# Theorem Proving for Metric Temporal Logic over the Naturals

Ullrich Hustadt[1], Ana Ozaki[2], and Clare Dixon[1]

[1] Department of Computer Science, University of Liverpool, UK
{cldixon,uhustadt}@liverpool.ac.uk
[2] Faculty of Computer Science, TU Dresden
Ana.Ozaki@tu-dresden.de

**Abstract** We study translations from Metric Temporal Logic (MTL) over the natural numbers to Linear Temporal Logic (LTL). In particular, we present two approaches for translating from MTL to LTL which preserve the ExpSpace complexity of the satisfiability problem for MTL. In each of these approaches we consider the case where the mapping between states and time points is given by (1) a strict monotonic function and by (2) a non-strict monotonic function (which allows multiple states to be mapped to the same time point). Our translations allow us to utilise LTL solvers to solve satisfiability and we empirically compare the translations, showing in which cases one performs better than the other.

## 1 Introduction

Linear and branching-time temporal logics have been used for the specification and verification of reactive systems. In linear-time temporal logic [19,11] we can, for example, express that a formula $\psi$ holds now or at some point in the future using the formula $\Diamond\psi$ ($\psi$ holds eventually). However, some applications require not just that a formula $\psi$ will hold eventually but that it holds within a particular time-frame, for example, between 3 and 7 moments from now.

To express such constraints, a range of Metric Temporal Logics (MTL) have been proposed [3,4], considering different underlying models of time and operators allowed. MTL has been used to formalise vehicle routing problems [17], monitoring of algorithms [22] and cyber-physical systems [1], among others [15]. A survey about MTL and its fragments can be found in [18]. It is known that MTL over the reals is undecidable, though, decidable fragments have been investigated [6,2,5]. Here we consider MTL with pointwise semantics over the natural numbers, following [3], where each state in the sequence is mapped to a time point on a time line isomorphic to the natural numbers. In this instance of MTL, temporal operators are annotated with intervals, which can be finite or infinite. For

example, $\Diamond_{[3,7]}$ means that $p$ should hold in a state that occurs in the interval $[3, 7]$ of time, while $\Box_{[2,\infty)}p$ means that $p$ should hold in all states that occur at least 2 moments from now. In contrast to LTL, where the time difference from one state to the next is always 1, in MTL, time is allowed to irregularly 'jump' from one state to the next. For example, using $\bigcirc_{[2,2]}p$ we can state that the time difference from the current state to the next state is 2. Furthermore, following Alur and Henzinger [3], the mapping between states and time points is given by a (weakly) monotonic function, which allows multiple states to be mapped to the same time point. In this work, we also consider the semantics where the mapping between states and time points is given by a strictly monotonic function, which forces time to progress from one state to another.

We provide two approaches for translating from MTL to LTL: in the first approach we introduce a fresh propositional variable that we call 'gap', which is used to encode the 'jumps' between states, as mentioned above; the second approach is inspired by [3], where fresh propositional variables encode time differences between states. In each approach we consider the case where the mapping between states and time points is given by (1) a strict monotonic function and by (2) a non-strict monotonic function (which allows multiple states to be mapped to the same time point). All translations are polynomial in the size of the MTL formula and the largest constant occurring in an interval (although exponential in the size of the MTL formula due to the binary encoding of the constants). Since the satisfiability problem for LTL is PSpace-complete [21], our translations preserve the ExpSpace complexity of the MTL satisfiability problem over the natural numbers [3].

Using these translations from MTL to LTL, we apply three temporal solvers, one resolution based [16], one tableau based [13] and the other based on the model checker NuSMV [7] to investigate the properties of the resulting formulae experimentally. To the best of our knowledge, there are no implementations of solvers for MTL with pointwise discrete semantics. In particular, our contributions are:

- translations from MTL to LTL which preserve the ExpSpace complexity of the MTL satisfiability problem;
- an experimental analysis of the behaviour of LTL solvers on the resulting formulae;
- to exemplify which kind of problems can be solved using MTL we also provide encodings of the classical Multiprocessor Job-Shop Scheduling problem [14,8] into MTL.

In the following we provide the syntax and semantics of LTL and MTL, show our translations from MTL to LTL and experimental results.

## 2 Preliminaries

We briefly state the syntax and semantics of LTL and MTL. Let $\mathcal{P}$ be a (countably infinite) set of propositional variables. Well formed formulae in LTL are formed according to the rule:

$$\varphi, \psi \quad := \quad p \mid \neg\varphi \mid (\varphi \wedge \psi) \mid \bigcirc\varphi \mid (\varphi \mathcal{U} \psi)$$

where $p \in \mathcal{P}$. We often omit parentheses if there is no ambiguity. We denote by $\bigcirc^c$ a sequence of $c$ next operators, i.e., $\bigcirc^0\varphi = \varphi$ and $\bigcirc^{n+1}\varphi = \bigcirc\bigcirc^n\varphi$, for every $n \in \mathbb{N}$.

An *LTL model* or *state sequence* $\sigma$ over $(\mathbb{N}, <)$ is an infinite sequence of states $\sigma_i \subseteq \mathcal{P}$, $i \in \mathbb{N}$. The semantics of LTL is defined as follows.

$(\sigma, i) \models p$ iff $p \in \sigma_i$
$(\sigma, i) \models (\varphi \wedge \psi)$ iff $(\sigma, i) \models \varphi$ and $(\sigma, i) \models \psi$
$(\sigma, i) \models \neg\varphi$ iff $(\sigma, i) \not\models \varphi$
$(\sigma, i) \models \bigcirc\varphi$ iff $(\sigma, i+1) \models \varphi$
$(\sigma, i) \models (\varphi \mathcal{U} \psi)$ iff $\exists k \geq i : (\sigma, k) \models \psi$ and $\forall j,\ i \leq j < k : (\sigma, j) \models \varphi$

Further connectives can be defined as usual: $\mathbf{true} \equiv p \vee \neg p$, $\mathbf{false} \equiv \neg(\mathbf{true})$, $\Diamond\varphi \equiv \mathbf{true}\mathcal{U}\varphi$ and $\Box\varphi \equiv \neg\Diamond\neg\varphi$. MTL formulae are constructed in a way similar to LTL, with the difference that temporal operators are now bounded by an interval $I$ with natural numbers as end-points or $\infty$ on the right side. Note that since we work with natural numbers as end-points we can assume w.l.o.g that all our intervals are of the form $[c_1, c_2]$ or $[c_1, \infty)$, where $c_1, c_2 \in \mathbb{N}$. Well formed formulae in MTL are formed according to the rule:

$$\varphi, \psi \quad := \quad p \mid \neg\varphi \mid (\varphi \wedge \psi) \mid \bigcirc_I\varphi \mid (\varphi \mathcal{U}_I \psi)$$

where $p \in \mathcal{P}$. A *timed state sequence* $\rho = (\sigma, \tau)$ over $(\mathbb{N}, <)$ is a pair consisting of an infinite sequence $\sigma$ of states $\sigma_i \subseteq \mathcal{P}$, $i \in \mathbb{N}$, and a function $\tau : \mathbb{N} \to \mathbb{N}$ that maps every $i$ corresponding to the $i$-th state to a time point $\tau(i)$ such that $\tau(i) < \tau(i+1)$. A *non-strict* timed state sequence $\rho = (\sigma, \tau)$ over $(\mathbb{N}, <)$ is a pair consisting of an infinite sequence $\sigma$ of states $\sigma_i \subseteq \mathcal{P}$, $i \in \mathbb{N}$, and a function $\tau : \mathbb{N} \to \mathbb{N}$ that maps every $i$ corresponding to the $i$-th state to a time point $\tau(i)$ such that $\tau(i) \leq \tau(i+1)$. We assume w.l.o.g. that $\tau(0) = 0$. The semantics of MTL is defined as follows (we omit propositional cases, which are as in LTL).

$$
\begin{array}{ll}
(\rho, i) \models \bigcirc_I \varphi & \text{iff } (\rho, i+1) \models \varphi \text{ and } \tau(i+1) - \tau(i) \in I \\
(\rho, i) \models (\varphi \mathcal{U}_I \psi) & \text{iff } \exists k \geq i : \tau(k) - \tau(i) \in I \text{ and } (\rho, k) \models \psi \\
 & \qquad \text{and } \forall j,\ i \leq j < k : (\rho, j) \models \varphi
\end{array}
$$

Further connectives can be defined as usual: $\Diamond_I \varphi \equiv \mathbf{true} \mathcal{U}_I \varphi$ and $\Box_I \varphi \equiv \neg \Diamond_I \neg \varphi$. To transform an MTL formula into Negation Normal Form, one uses the constrained dual until $\tilde{\mathcal{U}}_I$ operator [18], defined as $(\varphi \tilde{\mathcal{U}}_I \psi) \equiv \neg(\neg\varphi \mathcal{U}_I \neg\psi)$.

An MTL formula $\varphi$ is in *Negation Normal Form (NNF)* iff the negation operator ($\neg$) occurs only in front of propositional variables. One of the differences between MTL and LTL is that in LTL we have the equivalence $\neg(\bigcirc p) \equiv \bigcirc \neg p$, whereas in MTL $\neg(\bigcirc_{[2,2]} p) \not\equiv \bigcirc_{[2,2]} \neg p$. If $\neg(\bigcirc_{[2,2]} p)$ then either $p$ does not occur in the next state or the next state does not occur with time difference 2. We can express this as follows: $\neg(\bigcirc_{[2,2]} p) \equiv \bigcirc_{[2,2]} \neg p \vee \bigcirc_{[0,1]} \mathbf{true} \vee \bigcirc_{[3,\infty)} \mathbf{true}$.

An MTL formula $\varphi$ is in *Flat Normal Form (FNF)* iff it is of the form $p_0 \wedge \bigwedge_i \Box_{[0,\infty)}(p_i \rightarrow \psi_i)$ where $p_0, p_i$ are propositional variables or **true** and $\psi_i$ is either a formula of propositional logic or it is of the form $\bigcirc_I \psi_1$, $\psi_1 \mathcal{U}_I \psi_2$ or $\psi_1 \tilde{\mathcal{U}}_I \psi_2$ where $\psi_1, \psi_2$ are formulae of propositional logic.

One can transform an MTL formula into FNF by renaming subformulae with nested operators, as in [10,23]. For example, assume that we are given the following MTL formula: $\bigcirc_{[2,3]}(\neg \Box_{[1,2]} q)$. We first transform our formula into NNF and obtain: $\bigcirc_{[2,3]}(\Diamond_{[1,2]} \neg q)$. We then transform it into FNF: $p_0 \wedge \Box_{[0,\infty)}(p_0 \rightarrow \bigcirc_{[2,3]} p_1) \wedge \Box_{[0,\infty)}(p_1 \rightarrow \Diamond_{[1,2]} \neg q)$. The transformations into NNF and FNF are satisfiability preserving and can be performed in polynomial time.

## 3 From MTL to LTL: encoding 'gaps'

Assume that our MTL formulae are in NNF and FNF. The main idea for our approach is to map each timed state sequence $\rho = (\sigma, \tau)$ to a state sequence $\sigma'$ such that $\rho = (\sigma, \tau)$ is a model of an MTL formula if, and only if, $\sigma'$ is a model of our LTL translation. We first present our translation using the strict semantics and then show how to adapt it for the non-strict semantics, where multiple states are allowed to be mapped to the same time point.

**Strict Semantics** We translate MTL formulae for discrete time models into LTL using a new propositional variable *gap*. $\neg gap$ is true in those states $\sigma'_j$ of $\sigma'$ such that there is $i \in \mathbb{N}$ with $\tau(i) = j$ and *gap* is true in all

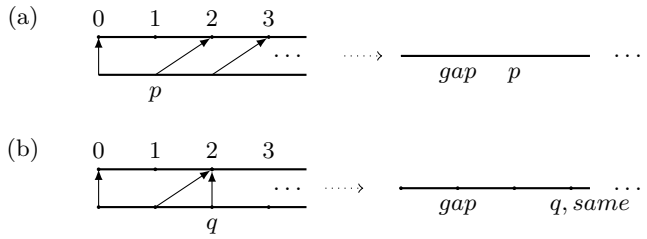


Figure 1: Examples illustrating Definitions 1 and 3

other states of $\sigma'$. We now define our mappings between MTL and LTL models.

**Definition 1.** *Given a timed state sequence $\rho = (\sigma, \tau)$, we define $\sigma' = \sigma'_0\sigma'_1\ldots$, where $\sigma'_j$ is as follows:*

$$\sigma'_j = \begin{cases} \sigma_i & \text{if there is } i \in \mathbb{N} \text{ such that } \tau(i) = j; \\ \{gap\} & \text{otherwise.} \end{cases}$$

Figure 1(a) illustrates the mapping given by Definition 1. For instance, if $\rho = (\sigma, \tau)$ is the timed state sequence on the left side of Figure 1(a) then $(\rho, 0) \models \bigcirc_{[2,3]} p$. As shown in Table 1, we translate $\bigcirc_{[2,3]} p$ into: $\bigvee_{2\leq l\leq 3}(\bigcirc^l(\neg gap \wedge p) \wedge \bigwedge_{1\leq k<l} \bigcirc^k gap)$. Note that the state sequence represented on the right side of Figure 1(a) is a model of the translation. Since *gap* is a propositional variable not occurring in $\sigma$, the time points mapped by the image of $\tau$ do not contain *gap*.

**Definition 2.** *Given a state sequence $\sigma'$ such that $(\sigma', 0) \models \neg gap \wedge \Box(\Diamond\neg gap)$, we inductively define $\rho = (\sigma_0, \tau(0))(\sigma_1, \tau(1))\ldots$, where $(\sigma_0, \tau(0)) = (\sigma'_0, 0)$ and, for $i, j, k \in \mathbb{N}$ and $i > 0$, $(\sigma_i, \tau(i))$ is as follows:*

$$\sigma_i = \sigma'_j \text{ and } \tau(i) = j \quad \text{if } j > \tau(i-1),\ gap \notin \sigma'_j \text{ and for all } k,\ \tau(i-1) < k < j,\ gap \in \sigma'_k.$$

As $\sigma'$ is such that $(\sigma', 0) \models \neg gap \wedge \Box(\Diamond\neg gap)$, for each $i \in \mathbb{N}$ we have $\tau(i) \in \mathbb{N}$. Also, for $i > 0$, $\tau(i) > \tau(i-1)$ and, so, $\tau : \mathbb{N} \to \mathbb{N}$ is well defined. We are ready for Theorem 1, which states the correctness of our translation from MTL to LTL using 'gap's.

**Theorem 1** *Let $\varphi = p_0 \wedge \bigwedge_i \Box_{[0,\infty)}(p_i \to \psi_i)$ be an MTL formula in NNF and FNF. Let $\varphi^\sharp = p_0 \wedge \bigwedge_i \Box(p_i \to (\neg gap \wedge \psi_i^\sharp))$ be the result of replacing each $\psi_i$ in $\varphi$ by $\psi_i^\sharp$ as in Table 1. Then, $\varphi$ is satisfiable if, and only if, $\varphi^\sharp \wedge \neg gap \wedge \Box(\Diamond\neg gap)$ is satisfiable.*

| MTL | (Strict) LTL Gap Translation |
|---|---|
| $(\bigcirc_{[0,\infty)}\alpha)^\sharp$ | $(\bigcirc_{[1,\infty)}\alpha)^\sharp$ |
| $(\bigcirc_{[c_1,\infty)}\alpha)^\sharp$ | $(\bigwedge_{1\leq k<c_1}\bigcirc^k gap)\wedge\bigcirc^{c_1}(gap\mathcal{U}(\alpha\wedge\neg gap))$ |
| $(\bigcirc_{[c_1,c_2]}\alpha)^\sharp$ | $\bigvee_{c_1\leq l\leq c_2}(\bigcirc^l(\neg gap\wedge\alpha)\wedge\bigwedge_{1\leq k<l}\bigcirc^k gap)$ |
| $(\bigcirc_{[0,0]}\alpha)^\sharp$ | **false** |
| $(\bigcirc_{[0,c_2]}\alpha)^\sharp$ | $(\bigcirc_{[1,c_2]}\alpha)^\sharp$ |
| $(\alpha\mathcal{U}_{[0,\infty)}\beta)^\sharp$ | $(gap\vee\alpha)\mathcal{U}(\neg gap\wedge\beta)$ |
| $(\alpha\mathcal{U}_{[c_1,\infty)}\beta)^\sharp$ | $(\bigwedge_{0\leq k<c_1}\bigcirc^k(gap\vee\alpha))\wedge\bigcirc^{c_1}((gap\vee\alpha)\mathcal{U}(\neg gap\wedge\beta))$ |
| $(\alpha\mathcal{U}_{[c_1,c_2]}\beta)^\sharp$ | $\bigvee_{c_1\leq l\leq c_2}(\bigcirc^l(\neg gap\wedge\beta)\wedge\bigwedge_{0\leq k<l}\bigcirc^k(gap\vee\alpha))$ |
| $(\alpha\mathcal{U}_{[0,0]}\beta)^\sharp$ | $\neg gap\wedge\beta$ |
| $(\alpha\mathcal{U}_{[0,c_2]}\beta)^\sharp$ | $(\neg gap\wedge\beta)\vee(\alpha\mathcal{U}_{[1,c_2]}\beta)^\sharp$ |

Table 1: (Strict) Gap Translation from MTL to LTL, where $\alpha, \beta$ are propositional formulae and $c_1, c_2 > 0$.

**Non-Strict Semantics** We now show how we modify the Gap translation for non-strict timed state sequences. We introduce a fresh propositional variable called 'same'. *same* is true exactly in those states $\sigma'_j$ of $\sigma'$ such that there is $i \in \mathbb{N}$ with $\tau(i) = j$ and, for $i > 0$, $\tau(i) = \tau(i-1)$. Note that *same* and *gap* cannot both be true in any state. We say that a state $s$ is a *gap state* if $gap \in s$. We now define our mappings between MTL and LTL models.

**Definition 3.** *Let $\rho = (\sigma, \tau)$ be a non-strict timed state sequence. We define $\sigma' = \sigma'_0\sigma'_1\ldots$ by initially setting $\sigma' = \sigma$ and then modifying $\sigma'$ with the two following steps:*

1. *For $i > 0$, if $\tau(i) - \tau(i-1) = 0$ then set $\sigma'_i := \sigma_i \cup \{same\}$;*
2. *For $i, j \geq 0$, if $\sigma'_j$ is the $i$-th non-gap state in $\sigma'$, $\sigma'_{j+1}$ is a non-gap state and $\tau(i+1) - \tau(i) = k > 1$ then add $k-1$ states of the form $\{gap\}$ between $\sigma'_j$ and $\sigma'_{j+1}$.*

Figure 1(b) illustrates the mapping given by Definition 3. For instance, if $\rho = (\sigma, \tau)$ is the non-strict timed state sequence on the left side of Figure 1(b) then $(\rho, 0) \models \Diamond_{[2,2]}q$. As shown in Table 2, we translate $\Diamond_{[2,2]}q$ into: $same\mathcal{U}(\neg same \wedge \bigcirc(same\mathcal{U}(\neg same \wedge \bigcirc((q \wedge \neg gap) \vee \bigcirc(same\mathcal{U}(q \wedge same))))))$. The main distinction from the translation presented in Table 1 is that here we use nested until operators to make progress in our encoding of the time line whenever we find a state with $\neg same$. Note that the state

sequence represented on the right side of Figure 1(b) is a model of the translation (recall that $\Diamond_{[2,2]}q \equiv \mathbf{true}\mathcal{U}_{[2,2]q}$).

**Definition 4.** *Let $\sigma'$ be a state sequence such that $(\sigma', 0) \models \neg gap \wedge \neg same \wedge \Box(\Diamond\neg gap) \wedge \Box(\neg same \vee \neg gap) \wedge \Box(gap \rightarrow \bigcirc\neg same)$. We first define $\tau : \mathbb{N} \rightarrow \mathbb{N}$ by setting $\tau(0) = 0$ and, for $i > 0$, $\tau(i)$ is as follows:*

$$\tau(i) = \begin{cases} \tau(i-1) & \text{if } \sigma'_j \text{ is the } i{+}1\text{-th non-gap state and } same \in \sigma'_j \\ \tau(i-1){+}k{+}1 & \text{otherwise,} \end{cases}$$

*where $k \geq 0$ is the number of gap states between the $i-1$-th and $i$-th non-gap states. We now define $\sigma$ as follows:*

$$\sigma_i = \sigma'_j \setminus \{same\}, \text{ where } \sigma'_j \text{ is the } i+1\text{-th non-gap state.}$$

One can use the mappings given by Definitions 3 and 4 to show Theorem 2 with ideas similar to those used in Theorem 1.

**Theorem 2** *Let $\varphi = p_0 \wedge \bigwedge_i \Box_{[0,\infty)}(p_i \rightarrow \psi_i)$ be an MTL formula in NNF and FNF. Let $\varphi^\sharp = p_0 \wedge \bigwedge_i \Box(p_i \rightarrow (\neg gap \wedge \psi_i^\sharp))$ be the result of replacing each $\psi_i$ in $\varphi$ by $\psi_i^\sharp$ as in Table 2. Then, $\varphi$ is satisfiable if, and only if, $\varphi^\sharp \wedge \neg gap \wedge \neg same \wedge \Box(\Diamond\neg gap) \wedge \Box(\neg same \vee \neg gap) \wedge \Box(gap \rightarrow \bigcirc\neg same)$ is satisfiable.*

## 4 From MTL to LTL: encoding time differences

Assume that our MTL formulae are in NNF and FNF. Similar to the previous section our proof strategy relies on mapping each timed state sequence $\rho = (\sigma, \tau)$ to a state sequence $\sigma'$ such that $\rho = (\sigma, \tau)$ is a model of an MTL formula if, and only if, $\sigma'$ is a model of our LTL translation. We first show a translation under the strict semantics and then we show how to adapt it for the non-strict semantics.

**Strict Semantics** Let $C - 1$ be the greatest number occurring in an interval in an MTL formula $\varphi$ or 1, if none occur. We say that a timed state sequence $\rho = (\sigma, \tau)$ is *$C$-bounded*, for a constant $C \in \mathbb{N}$, if $\tau(0) \leq C$ and, for all $i \in \mathbb{N}$, $\tau(i+1) - \tau(i) \leq C$. To map a timed state sequence $\rho = (\sigma, \tau)$ to a state sequence $\sigma'$ we employ the following result adapted from [4].

**Theorem 3** *Let $\varphi$ be an MTL formula. If there is a timed state sequence $\rho = (\sigma, \tau)$ such that $(\rho, 0) \models \varphi$ then there is a $C$-bounded timed state sequence $\rho_C$ such that $(\rho_C, 0) \models \varphi$.*

| MTL | Non-Strict LTL Gap Translation |
|---|---|
| $(\bigcirc_{[0,\infty)}\alpha)^\sharp$ | $(\bigcirc_{[0,0]}\alpha)^\sharp \vee (\bigcirc_{[1,\infty)}\alpha)^\sharp$ |
| $(\bigcirc_{[0,c_2]}\alpha)^\sharp$ | $(\bigcirc_{[0,0]}\alpha)^\sharp \vee (\bigcirc_{[1,c_2]}\alpha)^\sharp$ |
| $(\bigcirc_{[0,0]}\alpha)^\sharp$ | $\bigcirc(\alpha \wedge \mathit{same})$ |
| $(\alpha \mathcal{U}_{[c_1,\infty)}\beta)^\sharp$ | $\alpha \wedge \bigcirc((\alpha \wedge \mathit{same})\mathcal{U}(\neg \mathit{same} \wedge (\alpha \mathcal{U}_{[c_1-1,\infty)}\beta)^\sharp))$ |
| $(\alpha \mathcal{U}_{[0,\infty)}\beta)^\sharp$ | $(\mathit{gap} \vee \alpha)\mathcal{U}(\neg \mathit{gap} \wedge \beta)$ |
| $(\alpha \mathcal{U}_{[c_1,c_2]}\beta)^\sharp$ | $\alpha \wedge \bigcirc((\alpha \wedge \mathit{same})\mathcal{U}(\neg \mathit{same} \wedge (\alpha \mathcal{U}_{[c_1-1,c_2-1]}\beta)^\sharp))$ |
| $(\alpha \mathcal{U}_{[0,0]}\beta)^\sharp$ | $(\beta \wedge \neg \mathit{gap}) \vee (\alpha \wedge \bigcirc((\alpha \wedge \mathit{same})\mathcal{U}(\beta \wedge \mathit{same})))$ |
| $(\alpha \mathcal{U}_{[0,c_2]}\beta)^\sharp$ | $(\alpha \mathcal{U}_{[0,0]}\beta)^\sharp \vee (\alpha \mathcal{U}_{[1,c_2]}\beta)^\sharp$ |

Table 2: Non-Strict Gap Translation from MTL to LTL, using *gap* and *same*, where $\alpha, \beta$ are propositional logic formulae, $c_1, c_2 > 0$ and $(\bigcirc_{[c_1,\infty)}\alpha)^\sharp$ and $(\bigcirc_{[c_1,c_2]}\alpha)^\sharp$ are as in Table 1.

By Theorem 3, w.l.o.g., we can consider only timed state sequences where the time difference from a state to its previous state is bounded by $C$. Then, we can encode time differences with a set $\Pi_\delta = \{\delta_i^- \mid 1 \leq i \leq C\}$ of propositional variables where each $\delta_i^-$ represents a time difference of $i$ w.r.t. the previous state (one could also encode the time difference to the next state instead of the difference from the previous state). We also use propositional variables of the form $s_m^n$ with the meaning that 'the sum of the time differences from the last $n$ states to the current state is $m$'. For our translation, we only need to define these variables up to sums bounded by $2 \cdot C$. We can now define our mapping from an MTL model to an LTL model[3].

**Definition 5.** *Given a $C$-bounded timed state sequence $\rho = (\sigma, \tau)$, we define $\sigma' = \sigma'_0\sigma'_1 \ldots$ by setting $\sigma'_0 = \sigma_0$ and, for $i > 0$:*

$$\sigma'_i = \sigma_i \cup \{\delta_k^-, s_k^1 \mid \tau(i) - \tau(i-1) = k, 1 \leq k \leq C\}$$

$$\cup \{s^{j+1}_{\mathsf{min}(l+k,2\cdot C)} \mid s_k^1 \in \sigma'_i \text{ and } s_l^j \in \sigma'_{i-1}, 1 < j+1, l \leq 2 \cdot C, 1 \leq k \leq C\}$$

*where variables of the form $s_m^n$ and $\delta_n^-$, $n, m \in \mathbb{N}$, do not occur in $\sigma$.*

In Definition 5, if, for example, $\tau(2) - \tau(0) = 4$ then $(\sigma', 2) \models s_4^2$. Intuitively, the variable $s_4^2$ allow us to group together all the cases where

[3] We write $\mathsf{min}(l+k, 2 \cdot C)$ for the minimum between $l+k$ and $2 \cdot C$. If the minimum is $2 \cdot C$ then $s^{j+1}_{2\cdot C}$ means that the sum of the last $j+1$ variables is greater or equal to $2 \cdot C$.

the sum of the time differences from the last 2 states to the current state is 4. This happens when: $\tau(2) - \tau(1) = 3$ and $\tau(1) - \tau(0) = 1$; or $\tau(2) - \tau(1) = 1$ and $\tau(1) - \tau(0) = 3$; or $\tau(2) - \tau(1) = 2$ and $\tau(1) - \tau(0) = 2$.

The next lemma gives the main properties of $\sigma'$. First, we need some notation. We use two additional $n$-ary boolean operators $\oplus_{=1}$ and $\oplus_{\leq 1}$. If $S = \{\varphi_1, \ldots, \varphi_n\}$ is a finite set of LTL formulae, then $\oplus_{=1}(\varphi_1, \ldots, \varphi_n)$, also written $\oplus_{=1} S$, is an LTL formula. Let $\sigma'$ be a state sequence and $i \in \mathbb{N}$. Then $(\sigma', i) \models \oplus_{=1} S$ iff $(\sigma', i) \models \varphi_j \in S$ for exactly one $\varphi_j \in S$, $1 \leq j \leq n$. Similarly, $(\sigma', i) \models \oplus_{\leq 1} S$ iff $(\sigma', i) \models \varphi_j \in S$ for at most one $\varphi_j \in S$, $1 \leq j \leq n$. By definition of $\sigma'$ the following lemma is immediate.

**Lemma 1.** *Let $S_C$ be the conjunction of the following:*

1. $\bigcirc\Box \oplus_{=1} \Pi_\delta$, *for* $\Pi_\delta = \{\delta_k^- \mid 1 \leq k \leq C\}$;
2. $\Box(\delta_k^- \leftrightarrow s_k^1)$, *for* $1 \leq k \leq C$;
3. $\Box \oplus_{\leq 1} \Pi^i$, *for* $1 \leq i \leq 2 \cdot C$ *and* $\Pi^i = \{s_j^i \mid i \leq j \leq 2 \cdot C\}$;
4. $\Box((\bigcirc s_k^1 \wedge s_l^j) \rightarrow \bigcirc s^{j+1}_{\mathsf{min}(l+k, 2\cdot C)})$, *for* $1 < j+1, l \leq 2 \cdot C, 1 \leq k \leq C$.

*Given a $C$-bounded timed state sequence $\rho = (\sigma, \tau)$, let $\sigma' = \sigma'_0\sigma'_1 \ldots$ be as in Definition 5. Then, $(\sigma', 0) \models S_C$.*

Point 1 ensures that at all times, the time difference $k$ from the current state to the previous (if it exists) is uniquely encoded by the variable $\delta_k^-$. In Point 2 we have that the sum of the difference of the last state to the current, encoded by $s_k^1$, is exactly $\delta_k^-$. Point 3 ensures that at all times we cannot have more than one value for the sum of the time differences of the last $i$ states. Finally, Point 4 has the propagation of sum variables: if the sum of the last $j$ states is $l$ and the time difference to the next is $k$ then the next state should have that the sum of the last $j+1$ states is $l+k$. We now define our mapping from an LTL model of $S_C$ to an MTL model (for this mapping, we actually only need Point 1).

**Definition 6.** *Given a state sequence $\sigma' = \sigma'_0\sigma'_1 \ldots$ such that $(\sigma', 0) \models S_C$, we define a $C$-bounded timed state sequence $\rho = (\sigma, \tau)$ by setting $\sigma_i = \sigma'_i \setminus (\Pi_\delta \cup \bigcup_{1 \leq j \leq C} \Pi^j)$, for $i \in \mathbb{N}$, and:*

$$\tau(i) = \begin{cases} 0 & \text{if } i = 0 \\ \tau(i-1) + k & \text{if } i > 0,\ \delta_k^- \in \sigma'_i \end{cases}$$

Note that $\rho$, in particular, $\tau$, in Definition 6 is well-defined because for every $i \in \mathbb{N}$ there is exactly one $k$ such that $\delta_k^- \in \sigma'_i$. As shown in Table 3, we translate, for example, $\bigcirc_{[2,3]} p$ into $\bigcirc((\delta_2^- \vee \delta_3^-) \wedge p)$. We are ready for Theorem 4, which states the correctness of our translation using time differences.

| MTL | (Strict) LTL Time Difference Translation |
|---|---|
| $(\bigcirc_{[c_1,\infty)}\alpha)^\sharp$ | $\bigcirc((\bigvee_{c_1\leq i\leq C}\delta_i^-)\wedge\alpha)$ |
| $(\bigcirc_{[0,\infty)}\alpha)^\sharp$ | $\bigcirc\alpha$ |
| $(\bigcirc_{[c_1,c_2]}\alpha)^\sharp$ | $\bigcirc((\bigvee_{c_1\leq i\leq c_2}\delta_i^-)\wedge\alpha)$ |
| $(\bigcirc_{[0,c_2]}\alpha)^\sharp$ | $(\bigcirc_{[1,c_2]}\alpha)^\sharp$ |
| $(\bigcirc_{[0,0]}\alpha)^\sharp$ | **false** |
| $(\alpha\mathcal{U}_{[c_1,\infty)}\beta)^\sharp$ | $\bigvee_{1\leq i\leq c_1}(\bigcirc^i((\bigvee_{c_1\leq j\leq c_1+C}\ s_j^i)\wedge\alpha\mathcal{U}\beta)\wedge(\bigwedge_{0\leq k<i}\bigcirc^k\alpha))$ |
| $(\alpha\mathcal{U}_{[0,\infty)}\beta)^\sharp$ | $\alpha\mathcal{U}\beta$ |
| $(\alpha\mathcal{U}_{[c_1,c_2]}\beta)^\sharp$ | $\bigvee_{1\leq i\leq c_2}(\bigcirc^i((\bigvee_{c_1\leq j\leq c_2}\ s_j^i)\wedge\beta)\wedge(\bigwedge_{0\leq k<i}\bigcirc^k\alpha))$ |
| $(\alpha\mathcal{U}_{[0,c_2]}\beta)^\sharp$ | $\beta\vee(\alpha\mathcal{U}_{[1,c_2]}\beta)^\sharp$ |
| $(\alpha\mathcal{U}_{[0,0]}\beta)^\sharp$ | $\beta$ |

Table 3: (Strict) Time Difference Translation from MTL to LTL where $\alpha,\beta$ are propositional logic formulae and $c_1,c_2>0$.

**Theorem 4** *Let $\varphi=p_0\wedge\bigwedge_i\Box_{[0,\infty)}(p_i\rightarrow\psi_i)$ be an MTL formula in NNF and FNF. Let $\varphi^\sharp=p_0\wedge\bigwedge_i\Box(p_i\rightarrow\psi_i^\sharp)$ be the result of replacing each $\psi_i$ in $\varphi$ by $\psi_i^\sharp$ as in Table 3. Then, $\varphi$ is satisfiable if, and only if, $\varphi^\sharp\wedge S_C$ is satisfiable.*

**Non-Strict Semantics** We now show how we modify the Time Difference translation for non-strict timed state sequences. We extend the set $\Pi_\delta=\{\delta_i^-\mid 1\leq i\leq C\}$ of propositional variables representing time differences with $\delta_0^-$, which holds whenever the time difference to the previous state is 0. We say that a state is *non-zero* if the time difference to the previous state is non-zero. The meaning of the variables of the form $s_m^n$ also needs to change, it now indicates that 'the sum of the time differences from the last $n$ *non-zero* states to the current state is $m$'. As before, for our translation, we only need to define these variables up to sums bounded by $2\cdot C$. We can now define our mapping from an MTL model to an LTL model.

Given a $C$-bounded non-strict timed state sequence $(\sigma,\tau)$, we define a state sequence $\sigma'$ as in Definition 5, with the difference that, whenever $\tau(i)=\tau(i-1)$, we now make $\delta_0^-$ true in $\sigma'_i$ and copy all variables of the form $s_m^n$ in $\sigma'_{i-1}$ to $\sigma'_i$. Let $S'_C$ be the conjunction of the following:

1. $\bigcirc\Box\oplus_{=1}\Pi_\delta$, for $\Pi_\delta=\{\delta_k^-\mid 0\leq k\leq C\}$;
2. $\Box(\delta_k^-\leftrightarrow s_k^1)$, for $1\leq k\leq C$;

3. $\Box \oplus_{\leq 1} \Pi^i$, for $1 \leq i \leq 2 \cdot C$ and $\Pi^i = \{s^i_j \mid i \leq j \leq 2 \cdot C\}$;
4. $\Box((\bigcirc s^1_k \wedge s^j_l) \rightarrow \bigcirc s^{j+1}_{\mathsf{min}(l+k,2 \cdot C)})$, for $1 < j+1 \leq l+k \leq 2 \cdot C$.
5. $\Box((\bigcirc \delta^-_0 \wedge s^j_l) \rightarrow \bigcirc s^j_l)$, for $1 \leq j \leq l \leq 2 \cdot C$.

It is easy to see that $(\sigma', 0) \models S'_C$. Note that the only difference from $S'_C$ to $S_C$, defined in Lemma 1, is Point 5 which propagates the variables of the form $s^n_m$ to the next state if the time difference is zero. The mapping from an LTL model of $S'_C$ to an MTL model is defined in the same way as in Definition 6 (but now $k$ in $\delta^-_k$ can be zero). To simplify the notation, in Table 4 we write $\phi \mathcal{U}^n \gamma, \chi$ as a shorthand for $\phi \mathcal{U}(\gamma \wedge \bigcirc(\phi \mathcal{U}^{n-1} \gamma, \chi))$, where $\phi \mathcal{U}^1 \gamma, \chi = \phi \mathcal{U} \chi$. Theorem 5 states the correctness of our translation (Table 4) using non-strict time differences. It can be proved with ideas similar to those used in the proof of Theorem 4. The main distinction appears in the translation of the 'until' formulas, where we nest until operators so that we can count $n$ non-zero states and then check whether a variable of the form $s^n_m$ holds (in the strict case all states are non-zero, so in Table 3 we can count these states with next operators).

| MTL | (Non-Strict) LTL Time Difference Translation |
|---|---|
| $(\bigcirc_{[k_1,\infty)} \alpha)^\sharp$ | $\bigcirc((\bigvee_{k_1 \leq i \leq C} \delta^-_i) \wedge \alpha)$ |
| $(\bigcirc_{[k_1,k_2]} \alpha)^\sharp$ | $\bigcirc((\bigvee_{k_1 \leq i \leq k_2} \delta^-_i) \wedge \alpha)$ |
| $(\alpha \mathcal{U}_{[c_1,\infty)} \beta)^\sharp$ | $\alpha \wedge \bigcirc \bigvee_{1 \leq i \leq c_1} ((\alpha \wedge \delta^-_0) \mathcal{U}^i (\neg \delta^-_0 \wedge \alpha), (\neg \delta^-_0 \wedge (\bigvee_{c_1 \leq j \leq c_1 + C} s^i_j) \wedge \alpha \mathcal{U} \beta))$ |
| $(\alpha \mathcal{U}_{[0,\infty)} \beta)^\sharp$ | $\alpha \mathcal{U} \beta$ |
| $(\alpha \mathcal{U}_{[c_1,c_2]} \beta)^\sharp$ | $\alpha \wedge \bigcirc \bigvee_{1 \leq i \leq c_2} ((\alpha \wedge \delta^-_0) \mathcal{U}^i (\neg \delta^-_0 \wedge \alpha), (\neg \delta^-_0 \wedge (\bigvee_{c_1 \leq j \leq c_2} s^i_j) \wedge (\alpha \mathcal{U}_{[0,0]} \beta)^\sharp))$ |
| $(\alpha \mathcal{U}_{[0,c_2]} \beta)^\sharp$ | $(\alpha \mathcal{U}_{[0,0]} \beta)^\sharp \vee (\alpha \mathcal{U}_{[1,c_2]} \beta)^\sharp$ |
| $(\alpha \mathcal{U}_{[0,0]} \beta)^\sharp$ | $\beta \vee (\alpha \wedge \bigcirc((\alpha \wedge \delta^-_0) \mathcal{U} (\beta \wedge \delta^-_0)))$ |

Table 4: Non-Strict LTL Time Difference Translation from MTL to LTL where $\alpha, \beta$ are propositional logic formulae, $k_1, k_2 \geq 0$ and $c_1, c_2 > 0$.

**Theorem 5** *Let $\varphi = p_0 \wedge \bigwedge_i \Box_{[0,\infty)}(p_i \rightarrow \psi_i)$ be an MTL formula in NNF and FNF. Let $\varphi^\sharp = p_0 \wedge \bigwedge_i \Box(p_i \rightarrow \psi^\sharp_i)$ be the result of replacing each $\psi_i$ in $\varphi$ by $\psi^\sharp_i$ as in Table 4. Then, $\varphi$ is satisfiable if, and only if, $\varphi^\sharp \wedge S'_C$ is satisfiable.*

## 5 Experiments

In order to empirically evaluate the translations, we have used them together with three LTL satisfiability solvers, `TRP++`, `pltl` and `NuSMV`. `TRP++`

Table 5: Experiments with TRP++, pltl and NuSMV

(a) Performance on $\Diamond_{[0,b_1]}p \wedge \Box_{[0,\infty)}\neg p$

| $b_1$ | TD TRP++ | Gap TRP++ | TD pltl | Gap pltl | TD NuSMV | Gap NuSMV |
|---|---|---|---|---|---|---|
| Strict Semantics | | | | | | |
| 0 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 2 | 0.00 | 0.00 | OoM | 0.00 | 33.54 | 0.00 |
| 4 | 0.10 | 0.00 | OoM | 0.00 | T/O | 0.00 |
| 6 | 3.18 | 0.00 | OoM | 0.00 | T/O | 0.00 |
| 8 | 64.00 | 0.00 | OoM | 0.00 | T/O | 0.00 |
| Non-strict Semantics | | | | | | |
| 0 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 2 | 36.17 | 0.01 | OoM | 0.00 | 25.45 | 0.00 |
| 4 | T/O | 0.55 | OoM | 0.00 | T/O | 0.00 |
| 6 | T/O | 7.29 | OoM | 0.01 | T/O | 0.00 |
| 8 | T/O | 58.20 | OoM | 0.09 | T/O | 0.00 |

(b) Performance on $\bigcirc_{[10,\infty)}p \wedge \bigcirc_{[b_2,\infty)}\neg p$

| $b_2$ | TD TRP++ | Gap TRP++ | TD pltl | Gap pltl | TD NuSMV | Gap NuSMV |
|---|---|---|---|---|---|---|
| Strict Semantics | | | | | | |
| 10 | 0.00 | 0.33 | 0.01 | 0.00 | 0.63 | 0.04 |
| 12 | 0.00 | 0.77 | 0.01 | 0.00 | 2.76 | 0.07 |
| 14 | 0.00 | 1.68 | 0.01 | 0.00 | 55.91 | 0.15 |
| 16 | 0.00 | 3.15 | 0.01 | 0.00 | T/O | 0.24 |
| 18 | 0.01 | 5.68 | 0.01 | 0.00 | T/O | 0.40 |
| Non-strict Semantics | | | | | | |
| 10 | 0.00 | 0.32 | 0.00 | 0.00 | 0.93 | 0.05 |
| 12 | 0.00 | 0.74 | 0.00 | 0.00 | 6.06 | 0.05 |
| 14 | 0.00 | 1.62 | 0.00 | 0.00 | 42.44 | 0.08 |
| 16 | 0.00 | 3.03 | 0.00 | 0.00 | T/O | 0.13 |
| 18 | 0.00 | 5.43 | 0.00 | 0.00 | T/O | 0.14 |

implements an ordered resolution calculus for LTL [16], pltl implements a one-pass and-or tree tableau calculus [13], and NuSMV uses a reduction to model checking combined with symbolic model checking using BDDs [7]. These solvers performed well in the extensive comparison of LTL satisfiability solvers performed in [20].

We focus on formulae where differences between the two translations could lead to differences in the behaviour of solvers on these formulae. In particular, for $(\alpha \mathcal{U}_{[c_1,c_2]} \beta)$ the Strict and Non-Strict LTL Time Difference Translations contain disjunctive subformulae of the form $\bigvee_{c_1 \leq j \leq c_2} s^i_j$ that have no equivalence in the Strict and Non-Strict LTL Gap Translations of that formula. Each sum variable $s^i_j$ is also subject to the constraints expressed by $S_C$. It is a reasonable hypothesis that this will have a detrimental effect on the performance of a solver. On the other hand, for $\bigcirc_{[c_1,\infty)}\alpha$ both LTL Gap Translations contain an eventuality formula $gap\mathcal{U}(\alpha \wedge \neg gap)$ that is not present in the LTL Time Difference Translations of this formula. Here, the hypothesis is that the LTL Time Difference Translations lead to better behaviour of solvers.

To test our two hypotheses, we consider the unsatisfiable parameterised formulae $\theta^1_{b_1} := \Diamond_{[0,b_1]}p \wedge \Box_{[0,\infty)}\neg p$ for values of $b_1$ between 0 and 8, and $\theta^2_{b_2} := \bigcirc_{[10,\infty)}p \wedge \bigcirc_{[b_2,\infty)}\neg p$ for values of $b_2$ between 10 and 18. After transformation to Flat Normal Form, we apply one of the four translations, and run a solver five times on the resulting LTL formula (with a timeout of 1000 CPU seconds), and then determine the median CPU time over those

five runs. The repeated runs are necessary to moderate the fluctuations in runtime shown by all provers. The experiments were conducted on a PC with Intel i7-2600 CPU @ 3.40GHz and 16GB main memory. Tables 5a and 5b show the results with entries indicating the median CPU time in seconds ('T/O' indicates timeout). The combination TD + `pltl`, for both strict and non-strict semantics, runs out of memory for $\theta^1_{b_1}$ with $b_1 \geq 1$, after searching for a model for about 100 CPU seconds, indicated by 'OoM' in Table 5a. Strict and non-strict TD + `NuSMV` exceeds the timeout for $\theta^1_{b_1}$ with $b_1 > 3$ and the same is true for non-strict TD + `TRP++`.

Table 5a confirms our hypothesis that for $(\alpha \mathcal{U}_{[c_1,c_2]} \beta)$ the LTL Gap translations, independent of the semantics, lead to better performance. However, Table 5b only confirms that the LTL Time Difference Translations lead to better performance on $\bigcirc_{[c_1,\infty)} \alpha$ for `TRP++`. For `pltl` there is no significant difference between the translations while `NuSMV` again performs considerably better with the LTL Gap translations, independent of the semantics.

## 6 An Example: Multiprocessor Job-Shop Scheduling

We consider a generalisation of the classic job-shop scheduling problem, called the Multiprocessor Job-shop Scheduling (MJS) problem [14,8]. The representation provided is based on that in [9]. Here, a set of jobs have to be processed on a set of machines running in parallel. Each job might require a number of processor steps to complete. We first show how one can encode the problem in MTL with the strict semantics and then we show the encoding with the non-strict semantics.

**Strict Semantics** Assume we have $n$ jobs $j_1, j_2, \ldots, j_n$ and $k$ machines $m_1, m_2, \ldots, m_k$. Let

- $start\_run_{j_i}$, $run_{j_i}$ and $has\_run_{j_i}$ denote the start, the execution and the end of the execution of job $j_i$ on some machine, respectively;
- $start\_run_{j_i m_l}$ and $run_{j_i m_l}$ denote the start and the execution of job $j_i$ on machine $m_l$, respectively; and
- $t_{j_i m_l}$ to denote the time taken to run job $j_i$ on machine $m_l$.

The following equations state that (1) once a job starts running it must start running on one of the machines and that (2) once a job starts running on a machine it must run on that machine (where $\bigwedge_{1 \leq i \leq n}$ and $\bigwedge_{1 \leq i \leq n, 1 \leq l \leq k}$ in front of the formulas is omitted for brevity).

$$\Box(start\_run_{j_i} \rightarrow \bigvee_{l=1}^{k} start\_run_{j_i m_l}) \tag{1}$$

$$\Box(start\_run_{j_i m_l} \rightarrow run_{j_i m_l}) \tag{2}$$

Equation (3) states that: if a job is running on one machine then it cannot be running on another (integrity of jobs); and another job cannot be running on the same machine (integrity of machines). By Equation (4), once a job has started it cannot be started again.

$$\Box(run_{j_i m_l} \rightarrow (\bigwedge_{p=1,p\neq l}^{k} \neg run_{j_i m_p} \wedge \bigwedge_{q=1,q\neq i}^{n} \neg run_{j_q m_l})) \tag{3}$$

$$\Box(start\_run_{j_i} \rightarrow \bigcirc\Box\neg start\_run_{j_i}) \tag{4}$$

We write $\neg run_{j_i}$ as a short hand for $\bigwedge_{l=1}^{k} \neg run_{j_i m_l}$. We can use (5) to denote that once job $j_i$ is started to run on machine $m_l$ it takes time $t_{j_i m_l}$ and (6) to denote that once job $j_i$ has finished running on machine $m_l$ it will not run again. Further, Equation (7) denotes that job $j_i$ cannot be run until it has started.

$$\Box(start\_run_{j_i m_l} \rightarrow \Box_{[0,t_{j_i m_l}-1]} run_{j_i m_l} \wedge \neg has\_run_{j_i}) \tag{5}$$

$$\Box(start\_run_{j_i m_l} \rightarrow \Box_{[t_{j_i m_l},\infty)}(\neg run_{j_i} \wedge has\_run_{j_i})) \tag{6}$$

$$\Box(\neg run_{j_i} \mathcal{U} start\_run_{j_i}) \tag{7}$$

We assume initially that no jobs have run, i.e., $\bigwedge_{i=1}^{n} \neg has\_run_{j_i}$; and that (8) if a job has not run and is currently not running then it has not run in the next moment.

$$\Box((\neg has\_run_{j_i} \wedge \neg run_{j_i}) \rightarrow \bigcirc\neg has\_runj_i) \tag{8}$$

We can now check whether we can achieve a schedule after at most $t$ time points by adding $\Diamond_{[0,t]} \bigwedge_{i=1}^{n} has\_run_{j_i}$. We can also specify constraints on jobs such as

- $\Box(run_{j_i} \leftrightarrow run_{j_i,m_l})$: job $j_i$ must run on machine $m_l$;
- $\Diamond(start\_run_{j_i} \rightarrow \Diamond_{[1,\infty)} start\_run_{j_m})$: job $j_i$ must start before job $j_m$;
- $\Diamond_{[c,d]} start\_run_{j_i}$: job $j_i$ must start at a point within the interval $[c, d]$.

**Non-Strict Semantics** We again assume we have $n$ jobs $j_1, j_2, \ldots, j_n$ and $k$ machines $m_1, m_2, \ldots, m_k$. Let

- $start\_run_{j_i}$ and $has\_run_{j_i}$ denote the start and the end of job $j_i$ on some machine, respectively;
- $m_l$ denote a state of machine $m_l$;
- $run_{j_i}$ denote that job $j_i$ is running on some machine; and
- $t_{j_i m_l}$ denote the time taken to run job $j_i$ on machine $m_l$.

In each state exactly one of the variables of the form $m_l$ is true. Also, in each state at most one job is running, but now we may have multiple states at the same time. Let $\Pi_m = \{m_1, \ldots, m_k\}$ and $\Pi_j = \{run_{j_1}, \ldots, run_{j_n}\}$. The following states the constraints mentioned above (the meaning of $\oplus_{=1}$ and $\oplus_{\leq 1}$ is as described in Section 3):

$$\Box(\oplus_{=1}\Pi_m \wedge \oplus_{\leq 1}\Pi_j) \tag{9}$$

Equation (10) specifies that if a job is running on one machine then it cannot be running on another. Equation (11) states that once a job is started it cannot be started again (where $\bigwedge_{1\leq i\leq n, 1\leq l\leq k}$ and $\bigwedge_{1\leq i\leq n}$ in front of the formulas is omitted for brevity).

$$\Box((m_l \wedge run_{j_i}) \rightarrow \textstyle\bigwedge_{l'\neq l} \Box\neg(m_{l'} \wedge run_{j_i})) \tag{10}$$

$$\Box(start\_run_{j_i} \rightarrow \bigcirc\Box\neg start\_run_{j_i}) \tag{11}$$

We use the following

$$\begin{aligned}\Box((start\_run_{j_i} \wedge m_l) \rightarrow (\Box_{[0,t_{j_i m_l}-1]}(\neg has\_run_{j_i} \wedge (m_l \rightarrow run_{j_i})) \\ \wedge \Diamond_{[0,t_{j_i m_l}]} has\_run_{j_i}))\end{aligned} \tag{12}$$

to denote that once job $j_i$ started to run on machine $m_l$ it takes time $t_{j_i m_l}$ and (13) to denote that once job $j_i$ has finished running on machine $m_l$ it will not run again. Further, we use $\Box(\neg run_{j_i} \mathcal{U} start\_run_{j_i})$ to state a job $j_i$ cannot be run until it is started and $\Box(\neg has\_run_{j_i} \mathcal{U} start\_run_{j_i})$ to state that a job cannot have run before it starts (another rule above will make sure that $has\_run_{j_i}$ will hold after the run has finished).

$$\Box((start\_run_{j_i} \wedge m_l) \rightarrow \Box_{[t_{j_i m_l}+1,\infty)}(\neg run_{j_i} \wedge has\_run_{j_i})) \tag{13}$$

We assume initially that no jobs have run, i.e., $\bigwedge_{i=1}^{n} \neg has\_run_{j_i}$. We can now check whether we can achieve a schedule after at most $t$ time points by adding $\Diamond_{[0,t]} \bigwedge_{i=1}^{n} has\_run_{j_i}$.

## 7 Conclusion

We presented and evaluated experimentally four translations from MTL to LTL. These translations provide a route to practical reasoning about MTL over natural numbers via LTL solvers, which allow us to solve instances of classical problems such as the MJS problem (see `http://cgi.csc.liv.ac.uk/~ullrich/MTL/` for experimental results). As future work, we intend to investigate whether we can translate PDDL3.0 statements (see [12]) into MTL formulae and apply our translations so as to make use of LTL provers in the planning domain.

## References


1. Abbas, H., Fainekos, G., Sankaranarayanan, S., Ivančić, F., Gupta, A.: Probabilistic temporal logic falsification of cyber-physical systems. ACM Transactions on Embedded Computing Systems (TECS) 12(2s), 95:1–95:30 (2013)
2. Alur, R., Feder, T., Henzinger, T.A.: The benefits of relaxing punctuality. J. ACM 43(1), 116–146 (1996)
3. Alur, R., Henzinger, T.A.: Real-time logics: Complexity and expressiveness. Inf. Comput. 104(1), 35–77 (1993)
4. Alur, R., Henzinger, T.A.: A really temporal logic. J. ACM 41(1), 181–204 (1994)
5. Bersani, M.M., Rossi, M., San Pietro, P.: A tool for deciding the satisfiability of continuous-time metric temporal logic. Acta Informatica 53(2), 171–206 (2016)
6. Bouyer, P., Markey, N., Ouaknine, J., Worrell, J.: The cost of punctuality. In: Proc. LICS 2007. pp. 109–120. IEEE (2007)
7. Cimatti, A., Clarke, E.M., Giunchiglia, E., Giunchiglia, F., Pistore, M., Roveri, M., Sebastiani, R., Tacchella, A.: NuSMV 2: An OpenSource tool for symbolic model checking. In: Proc. CAV 2002. LNCS, vol. 2404, pp. 359–364. Springer (2002)
8. Dauzère-Pérès, S., Paulli, J.: An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem using tabu search. Annals of Operations Research 70, 281–306 (1997)
9. Dixon, C., Fisher, M., Konev, B.: Temporal Logic with Capacity Constraints. In: Proc. FroCoS 2007. LNCS, vol. 4720, pp. 163–177. Springer (2007)
10. Fisher, M.: A normal form for temporal logics and its applications in theorem-proving and execution. Journal of Logic and Computation 7(4), 429–456 (1997)
11. Gabbay, D., Pnueli, A., Shelah, S., Stavi, J.: On the temporal analysis of fairness. In: Proc. POPL '80. pp. 163–173. ACM (1980)
12. Gerevini, A., Haslum, P., Long, D., Saetti, A., Dimopoulos, Y.: Deterministic planning in the fifth international planning competition: PDDL3 and experimental evaluation of the planners. Artificial Intelligence 173(5-6) (2009)
13. Goré, R.: And-or tableaux for fixpoint logics with converse: LTL, CTL, PDL and CPDL. In: Proc. IJCAR 2014. LNCS, vol. 8562, pp. 26–45. Springer (2014)
14. Graham, R.L.: Bounds for certain multiprocessing anomalies. Bell Labs Technical Journal 45(9), 1563–1581 (1966)
15. Gunadi, H., Tiu, A.: Efficient runtime monitoring with metric temporal logic: A case study in the Android operating system. In: Proc. FM 2014. LNCS, vol. 8442, pp. 296–311. Springer (2014)
16. Hustadt, U., Konev, B.: TRP++2.0: A temporal resolution prover. In: Proc. CADE-19. LNCS, vol. 2741, pp. 274–278. Springer (2003)
17. Karaman, S., Frazzoli, E.: Vehicle routing problem with metric temporal logic specifications. In: Proc. CDC 2008. pp. 3953–3958. IEEE (2008)
18. Ouaknine, J., Worrell, J.: Some recent results in metric temporal logic. In: Proc. FORMATS 2008. LNCS, vol. 5215, pp. 1–13. Springer (2008)
19. Pnueli, A.: The temporal logic of programs. In: Proc. SFCS '77. pp. 46–57. IEEE (1977)
20. Schuppan, V., Darmawan, L.: Evaluating LTL satisfiability solvers. In: Proc. ATVA 2011. LNCS, vol. 6996, pp. 397–413. Springer (2011)
21. Sistla, A.P., Clarke, E.M.: The complexity of propositional linear temporal logics. J. ACM 32(3), 733–749 (1985)
22. Thati, P., Roşu, G.: Monitoring algorithms for metric temporal logic specifications. Electronic Notes in Theoretical Computer Science 113, 145–162 (2005)
23. Tseitin, G.S.: On the complexity of derivation in propositional calculus. In: Automation of reasoning, pp. 466–483. Springer (1983)

## A Proofs for 'MTL to LTL translation: encoding 'gaps' '

The following two propositions are useful for the proof of Theorem 1. Since *gap* is a propositional symbol not occurring in $\sigma$, the time points mapped by the image of $\tau$ do not contain *gap*. Then, it is easy to see the following.

**Proposition 1.** *Given a timed state sequence $\rho = (\sigma, \tau)$, let $\sigma'$ be as in Definition 1. Then, $(\sigma', 0) \models \Box(\Diamond \neg gap)$.*

The state sequence $\sigma'$ in Definition 2 is such that $(\sigma', 0) \models \neg gap \wedge \Box(\Diamond \neg gap)$. Consequently, for the timed state sequence constructed in Definition 2 we have that for each $i \in \mathbb{N}$, $\tau(i) \in \mathbb{N}$. Also, for $i > 0$, $\tau(i) > \tau(i-1)$ and, so, $\tau : \mathbb{N} \to \mathbb{N}$ is well defined. The following proposition states this property.

**Proposition 2.** *Given a state sequence $\sigma'$ such that $(\sigma', 0) \models \neg gap \wedge \Box(\Diamond \neg gap)$, let $\rho = (\sigma, \tau)$ be as in Definition 2. Then, $\tau : \mathbb{N} \to \mathbb{N}$ is a function such that $\tau(i+1) > \tau(i)$, for all $i \in \mathbb{N}$.*

We now show Theorem 1, one can use similar arguments to show Theorem 2.

**Theorem 1 (restated).** Let $\varphi = p_0 \wedge \bigwedge_i \Box_{[0,\infty)}(p_i \to \psi_i)$ be an MTL formula in NNF and FNF. Let $\varphi^\sharp = p_0 \wedge \bigwedge_i \Box(p_i \to (\neg gap \wedge \psi_i^\sharp))$ be the result of replacing each $\psi_i$ in $\varphi$ by $\psi_i^\sharp$ as in Table 1. Then, $\varphi$ is satisfiable if, and only if, $\varphi^\sharp \wedge \neg gap \wedge \Box(\Diamond \neg gap)$ is satisfiable.

*Proof.* ($\Rightarrow$) If $\varphi$ is satisfiable then there is a timed state sequence $\rho = (\sigma, \tau)$ such that $(\rho, 0) \models \varphi$. Let $\sigma' = \sigma'_0 \sigma'_1 \ldots$ be as in Definition 1. We need to show that $(\sigma', \tau(0)) \models \varphi^\sharp \wedge \neg gap \wedge \Box(\Diamond \neg gap)$. By Definition 1 we have that $(\sigma', \tau(0)) \models \neg gap$ and, by Proposition 1, $(\sigma', \tau(0)) \models \Box(\Diamond \neg gap)$. Also, by definition of $\sigma'$, for all propositional variables $p_i$ occurring in $\sigma$ and all $j \in \mathbb{N}$, we have $(\rho, j) \models p_i$ iff $(\sigma', \tau(j)) \models p_i$. Clearly, for any propositional formula $\alpha$, $(\rho, j) \models \alpha$ iff $(\sigma', \tau(j)) \models \alpha$. Then, following our translation in Table 1, we only need to show that $(\rho, j) \models \psi_i$ implies $(\sigma', \tau(j)) \models \neg gap \wedge \psi_i^\sharp$, $i, j \in \mathbb{N}$. This follows from Claims 1, 2 and 3 below (other cases are similar).

**Claim 1** *For $c_1 > 0$, if $(\rho, j) \models \bigcirc_{[c_1,c_2]} \alpha$ then $(\sigma', \tau(j)) \models \neg gap \wedge (\bigvee_{c_1 \leq l \leq c_2} (\bigcirc^l (\neg gap \wedge \alpha) \wedge \bigwedge_{1 \leq k < l} \bigcirc^k gap))$.*

If $(\rho, j) \models \bigcirc_{[c_1,c_2]} \alpha$ then $(\rho, j+1) \models \alpha$ and $\tau(j+1) \in \tau(j) + [c_1, c_2]$. By definition of $\sigma'$, $(\sigma', \tau(j+1)) \models \neg gap \wedge \alpha$. Also, there is no $n \in \mathbb{N}$ such that $\tau(n) = m$, for $\tau(j) < m < \tau(j+1)$, meaning that

for all such $m$, $(\sigma', m) \models gap$. As $\tau(j) + c_1 \leq \tau(j+1) \leq \tau(j) + c_2$, we have $(\sigma', \tau(j)) \models \bigvee_{c_1 \leq l \leq c_2}(\bigcirc^l(\neg gap \wedge \alpha) \wedge \bigwedge_{1 \leq k < l} \bigcirc^k gap)$. By definition of $\sigma'$, we also have that $(\sigma', \tau(j)) \models \neg gap$. So $(\sigma', \tau(j)) \models \neg gap \wedge (\bigvee_{c_1 \leq l \leq c_2}(\bigcirc^l(\neg gap \wedge \alpha) \wedge \bigwedge_{1 \leq k < l} \bigcirc^k gap))$.

**Claim 2** *For $c_1 > 0$, if $(\rho, j) \models \alpha \mathcal{U}_{[c_1,c_2]} \beta$ then $(\sigma', \tau(j)) \models \neg gap \wedge (\bigvee_{c_1 \leq l \leq c_2}(\bigcirc^l(\neg gap \wedge \beta) \wedge \bigwedge_{0 \leq k < l} \bigcirc^k(gap \vee \alpha)))$.*

If $(\rho, j) \models \alpha \mathcal{U}_{[c_1,c_2]} \beta$ then there is $k \in \mathbb{N}$ such that $\tau(k) \in \tau(j) + [c_1, c_2]$ and $(\rho, k) \models \beta$ and for all $l \in \mathbb{N}$ with $j \leq l < k$ we have $(\rho, l) \models \alpha$. By definition of $\sigma'$, $(\sigma', \tau(k)) \models \neg gap \wedge \beta$ and, as $\tau(j) + c_1 \leq \tau(k) \leq \tau(j) + c_2$, we have that $(*)$: $(\sigma', \tau(j)) \models \bigvee_{c_1 \leq m \leq c_2}(\bigcirc^m(\neg gap \wedge \beta))$. Also, by definition of $\sigma'$, for all $l \in \mathbb{N}$ with $j \leq l < k$ we have $(\sigma', \tau(l)) \models \alpha$. If $n \in \mathbb{N}$ is such that $\tau(j) \leq n < \tau(k)$ and there is no $l \in \mathbb{N}$ with $n = \tau(l)$ then, by definition of $\sigma'$, $(\sigma', n) \models gap$. Then, $(\sigma', \tau(j)) \models \bigwedge_{0 \leq k < l} \bigcirc^k(gap \vee \alpha)$ and, by $(*)$, $(\sigma', \tau(j)) \models \bigvee_{c_1 \leq l \leq c_2}(\bigcirc^l(\neg gap \wedge \beta) \wedge \bigwedge_{0 \leq k < l} \bigcirc^k(gap \vee \alpha))$.

**Claim 3** *For $c_1 > 0$, if $(\rho, j) \models \alpha \tilde{\mathcal{U}}_{[c_1,c_2]} \beta$ then $(\sigma', \tau(j)) \models \neg(\neg(\alpha)\mathcal{U}_{[c_1,c_2]}\neg(\beta))^\sharp$.*

If $(\rho, j) \models \alpha \tilde{\mathcal{U}}_{[c_1,c_2]} \beta$ then, by semantics of $\tilde{\mathcal{U}}$, $(\rho, j) \models \neg(\neg(\alpha)\mathcal{U}_{[c_1,c_2]}\neg(\beta))$. By Claim 5, if $(\rho, j) \not\models \neg(\alpha)\mathcal{U}_{[c_1,c_2]}\neg(\beta)$ then $(\sigma', \tau(j)) \not\models \neg gap \wedge (\bigvee_{c_1 \leq l \leq c_2}(\bigcirc^l(\neg gap \wedge \neg\beta) \wedge \bigwedge_{0 \leq k < l} \bigcirc^k(gap \vee \neg\alpha))$. That is, $(\sigma', \tau(j)) \models \neg(\neg(\alpha)\mathcal{U}_{[c_1,c_2]}\neg(\beta))^\sharp$.

($\Leftarrow$) If $\varphi^\sharp \wedge \neg gap \wedge \Box(\Diamond \neg gap)$ is satisfiable then there is a state sequence $\sigma'$ such that $(\sigma', 0) \models \varphi^\sharp \wedge \neg gap \wedge \Box(\Diamond \neg gap)$. Let $\rho = (\sigma, \tau)$ be a timed state sequence as in Definition 2. By Proposition 2, $\tau : \mathbb{N} \to \mathbb{N}$ is well defined. We need to show that $(\rho, 0) \models \varphi$. By definition of $\sigma$, for all propositional variables $p_i$ occurring in $\sigma'_n$ with $gap \notin \sigma'_n$ there is $j \in \mathbb{N}$ such that $\tau(j) = n$ and $(\rho, j) \models p_i$ iff $(\sigma', \tau(j)) \models p_i$. Clearly, for any propositional formula $\alpha$, $(\rho, j) \models \alpha$ iff $(\sigma', \tau(j)) \models \alpha$. Then, following our translation in Table 1, we only need to show that $(\sigma', \tau(j)) \models \neg gap \wedge \psi_i^\sharp$ implies $(\rho, j) \models \psi_i$, $i, j \in \mathbb{N}$. This follows from Claims 4, 5 and 6 below (other cases are similar).

**Claim 4** *For $c_1 > 0$, if $(\sigma', j) \models \neg gap \wedge (\bigvee_{c_1 \leq l \leq c_2}(\bigcirc^l(\neg gap \wedge \alpha) \wedge \bigwedge_{1 \leq k < l} \bigcirc^k gap))$ then $(\rho, i) \models \bigcirc_{[c_1,c_2]} \alpha$, where $j = \tau(i)$.*

If $(\sigma', j) \models \neg gap \wedge (\bigvee_{c_1 \leq l \leq c_2}(\bigcirc^l(\neg gap \wedge \alpha) \wedge \bigwedge_{1 \leq k < l} \bigcirc^k gap))$ then there is $l$, $c_1 \leq l \leq c_2$, such that $(\sigma', j + l) \models \neg gap \wedge \alpha$ and for all $k$, $1 \leq k < l$, $(\sigma', j + k) \models gap$. If $(\sigma', j) \models \neg gap$ then there is $i \in \mathbb{N}$ such that $\tau(i) = j$. As $(\sigma', j + l) \models \neg gap$ and, for all $k$, $1 \leq k < l$, $(\sigma', j + k) \models gap$, we have that $\tau(i + 1) = j + l$. As

$(\sigma', j+l) \models \alpha$, by Definition 2, $(\rho, i+1) \models \alpha$. Since $c_1 \leq l \leq c_2$, we have that $(\rho, i) \models \bigcirc_{[c_1,c_2]}\alpha$.

**Claim 5** *For $c_1 > 0$, if $(\sigma', j) \models \neg gap \wedge (\bigvee_{c_1 \leq l \leq c_2}(\bigcirc^l(\neg gap \wedge \beta) \wedge \bigwedge_{0 \leq k < l} \bigcirc^k(gap \vee \alpha)))$ then $(\rho, i) \models \alpha \mathcal{U}_{[c_1,c_2]}\beta$, where $j = \tau(i)$.*

If $(\sigma', j) \models \neg gap \wedge (\bigvee_{c_1 \leq l \leq c_2}(\bigcirc^l(\neg gap \wedge \beta) \wedge \bigwedge_{0 \leq k < l} \bigcirc^k(gap \vee \alpha)))$ then there is $l$, $c_1 \leq l \leq c_2$, such that $(\sigma', j+l) \models \neg gap \wedge \beta$ and for all $k$, $1 \leq k < l$, $(\sigma', j+k) \models gap \vee \alpha$. As $(\sigma', j) \models \neg gap$, by definition of $\rho$, there is $i \in \mathbb{N}$ such that $\tau(i) = j$. Also, as $(\sigma', j+l) \models \neg gap \wedge \beta$, there is $i_0 \in \mathbb{N}$ such that $\tau(i_0) = j + l$ and $(\rho, i_0) \models \neg gap \wedge \beta$. Since for all $k$, $1 \leq k < l$, $(\sigma', j+k) \models gap \vee \alpha$, we also have that for all $n$, $j+1 \leq n < j+l$, if there is $m \in \mathbb{N}$ such that $\tau(m) = n$ then $(\rho, m) \models \alpha$. Then, $(\rho, i) \models \alpha \mathcal{U}_{[c_1,c_2]}\beta$.

**Claim 6** *For $c_1 > 0$, if $(\sigma', j) \models \neg(\neg(\alpha)\mathcal{U}_{[c_1,c_2]}\neg(\beta))^\sharp$ then $(\rho, i) \models \alpha \tilde{\mathcal{U}}_{[c_1,c_2]}\beta$, where $j = \tau(i)$.*

If $(\sigma', j) \not\models (\neg(\alpha)\mathcal{U}_{[c_1,c_2]}\neg(\beta))^\sharp$ then, by Claim 2, $(\rho, i) \not\models \neg(\alpha)\mathcal{U}_{[c_1,c_2]}\neg(\beta)$, where $j = \tau(i)$. By semantics of $\tilde{\mathcal{U}}$, we have that $(\rho, i) \models \alpha \tilde{\mathcal{U}}_{[c_1,c_2]}\beta$.

❑

**Example** Assume that we are given the following MTL formula in NNF and FNF: $\varphi = p_0 \wedge \Box_{[0,\infty)}(p_0 \rightarrow \bigcirc_{[2,3]}p_1) \wedge \Box_{[0,\infty)}(p_1 \rightarrow \Diamond_{[1,2]}\neg q)$. Using Table 1, we translate $\varphi$ into LTL as follows (recall that $\Diamond_I \psi \equiv \mathbf{true}\mathcal{U}_I\psi$):

$$\begin{aligned}\varphi^\sharp = p_0 \wedge \Box_{[0,\infty)}(p_0 \rightarrow (\neg gap \wedge (\textstyle\bigvee_{2 \leq l \leq 3}(\bigcirc^l(\neg gap \wedge p_1) \wedge \bigwedge_{1 \leq k < l} \bigcirc^k gap)) \\ \wedge \Box_{[0,\infty)}(p_1 \rightarrow (\neg gap \wedge (\textstyle\bigvee_{1 \leq l \leq 2}(\bigcirc^l(\neg gap \wedge \neg q))))\end{aligned}$$

By Theorem 1, $\varphi$ is satisfiable iff $\varphi^\sharp \wedge \neg gap \wedge \Box(\Diamond \neg gap)$ is satisfiable.

## B Proofs for 'MTL to LTL translation: encoding time differences'

For the sake of completeness, we restate and prove Theorem 3. Let $C - 1$ be the greatest number occurring in an interval in an MTL formula $\varphi$ or 1, if none occur.

**Definition 7.** *Given a timed sequence $\rho = (\sigma, \tau)$ and $C \in \mathbb{N}$, we define a timed sequence $\rho_C = (\sigma_C, \tau_C)$ as follows:*

- $\sigma_C = \sigma$;

– $\tau_C(0) = min(\tau(0), C)$ *and, for* $i > 0$, $\tau_C(i) = \tau_C(i-1) + min(C, \tau(i) - \tau(i-1))$.

The following proposition states the main property of Definition 7.

**Proposition 3.** *Let* $\rho = (\sigma, \tau)$ *be a timed state sequence and let* $\rho_C = (\sigma_C, \tau_C)$ *be as in Definition 7. For all* $i, j \in \mathbb{N}$ *and all intervals of the form* $I = [c_1, c_2]$ *or* $I = [c_1, \infty)$ *with* $c_1, c_2 < C$*, the following holds:*

$$\tau(j) \in \tau(i) + I \Leftrightarrow \tau_C(j) \in \tau_C(i) + I$$

*Proof.* First assume $I = [c_1, c_2]$. If $\tau(j) \in \tau(i) + [c_1, c_2]$ then $\tau(j) - \tau(i) \leq c_2 < C$ and, so, $\tau_C(j) - \tau_C(i) = \tau(j) - \tau(i)$. Thus, $\tau_C(j) \in \tau_C(i) + [c_1, c_2]$. Conversely, if $\tau_C(j) \in \tau_C(i) + [c_1, c_2]$ then $\tau_C(j) - \tau_C(i) \leq c_2 < C$ and, so, $\tau_C(j) - \tau_C(i) = \tau(j) - \tau(i)$. Thus, $\tau(j) \in \tau(i) + [c_1, c_2]$.

Now assume $I = [c_1, \infty)$. If $\tau(j) \in \tau(i) + [c_1, \infty)$ then $\tau(j) - \tau(i) \geq c_1$. If $\tau(j) - \tau(i) < C$ then $\tau_C(j) - \tau_C(i) = \tau(j) - \tau(i)$ and, so, $\tau_C(j) \in \tau_C(i) + [c_1, \infty)$. Otherwise $\tau(j) - \tau(i) \geq C$. Then, $\tau_C(j) - \tau_C(i) = C$. As $C > c_1$, we have that $\tau_C(j) \in \tau_C(i) + [c_1, \infty)$. Conversely, if $\tau_C(j) \in \tau_C(i) + [c_1, \infty)$ then $\tau_C(j) - \tau_C(i) \geq c_1$. If $\tau_C(j) - \tau_C(i) < C$ then $\tau_C(j) - \tau_C(i) = \tau(j) - \tau(i)$ and, so, $\tau(j) \in \tau(i) + [c_1, \infty)$. Otherwise $\tau_C(j) - \tau_C(i) = C$. Then, $\tau(j) - \tau(i) \geq C$. As $C > c_1$, we have that $\tau(j) \in \tau(i) + [c_1, \infty)$. ❑

We are now ready for Theorem 3.

**Theorem 3 Adaptation from [4] (restated).** If there is a timed state sequence $\rho = (\sigma, \tau)$ such that $(\rho, 0) \models \varphi$ then there is a $C$-bounded timed state sequence $\rho_C$ such that $(\rho_C, 0) \models \varphi$.

*Proof.* Let $\rho = (\sigma, \tau)$ be a timed state sequence and let $\rho_C = (\sigma_C, \tau_C)$ be as in Definition 7. By definition of $\rho_C$ we have that $\rho_C$ is $C$-bounded.

Assume w.l.o.g. that $\varphi$ is in NNF. Let $\mathsf{sub}(\varphi)$ be the set of all subformulae of $\varphi$. To prove this lemma, we argue by structural induction and show that for all $\varphi' \in \mathsf{sub}(\varphi)$, if $(\rho, i) \models \varphi'$ then $(\rho_C, i) \models \varphi'$, $i \in \mathbb{N}$.

In the base case, $\varphi'$ is a propositional formula. Then, as $\sigma_C = \sigma$, we have that $(\rho, i) \models \varphi'$ implies $(\rho_C, i) \models \varphi'$, $i \in \mathbb{N}$. Suppose that, for $\chi, \psi \in \mathsf{sub}(\varphi)$ and $i \in \mathbb{N}$, $(\rho, i) \models \chi$ implies $(\rho_C, i) \models \chi$, and, $(\rho, i) \models \psi$ implies $(\rho_C, i) \models \psi$. We explain for $\bigcirc_I, \mathcal{U}_I$ and $\tilde{\mathcal{U}}_I$ (other cases are similar):

– $\varphi'$ is of the form $\bigcirc_I \chi$: if $(\rho, i) \models \bigcirc_I \chi$ then $\tau(i+1) \in \tau(i) + I$ and $(\rho, i+1) \models \psi$. By induction hypothesis, $(\rho_C, i+1) \models \psi$. By Proposition 3, $\tau_C(i+1) \in \tau_C(i) + I$. Then, $(\rho_C, i) \models \bigcirc_I \chi$.

- $\varphi'$ is of the form $\chi \mathcal{U}_I \psi$: if $(\rho, i) \models \chi \mathcal{U}_I \psi$ then there is $k \in \mathbb{N}$ such that $k \geq i$, $\tau(k) \in \tau(i) + I$ and $(\rho, k) \models \psi$ and for all $j \in \mathbb{N}$, if $i \leq j < k$ then $(\rho, j) \models \chi$. By induction hypothesis, $(\rho_C, k) \models \psi$ and $(\rho_C, j) \models \chi$, for all $j \in \mathbb{N}$ with $i \leq j < k$. By Proposition 3, $\tau_C(k) \in \tau_C(i) + I$. Then, $(\rho_C, i) \models \chi \mathcal{U}_I \psi$.
- $\varphi'$ is of the form $\chi \tilde{\mathcal{U}}_I \psi$: if $(\rho, i) \models \chi \tilde{\mathcal{U}}_I \psi$ then either:
  1. for all $j \in \mathbb{N}$, if $\tau(j) \in \tau(i) + I$ then $(\rho, j) \models \psi$; or
  2. (if $c_1 > 0$) $(\rho, k) \models \chi$ for some $k \in \mathbb{N}$ such that $\tau(k) \in \tau(i) + [0, c_1 - 1]$, where $c_1$ is the left end-point of $I$; or
  3. $(\rho, l) \models \chi$ for some $l \in \mathbb{N}$ such that $\tau(l) \in \tau(i) + I$ and for all $l' \in \mathbb{N}$ such that $\tau(i) + c_1 \leq \tau(l') \leq \tau(l)$, we have $(\rho, l') \models \psi$.

  We make a case distinction. In Case (1), we have to establish that, for all $j \in \mathbb{N}$, if $\tau_C(j) \in \tau_C(i) + I$ then $(\rho_C, j) \models \psi$. By Proposition 3, if $\tau_C(j) \in \tau_C(i) + I$ then $\tau(j) \in \tau(i) + I$ and by induction hypothesis, $(\rho_C, j) \models \psi$, for all $j \in \mathbb{N}$ with $\tau(j) \in \tau(i) + I$. Therefore, $(\rho_C, i) \models \chi \tilde{\mathcal{U}}_I \psi$. Cases (2) and (3) can be proved with similar arguments.

❑

The following two propositions are useful for the proof of Theorem 4.

**Proposition 4.** *Given a $C$-bounded timed state sequence $\rho = (\sigma, \tau)$, let $\sigma'$ be as in Definition 5. For all $i < j \in \mathbb{N}$, if $\tau(j) - \tau(i) = m \leq 2 \cdot C$ then $(\sigma', j) \models s^n_m$ where $n = j - i$.*

*Proof.* In the base case $j = i + 1$, that is, $n = 1$. As $\rho$ is $C$-bounded, $\tau(i+1) - \tau(i) = k$, for $1 \leq k \leq C$. Then, by Definition 5, $(\sigma', j) \models s^1_k$. Suppose that, for $j = i + n$, if $\tau(i+n) - \tau(i) = l$ then $(\sigma', i+n) \models s^n_l$ where $l \leq 2 \cdot C$. In the induction step we consider $j = i + n + 1$. Let $k'$ be such that $\tau(i+n+1) - \tau(i+n) = k'$. As $\tau(i+n) - \tau(i) = l$ and $\tau(i+n+1) - \tau(i+n) = k'$, we have that $\tau(i+n+1) - \tau(i) = l + k'$. By Definition 5, $(\sigma', i+n+1) \models s^1_{k'}$. By induction hypothesis, $(\sigma', j) \models s^n_l$. By Definition 5, if $l + k' \leq 2 \cdot C$, then $(\sigma', i+n+1) \models s^{n+1}_{l+k'}$. ❑

**Proposition 5.** *Given a state sequence $\sigma' = \sigma'_0 \sigma'_1 \ldots$ such that $(\sigma', 0) \models S_C$, let $\rho$ be as in Definition 6. For all $i < j \in \mathbb{N}$, if $(\sigma', j) \models s^n_m$ with $n = j - i$ then $\tau(j) - \tau(i) = \mathsf{min}(m, 2 \cdot C)$.*

*Proof.* In the base case $j = i + 1$, that is, $n = 1$. Assume that $(\sigma', i+1) \models s^1_k$. By definition of $S_C$, $(\sigma', i+1) \models \delta^-_k$. Then, by Definition 6, $\tau(i+1) - \tau(i) = k \leq C$, $i \in \mathbb{N}$. Suppose that, for $j = i + n$, $(\sigma', i+n) \models s^n_m$ implies $\tau(i+n) - \tau(i) = \mathsf{min}(m, 2 \cdot C)$. In the induction step we consider $j = i + n + 1$. Assume that $(\sigma', i+n+1) \models s^{n+1}_l$. By Points 1 and 2 of

the definition of $S_C$, there is $1 \leq k' \leq C$ such that $(\sigma', i+n+1) \models s^1_{k'}$. As $(\sigma', i+n) \models s^n_m$ and $(\sigma', i+n+1) \models s^1_{k'}$, we have that, by Point 4 of the definition of $S_C$, $(\sigma', i+n+1) \models s^{n+1}_{\mathsf{min}(m+k', 2\cdot C)}$. Also, by Point 3 of the definition, there is no $l \neq m+k'$ such that $(\sigma', i+n+1) \models s^{n+1}_l$. Then, we can assume that $l = \mathsf{min}(m+k', 2\cdot C)$. By induction hypothesis, $\tau(i+n) - \tau(i) = \mathsf{min}(m, 2\cdot C)$ and, by Definition 6, $\tau(i+n+1) - \tau(i+n) = k'$. Then $\tau(i+n+1) - \tau(i) = \mathsf{min}(m+k', 2\cdot C)$. ❑

We now show Theorem 4, one can use similar arguments to show Theorem 5.

**Theorem 4 (restated).** Let $\varphi = p_0 \wedge \bigwedge_i \Box_{[0,\infty)}(p_i \rightarrow \psi_i)$ be an MTL formula in NNF and FNF. Let $\varphi^\sharp = p_0 \wedge \bigwedge_i \Box(p_i \rightarrow \psi_i^\sharp)$ be the result of replacing each $\psi_i$ in $\varphi$ by $\psi_i^\sharp$ as in Table 3. Then, $\varphi$ is satisfiable if, and only if, $\varphi^\sharp \wedge S_C$ is satisfiable.

*Proof.* ($\Rightarrow$) If $\varphi$ is satisfiable then there is a timed state sequence $\rho = (\sigma, \tau)$ such that $(\rho, 0) \models \varphi$. By Theorem 3, we can assume w.l.g. that $\rho$ is $C$-bounded, where $C-1$ is the largest constant in $\varphi$. Let $\sigma' = \sigma'_0\sigma'_1\ldots$ be as in Definition 5. By Lemma 1, $(\sigma', 0) \models S_C$. So we need to show that $(\sigma', 0) \models \varphi^\sharp$. By definition of $\sigma'$, for all propositional variables $p_i$ occurring in $\sigma$ and all $j \in \mathbb{N}$, we have $(\rho, j) \models p_i$ iff $(\sigma', j) \models p_i$. Clearly, for any propositional formula $\alpha$, $(\rho, j) \models \alpha$ iff $(\sigma', j) \models \alpha$. Then, following our translation in Table 3, we only need to show that $(\rho, j) \models \psi_i$ implies $(\sigma', j) \models \psi_i^\sharp$, $i, j \in \mathbb{N}$. This follows from Claims 7, 8, 9 and 10 below (other cases are similar).

**Claim 7** *For $c_1 > 0$, if $(\rho, j) \models \bigcirc_{[c_1,c_2]}\alpha$ then $(\sigma', j) \models \bigcirc((\bigvee_{c_1 \leq i \leq c_2} \delta_i^-) \wedge \alpha)$.*

If $(\rho, j) \models \bigcirc_{[c_1,c_2]}\alpha$, with $c_1 > 0$, then $(\rho, j+1) \models \alpha$ and $\tau(j+1) \in \tau(j) + [c_1, c_2]$. By definition of $\sigma'$, $(\sigma', j+1) \models \alpha$ and $(\sigma', j+1) \models \bigvee_{c_1 \leq i \leq c_2} \delta_i^-$. Then, $(\sigma', j) \models \bigcirc((\bigvee_{c_1 \leq i \leq c_2} \delta_i^-) \wedge \alpha)$.

**Claim 8** *For $c_1 > 0$, if $(\rho, j) \models \alpha \mathcal{U}_{[c_1,c_2]}\beta$ then $(\sigma', j) \models \bigvee_{1 \leq i \leq c_2}(\bigcirc^i((\bigvee_{c_1 \leq l \leq c_2} s^i_l) \wedge \beta) \wedge \bigwedge_{0 \leq k < i} \bigcirc^k \alpha)$.*

If $(\rho, j) \models \alpha \mathcal{U}_{[c_1,c_2]}\beta$, with $c_1 > 0$, then there is $m \in \mathbb{N}$ such that $m \geq j$, $\tau(m) \in \tau(j) + [c_1, c_2]$ and $(\rho, m) \models \beta$ and for all $n \in \mathbb{N}$, if $j \leq n < m$ then $(\rho, n) \models \alpha$. By definition of $\sigma'$, $(\sigma', m) \models \beta$ and for all $n \in \mathbb{N}$, if $j \leq n < m$ then $(\sigma', n) \models \alpha$. As $\tau(m) - \tau(j) \leq c_2 < C$, by Proposition 4, $(\sigma', m) \models s^{m-j}_{\tau(m)-\tau(j)}$. As $c_1 \leq \tau(m) - \tau(j) \leq c_2$, we have $(\sigma', m) \models \bigvee_{c_1 \leq l \leq c_2} s^{m-j}_l$. Since $c_1 > 0$ and $\tau$ is strictly monotonically increasing, we have $1 \leq m - j \leq c_2$. Then, $(\sigma', j) \models \bigvee_{1 \leq i \leq c_2}(\bigcirc^i(\bigvee_{c_1 \leq l \leq c_2} s^i_l) \wedge \beta)$. As $(\sigma', n) \models \alpha$ for all $n \in \mathbb{N}$ such

that $j \leq n < m$, we have that $(\sigma', j) \models \bigwedge_{0 \leq k < m-j} \bigcirc^k \alpha$. Thus, $(\sigma', j) \models \bigvee_{1 \leq i \leq c_2} (\bigcirc^i ((\bigvee_{c_1 \leq l \leq c_2} s^i_l) \wedge \beta) \wedge \bigwedge_{0 \leq k < i} \bigcirc^k \alpha)$.

**Claim 9** *For $c_1 > 0$, if $(\rho, j) \models \alpha \mathcal{U}_{[c_1,\infty)} \beta$ then $(\sigma', j) \models \bigvee_{1 \leq i \leq c_1} (\bigcirc^i ((\bigvee_{c_1 \leq l \leq c_1 + C} s^i_l) \wedge \alpha \mathcal{U} \beta) \wedge \bigwedge_{0 \leq k < i} \bigcirc^k \alpha)$.*

If $(\rho, j) \models \alpha \mathcal{U}_{[c_1,\infty)} \beta$, with $c_1 > 0$, then there is $m \in \mathbb{N}$ such that $m \geq j$, $\tau(m) \in \tau(j) + [c_1, \infty)$ and $(\rho, m) \models \beta$ and for all $n \in \mathbb{N}$, if $j \leq n < m$ then $(\rho, n) \models \alpha$. Let $i \in \mathbb{N}$ be minimum such that $\tau(i) \geq \tau(j) + c_1$. That is, $\tau(i-1) < \tau(j) + c_1$ (as $c_1 > 0$ we know that such $i$ exists). As $\rho$ is $C$-bounded, by minimality of $i$, we have that $\tau(i) - \tau(j) \leq c_1 + C$, $(\rho, i) \models \alpha \mathcal{U}_{[0,\infty)} \beta$ and for all $n \in \mathbb{N}$, if $j \leq n < i$ then $(\rho, n) \models \alpha$.

By definition of $\sigma'$, $(\sigma', i) \models \alpha \mathcal{U} \beta$ and for all $n \in \mathbb{N}$, if $j \leq n < i$ then $(\sigma', n) \models \alpha$. As $\tau(i) - \tau(j) \leq c_1 + C < 2 \cdot C$, by Proposition 4, $(\sigma', i) \models s^{i-j}_{\tau(i)-\tau(j)}$. As $c_1 \leq \tau(i) - \tau(j) \leq c_1 + C$, we have $(\sigma', i) \models \bigvee_{c_1 \leq l \leq c_1 + C} s^{i-j}_l$. Since $\tau$ is strictly monotonically increasing, we have that $j < i \leq j + c_1$. Then, $(\sigma', j) \models \bigvee_{1 \leq i' \leq c_1} (\bigcirc^{i'} (\bigvee_{c_1 \leq l \leq c_1 + C} s^{i'}_l) \wedge \alpha \mathcal{U} \beta)$. As $(\sigma', n) \models \alpha$ for all $n \in \mathbb{N}$ such that $j \leq n < i$, we have that $(\sigma', j) \models \bigwedge_{0 \leq k < i-j} \bigcirc^k \alpha$. Thus, $(\sigma', j) \models \bigvee_{1 \leq i' \leq c_1} (\bigcirc^{i'} (\bigvee_{c_1 \leq l \leq c_1 + C} s^{i'}_l) \wedge \alpha \mathcal{U} \beta) \wedge \bigwedge_{0 \leq k < i'} \bigcirc^k \alpha)$.

**Claim 10** *For $c_1 > 0$, if $(\rho, j) \models \alpha \tilde{\mathcal{U}}_{[c_1,c_2]} \beta$ then $(\sigma', j) \models \neg(\neg(\alpha) \mathcal{U}_{[c_1,c_2]} \neg(\beta))^\sharp$.*

If $(\rho, j) \models \alpha \tilde{\mathcal{U}}_{[c_1,c_2]} \beta$ then, by semantics of $\tilde{\mathcal{U}}$, $(\rho, j) \models \neg(\neg(\alpha) \mathcal{U}_{[c_1,c_2]} \neg(\beta))$. By Claim 12, if $(\rho, j) \not\models \neg(\alpha) \mathcal{U}_{[c_1,c_2]} \neg(\beta)$ then $(\sigma', j) \not\models \bigvee_{1 \leq i \leq c_2} (\bigcirc^i ((\bigvee_{c_1 \leq l \leq c_2} s^i_l) \wedge \neg(\beta)) \wedge \bigwedge_{0 \leq k < i} \bigcirc^k \neg(\alpha))$. That is, $(\sigma', j) \models \neg(\neg(\alpha) \mathcal{U}_{[c_1,c_2]} \neg(\beta))^\sharp$.

($\Leftarrow$) If $\varphi^\sharp \wedge S_C$ is satisfiable then there is a state sequence $\sigma'$ such that $(\sigma', 0) \models \varphi^\sharp \wedge S_C$. Let $\rho = (\sigma, \tau)$ be a $C$-bounded timed state sequence as in Definition 6. We need to show that $(\rho, 0) \models \varphi$. By definition of $\sigma$, for all propositional variables $p_i$ occurring in $\sigma'$ but not in $\Pi_\delta \cup \bigcup_{1 \leq i' \leq C} \Pi^{i'}$ and all $j \in \mathbb{N}$, we have $(\rho, j) \models p_i$ iff $(\sigma', j) \models p_i$. Clearly, for any propositional formula $\alpha$, $(\rho, j) \models \alpha$ iff $(\sigma', j) \models \alpha$. Then, following our translation in Table 3, we only need to show that $(\sigma', j) \models \psi^\sharp_i$ implies $(\rho, j) \models \psi_i$, $i, j \in \mathbb{N}$. This follows from Claims 11 and 12 below (other cases are similar).

**Claim 11** *For $c_1 > 0$, if $(\sigma', j) \models \bigcirc((\bigvee_{c_1 \leq i \leq c_2} \delta^-_i) \wedge \alpha)$ then $(\rho, j) \models \bigcirc_{[c_1,c_2]} \alpha$.*

If $(\sigma', j) \models \bigcirc((\bigvee_{c_1 \leq i \leq c_2} \delta^-_i) \wedge \alpha)$, with $c_1 > 0$, then, by definition of $\rho$, $c_1 \leq \tau(j+1) - \tau(j) \leq c_2$ and $(\rho, j+1) \models \alpha$. Then, $(\rho, j) \models \bigcirc_{[c_1,c_2]} \alpha$.

**Claim 12** *For $c_1 > 0$, if $(\sigma', j) \models \bigvee_{1\leq i\leq c_2}(\bigcirc^i((\bigvee_{c_1\leq l\leq c_2} s_l^i) \wedge \beta) \wedge \bigwedge_{0\leq k<i} \bigcirc^k \alpha)$ then $(\rho, j) \models \alpha \mathcal{U}_{[c_1,c_2]}\beta$.*

If $(\sigma', j) \models \bigvee_{1\leq i\leq c_2}(\bigcirc^i((\bigvee_{c_1\leq l\leq c_2} s_l^i) \wedge \beta) \wedge \bigwedge_{0\leq k<i} \bigcirc^k \alpha)$, with $c_1 > 0$, then there is $i$, $1 \leq i \leq c_2$, such that $(\sigma', j+i) \models \beta$ and, for all $k$, $0 \leq k < i$, $(\sigma', j+k) \models \alpha$. By definition of $\rho$, $(\rho, j+i) \models \beta$ and, for all $0 \leq k < i$, $(\rho, j+k) \models \alpha$. So it remains to show that $\tau(j+i) \in \tau(j) + [c_1, c_2]$. By Proposition 5, if $(\sigma', j+i) \models s_l^i$ and $c_1 \leq l \leq c_2 < 2 \cdot C$ then $l = \tau(j+i) - \tau(j)$. By definition of $S_C$, we can assume that there is only one $l$ such that $(\sigma', j+i) \models s_l^i$. As $c_1 \leq l \leq c_2$, $\tau(j+i) \in \tau(j) + [c_1, c_2]$. Thus, $(\rho, j) \models \alpha \mathcal{U}_{[c_1,c_2]}\beta$.

**Claim 13** *For $c_1 > 0$, if $(\sigma', j) \models \bigvee_{1\leq i\leq c_1}(\bigcirc^i((\bigvee_{c_1\leq l\leq c_1+C} s_l^i) \wedge \alpha\mathcal{U}\beta) \wedge \bigwedge_{0\leq k<i} \bigcirc^k \alpha)$ then $(\rho, j) \models \alpha \mathcal{U}_{[c_1,\infty)}\beta$.*

If $(\sigma', j) \models \bigvee_{1\leq i\leq c_1}(\bigcirc^i((\bigvee_{c_1\leq l\leq c_1+C} s_l^i) \wedge \alpha\mathcal{U}\beta) \wedge \bigwedge_{0\leq k<i} \bigcirc^k \alpha)$, with $c_1 > 0$, then there is $i$, $1 \leq i \leq c_1$, such that $(\sigma', j+i) \models \alpha\mathcal{U}\beta$ and, for all $k$, $0 \leq k < i$, $(\sigma', j+k) \models \alpha$. By definition of $\rho$, $(\rho, j+i) \models \alpha\mathcal{U}\beta$ and, for all $0 \leq k < i$, $(\rho, j+k) \models \alpha$. So, it remains to show that $\tau(j+i) \in \tau(j) + [c_1, \infty)$. By Proposition 5, if $(\sigma', j+i) \models s_l^i$ and $c_1 \leq l \leq c_1 + C < 2 \cdot C$ then $l = \tau(j+i) - \tau(j)$. By definition of $S_C$, we can assume that there is only one $l$ such that $(\sigma', j+i) \models s_l^i$. As $c_1 \leq l \leq c_1 + C$, $\tau(j+i) \in \tau(j) + [c_1, \infty)$. Thus, $(\rho, j) \models \alpha \mathcal{U}_{[c_1,\infty)}\beta$.

**Claim 14** *For $c_1 > 0$, if $(\sigma', j) \models \neg(\neg(\alpha)\mathcal{U}_{[c_1,c_2]}\neg(\beta))^\sharp$ then $(\rho, j) \models \alpha\tilde{\mathcal{U}}_{[c_1,c_2]}\beta$.*

If $(\sigma', j) \not\models (\neg(\alpha)\mathcal{U}_{[c_1,c_2]}\neg(\beta))^\sharp$ then, by Claim 8, $(\rho, j) \not\models \neg(\alpha)\mathcal{U}_{[c_1,c_2]}\neg(\beta)$. By semantics of $\tilde{\mathcal{U}}$, we have that $(\rho, j) \models \alpha\tilde{\mathcal{U}}_{[c_1,c_2]}\beta$.

❑

**Example** We illustrate the Time Difference translation by carrying out a simple example. Assume that we are given the following MTL formula (in NNF and FNF):

$$\begin{aligned}\varphi = p_0 \wedge \Box_{[0,\infty)}(p_0 \rightarrow \bigcirc_{[2,3]} p_1) \\ \wedge \Box_{[0,\infty)}(p_1 \rightarrow \Diamond_{[1,2]} \neg q).\end{aligned}$$

Using Table 3, we translate $\varphi$ into LTL as follows:

$$\begin{aligned}\varphi^\sharp = p_0 \wedge \Box_{[0,\infty)}(p_0 \rightarrow (\neg gap \wedge (\bigcirc_{[2,3]} p_1)^\sharp)) \\ \wedge \Box_{[0,\infty)}(p_1 \rightarrow (\neg gap \wedge (\Diamond_{[1,2]} \neg q)^\sharp)),\end{aligned}$$

where

$$(\bigcirc_{[2,3]} p_1)^\sharp = \bigcirc((\bigvee_{2 \le i \le 3} \delta_i^-) \wedge p_1)$$
$$(\Diamond_{[1,2]} \neg q)^\sharp = \bigvee_{1 \le i \le 2} (\bigcirc^i ((\bigvee_{1 \le j \le 2} s_j^i) \wedge \neg q))$$

(recall that $\Diamond_I \psi \equiv \mathbf{true} \mathcal{U}_I \psi$). By Theorem 4, $\varphi$ is satisfiable iff $\varphi^\sharp \wedge S_4$ is satisfiable, where $S_4$ is the conjunction of the following:

1. $\bigcirc\Box \oplus_{=1} \Pi_\delta$, for $\Pi_\delta = \{\delta_k^- \mid 1 \le k \le 4\}$;
2. $\Box(\delta_k^- \leftrightarrow s_k^1)$, for $1 \le k \le 4$;
3. $\Box \oplus_{\le 1} \Pi^i$, for $1 \le i \le 8$ and $\Pi^i = \{s_j^i \mid i \le j \le 8\}$;
4. $\Box(\bigcirc s_k^1 \wedge s_l^j \rightarrow \bigcirc s^{j+1}_{\mathsf{min}(l+k,8)})$, for $1 < j+1 \le l+k \le 8$.

## C Additional Experimental Results

We have empirically evaluated the translation presented in Sections 3 and 4 on a much wider range of LTL solvers than we were able to report in Section 5, namely, `Aalta`, `Leviathan`, `LS4`, `LWB`, `NuSMV`, `pltl`, `TRP++`, and `TSPASS`.

`Aalta` [24] implements the obligation-based LTL satisfiability checking algorithm for finite and infinite traces devised by Li et al. [29]. We have performed experiments with both versions 1.2 and 2.0 of `Aalta`. As `Aalta` 2.0 was consistently several orders of magnitude faster than `Aalta` 1.2, we will only report the results for the latest version of `Aalta`.

`NuSMV` 2.6.0 [34] provides a reduction of the LTL satisfiability problem to the LTL model checking problem [7]. It is then possible to decide the latter problem either using a BDD-based algorithm (`NuSMV-BDD`) or a SAT-based algorithm (`NuSMV-SBMC`). We use `NuSMV-BDD` with the options `dynamic` for dynamic reordering and `elbwd` for backward image computation, for `NuSMV-SBMC` we have enabled the completeness check.

The Logics Workbench 1.1 (`LWB`) [30] contains implementations of two different LTL solvers. The first is a two-pass tableau-based decision procedure developed by Janssen [27] which underlies the `provable` and `satisfiable` functions of the `pltl` module of `LWB`. In the following we denote this procedure by `LWB-SAT`. The second is a one-pass tableau calculus by Schwendimann [37] which underlies the `model` function of the `pltl` module. In the following we denote this procedure by `LWB-MOD`.

`Leviathan` [28] is an LTL satisfiability checking and model building tool that implements a novel one-pass tableau calculus by Reynolds [25,36].

The `pltl` [35] system also implements two tableau-based methods. The `tree` method is again based on Schwendimann's one-pass tableau

calculus, the `graph` method is based on a one-pass and-or tree tableau calculus [13] resulting in a time complexity optimal decision procedure for LTL.

`TRP++` 2.2 [39] is based on an ordered resolution calculus that operates on LTL formulae in a clausal normal form [16], while `TSPASS` [40] extends that calculus to monodic first-order linear time temporal logic over expanding domains [33]. We use `TSPASS` with the `ModelConstruction` option so that it returns a model for satisfiable formulae [32].

`LS4` [31] is an LTL-prover based on labelled superposition with partial model guidance developed by Suda and Weidenbach [38]. It operates on LTL formulae in the same clausal normal form as `TRP++`.

We again use the unsatisfiable parameterised formulae $\theta^1_{b_1} := \Diamond_{[0,b_1]}p \wedge \Box_{[0,\infty)}\neg p$ and $\theta^2_{b_2} := \bigcirc_{[10,\infty)}p \wedge \bigcirc_{[c_1,\infty)}\neg p$ to highlight differences between the Time Difference Translations and the Gap translations. We expect that on $\theta^1_{b_1}$ combinations of the Time Difference Translations with LTL solvers perform worse than combinations of the Gap Translations with LTL solvers, while we expect the opposite for $\theta^2_{b_2}$.

Tables 6 and 7 show the median CPU times for the combination of our translations with the various LTL solvers on $\theta^1_{b_1}$, with values of $b_1$ between 0 and 10, and on $\theta^2_{b_2}$, with values of $b_2$ between 10 and 20, respectively. An entry 'T/O' indicates that the timeout of 1000 CPU seconds was exceeded by a prover, an entry 'OoM' indicates that the LTL solver ran out of memory and stopped, while entry 'Fail' indicates that the LTL solver encountered some other error condition and stopped.

Figure 2 and 3 show the same data in the form of 'heat maps' where different colours are used to represent different ranges of runtimes. This allows to easily recognise significant differences in the performance of the various combinations of translations and provers while playing down insignificant differences.

The results in Table 6 and Figure 2 confirm our hypothesis that on $\theta^1_{b_1}$ the Gap translations will lead to better performance than the Time Difference translations. For both the strict and non-strict semantics, for all LTL solvers, and for all values of $b_1$ considered, the Time Difference translations result in equal or worse performance than the Gap translation, with the vast majority of entries indicating a significantly worse performance for the Time Difference translations.

Regarding the strict versus the non-strict translations, we would expect that a non-strict translation results in worse performance compared to the corresponding strict translation as the search space for an LTL solver is larger. For the non-strict versus the strict Gap translation we see that this

Table 6: Performance on $\Diamond_{[0,b_1]} p \wedge \Box_{[0,\infty)} \neg p$

| $b_1$ | TD Aalta | Gap Aalta | TD Leviath an | Gap Leviath an | TD LS4 | Gap LS4 | TD LWB MOD | Gap LWB MOD | TD LWB SAT | Gap LWB SAT | TD NuSMV BDD | Gap NuSMV BDD | TD NuSMV SBMC | Gap NuSMV SBMC | TD plt1 | Gap plt1 | TD TRP++ | Gap TRP++ | TD TSPASS | Gap TSPASS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Strict Semantics | | | | | | | | | | | | | | | | | | | | |
| 0 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.01 | 0.01 | 0.02 | 0.00 | 0.00 | 0.01 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 1 | 0.00 | 0.00 | T/O | 0.00 | 0.00 | 0.00 | 0.01 | 0.01 | 0.01 | 0.02 | 0.32 | 0.00 | 0.06 | 0.01 | 0.30 | 0.00 | 0.00 | 0.00 | 0.01 | 0.00 |
| 2 | 0.00 | 0.00 | T/O | T/O | 0.00 | 0.00 | 0.02 | 0.01 | T/O | 0.02 | 33.54 | 0.00 | 0.31 | 0.02 | OoM | 0.00 | 0.00 | 0.00 | 0.03 | 0.00 |
| 3 | 0.00 | 0.00 | T/O | T/O | 0.00 | 0.00 | 0.11 | 0.01 | T/O | 0.02 | T/O | 0.00 | 1.37 | 0.02 | OoM | 0.00 | 0.02 | 0.00 | 0.09 | 0.00 |
| 4 | 0.00 | 0.00 | T/O | T/O | 0.00 | 0.00 | 0.80 | 0.01 | T/O | 0.02 | T/O | 0.00 | 4.93 | 0.02 | OoM | 0.00 | 0.10 | 0.00 | 0.22 | 0.00 |
| 5 | 0.00 | 0.00 | T/O | T/O | 0.01 | 0.00 | 3.92 | 0.01 | T/O | 0.02 | T/O | 0.00 | 15.02 | 0.03 | OoM | 0.00 | 0.60 | 0.00 | 0.41 | 0.00 |
| 6 | 0.00 | 0.00 | T/O | T/O | 0.01 | 0.00 | 15.49 | 0.01 | T/O | 0.02 | T/O | 0.00 | 40.14 | 0.04 | OoM | 0.00 | 3.18 | 0.00 | 0.62 | 0.01 |
| 7 | Fail | 0.01 | T/O | T/O | 0.01 | 0.00 | 51.51 | 0.01 | T/O | 0.02 | T/O | 0.00 | 96.47 | 0.05 | OoM | 0.00 | 15.13 | 0.00 | 2.94 | 0.01 |
| 8 | Fail | 0.01 | T/O | T/O | 0.02 | 0.00 | Fail | 0.01 | T/O | 0.02 | T/O | 0.00 | 213.86 | 0.06 | OoM | 0.00 | 64.00 | 0.00 | 7.86 | 0.01 |
| 9 | Fail | 0.01 | T/O | T/O | 0.02 | 0.00 | Fail | 0.01 | T/O | 0.02 | T/O | 0.01 | 440.59 | 0.06 | OoM | 0.00 | 224.83 | 0.00 | 27.28 | 0.01 |
| 10 | Fail | 0.01 | T/O | T/O | 0.03 | 0.00 | Fail | 0.01 | T/O | 0.03 | T/O | 0.01 | 856.34 | 0.08 | OoM | 0.00 | 760.25 | 0.00 | 110.19 | 0.01 |
| Non-strict Semantics | | | | | | | | | | | | | | | | | | | | |
| 0 | 0.00 | 0.00 | 0.53 | T/O | 0.00 | 0.00 | 0.01 | 0.01 | 0.01 | 0.01 | 0.00 | 0.00 | 0.03 | 0.03 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.00 |
| 1 | 0.04 | 0.00 | T/O | T/O | 0.00 | 0.00 | 0.01 | 0.01 | 139.70 | 0.01 | 0.40 | 0.00 | 3.21 | 0.06 | OoM | 0.00 | 0.37 | 0.00 | 0.32 | 0.01 |
| 2 | 0.49 | 0.00 | T/O | T/O | 0.00 | 0.00 | 0.41 | 0.01 | OoM | 0.01 | 25.45 | 0.00 | T/O | 0.12 | OoM | 0.00 | T/O | 0.01 | 36.17 | 0.02 |
| 3 | 2.25 | 0.00 | T/O | T/O | 0.01 | 0.00 | 13.94 | 0.01 | OoM | 0.01 | T/O | 0.00 | T/O | 0.24 | OoM | 0.00 | T/O | 0.11 | T/O | 0.03 |
| 4 | 12.71 | 0.00 | T/O | T/O | 0.02 | 0.00 | 512.97 | 0.01 | OoM | 0.01 | T/O | 0.00 | T/O | 0.49 | OoM | 0.00 | T/O | 0.55 | T/O | 0.05 |
| 5 | 35.02 | 0.00 | T/O | T/O | 0.03 | 0.01 | T/O | 0.01 | OoM | 0.01 | T/O | 0.00 | T/O | 0.95 | OoM | 0.00 | T/O | 2.20 | T/O | 0.08 |
| 6 | 113.01 | 0.00 | T/O | T/O | 0.04 | 0.02 | T/O | 0.01 | OoM | 0.02 | T/O | 0.00 | T/O | 1.97 | OoM | 0.01 | T/O | 7.29 | T/O | 0.12 |
| 7 | Fail | 0.00 | T/O | T/O | 0.10 | 0.03 | T/O | 0.02 | OoM | 0.05 | T/O | 0.00 | T/O | 3.96 | OoM | 0.04 | T/O | 21.61 | T/O | 0.16 |
| 8 | Fail | 0.00 | T/O | T/O | 0.15 | 0.04 | Fail | 0.03 | OoM | 0.13 | T/O | 0.00 | T/O | 6.16 | OoM | 0.09 | T/O | 58.20 | T/O | 0.22 |
| 9 | Fail | 0.00 | T/O | T/O | 0.19 | 0.06 | Fail | 0.06 | OoM | 0.38 | T/O | 0.01 | T/O | 10.63 | OoM | 0.20 | T/O | 148.41 | T/O | 0.30 |
| 10 | Fail | 0.00 | T/O | T/O | 0.25 | 0.09 | Fail | 0.12 | OoM | 1.14 | T/O | 0.01 | T/O | 17.04 | OoM | 0.53 | T/O | 328.81 | T/O | 0.40 |

Table 7: Performance on $\bigcirc_{[10,\infty)} p \wedge \bigcirc_{[b_2,\infty)} \neg p$

| $b_2$ | TD Aalta | Gap Aalta | TD Leviathan | Gap Leviathan | TD LS4 | Gap LS4 | TD LWB MOD | Gap LWB MOD | TD LWB SAT | Gap LWB SAT | TD NuSMV BDD | Gap NuSMV BDD | TD NuSMV SBMC | Gap NuSMV SBMC | TD pltl | Gap pltl | TD TRP++ | Gap TRP++ | TD TSPASS | Gap TSPASS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Strict Semantics | | | | | | | | | | | | | | | | | | | | |
| 10 | 0.00 | 0.06 | 0.04 | T/O | 0.00 | 0.00 | 0.01 | 0.01 | 0.02 | 0.02 | 0.63 | 0.04 | 0.08 | 0.18 | 0.01 | 0.00 | 0.00 | 0.33 | 0.03 | 489.28 |
| 11 | 0.00 | 0.09 | 0.08 | T/O | 0.00 | 0.00 | 0.01 | 0.01 | 0.02 | 0.02 | 0.91 | 0.06 | 0.10 | 0.20 | 0.01 | 0.00 | 0.00 | 0.50 | 0.04 | 826.80 |
| 12 | 0.00 | 0.14 | 0.12 | T/O | 0.00 | 0.00 | 0.01 | 0.01 | 0.02 | 0.02 | 2.76 | 0.07 | 0.12 | 0.24 | 0.01 | 0.00 | 0.00 | 0.77 | 0.05 | T/O |
| 13 | 0.00 | 0.22 | 0.18 | T/O | 0.00 | 0.01 | 0.01 | 0.01 | 0.02 | 0.02 | 9.74 | 0.14 | 0.14 | 0.27 | 0.01 | 0.00 | 0.00 | 1.18 | 0.05 | T/O |
| 14 | 0.00 | 0.29 | 0.22 | T/O | 0.00 | 0.01 | 0.01 | 0.01 | 0.02 | 0.02 | 55.91 | 0.15 | 0.17 | 0.32 | 0.01 | 0.00 | 0.00 | 1.68 | 0.07 | T/O |
| 15 | 0.00 | 0.40 | 0.30 | T/O | 0.00 | 0.01 | 0.01 | 0.01 | 0.02 | 0.02 | 53.49 | 0.15 | 0.20 | 0.36 | 0.01 | 0.00 | 0.00 | 2.33 | 0.08 | T/O |
| 16 | 0.00 | 0.57 | 0.42 | T/O | 0.00 | 0.01 | 0.01 | 0.01 | 0.02 | 0.02 | T/O | 0.24 | 0.24 | 0.41 | 0.01 | 0.00 | 0.00 | 3.15 | 0.09 | T/O |
| 17 | 0.00 | 0.56 | 0.60 | T/O | 0.00 | 0.02 | 0.01 | 0.01 | 0.02 | 0.02 | T/O | 0.27 | 0.28 | 0.47 | 0.01 | 0.00 | 0.01 | 4.38 | 0.10 | T/O |
| 18 | 0.00 | 0.78 | 0.82 | T/O | 0.00 | 0.02 | 0.01 | 0.01 | 0.02 | 0.02 | T/O | 0.40 | 0.34 | 0.54 | 0.01 | 0.00 | 0.01 | 5.68 | 0.13 | T/O |
| 19 | 0.00 | 1.09 | 1.10 | T/O | 0.00 | 0.02 | 0.01 | 0.01 | 0.02 | 0.02 | T/O | 0.13 | 0.39 | 0.61 | 0.01 | 0.00 | 0.01 | 7.35 | 0.15 | T/O |
| 20 | 0.00 | 1.31 | 1.52 | T/O | 0.00 | 0.03 | 0.01 | 0.01 | 0.02 | 0.02 | T/O | 0.14 | 0.46 | 0.68 | 0.01 | 0.00 | 0.01 | 9.41 | 0.18 | T/O |
| Non-strict Semantics | | | | | | | | | | | | | | | | | | | | |
| 10 | 0.00 | 0.06 | 0.08 | T/O | 0.00 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.93 | 0.05 | 0.10 | 0.19 | 0.00 | 0.00 | 0.00 | 0.32 | 0.00 | 511.56 |
| 11 | 0.00 | 0.10 | 0.12 | T/O | 0.00 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 1.82 | 0.05 | 0.12 | 0.23 | 0.00 | 0.00 | 0.00 | 0.49 | 0.00 | 878.00 |
| 12 | 0.00 | 0.16 | 0.18 | T/O | 0.00 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 6.06 | 0.05 | 0.14 | 0.26 | 0.00 | 0.00 | 0.00 | 0.74 | 0.00 | T/O |
| 13 | 0.00 | 0.23 | 0.24 | T/O | 0.00 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 11.33 | 0.06 | 0.17 | 0.31 | 0.00 | 0.00 | 0.00 | 1.13 | 0.00 | T/O |
| 14 | 0.00 | 0.28 | 0.28 | T/O | 0.00 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 42.44 | 0.08 | 0.20 | 0.35 | 0.00 | 0.00 | 0.00 | 1.62 | 0.00 | T/O |
| 15 | 0.00 | 0.38 | 0.42 | T/O | 0.00 | 0.02 | 0.01 | 0.01 | 0.01 | 0.01 | 959.16 | 0.12 | 0.24 | 0.40 | 0.00 | 0.00 | 0.00 | 2.23 | 0.00 | T/O |
| 16 | 0.00 | 0.52 | 0.57 | T/O | 0.00 | 0.02 | 0.01 | 0.01 | 0.01 | 0.01 | T/O | 0.13 | 0.28 | 0.45 | 0.00 | 0.00 | 0.00 | 3.03 | 0.00 | T/O |
| 17 | 0.00 | 0.68 | 0.86 | T/O | 0.00 | 0.02 | 0.01 | 0.01 | 0.01 | 0.01 | T/O | 0.13 | 0.33 | 0.52 | 0.00 | 0.00 | 0.00 | 4.19 | 0.00 | T/O |
| 18 | 0.00 | 0.88 | 1.00 | T/O | 0.00 | 0.03 | 0.01 | 0.01 | 0.01 | 0.01 | T/O | 0.14 | 0.39 | 0.58 | 0.00 | 0.00 | 0.00 | 5.43 | 0.00 | T/O |
| 19 | 0.00 | 0.99 | 1.44 | T/O | 0.00 | 0.03 | 0.01 | 0.01 | 0.01 | 0.01 | T/O | 0.18 | 0.46 | 0.65 | 0.00 | 0.00 | 0.00 | 7.01 | 0.01 | T/O |
| 20 | 0.00 | 1.26 | 2.18 | T/O | 0.00 | 0.04 | 0.01 | 0.01 | 0.01 | 0.01 | T/O | 0.15 | 0.53 | 0.74 | 0.00 | 0.00 | 0.00 | 8.94 | 0.01 | T/O |

is indeed the case for all provers except `Aalta`. Likewise, the non-strict Time Difference translation results in worse performance than the strict Time Difference translation with the exception of `NuSMV-BDD`. It would be of interest to investigate what causes these deviations from expectations.

The best performing combination is Gap + `Aalta`. The most 'robust' LTL solver, in the sense of producing the lowest total runtime across all translations and all instances of $\theta^1_{b_1}$ considered, is `LS4`.

The results in Table 7 and Figure 3 also largely confirm our hypothesis that on $\theta^2_{b_2}$ the Time Difference translations will lead to better performance than the Gap translations. In total we have considered ten LTL solvers and translations under two different semantics, giving us twenty points of comparison between a Time Diference translation and a Gap translation, each over eleven instances of $\theta^2_{b_2}$. For twelve of these comparison points a Time Difference translation results in better performance than a Gap translation (highlighted in green in Table 7), for five comparison points the performance is the same and only for three comparison points the Time Difference translation results in worse performance than the Gap translation (highlighted in pink in Table 7).

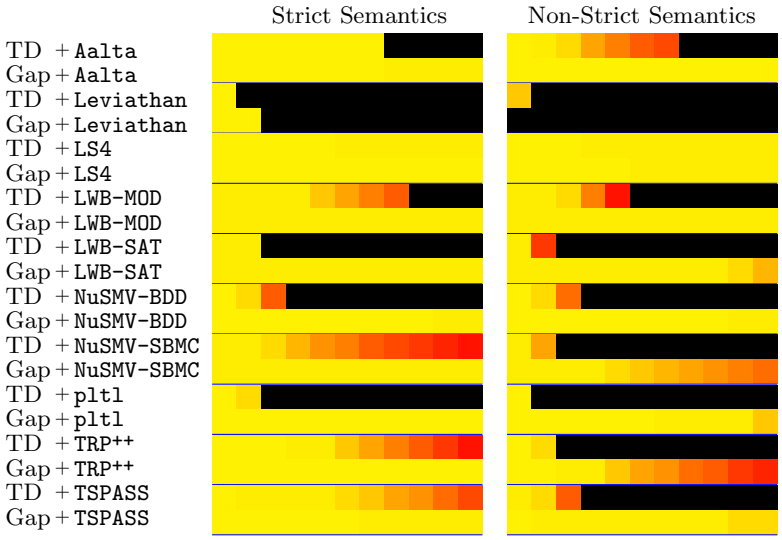


Figure 2: Heat map of the runtimes on $\theta^1_{b_1} = \Diamond_{[0,b_1]} p \wedge \Box_{[0,\infty)} \neg p$. Each rectangle represents the runtime of one approach on one instance of $\theta^1_{b_1}$ with runtimes given colours as follows:

< 0.01 sec | > 0.01 sec, ≤ 0.25 sec | > 0.25 sec, ≤ 0.50 sec | > 0.50 sec, ≤ 1 sec
> 1 sec, ≤ 2 sec | > 2 sec, ≤ 4 sec | > 4 sec, ≤ 8 sec | > 8 sec, ≤ 16 sec
> 16 sec, ≤ 32 sec | > 32 sec, ≤ 64 sec | > 64 sec, ≤ 125 sec | > 125 sec, ≤ 250 sec
> 250 sec, ≤ 500 sec | > 500 sec, ≤ 1000 sec | T/O, OoM or Fail

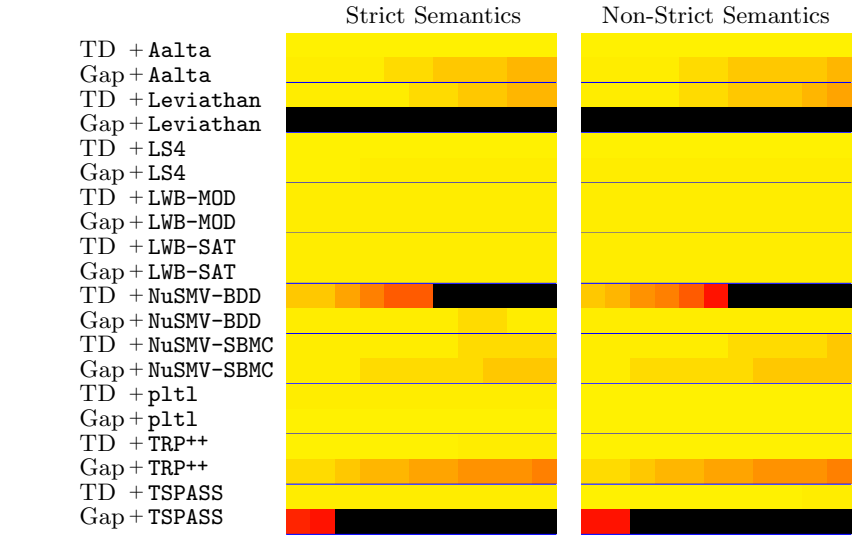


Figure 3: Heat map of the runtimes on $\theta^2_{b_2} := \bigcirc_{[10,\infty)} p \wedge \bigcirc_{[c_1,\infty)} \neg p$. Each rectangle represents the runtime of one approach on one instance of $\theta^2_{b_2}$ with runtimes given colours as follows:

There are several LTL solvers that perform very well on $\theta^2_{b_2}$, the most 'robust' LTL solver on $\theta^2_{b_2}$ is `pltl`.

Overall the experiments presented in this section confirm that no translation is strictly 'better' than the other. It will depend on the characteristics of the formula and sometimes on the LTL solver whether a Time Difference translation or a Gap translation results in better performance and therefore in a greater likelihood of deciding a formula in reasonable time. However, the experiments also show the significant performance improvements that can be achieved by choosing the 'right' translation and, arguably, the simplicity of the Gap translations means that more often than not it is the translation to use. The experimental results for Multiprocessor Job-Shop Scheduling problems presented in the next section provide additional support for this.

## D Experiments with MJS problems

We have performed an experimental evaluation of the combination of our translations with a wide range of LTL solvers on MJS problems. The

provers we have used are `Aalta`, `Leviathan`, `LS4`, `LWB`, `NuSMV`, `TRP++`, and `TSPASS`. For a short description of each of these provers see Section C.

Regarding the Multiprocessor Job-shop Scheduling problems we made the simplifying assumption that a job $j_i$, for each $i$, $1 \leq i \leq n$, takes the same amount of time $t_i$ on whichever machine it is processed on. We can then characterise a MJS problem by stating (i) a *job list* $J$ consisting of a list of durations $(t'_1, \ldots, t'_n)$, (ii) the number $k$ of machines available, and (iii) the time bound $t$. In equations 5, 6, 12 and 13, for every $i$, $1 \leq i \leq n$, and every $l$, $1 \leq l \leq k$, $t_{j_i m_l}$ will be given by $t'_{j_i}$. The time bound $t$ is used in the formula $\Diamond_{[0,t]} \bigwedge_{i=1}^{n} has\_run_{j_i}$ that expresses the requirement for a schedule that completes all $n$ jobs on $k$ machines in at most $t$ time points.

It is an interesting characteristic of the MTL formulae for MJS problems, in contrast to, say, random 3CNF formulae or random modal $K$3CNF formulae that are often used for benchmarking SAT solvers or modal logic provers, unsatisfiable MTL formulae are smaller than satisfiable MTL formulae: A time bound $t$ that is too small leads to a MJS problem for which no schedule exists that can complete all jobs within the given time bound while a sufficiently large time bound guarantees that such a schedule can be found. By our encodings a small time bound results in a smaller LTL formula than a large time bound. The difference in size is more pronounced for the Time Difference Translations than for the Gap Translations.

In our experiments we varied the number $n$ of jobs between 1 and 4, the duration $t'_i$ of a job between 1 and 4, the number $k$ of machines between 1 and 3 and finally the time bound $t$ between 0 and 4. We then constructed MTL formulae for the strict semantics and the non-strict semantics according to the formalisations in Section 6. Each formula was transformed to Flat Normal Form, translated to LTL using one of the encodings, and each solver run five times on the resulting LTL formula (with a timeout of 1000 CPU seconds), and the median CPU time over those five runs determined. We refer to that median CPU time as the runtime.

If an LTL solver reports that the formula resulting from a MJS problem with $n$ jobs, $k$ machines and time bound $t$ is satisfiable, then a schedule exists that completes all $n$ jobs on $k$ machines within $t$ time points. However, this provides us with no information about what that schedule might be. If for a satisfiable formula the LTL solver also returns a model, then the information at which time point $start\_run_{j_i m_l}$ (or $start\_run j_i \wedge m_l$ for the non-strict semantics) becomes true tells when a particular job has to be started on a particular machine in order to complete all jobs within the time bound, thus, the model provides us with a schedule.

Among the systems included in our experiments, only `Leviathan`, `NuSMV-BDD`, `NuSMV-SBMC`, `LWB-MOD` and `TSPASS` can produce models and have been used with the command line options that require them to do so. It should be noted that the model construction in `TSPASS` potentially requires significantly more resolution inferences to be performed than are necessary to just decide the satisfiability of a formula. Thus, the way we use `TSPASS` puts it as a distinct performance disadvantage over `TRP++` although both perform very similar inference steps when just deciding the satisfiability of an LTL formula.

Tables 8 and 9 show the results for the formalisation of MJS problems in MTL with strict semantics and non-strict semantics, respectively. The first three columns in each table show the job list $J$, the number $k$ of machines and the time bound $t$. The fourth column indicates whether the corresponding MTL formula is satisfiable (S) or unsatisfiable (U). The remaining columns of the table show the runtime for each translation and each prover ('T/O' indicates timeout). Figure 4 shows the same data in the form of a heat map.

Table 8: Performance of LTL solvers on MJS problems (strict semantics)

| | | | | TD | Gap | TD | Gap | TD | Gap | TD | Gap | TD | Gap | TD | Gap | TD | Gap |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Aalta | Aalta | LS4 | LS4 | LWB | LWB | NuSMV | NuSMV | NuSMV | NuSMV | TRP++ | TRP++ | TSPASS | TSPASS |
| $J$ | $k$ | $t$ | $S$ | | | | | MOD | MOD | BDD | BDD | SBMC | SBMC | | | | |
| 1 | 1 | 0 | U | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.02 | 0.42 | 0.03 | 0.04 | 0.02 | 0.00 | 0.00 | 0.00 | 0.02 |
| 1 | 1 | 1 | S | 0.00 | 0.00 | 0.00 | 0.00 | 0.02 | 0.02 | 0.75 | 0.03 | 0.13 | 0.05 | 0.01 | 0.06 | 0.02 | 1.12 |
| 1 | 1 | 2 | S | 0.02 | 0.00 | 0.00 | 0.00 | 0.23 | 0.02 | 11.83 | 0.03 | 0.48 | 0.05 | 3.92 | 0.08 | 0.10 | 2.01 |
| 1 | 1 | 3 | S | 0.24 | 0.00 | 0.00 | 0.00 | 2.25 | 0.02 | 180.48 | 0.03 | 1.56 | 0.05 | T/O | 0.12 | 0.53 | 7.35 |
| 1,2 | 1 | 0 | U | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.02 | 9.11 | 0.10 | 0.23 | 0.05 | 0.00 | 0.00 | 0.02 | 0.02 |
| 1,2 | 1 | 1 | U | 0.01 | 0.00 | 0.00 | 0.00 | 0.01 | 0.02 | 11.68 | 0.12 | 0.33 | 0.09 | 0.02 | 0.00 | 0.03 | 0.05 |
| 1,2 | 1 | 2 | U | 0.31 | 0.00 | 0.00 | 0.00 | 1.09 | 0.02 | 16.50 | 0.20 | 0.47 | 0.12 | 157.39 | 0.01 | 0.28 | 0.10 |
| 1,2 | 1 | 3 | S | 1.13 | 0.01 | 0.00 | 0.00 | 11.22 | 0.03 | T/O | 0.20 | 2.40 | 0.22 | T/O | 2.69 | 315.98 | 950.48 |
| 1,2 | 2 | 0 | U | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.02 | 10.07 | 0.28 | 0.26 | 0.07 | 0.00 | 0.00 | 0.02 | 0.02 |
| 1,2 | 2 | 1 | U | 0.01 | 0.01 | 0.00 | 0.00 | 0.11 | 0.02 | 19.67 | 0.44 | 0.38 | 0.12 | 0.03 | 0.01 | 0.05 | 0.09 |
| 1,2 | 2 | 2 | S | 0.06 | 0.02 | 0.00 | 0.00 | 0.61 | 0.04 | 21.97 | 0.70 | 0.58 | 0.26 | T/O | 1.07 | 31.92 | 70.68 |
| 1,2 | 2 | 3 | S | 0.90 | 0.02 | 0.01 | 0.00 | 2.63 | 0.04 | T/O | 0.99 | 1.68 | 0.26 | T/O | 1.55 | 112.66 | 422.40 |
| 1,1,2 | 2 | 0 | U | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.02 | 15.00 | 1.26 | 0.50 | 0.14 | 0.00 | 0.00 | 0.02 | 0.03 |
| 1,1,2 | 2 | 1 | U | 0.02 | 0.02 | 0.00 | 0.00 | 0.02 | 0.02 | 21.65 | 1.31 | 0.55 | 0.20 | 0.04 | 0.01 | 0.05 | 0.25 |
| 1,1,2 | 2 | 2 | S | 0.15 | 0.03 | 0.00 | 0.00 | 1.04 | 0.13 | 44.82 | 2.07 | 1.11 | 0.39 | T/O | 12.69 | 190.13 | T/O |
| 1,1,2 | 2 | 3 | S | 0.49 | 0.03 | 0.00 | 0.00 | 31.01 | 0.12 | 852.46 | 1.95 | 3.84 | 0.40 | T/O | 33.93 | T/O | T/O |
| 1,1,2 | 3 | 0 | U | 0.00 | 0.00 | 0.00 | 0.00 | 0.02 | 0.02 | 17.91 | 2.59 | 0.50 | 0.19 | 0.00 | 0.00 | 0.03 | 0.04 |
| 1,1,2 | 3 | 1 | U | 0.02 | 0.02 | 0.00 | 0.00 | 0.46 | 0.04 | 92.28 | 2.89 | 0.89 | 0.27 | 0.13 | 0.02 | 0.09 | 0.29 |
| 1,1,2 | 3 | 2 | S | 0.21 | 0.04 | 0.00 | 0.00 | 2.39 | 0.44 | 116.92 | 6.60 | 1.41 | 0.43 | T/O | 7.62 | 88.98 | 149.05 |
| 1,1,2 | 3 | 3 | S | 0.32 | 0.04 | 0.00 | 0.00 | 3.68 | 0.19 | T/O | 14.88 | 2.84 | 0.55 | T/O | 9.78 | 656.16 | 640.64 |
| 1,1,2,2 | 2 | 0 | U | 0.00 | 0.00 | 0.00 | 0.00 | 0.02 | 0.02 | 25.62 | 2.53 | 0.59 | 0.20 | 0.00 | 0.00 | 0.03 | 0.05 |
| 1,1,2,2 | 2 | 1 | U | 0.02 | 0.02 | 0.00 | 0.00 | 0.02 | 0.03 | 53.98 | 4.40 | 0.98 | 0.29 | 0.10 | 0.02 | 0.07 | 0.55 |
| 1,1,2,2 | 2 | 2 | U | 0.34 | 0.09 | 0.00 | 0.00 | 9.06 | 0.77 | 68.83 | 8.68 | 1.24 | 0.39 | T/O | 0.17 | 2.55 | 1.84 |
| 1,1,2,2 | 2 | 3 | S | 2.30 | 0.07 | 0.01 | 0.00 | 239.81 | 1.93 | T/O | 8.30 | 4.64 | 0.73 | T/O | T/O | T/O | T/O |
| 1,1,2,2 | 3 | 0 | U | 0.00 | 0.00 | 0.00 | 0.00 | 0.02 | 0.02 | 167.08 | 7.04 | 0.49 | 0.29 | 0.00 | 0.00 | 0.04 | 0.06 |
| 1,1,2,2 | 3 | 1 | U | 0.03 | 0.04 | 0.00 | 0.00 | 0.09 | 0.06 | 113.48 | 26.05 | 2.40 | 0.42 | 0.33 | 0.03 | 0.15 | 0.78 |
| 1,1,2,2 | 3 | 2 | S | 0.34 | 0.09 | 0.00 | 0.00 | 8.62 | 4.06 | 342.19 | 17.49 | 1.28 | 0.82 | T/O | 696.66 | T/O | T/O |
| 1,1,2,2 | 3 | 3 | S | 1.29 | 0.08 | 0.01 | 0.00 | 274.13 | 2.87 | T/O | 132.39 | 2.93 | 0.79 | T/O | T/O | T/O | T/O |
| 1,2,2,3 | 2 | 0 | U | 0.01 | 0.00 | 0.00 | 0.00 | 0.03 | 0.02 | 410.67 | 3.42 | 1.12 | 0.12 | 0.00 | 0.00 | 0.06 | 0.05 |
| 1,2,2,3 | 2 | 1 | U | 0.06 | 0.02 | 0.00 | 0.00 | 0.05 | 0.03 | 990.74 | 8.23 | 1.90 | 0.16 | 0.32 | 0.04 | 0.19 | 0.74 |
| 1,2,2,3 | 2 | 2 | U | 0.44 | 0.04 | 0.01 | 0.00 | 73.57 | 0.78 | T/O | 7.74 | 2.38 | 0.44 | T/O | 0.21 | 3.30 | 1.33 |
| 1,2,2,3 | 2 | 3 | U | 4.66 | 0.26 | 0.01 | 0.00 | T/O | 4.30 | T/O | 28.87 | 3.81 | 0.58 | T/O | 14.71 | T/O | 594.90 |
| 1,2,2,3 | 2 | 4 | S | 20.09 | 0.30 | 0.02 | 0.00 | T/O | 8.17 | T/O | 26.49 | 15.78 | 0.99 | T/O | T/O | T/O | T/O |
| 1,2,3,4 | 2 | 4 | U | 31.50 | 0.88 | 0.03 | 0.01 | T/O | 13.73 | T/O | 17.41 | 8.55 | 0.79 | T/O | 56.04 | T/O | T/O |
| 1,2,3,4 | 2 | 5 | S | 161.79 | 1.05 | 0.03 | 0.01 | T/O | 29.64 | T/O | 25.26 | 29.35 | 1.39 | T/O | T/O | T/O | T/O |

`Leviathan` was only able to solve the simplest MJS problem using the Strict Gap Translation and otherwise exceeded the timeout. We have not included its results in the tables and the heat map.

Regarding the formalisation of MJS problems in the strict semantics, we see in Figure 4 and Table 8 that for every prover except `TSPASS`, the Gap Translation results in equal or better performance than the Time Difference Translation on every single problem. The Gap Translation together with `LS4` offers the best performance for every instance but does not provide models for satisfiable problems. The Gap Translation together with `NuSMV-SBMC` offers the best performance among the approaches that return models.

Regarding the formalisation of MJS problems in the non-strict semantics, the most striking observation we can make from Figure 4 and from the data in Table 9 is how much more challenging the corresponding LTL satisfiability problems are for all the provers. In total there are 35 MJS problems on which we benchmarked 7 provers with 2 different translations producing a total of 490 results. Of these 244 were timeouts, compared

Table 9: Performance of LTL solvers on MJS problems (non-strict semantics)

| | | | | TD + | Gap + | TD + | Gap + | TD + | Gap + | TD + | Gap + | TD + | Gap + | TD + | Gap + | Delta + | Gap + |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Aalta | Aalta | LS4 | LS4 | LWB | LWB | NuSMV | NuSMV | NuSMV | NuSMV | TRP++ | TRP++ | TSPASS | TSPASS |
| $J$ | $m$ | $t$ | $S$ | | | | | MOD | MOD | BDD | BDD | SBMC | SBMC | | | | |
| 1 | 1 | 0 | U | 0.37 | 0.01 | 0.00 | 0.00 | 0.14 | 0.01 | 45.15 | 0.00 | T/O | 4.31 | T/O | 0.37 | 0.02 | 71.43 |
| 1 | 1 | 1 | S | 0.56 | 0.02 | 0.00 | 0.00 | 0.30 | 0.01 | 39.00 | 0.01 | 0.47 | 0.04 | T/O | 1.21 | T/O | 77.45 |
| 1 | 1 | 2 | S | 0.13 | 0.00 | 0.00 | 0.00 | 0.45 | 0.02 | 59.44 | 0.03 | 0.48 | 0.04 | T/O | 1.79 | T/O | 446.46 |
| 1 | 1 | 3 | S | 1.30 | 0.01 | 0.00 | 0.00 | 2.83 | 0.01 | T/O | 0.04 | 1.55 | 0.04 | T/O | 3.42 | T/O | T/O |
| 1,2 | 1 | 0 | U | 0.86 | 0.06 | 0.01 | 0.00 | 3.61 | 0.02 | Fail | 0.11 | Fail | T/O | T/O | 522.45 | T/O | T/O |
| 1,2 | 1 | 1 | U | 3.18 | 0.20 | 0.01 | 0.03 | 11.79 | 0.11 | Fail | 0.17 | Fail | T/O | T/O | T/O | T/O | T/O |
| 1,2 | 1 | 2 | U | 9.39 | Fail | 0.03 | 2.36 | 56.21 | 0.53 | T/O | 0.19 | Fail | T/O | T/O | T/O | T/O | T/O |
| 1,2 | 1 | 3 | S | 2.02 | 0.10 | 0.01 | 0.00 | 17.51 | 0.33 | T/O | 0.52 | 2.67 | 0.14 | T/O | T/O | T/O | T/O |
| 1,2 | 2 | 0 | U | 1.62 | 0.31 | 0.03 | 0.02 | 165.43 | 1.06 | T/O | 0.53 | Fail | T/O | T/O | T/O | T/O | T/O |
| 1,2 | 2 | 1 | U | 39.60 | 4.10 | 0.14 | 89.65 | 430.17 | 10.72 | T/O | 0.68 | T/O | T/O | T/O | T/O | T/O | T/O |
| 1,2 | 2 | 2 | S | 4.24 | 0.76 | 0.01 | 0.01 | 161.83 | 4.06 | T/O | 1.90 | 2.27 | 0.16 | T/O | T/O | T/O | T/O |
| 1,2 | 2 | 3 | S | 1.57 | 0.19 | 0.02 | 0.01 | 170.01 | 14.02 | T/O | 4.95 | 2.30 | 0.16 | T/O | T/O | T/O | T/O |
| 1,1,2 | 2 | 0 | U | 18.06 | 2.82 | 0.03 | 0.50 | T/O | 12.96 | T/O | 2.00 | Fail | 503.72 | T/O | T/O | T/O | T/O |
| 1,1,2 | 2 | 1 | U | 299.36 | 12.37 | 0.20 | T/O | T/O | 806.58 | T/O | 7.62 | Fail | T/O | T/O | T/O | T/O | T/O |
| 1,1,2 | 2 | 2 | S | Fail | Fail | 0.04 | 0.02 | 910.67 | 167.88 | T/O | 24.65 | 3.08 | 0.32 | T/O | T/O | T/O | T/O |
| 1,1,2 | 2 | 3 | S | 26.20 | 1.19 | 0.03 | 0.02 | 928.95 | T/O | T/O | T/O | 61.32 | 3.11 | 0.32 | T/O | T/O | T/O |
| 1,1,2 | 3 | 0 | U | 7.96 | 6.52 | 0.61 | T/O | T/O | T/O | T/O | 4.66 | Fail | T/O | T/O | T/O | T/O | T/O |
| 1,1,2 | 3 | 1 | U | 472.29 | T/O | T/O | T/O | T/O | T/O | T/O | 29.01 | Fail | T/O | T/O | T/O | T/O | T/O |
| 1,1,2 | 3 | 2 | S | 32.26 | 1.25 | 0.05 | 0.01 | T/O | T/O | T/O | 253.91 | 3.38 | 0.42 | T/O | T/O | T/O | T/O |
| 1,1,2 | 3 | 3 | S | 11.77 | T/O | 0.06 | 0.02 | T/O | T/O | T/O | 450.82 | 3.42 | 0.42 | T/O | T/O | T/O | T/O |
| 1,1,2,2 | 2 | 0 | U | 13.17 | 3.42 | 0.06 | 4.84 | 53.79 | T/O | 3.18 | Fail | T/O | T/O | T/O | T/O | T/O | T/O |
| 1,1,2,2 | 2 | 1 | U | T/O | Fail | 1.51 | T/O | T/O | T/O | T/O | 12.84 | Fail | T/O | T/O | T/O | T/O | T/O |
| 1,1,2,2 | 2 | 2 | U | Fail | Fail | T/O | T/O | T/O | T/O | T/O | 146.48 | Fail | T/O | T/O | T/O | T/O | T/O |
| 1,1,2,2 | 2 | 3 | S | Fail | Fail | 0.15 | 0.06 | T/O | T/O | T/O | 930.71 | 4.22 | 0.60 | T/O | T/O | T/O | T/O |
| 1,1,2,2 | 3 | 0 | U | 18.68 | 222.25 | T/O | T/O | T/O | T/O | T/O | 25.77 | Fail | Fail | T/O | T/O | T/O | T/O |
| 1,1,2,2 | 3 | 1 | U | 910.03 | Fail | T/O | T/O | T/O | T/O | T/O | 253.55 | Fail | Fail | T/O | T/O | T/O | T/O |
| 1,1,2,2 | 3 | 2 | S | Fail | Fail | 0.16 | 0.10 | T/O | T/O | T/O | T/O | 4.63 | 0.83 | T/O | T/O | T/O | T/O |
| 1,1,2,2 | 3 | 3 | S | Fail | 5.41 | 0.20 | 0.09 | T/O | T/O | T/O | T/O | 4.69 | 0.83 | T/O | T/O | T/O | T/O |
| 1,2,2,3 | 2 | 0 | U | 29.36 | 15.60 | 0.07 | 3.71 | T/O | 58.14 | T/O | 7.04 | Fail | 4.90 | T/O | T/O | T/O | T/O |
| 1,2,2,3 | 2 | 1 | U | 120.02 | Fail | 1.67 | T/O | T/O | T/O | T/O | 9.05 | Fail | T/O | T/O | T/O | T/O | T/O |
| 1,2,2,3 | 2 | 2 | U | T/O | Fail | T/O | T/O | T/O | T/O | T/O | 62.53 | Fail | T/O | T/O | T/O | T/O | T/O |
| 1,2,2,3 | 2 | 3 | U | Fail | Fail | T/O | T/O | T/O | T/O | T/O | 482.93 | Fail | T/O | T/O | T/O | T/O | T/O |
| 1,2,2,3 | 2 | 4 | S | Fail | Fail | 0.24 | 0.08 | T/O | T/O | T/O | T/O | 12.65 | 0.97 | T/O | T/O | T/O | T/O |
| 1,2,3,4 | 2 | 4 | U | Fail | Fail | T/O | T/O | T/O | T/O | T/O | T/O | Fail | T/O | T/O | T/O | T/O | T/O |
| 1,2,3,4 | 2 | 5 | S | Fail | Fail | 0.54 | 0.15 | T/O | T/O | T/O | T/O | 34.28 | 1.32 | T/O | T/O | T/O | T/O |

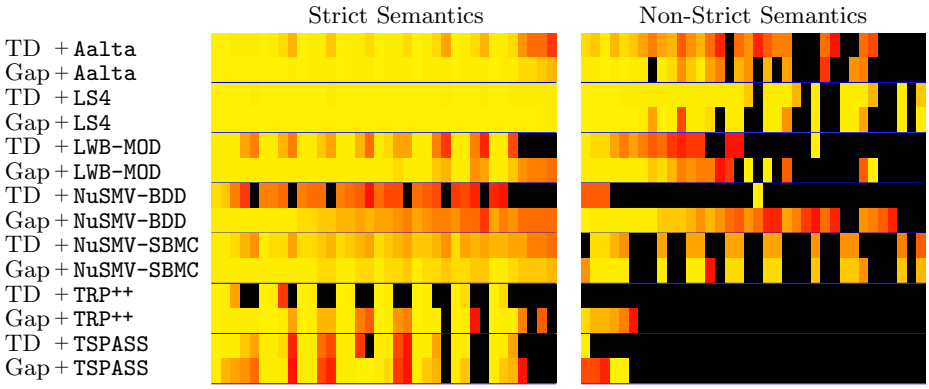


Figure 4: Heat map for the performance of LTL provers on MJS problems. Each rectangle represents the runtime of one approach on a MJS problem with runtimes given colours as follows:

to 51 timeouts for MJS problems in the strict semantics. In a further 44 instances (indicated by 'Fail' in the table) the provers failed, that is, they encountered a memory allocation error, exhausted the limited size of some internal data structure, or produced some other internal failure before reaching the timeout.

Overall, the Non-Strict Gap Translation results in better performance than the Non-Strict Time Difference Translation. The combination of the Non-Strict Gap Translation and `LS4` is still the best performing single approach, but exceeds the timeout for most of the unsatisfiable MJS problems. The combination of the Non-Strict Gap Translation and `NuSMV-SBMC` shows the same pattern, only slower, but produces models. In contrast to `LS4` and `NuSMV-SBMC`, `NuSMV-BDD` with the Non-Strict Gap Translation solves almost all satisfiable MJS problems but exceeds the timeout on unsatisfiable problems. The best results can therefore be obtained by a combination of the Non-Strict Gap Translation with a portfolio consisting of `LS4` and `NuSMV-BDD` or `NuSMV-SBMC` and `NuSMV-BDD`.

In summary, the experimental results presented in this section provide further evidence of the significant performance improvements that can be gained from the use of the Gap Translations over Time Difference Translations.

It is worth pointing out that Multiprocessor Job-shop Scheduling is not our intended domain of application for the translations we have presented.

There is a lot of symmetry in these problems that a more specialised solver can take advantage of while LTL solvers are oblivious to it, e.g., the order in which jobs are executed on a particular machine does not affect the overall time to completion nor does the choice of machine. However, for problems containing less symmetry that can naturally be formalised in MTL, and for which scheduling or planning is just one part of it, the approach used here can be beneficial.

## References


24. Aalta, http://lab205.org/aalta/
25. Bertello, M., Gigante, N., Montanari, A., Reynolds, M.: Leviathan: A new LTL satisfiability checking tool based on a one-pass tree-shaped tableau. In: Proc. IJCAI 2016. pp. 950–956. IJCAI/AAAI Press (2016)
26. Goré, R.: And-or tableaux for fixpoint logics with converse: LTL, CTL, PDL and CPDL. In: Proc. IJCAR 2014. LNCS, vol. 8562, pp. 26–45. Springer (2014)
27. Janssen, G.: Logics for Digital Circuit Verification: Theory, Algorithms, and Applications. Ph.D. thesis, Eindhoven University of Technology, The Netherlands (1999)
28. Leviathan, https://github.com/Corralx/leviathan
29. Li, J., Yao, Y., Pu, G., Zhang, L., He, J.: Aalta: an LTL satisfiability checker over infinite/finite traces. In: Proc. FSE 2014. pp. 731–734. ACM (2014)
30. Logics Workbench, http://www.lwb.unibe.ch/index.html
31. LS4, https://github.com/quickbeam123/ls4
32. Ludwig, M., Hustadt, U.: Resolution-based model construction for PLTL. In: Proc. TIME 2009. pp. 73–80. IEEE (2009)
33. Ludwig, M., Hustadt, U.: Implementing a fair monodic temporal logic prover. AI Communications 23(2–3), 69–96 (2010)
34. NuSMV, http://nusmv.fbk.eu/
35. pltl, http://users.cecs.anu.edu.au/~rpg/PLTLProvers/
36. Reynolds, M.: A new rule for LTL tableaux. In: Proc. GandALF 2016. EPTCS, vol. 226, pp. 287–301 (2016)
37. Schwendimann, S.: A new one-pass tableau calculus for PLTL. In: Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX '98. Lecture Notes in Computer Science, vol. 1397, pp. 277–292. Springer (1998)
38. Suda, M., Weidenbach, C.: A PLTL-prover based on labelled superposition with partial model guidance. In: Proc. IJCAR. LNCS, vol. 7364, pp. 537–543. Springer (2012)
39. TRP++, http://cgi.csc.liv.ac.uk/~konev/software/trp++/
40. TSPASS, http://cgi.csc.liv.ac.uk/~michael/TLBook/TSPASS-System/