2}$. It is important to note that this condition provides no information on the truth value of A prior to $t + i_1$, nor beyond $t + i_1 + \frac{i_2 - i_1}{2}$.

The same method of interpretation applies to Formula (2). In this case, the upper bound interval can be similarly rewritten, i.e.,

$$t + \frac{3i_2 - i_1}{2} = t + \frac{3i_2}{2} - \frac{i_1}{2} = t + i_2 + \frac{i_2}{2} - \frac{i_1}{2} = t + i_2 + \frac{i_2 - i_1}{2}$$

From any time point within the range $[t + i_2, t + i_2 + \frac{i_2 - i_1}{2}]$, the formula in Equation (2) requires a backwards “jump” of at least $i_2 - i_1$ time units. As such, this jump back ends at the latest at $t + i_2$ and starts at the latest at $t + i_2 + \frac{i_2 - i_1}{2} - (i_2 - i_1) = t + i_2 - \frac{i_2 - i_1}{2}$, i.e., the midpoint of the interval $[t + i_1, t + i_2]$. Again, A must continuously hold during this jump. As such, A holds continuously from $t + i_2 - \frac{i_2 - i_1}{2}$ to $t + i_2$, which is the latter half of the interval $[t + i_1, t + i_2]$.

As a result, if the conjunction of (1) and (2) holds for a model M at time point t , then A holds in M over (at least) the interval $[t + i_1, t + i_2]$, i.e., $M, t \models \boxplus_{[i_1, i_2]} A$. Crucially, the endpoints we use are minimal and maximal in the following precise sense: any earlier start would shift the $\mathcal{U}$ -jump too far left and fail to reach the midpoint, and any later start would be outside the allowed interval for the witness. Likewise, any later end for the $\mathcal{S}$ -jump would fail to reach back to the midpoint. Therefore, no strictly larger interval than $[t + i_1, t + i_2]$ is enforced by the construction.

Suppose that $M, t \models \boxplus_{[i_1, i_2]} A$. By the semantics of the $\boxplus$ operator, A holds at every time point within the interval $[t + i_1, t + i_2]$. Furthermore, since $\top$ is valid at all time points, it trivially holds that $M, t + i_2 \models \top$. Observe that $t + i_2$ lies within the interval $[t + i_2, t + i_2 + \kappa]$, and thus the condition $A\mathcal{U}_{[i_2 - i_1, i_2 - i_1 + \kappa]} \top$ is satisfied at time $t + i_1$. Moreover, since $t + i_1$ belongs to the interval $[t + \frac{3i_1 - i_2}{2}, t + i_1]$, it follows that Formula (1) holds at time t .

We now turn our attention to Equation (2). Note that $t + i_1$ lies within the interval $[t + i_1 - \lambda, t + i_1]$, which can be equivalently expressed as $[(t + i_2) - (i_2 - i_1 + \lambda), (t + i_2) - (i_2 - i_1)]$. Given that A holds throughout $[t + i_1, t + i_2]$ and $\top$ holds at $t + i_2$, it follows that $M, t + i_2 \models A\mathcal{S}_{[i_2 - i_1, i_2 - i_1 + \lambda]} \top$, and consequently, Formula (2) holds in M at time t . Since $t + i_2$ lies within the interval $[t + i_2, t + \frac{3i_2 - i_1}{2}]$, we conclude that there exists a time point t^+ in the interval $[t + i_2, t + \frac{3i_2 - i_1}{2}]$ for which $A\mathcal{S}_{[i_2 - i_1, i_2 - i_1 + \lambda]} \top$ holds in M at time t^+ . As a result, Formula (2) holds in M at time point t .

$M, t \models \boxplus_{[i_1, i_2]} A$ thus induces that both Formulae (2) and (1) hold in M at time t , and therefore their conjunction as well.

□

Analogously, $\boxminus_{[i_1, i_2]} A$ can be rewritten into a similar expression.

Theorem 4. Suppose $t \in T$, $I = [i_1, i_2]$ is a non-negative interval, κ, λ are positive time points and A is a formula. Then,

$$\boxminus_{[i_1, i_2]} A \equiv \left(\boxplus_{[\frac{3i_1 - i_2}{2}, i_1]} (AS_{[i_2 - i_1, i_2 - i_1 + \lambda]} \top) \right) \wedge \left(\boxplus_{[i_2, \frac{3i_2 - i_1}{2}]} (AU_{[i_2 - i_1, i_2 - i_1 + \kappa]} \top) \right)$$

Proof. The proof of Theorem 4 is analogous to that of Theorem 3. $\square$

It follows from Theorems 3 and 4 that any formula with bounded interval subscripts can be rewritten into formulae containing only the temporal operators $\boxplus$, $\boxminus$, $\mathcal{S}$ and $\mathcal{U}$. Combined with Proposition 1, we conclude that negation-free BMTL and Bounded MITL can be defined using only temporal operators $\mathcal{S}$ and $\mathcal{U}$ without compromising on temporal expressivity compared to the full BMTL and Bounded MITL respectively.

4. Discussion

The results above lead to a surprising and counterintuitive conclusion: even without negation, the universal temporal operators can be eliminated. While classical transformations of “always” operators into “once” and negation are well-known in temporal logic, our findings reveal that similar reductions are possible even in negation-free settings. This challenges the common intuition that negation is essential for expressing “always” statements. Given the influence of MTL on recent temporal reasoning frameworks, the obtained results can be carried over to MTL-based frameworks such as MEL and DatalogMTL.

The constructions following Theorems 1 and 2 rely on punctual intervals, i.e., singleton bounds of the form $[i, i]$. While these are admissible in BMTL, their use is well known to introduce subtle semantic and computational issues. In dense time domains, punctual requirements make formula satisfaction fragile: an event must occur at exactly one time point, so even infinitesimal timestamp perturbations may invalidate the formula. This lack of robustness has been repeatedly highlighted in the MTL literature. More importantly, as shown by Alur et al. [34] and by Ouaknine and Worrell [35], allowing punctual constraints renders satisfiability and model checking for MTL undecidable. This motivated the development of MITL, which explicitly excludes singleton intervals in order to recover decidability. Consequently, while Theorems 1 and 2 establish that $\boxplus$ and $\boxminus$ can be eliminated within negation-free BMTL using punctual constraints, these transformations are not applicable to MITL or its well-behaved subfragments. This observation motivates the alternative equivalences presented in Theorems 3 and 4. These results show that, even under the stricter MITL syntax, the universal operators remain eliminable, albeit via more intricate constructions. This not only preserves expressive completeness in the absence of punctuality but also aligns the syntactic simplification results with the well-studied decidable fragments of MTL.

Looking forward, the syntactical simplifications lead to several avenues for future research. From a theoretical perspective, it is an open question whether analogous syntactic reductions can be achieved in temporal reasoning frameworks with different temporal operators or semantics—for example, LARS. This would further prompt us to reconsider the theoretical minimal requirements of temporal logic frameworks for open-world infrastructures such as IoT and the Semantic Web. These theoretical results help us better understand the strengths and weaknesses of stream reasoning systems, which in turn allows us to better target research efforts toward more efficient, scalable, and predictable reasoning systems for dynamic and data-intensive environments.

Approached from a more practical angle, the expressions obtained in this paper can be leveraged in MTL-based frameworks; via the obtained expressions, the more complex temporal operators $\boxplus$, $\boxminus$, $\boxplus$, $\boxminus$ can be implemented as “built-ins” on top of the existing theory to improve readability (and thereby user convenience) without modifying the un-

derlying theory. Finally, an interesting direction for future work is to explore whether similar syntactic simplifications can be extended to unbounded intervals (i.e., $[0, +\infty[$). In the present study, the key idea relies on reconstructing a bounded interval by covering it from both endpoints. This approach fundamentally presupposes boundedness, since there is no “right endpoint” in intervals such as $[0, +\infty)$. Extending the method would therefore require qualitatively different techniques—likely ones that reason about asymptotic behavior or employ limit-style constraints rather than interval reconstruction. Exploring such alternative approaches opens promising possibilities for broadening the applicability of the results presented here.

5. Conclusions

In this work, we have shown that the syntax of negation-free MTL with bounded intervals can be significantly simplified without compromising its expressive power. Specifically, we demonstrated that all occurrences of the temporal operators $\boxplus$ and $\boxminus$ can be syntactically eliminated in favor of combinations of $\Diamond$, $\Diamond$, $\mathcal{U}$, and $\mathcal{S}$. This is the case for negation-free BMTL, as follows from Theorems 1 and 2, but also for the more restrictive Bounded MITL, as illustrated by Theorems 3 and 4. Furthermore, we established that even the “once” operators $\Diamond$ and $\Diamond$ can be removed, yielding a fragment that relies solely on the $\mathcal{S}$ and $\mathcal{U}$ operators.

Author Contributions: Conceptualization, M.v.N.; methodology, M.v.N., F.O. and P.B.; formal analysis, M.v.N.; investigation, M.v.N.; writing—original draft, M.v.N.; writing—review & editing, F.O. and P.B.; supervision, F.O. and P.B.; project administration, F.O. and P.B.; funding acquisition, F.O. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by the FRACTION project (Nr. G086822N), funded by the Fonds voor Wetenschappelijk Onderzoek (FWO) organization.

Data Availability Statement: No new data were created or analyzed in this study.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Baldazzi, T.; Bellomarini, L.; Sallinger, E. *Reasoning over Financial Scenarios with the Vadalog System*; OpenProceedings.org: Konstanz, Germany, 2023; Volume 26, pp. 782–791.
2. Doherty, P.; Kvarnstrom, J.; Wzorek, M.; Rudol, P.; Heintz, F.; Conte, G. Hdrc3: A Distributed Hybrid Deliberative/Reactive Architecture for Unmanned Aircraft Systems. In *Handbook of Unmanned Aerial Vehicles*; Springer Science+Business Media B.V.: Dordrecht, The Netherlands, 2015; pp. 849–952.
3. De Brouwer, M.; Steenwinkel, B.; Fang, Z.; Stojchevska, M.; Bonte, P.; De Turck, F.; Van Hoecke, S.; Ongenae, F. Context-Aware and Privacy-Preserving Homecare Monitoring through Adaptive Query Derivation for IoT Data Streams with DIVIDE. *Semant. Web J.* **2019**.
4. Abiteboul, S.; Hull, R.; Vianu, V. *Foundations of Databases*; Addison-Wesley: Boston, MA, USA, 1995.
5. Hayes, P. Re: Why Must the Web Be Monotonic? from Pat Hayes on 2001-07-24 (www-rdf-logic@w3.org from July 2001). Available online: <https://lists.w3.org/Archives/Public/www-rdf-logic/2001Jul/0067.html> (accessed on 16 August 2023).
6. Hochstenbach, P.; van Noort, M.; Arndt, D.; Martens, R.; De Roo, J.; Verborgh, R.; Bonte, P.; Ongenae, F. RDF Surfaces: Enabling Classical Negation on the Semantic Web. *arXiv* **2024**, arXiv:2406.10659. [[CrossRef](#)]
7. Marinier, P.; Anepu, B.M.; Pelletier, G.; Olesen, R.L. Maintaining Time Alignment with Multiple Uplink Carriers, 2015. U.S. Patent 8,934,459, 13 January 2015.
8. Tu, D.Q.; Kayes, A.; Rahayu, W.; Nguyen, K. IoT Streaming Data Integration from Multiple Sources. *Computing* **2020**, *102*, 2299–2329. [[CrossRef](#)]
9. Bonte, P.; Callé, C.; Curé, O.; Kondylakis, H.; Tommasini, R. Languages and Systems for RDF Stream Processing, a Survey. *VLDB J.* **2025**, *34*, 50. [[CrossRef](#)]
10. Bonte, P.; Calbimonte, J.P.; de Leng, D.; Dell’Aglia, D.; Della Valle, E.; Eiter, T.; Giannini, F.; Heintz, F.; Schekotihin, K.; Le-Phuoc, D.; et al. Grounding Stream Reasoning Research. *Trans. Graph Data Knowl. (TGDK)* **2024**, *2*, 1–47.

11. Santipantakis, G.M.; Vlachou, A.; Doukeridis, C.; Artikis, A.; Kontopoulos, I.; Vouros, G.A. A Stream Reasoning System for Maritime Monitoring. In *Proceedings of the 25th International Symposium on Temporal Representation and Reasoning (TIME 2018)*; Schloss Dagstuhl–Leibniz-Zentrum für Informatik: Wadern, Germany, 2018; pp. 20:1–20:17.
12. Bellomarini, L.; Blasi, L.; Nissl, M.; Sallinger, E. The Temporal Vatalog System: Temporal Datalog-Based Reasoning. *Theory Pract. Log. Program.* **2025**, *25*, 168–196. [\[CrossRef\]](#)
13. Dehkordi, A.H.; Alizadeh, M.; Movaghar, A. Linear Temporal Public Announcement Logic: A New Perspective for Reasoning the Knowledge of Multi-Classifiers. *arXiv* **2020**, arXiv:2009.03793.
14. De Leng, D.; Heintz, F. Qualitative Spatio-Temporal Stream Reasoning with Unobservable Intertemporal Spatial Relations using Landmarks. In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI Press: Palo Alto, CA, USA, 2016; Volume 30.
15. Beck, H.; Dao-Tran, M.; Eiter, T. LARS: A Logic-Based Framework for Analytic Reasoning over Streams. *Artif. Intell.* **2018**, *261*, 16–70. [\[CrossRef\]](#)
16. Brandt, S.; Güzel Kalaycı, E.; Zakharyashev, M.; Xiao, G. Querying Log Data with Metric Temporal Logic. *Artif. Intell. Res.* **2018**, *62*, 829–877. [\[CrossRef\]](#)
17. Della Valle, E.; Ceri, S.; Van Harmelen, F.; Fensel, D. It's a Streaming World! Reasoning upon Rapidly Changing Information. *IEEE Intell. Syst.* **2009**, *24*, 83–89. [\[CrossRef\]](#)
18. Wałęga, P.A.; Cucala, D.J.T.; Kostylev, E.V.; Grau, B.C. DatalogMTL with Negation Under Stable Models Semantics. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, Online Event, 3–12 November 2021; Volume 18, pp. 609–618.
19. Ketsman, B.; Koch, C. Datalog with Negation and Monotonicity. In *Proceedings of the 23rd International Conference on Database Theory (ICDT 2020)*; Schloss-Dagstuhl-Leibniz Zentrum für Informatik: Wadern, Germany, 2020.
20. Cucala, D.J.T.; Wałęga, P.A.; Grau, B.C.; Kostylev, E. Stratified Negation in Datalog with Metric Temporal Operators. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Virtually, 19–21 May 2021; Volume 35, pp. 6488–6495.
21. Trakhtenbrot, B. On Recursive Separability. *Dokl. Akad. Nauk. SSSR* **1953**, *88*, 953–956.
22. Urbani, J.; Krötzsch, M.; Eiter, T. Chasing Streams with Existential Rules. In *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning, KR 2022*. International Joint Conferences on Artificial Intelligence Organization, Haifa, Israel, 31 July–5 August 2022; pp. 415–419.
23. Bouyer, P.; Chevalier, F.; Markey, N. On the Expressiveness of TPTL and MTL. *Inf. Comput.* **2010**, *208*, 97–116. [\[CrossRef\]](#)
24. Basin, D.; Klaedtke, F.; Müller, S.; Zălinescu, E. Monitoring Metric First-Order Temporal Properties. *J. ACM* **2015**, *62*, 15. [\[CrossRef\]](#)
25. Finger, M.; Gabbay, D.M. Adding a Temporal Dimension to a Logic System. *J. Logic Lang. Inf.* **1992**, *1*, 203–233. [\[CrossRef\]](#)
26. Beck, H.; Dao-Tran, M.; Eiter, T.; Fink, M. LARS: A Logic-Based Framework for Analyzing Reasoning over Streams. *Proc. AAAI Conf. Artif. Intell.* **2015**, *29*. [\[CrossRef\]](#)
27. Koymans, R. Specifying Real-time Properties with Metric Temporal Logic. *Real-Time Syst.* **1990**, *2*, 255–299. [\[CrossRef\]](#)
28. Gutiérrez-Basulto, V.; Jung, J.C.; Ozaki, A. On Metric Temporal Description Logics. In *Proceedings of the Twenty-second European Conference on Artificial Intelligence*, The Hague, The Netherlands, 29 August–2 September 2016; pp. 837–845.
29. Kulpeshov, B.S. Algebras of binary formulas for weakly circularly minimal theories with trivial definable closure. *Sib. Math. J.* **2024**, *65*, 316–327. [\[CrossRef\]](#)
30. Kumduang, T.; Sriwongsa, S.; Wattanatripop, K. Combinatorial results to measure the complexity of quantifier free formulas in algebraic systems. *Asian-Eur. J. Math.* **2025**, 2550126. [\[CrossRef\]](#)
31. Denecke, K. Partial clones. *Asian-Eur. J. Math.* **2020**, *13*, 2050161. [\[CrossRef\]](#)
32. Tiger, M.; Heintz, F. Stream Reasoning using Temporal Logic and Predictive Probabilistic State Models. In *Proceedings of the 2016 23rd International Symposium on Temporal Representation and Reasoning (TIME)*; IEEE: New York, NY, USA, 2016; pp. 196–205.
33. Cabalar, P.; Diéguez, M.; Schaub, T.; Schuhmann, A. Metric Temporal Answer Set Programming over Timed Traces. In *Proceedings of the International Conference on Logic Programming and Nonmonotonic Reasoning*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 117–130.
34. Alur, R.; Feder, T.; Henzinger, T.A. The Benefits of Relaxing Punctuality. *J. ACM* **1996**, *43*, 116–146. [\[CrossRef\]](#)
35. Ouaknine, J.; Worrell, J. Some Recent Results in Metric Temporal Logic. In *Proceedings of the International Conference on Formal Modeling and Analysis of Timed Systems*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 1–13.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.