

Note on the characterization of until as a fixed point under clocked semantics

Dana Fisman^{1,2}

¹ Weizmann Institute of Science

² IBM Haifa Research Laboratory

Abstract. Modern hardware designs are typically based on multiple clocks. While a singly-clocked hardware design is easily described in standard temporal logics, describing a multiply-clocked design is cumbersome. Thus, it is desirable to have an easier way to formulate properties related to clocks in a temporal logic. In [2] a relatively simple solution built on top of the traditional LTL semantics was suggested. The suggested semantics was examined relative to a list of design goals, and it was shown that it answered all requirements except for preserving the least fixed point characterization of the until operator under multiple clocks. In this work we show that with a minor addition to the semantics of [2] this requirement is met as well.

1 Introduction

Synchronous hardware designs are based on a notion of discrete time, in which the flip-flop (or latch) takes the system from the current state to the next state. A flip-flop or latch is a memory element, which passes on some function of its inputs to its outputs, but only when its clock input is active. The signal that causes the flip-flop (or latch) to transition is termed the *clock*. In a singly-clocked hardware design, the behavior of hardware in terms of the clock naturally maps to the notion of the next-time operator in temporal logics such as LTL [4], so that the following LTL formula: $G(p \rightarrow X q)$ can be interpreted as “globally, if p then *at the next clock cycle*, q ”. Mapping between a state of a model for the temporal logic and a clock cycle of hardware can then be dealt with by the tool which builds a model from the source code (written in some hardware description language, or HDL).

Modern hardware designs, however, are typically based on multiple clocks. In such a design, for instance, some flip-flops may be clocked with $clka$, while others are clocked with $clkb$. In this case, the mapping between states and clock cycles cannot be done automatically; rather, the formula itself must contain some indication of which clock to use. Thus, it is desirable to have an easier way to formulate properties related to clocks in a temporal logic. For example, the linear temporal logic LTL can be extended with a clock operator, denoted \mathcal{C} , so that the formula

$$(G(p \rightarrow X q))\mathcal{C}clka \tag{1}$$

stating that “globally, if p during a cycle of $clka$, then at the next cycle of $clka$, q ” will be equivalent to the LTL formula

$$G ((clka \wedge p) \rightarrow X[\neg clka W (clka \wedge q)]) \tag{2}$$

In [2] a relatively simple solution built on top of the traditional LTL semantics is given. This solution is based on the idea that the only role of the clock operator should be to define a projection of the path onto those states where the clock “ticks”. Actually, referring to a projection of the path is not precisely correct, as we allow access to states in between consecutive states of a projection in the event of a clock switch. However, the word “projection” conveys the intuitive function of the clock operator in the case that the formula is singly-clocked. Achieving this introduces a problem for paths on which the clock never ticks. This problem is solved in [2] by introducing a propositional strength operator that extends the semantics from non-empty paths to empty paths in the same way that the strong next operator [3, pp. 272-273] extends the semantics from infinite to finite paths.

The logic given in [2], is measured against a list of design goals. It is shown that all design goals are met, but that the least fixed point characterization of *until* is not preserved when multiple clocks are involved. In this work we show that with a minor addition to the semantics of [2] the *until* operator preserves its least fixed point characterization (as well as the other design goals).

The addition suggest herein can be thought of as *alinement* operators, that takes you to the closest clock tick, when the current cycle is not a clock tick. Note that the *next* operators takes you to the *second* clock tick when the current cycle is not a clock tick. This is ok since on the projected path, the second clock tick is the second letter – exactly the place where the *next* operator will take you in standard LTL. There are two alignment operators, weak and strong, in order to deal with the possibility the clock may stop ticking; The strong alignment operator demands the clock to tick at least once more, while the weak alignment operators makes no such requirement. On singly-clocked formulas there is no need for alinement operators, since you are always on a clock tick (the clock operator takes you to the projected path, and all other operators keep you on this path). On multiply-clocked formulas, however, on the event of a clock switch you may get to a cycle which is not a clock tick - the alignment operators, in this case takes you to the closest relevant tick.

The remainder of the paper is organized as follows. In Section 2.2 we give the semantics of the logic, and explain the difference with [2]. In Section 3 we prove that the least fixed point characterization of *until* is preserved, as well as the other design goals of [2]. In Section 4 we conclude.

2 The Definition of LTL[®]

The semantics given in [2] solves the problem of finite path introduced by clocks in LTL-based semantics by providing a propositional strength operator, similar to the strength given to the *next* operator in [3, pp. 272-273]. That is, given a proposition p , both $p!$ and p are LTL[®] formulas – the strong version $p!$ holds if on every non-empty path w , the first letter of w satisfies p ; and the weak version holds also on the empty path.

In this work we solve the problem of finite paths by generalizing the *next* operator with a “power”. That is, the *next* operators comes with a non-negative integer m , so that $X!^m$ holds on the next m cycles in the unclocked semantics and on

the next m tick of the clock in the clocked semantics. Similarly, X^m holds in the unclocked semantics on the next m cycles, if there are next m cycles; and in the clocked semantics on the next m tick of the clock if there are next m ticks. The operators obtained by instantiating m with zero (i.e. $X!^0$ and X^0) can be seen as alinement operators. The operator $X!^0$ takes you to the closest clock tick, when the current cycle is not a clock tick in the clocked semantics, and in the unclocked semantics it leaves you exactly where you are. Together with its weak version X^0 , they thus supply a solution to empty paths, which is similar to [2].

2.1 Syntax

The syntax of LTL° is defined below, where we use the term *boolean expression* to refer to any application of the standard boolean operators to atomic propositions.

Definition 1 (Formulas of LTL°).

- If b is a boolean expression, then $b!$ and b are LTL° formulas.
- If clk is a boolean expression, m is a non-negative integer, and f , f_1 , and f_2 are LTL° formulas, then the following are LTL° formulas:
 - $\neg f$ • $f_1 \wedge f_2$ • $X!^m f$ • $[f_1 U f_2]$ • $f @ clk$

Additional operators are derived from the basic operators defined above:¹

$$\begin{array}{ll}
 \bullet f_1 \vee f_2 \stackrel{\text{def}}{=} \neg(\neg f_1 \wedge \neg f_2) & \bullet f_1 \rightarrow f_2 \stackrel{\text{def}}{=} \neg f_1 \vee f_2 \\
 \bullet X^m f \stackrel{\text{def}}{=} \neg X!^m \neg f & \bullet F f \stackrel{\text{def}}{=} [T U f] \\
 \bullet G f \stackrel{\text{def}}{=} \neg F \neg f & \bullet [f_1 W f_2] \stackrel{\text{def}}{=} [f_1 U f_2] \vee G f_1 \\
 \bullet X f \stackrel{\text{def}}{=} X^1 f & \bullet X! f \stackrel{\text{def}}{=} X!^1 f
 \end{array}$$

We refer to the subset of LTL° consisting of the formulas that have no clock operator, by LTL . This subset is a slight generalization of LTL as defined in [4] – it consists of two version of boolean expressions as well as the generalized version of the next operator. The important thing, however, is that it agrees with the semantics of [4] on the common operators (on non-empty paths, as in [4] the definition is restricted to non-empty paths).

2.2 Semantics

We denote a letter by ℓ , and an empty, finite, or infinite word by u , v , or w . The *concatenation* of u and v is denoted by uv . If u is infinite, then $uv = u$. The empty word is denoted by ϵ , so that $w\epsilon = \epsilon w = w$. We denote the *length* of word v as $|v|$. The empty word ϵ has length 0, a finite word $v = (\ell_0 \ell_1 \cdots \ell_n)$ has length $n + 1$, and an infinite word has length ∞ . We use i , j , and k to denote non-negative integers. For $i < |v|$ we use v^i to denote the $(i + 1)^{st}$ letter of v (since counting of letters

¹ Where T is an atomic proposition that holds on every letter. In the sequel, we also use F , which is an atomic proposition that does not hold for any letter.

starts at zero). We denote by $v^{i\cdot}$ the suffix of v starting at v^i . We denote by ℓ^k the word of length k , each letter of which is ℓ , and by ℓ^ω the infinite-length word, each letter of which is ℓ .

The semantics of LTL° is defined inductively with respect to the alphabet $\Sigma = 2^P$. For a boolean expression $b \in B = 2^{2^P}$ and a letter $\ell \in \Sigma = 2^P$ we define the boolean satisfaction relation \models by $\ell \models b$ iff $\ell \in b$.

We first present the semantics of LTL° minus the clock operator over infinite, finite, and empty words (*unclocked semantics*). We then present the semantics of LTL° over infinite, finite, and empty words (*clocked semantics*). Later, we relate the two.

Unclocked semantics We now present semantics for LTL° minus the clock operator. The semantics are defined with respect to an infinite, finite, or empty word. The notation $w \models f$ means that formula f holds along the word w . The semantics are defined as follows, where b denotes a boolean expression, f , f_1 , and f_2 denote formulas, and m , j and k denote natural numbers (i.e., non-negative integers).

- $w \models b \iff |w| = 0$ or $w^0 \models b$
- $w \models b! \iff |w| > 0$ and $w^0 \models b$
- $w \models \neg f \iff w \not\models f$
- $w \models f_1 \wedge f_2 \iff w \models f_1$ and $w \models f_2$
- $w \models X!^m f \iff |w| > m$ and $w^{m\cdot} \models f$
- $w \models [f_1 \cup f_2] \iff \exists k < |w|$ such that $w^{k\cdot} \models f_2$, and $\forall j < k$ $w^{j\cdot} \models f_1$

Clocked semantics We define the semantics of an LTL° formula with respect to an infinite, finite, or empty word w and a context c , where c is a boolean expression over P . We say that a finite word w is a *clock tick of* clock c if c holds at the last letter of w and does not hold at any previous letter of w . Formally,

Definition 1 (clock ticks).

- We say that finite word w is a clock tick of c iff $|w| > 0$ and $w^{|w|-1} \models c$ and for every natural number $i < |w| - 1$, $w^i \not\models c$.
- For $m > 0$, we say that finite word w is m clock ticks of c iff there exists m words w_1, w_2, \dots, w_m such that $w = w_1 w_2 \dots w_m$ and for every $1 \leq i \leq m$ the word w_i is a clock tick of c .

The notation $w \models^c f$ means that formula f holds along the word w in the context of clock c . The semantics are defined as follows, where b , c and c_1 denote boolean expressions, f , f_1 , and f_2 denote formulas, and m , j and k denote natural numbers (i.e., non-negative integers).

- $w \models^c b \iff$ if $\exists k < |w|$ s.t. $w^{0..k}$ is a clock tick of c then $w^k \models b$
- $w \models^c b! \iff \exists k < |w|$ s.t. $w^{0..k}$ is a clock tick of c and $w^k \models b$
- $w \models^c \neg f \iff w \not\models^c f$

- $w \models^c f_1 \wedge f_2 \iff w \models^c f_1$ and $w \models^c f_2$
- $w \models^c X!^m f \iff \exists j < |w|$ s.t. $w^{0..j}$ are $m + 1$ clock ticks of c and $w^{j..} \models^c f$
- $w \models^c [f_1 \text{ U } f_2] \iff \exists k < |w|$ s.t. $w^k \models c$ and $w^{k..} \models^c f_2$ and $\forall j < k$ s.t. $w^j \models c$, $w^{j..} \models^c f_1$
- $w \models^c f @_{c_1} \iff w \models^{c_1} f$

The following claims provide some simple observation regarding the weak/strong next operators and their relation to weak/strong boolean expressions. The first claim states the direct semantics of the weak next operator.

Claim 1 (Weak next operator) *Let b be a boolean expression, m a non-negative integer and f be an LTL formula. Then*

$$w \models^c X^m f \iff \text{if } \exists j < |w| \text{ s.t. } w^{0..j} \text{ are } m + 1 \text{ clock ticks of } c \text{ then } w^{j..} \models^c f$$

The following claim states that the weak/strong boolean expressions can be stated in terms of the weak/strong next operator by setting m to zero. Note that this does not mean that the X^0 and $X!^0$ are redundant in the presence of weak and strong boolean expressions. They are needed since there is no way to get to the closest tick (when the current cycle is not a clock tick) for general formulas.

Claim 2 (Weak/strong Boolean expressions) *Let b be a boolean expression. Then*

1. $w \models^c b! \iff w \models^c X!^0 b$
2. $w \models^c b \iff w \models^c X^0 b$

The following claim shows that the “simple” strong and weak next operators, obtained by setting m to one in $X!^m$ and X^m , have the same semantics as defined in [2].

Claim 3 (Weak/strong simple next operators) *Let c be a boolean expression and f be an LTL formula.*

1. $w \models^c X!f \iff \exists j < k < |w|$ s.t. $w^{0..j}$ is a clock tick of c and $w^{j+1..k}$ is a clock tick of c and $w^{k..} \models^c f$
2. $w \models^c Xf \iff \text{if } \exists j < k < |w|$ s.t. $w^{0..j}$ is a clock tick of c and $w^{j+1..k}$ is a clock tick of c then $w^{k..} \models^c f$

The following claim states that $X!^m$ can be obtained by m applications of $X!$ and similarly for the weak version.

Claim 4 (Power characterization) *Let c be a boolean expression, m a positive integer, and f be an LTL formula.*

1. $w \models^c X!^m f \iff w \models^c \underbrace{X!X!\dots X!}_m f$
2. $w \models^c X^m f \iff w \models^c \underbrace{XX\dots X}_m f$

The following claim states that the $X!^m$ and X^m are additive as expected (even if n or m are zero).

Claim 5 *Let m and n be non-negative integers and f and LTL^{\circledast} formula. Then*

1. $X!^m X!^n f \equiv X!^{m+n} f$
2. $X^m X^n f \equiv X^{m+n} f$

The proof of these claims are given in the appendix.

3 Meeting the goals

In this section, we show that the logic LTL^{\circledast} satisfies all the goals of [2], as well as the least fixed point characterization. We make use of the following definitions.

Definition 2 (Projection). *The projection of word w onto clock c , denoted $w|_c$, is the word obtained from w after leaving only the letters which satisfy c .*

Definition 3 (Unclocked equivalent). *Two LTL^{\circledast} formulas f and g with no clock operator are unclocked equivalent ($f \equiv g$) if for all words w , $w \models f$ if and only if $w \models g$.*

Definition 4 (Clocked equivalent). *Two LTL^{\circledast} formulas f and g are clocked equivalent ($f \equiv^c g$) if for all words w and all contexts c , $w|_c \models f$ if and only if $w|_c \models g$.*

3.1 The until fixed point characterization

In standard LTL, $[f \text{ U } g]$ can be defined as a least fixed point of the equation $S = g \vee (f \wedge X! S)$. In [2] the proposed least fixed point characterization of until is by the equation: $S = ((T! \wedge g) \vee (f \wedge X! S))@c$, where $T!$ is the strong proposition T asserting that there is a current cycle (and T holds on it). This characterization holds for singly-clocked formulas but not for multiply-clocked formulas. The following counter examples shows that the characterization breaks when multiple clocks are involved. Let p, q and d be atomic propositions, and let $f = q@d$. Consider a word w such that $w^0 \models d \wedge q$ and for all $i > 0$, $w^i \not\models d \wedge q$, and $w^0 \not\models c$. Then $w \not\models^c f$ hence $w \not\models (T! \wedge f) \vee (p \wedge X! [p \text{ U } f])$. However, since $w^0 \not\models c$, and there is no state other than w^0 where $d \wedge q$ holds, $w \models^c [p \text{ U } f]$.

The following claim states that under the semantics given here, the until operator can be defined as a least fixed point of the equation $S = X!^0(g \vee (f \wedge X! S))$ (even in the presence of multiple clocks). Since by definition

$$w \models^T X!^0 f \iff |w| > 0 \text{ and } w \models f$$

this can be seen as a generalization of the standard characterization: The standard characterization works with no clock context, or equivalently with T as the clock context - thus the $X!^0$ operators can be removed. The obtained equation $S = g \vee (f \wedge X! S)$ is then the standard characterization. If we restrict also to infinite paths, the strength form the $X!$ operator can be taken away as well.

Claim 6 *Let f and g be $\text{LTL}^{\textcircled{c}}$ formulas. Then*

$$[f \text{ U } g] \stackrel{\textcircled{c}}{\equiv} \text{X}!^0 (g \vee (f \wedge \text{X}! [f \text{ U } g]))$$

Proof.

$$\begin{aligned} & w \models^c \text{X}!^0 (g \vee (f \wedge \text{X}! [f \text{ U } g])) \\ \iff & \exists j < |w| \text{ s.t. } w^{0..j} \text{ is a clock tick of } c \text{ and } w^{j..} \models^c (g \vee (f \wedge \text{X}! [f \text{ U } g])) \\ \iff & \exists j < |w| \text{ s.t. } w^{0..j} \text{ is a clock tick of } c \text{ and either } w^{j..} \models^c g \text{ or } w^{j..} \models^c (f \wedge \\ & \quad \text{X}! [f \text{ U } g]) \\ \iff & \exists j_1 < |w| \text{ s.t. } w^{0..j_1} \text{ is a clock tick of } c \text{ and either } w^{j_1..} \models^c g \text{ or } (w^{j_1..} \models^c f \\ & \quad \text{and } \exists j_2 > j_1 \text{ s.t. } w^{j_1+1..j_2} \text{ is a clock tick of } c \text{ and } w^{j_2..} \models^c [f \text{ U } g]) \\ \iff & \exists j_1 < |w| \text{ s.t. } w^{0..j_1} \text{ is a clock tick of } c \text{ and either } w^{j_1..} \models^c g \text{ or } (w^{j_1..} \models^c f \\ & \quad \text{and } \exists j_2 > j_1 \text{ s.t. } w^{j_1+1..j_2} \text{ is a clock tick of } c \text{ and } \exists k < |w^{j_2+2..}| \text{ such that} \\ & \quad w^{j_2+k} \models^c c \text{ and } w^{j_2+k..} \models^c g \text{ and for every } j < k \text{ such that } w^{j_2+j} \models^c c, \\ & \quad w^{j_2+j..} \models^c f \\ \iff & \exists k < |w| \text{ s.t. } w^k \models^c c \text{ and } w^{k..} \models^c g \text{ and } \forall j < k \text{ s.t. } w^j \models^c c, w^{j..} \models^c f \\ \iff & w \models^c f \text{ U } g \end{aligned}$$

3.2 The other goals

Below we state that the semantics given here preserve the goals met in [2]. For explanation and motivation of the goals, as well as relation to other works, see [2]. The proofs are a slight modification of those of [2] and are given in the appendix.

1. When singly-clocked, the semantics should be that of the projection view.

Proposition 1. *For any $\text{LTL}^{\textcircled{c}}$ formula f with no clock operator, a boolean expression c and an infinite, finite, or empty word w , the following holds:*

$$w \models^c f \quad \text{if and only if} \quad w|_c \models f$$

The following is an immediate consequence of this.

Corollary 1. *for an $\text{LTL}^{\textcircled{c}}$ formula with no clock operator f , and a word w .*

$$w \models^{\text{T}} f \quad \text{if and only if} \quad w \models f$$

2. Clocks should not accumulate.

Proposition 2. *For any $\text{LTL}^{\textcircled{c}}$ formula f and boolean expressions c_1 and c_2 the following holds:*

$$f @_{c_1} @_{c_2} \stackrel{\textcircled{c}}{\equiv} f @_{c_1}$$

3. The clock operator should be its own dual.

Proposition 3. *For any $\text{LTL}^{\textcircled{c}}$ formula f and boolean expression b the following holds:*

$$(\neg f) @ b \stackrel{\textcircled{c}}{\equiv} \neg(f @ b)$$

4. There should be a clocked version of $(F p) \wedge (G q)$ that is meaningful on paths with a finite number of clock ticks. Indeed, $((F p) \wedge (G q))@c$, holds if p holds for some state and q holds for all states on the projected path.
5. For any atomic proposition p , if $(F p)@clk$ holds on a path, it should hold on any extension of that path.

Proposition 4. *For boolean expressions b , clk and c , a finite word w , and an infinite or finite word w' , the following holds:*

$$w \models^c (F b)@clk \implies ww' \models^c (F b)@clk$$

6. For any clock c , two equivalent LTL formulas should remain equivalent when clocked with c .

Proposition 5. *For $LTL^@$ formulas f and g with no clock operators, and a boolean expression b , the following holds:*

$$f \equiv g \implies f@b \equiv g@b$$

7. Substituting sub-formula g for an equivalent sub-formula h should not change the truth value of the original formula.

Proposition 6. *If g is a sub-formula of f , and $g' \equiv g$, then the following holds:*

$$f \equiv f@g[g \leftarrow g']$$

where $\varphi[\psi \leftarrow \psi']$ denotes the formula obtained from φ by replacing sub-formula ψ with ψ' .

8. The truth value of $LTL^@$ Formula 1 should be the same as the truth value of LTL Formula 2 for every path.

Proposition 7. *For every word w ,*

$$w \models^T (G(p \rightarrow X q))@clka \iff w \models G((clka \wedge p) \rightarrow X[\neg clka \ W (clka \wedge q)])$$

3.3 Rewrite Rules

In [2] it was shown that the clock operator does not add expressive power. In fact there are rewrite rules that given an $LTL^@$ formula f return an equivalent LTL formula. The rewrite rules forms a recursive procedure $\mathcal{T}^{clk}()$, whose application starting with $clk = T$ results in an LTL formula with the same truth value in context T . The rewrite rules are given below. Note that by Claim 4 it suffices to provide rewrite rules for $X!^0$ and $X!$ instead of $X!^m$.

- $\mathcal{T}^{clk}(b) = [\neg clk \ W (clk \wedge b)]$
- $\mathcal{T}^{clk}(b!) = [\neg clk \ U (clk \wedge b)]$
- $\mathcal{T}^{clk}(X!^0 f) = [\neg clk \ U (clk \wedge f)]$
- $\mathcal{T}^{clk}(\neg f) = \neg \mathcal{T}^{clk}(f)$
- $\mathcal{T}^{clk}(f_1 \wedge f_2) = \mathcal{T}^{clk}(f_1) \wedge \mathcal{T}^{clk}(f_2)$

- $\mathcal{T}^{clk}(\mathbf{X}! f) = [\neg clk \cup (clk \wedge \mathbf{X}![\neg clk \cup (clk \wedge \mathcal{T}^{clk}(f))])]$
- $\mathcal{T}^{clk}([f_1 \cup f_2]) = [(clk \rightarrow \mathcal{T}^{clk}(f_1)) \cup (clk \wedge \mathcal{T}^{clk}(f_2))]$
- $\mathcal{T}^{clk}(f @ clk_1) = \mathcal{T}^{clk_1}(f)$

Proposition 8. *Let f be any LTL[@] formula, c a boolean expression, and w a word.*

$$w \models^c f \quad \text{if and only if} \quad w \models \mathcal{T}^c(f)$$

The proof of this proposition as well as some additional rewrite rules are given in the Appendix.

4 Conclusions

In [2] a relatively simple definition of LTL augmented with a clock operator was given. The augmented logic is suitable for specifying properties in multiply-clocks designs [1, Chapter 14]. In this definition, the only role of the clock operator is to define a projection of the path, and it is its own dual. This definition was shown to answer a list of design goals. However it does not preserve the least fixed point characterization of the until operator.

In this work we fix this problem with a minor addition to the semantics of [2]. The addition introduces a “power” to the next operator. The key of this solution is that by taking the zero power we get the operators $\mathbf{X}!^0$ and \mathbf{X}^0 which can be thought of as alignment operators - taking us to the closest clock tick, if the current cycle is not a clock tick.

The suggested semantics can be seen as a way to abstract the path when multiple clocks are involved. The clock operator $@$ defines the current clock context, so that the temporal operators move according to this context. For example $[f \cup g]$ demands that g hold on some future tick of the context clock, and f holds on all ticks preceding the tick where g holds. The alignment operators $\mathbf{X}!^0$ and \mathbf{X}^0 allow you to move to the closest tick of a clock, which is needed in the event of a clock switch.

The alignment operators $\mathbf{X}!^0$ and \mathbf{X}^0 move to the nearest concurrent or future clock tick. It might be practically useful to include also alignment operators that are strictly future. That is, while $\mathbf{X}!^0$ and \mathbf{X}^0 do not advance when the current cycle is a clock tick, the strictly future alignment operators will advance to the next clock tick, when the current cycle is a clock tick (and to the closest clock tick when the current cycle is not a clock tick). These can be defined as syntactic sugaring by means of the existing operators, using \mathbf{T} as the clock context.

Acknowledgements

I would like to thank Cindy Eisner and John Havlicek for their comments on an early draft of this paper.

References

1. C. Eisner and D. Fisman. *A practical introduction to PSL*. Springer, July 2006.
2. C. Eisner, D. Fisman, J. Havlicek, A. McIsaac, and D. Van Campenhout. The definition of a temporal clock operator. In *Proc. 30th Int. Colloq. Aut. Lang. Prog. (ICALP'03)*, LNCS 2719, pages 857–870, June 2003.
3. Z. Manna and A. Pnueli. *Temporal Verification of Reactive Systems: Specification*. Springer-Verlag, New York, 1992.
4. A. Pnueli. In transition from global to modular temporal reasoning about programs. In K. Apt, editor, *Logics and Models of Concurrent Systems*, volume F-13 of NATO Advanced Summer Institutes, pages 123–144. Springer-Verlag, a985.

A Proofs of Claims of Section 2.2

Claim 1. *Let b be a boolean expression and f be an LTL formula. Then*

$$w \models^c X^m f \iff \text{if } \exists j < |w| \text{ s.t. } w^{0..j} \text{ are } m+1 \text{ clock ticks of } c \text{ then } w^{j..} \models^c f$$

Proof.

$$\begin{aligned} & w \models^c X^m f \\ \iff & w \models^c \neg X!^m \neg f \\ \iff & w \not\models^c X!^m \neg f \\ \iff & \text{not (there exist } j < |w| \text{ such that } w^{0..j} \text{ are } m+1 \text{ clock ticks of } c \text{ and } \\ & \quad w^{j..} \models^c \neg f) \\ \iff & \text{either there does not exist } j < |w| \text{ such that } w^{0..j} \text{ are } m+1 \text{ clock ticks of } c \\ & \quad \text{or } w^{j..} \not\models^c \neg f \\ \iff & \text{either there does not exist } j < |w| \text{ such that } w^{0..j} \text{ are } m+1 \text{ clock ticks of } c \\ & \quad \text{or } w^{j..} \models^c f \\ \iff & \text{if there exist } j < |w| \text{ such that } w^{0..j} \text{ are } m+1 \text{ clock ticks of } c \text{ then } w^{j..} \models^c f \end{aligned}$$

Claim 2. *Let b be a boolean expression. Then*

1. $w \models^c b! \iff w \models^c X!^0 b$
2. $w \models^c b \iff w \models^c X^0 b$

Proof.

1. $w \models X^0 b$
 - \iff [By definition of X^m]
 - if there exist $j < |w|$ such that $w^{0..j}$ is 1 clock tick of c then $w^{j..} \models^c b$
 - \iff [By definition of weak boolean b]
 - if there exist $j < |w|$ such that $w^{0..j}$ is 1 clock tick of c then if there exist $k < |w^{j..}|$ such that $w^{j..k}$ is a clock tick of c then $w^k \models b$
 - \iff [If $w^{0..j}$ is a clock tick then so is $w^{j..j}$]
 - if there exist $k < |w|$ such that $w^{0..k}$ is a clock tick of c then $w^k \models b$
 - $\iff w \models^c b$
2. $w \models^c X!^0 b$
 - \iff [By definition of $X!^m$]
 - there exist $j < |w|$ such that $w^{0..j}$ is 1 clock tick of c and $w^{j..} \models^c b$
 - \iff [By definition of strong boolean $b!$]
 - there exist $j < |w|$ such that $w^{0..j}$ is 1 clock tick of c and there exist $k < |w^{j..}|$ such that $w^{j..k}$ is a clock tick of c and $w^k \models b$
 - \iff [if $w^{0..j}$ is a clock tick then so is $w^{j..j}$]
 - there exist $k < |w|$ such that $w^{0..k}$ is a clock tick of c and $w^k \models b$
 - $\iff w \models^c b!$

Claim 3. Let c be a boolean expression and f be an LTL formula.

1. $w \models^c X!f \iff \exists j < k < |w|$ s.t. $w^{0..j}$ is a clock tick of c and $w^{j+1..k}$ is a clock tick of c and $w^{k..} \models^c f$
2. $w \models^c Xf \iff$ if $\exists j < k < |w|$ s.t. $w^{0..j}$ is a clock tick of c and $w^{j+1..k}$ is a clock tick of c then $w^{k..} \models^c f$

Proof.

1. Follows directly from the definition.
2. Follows directly from the definition and Claim 1.

Claim 4. Let c be a boolean expression and f be an LTL formula.

1. $w \models^c X!^m f \iff w \models^c \underbrace{X!X!\dots X!}_m f$
2. $w \models^c X^m f \iff w \models^c \underbrace{XX\dots X}_m f$

Proof. The proof is by induction on m .

1. For $m = 1$ the claim holds by definition of the semantics of $X!^m$ and of $X!$.
Assume the claim holds for an arbitrary $m > 1$. Then

$$\begin{aligned} & w \models^c X!^m f \\ \iff & \exists j < |w| \text{ s.t. } w^{0..j} \text{ are } m+1 \text{ clock ticks of } c \text{ and } w^{j..} \models^c f \\ \iff & \exists i < j < |w| \text{ s.t. } w^{0..i} \text{ is a clock tick of } c \text{ and } w^{i+1..j} \text{ are } m \text{ clock ticks of } c \\ & \text{and } w^{j..} \models^c f \\ \iff & \text{[By induction]} \\ & \exists i < |w| \text{ s.t. } w^{0..i} \text{ is a clock tick of } c \text{ and } w^{i..} \models^c X!^{m-1} f \\ \iff & \exists i < |w| \text{ s.t. } w^{0..i} \text{ is a clock tick of } c \text{ and } w^{i..} \models^c \underbrace{X!X!\dots X!}_{m-1} f \\ \iff & w \models^c X! \underbrace{X!X!\dots X!}_{m-1} f \\ \iff & w \models^c \underbrace{X!X!\dots X!}_m f \end{aligned}$$
2. The proof is similar to the above only that it uses the definition of X^m as given in Claim 1.

Claim 5. Let m and n be non-negative integers and f and LTL^Q formula. Then

1. $X!^m X!^n f \equiv X!^{m+n} f$
2. $X^m X^n f \equiv X^{m+n} f$

Proof.

1. $w \models^c X!^m X!^n f$

$$\begin{aligned}
&\Leftrightarrow \exists j_1 < |w| \text{ s.t. } w^{0..j_1} \text{ are } m+1 \text{ clock ticks of } c \text{ and } w^{j_1..} \models^c \mathbf{X}!^n f \\
&\Leftrightarrow \exists j_1 < |w| \text{ s.t. } w^{0..j_1} \text{ are } m+1 \text{ clock ticks of } c \text{ and } \exists j_2 < |w^{j_1..}| \text{ s.t. } w^{j_1..j_2} \\
&\quad \text{are } n+1 \text{ clock ticks of } c \text{ and } w^{j_2..} \models^c f \\
&\Leftrightarrow \left[\begin{array}{l} \text{since } w^{0..j_1} \text{ are } m+1 \text{ clock ticks of } c, \text{ it follows that } w^{j_1} \models c \\ \text{and so } w^{j_1..j_1} \text{ is a clock tick of } c \end{array} \right] \\
&\quad \exists j_1 < |w| \text{ s.t. } w^{0..j_1} \text{ are } m+1 \text{ clock ticks of } c \text{ and } \exists j_2 < |w^{j_1+1..}| \text{ s.t. } w^{j_1+1..j_2} \\
&\quad \text{are } n \text{ clock ticks of } c \text{ and } w^{j_2..} \models^c f \\
&\Leftrightarrow \exists j < |w| \text{ s.t. } w^{0..j} \text{ are } m+1+n \text{ clock ticks of } c \text{ and } w^{j..} \models^c f \\
&\Leftrightarrow w \models^c \mathbf{X}!^{m+n} f \\
2. & w \models^c \mathbf{X}^m \mathbf{X}^n f \\
&\Leftrightarrow \text{if } \exists j_1 < |w| \text{ s.t. } w^{0..j_1} \text{ are } m+1 \text{ clock ticks of } c \text{ then } w^{j_1..} \models^c \mathbf{X}^n f \\
&\Leftrightarrow \text{if } \exists j_1 < |w| \text{ s.t. } w^{0..j_1} \text{ are } m+1 \text{ clock ticks of } c \text{ then if } \exists j_2 < |w^{j_1..}| \text{ s.t. } \\
&\quad w^{j_1..j_2} \text{ are } n+1 \text{ clock ticks of } c \text{ then } w^{j_2..} \models^c f \\
&\Leftrightarrow \left[\begin{array}{l} \text{if } w^{0..j_1} \text{ are } m+1 \text{ clock ticks of } c, \text{ then } w^{j_1} \models c \\ \text{and so } w^{j_1..j_1} \text{ is a clock tick of } c \end{array} \right] \\
&\quad \text{if } \exists j_1 < |w| \text{ s.t. } w^{0..j_1} \text{ are } m+1 \text{ clock ticks of } c \text{ then if } \exists j_2 < |w^{j_1+1..}| \text{ s.t. } \\
&\quad w^{j_1+1..j_2} \text{ are } n \text{ clock ticks of } c \text{ then } w^{j_2..} \models^c f \\
&\Leftrightarrow \text{if } \exists j < |w| \text{ s.t. } w^{0..j} \text{ are } m+1+n \text{ clock ticks of } c \text{ then } w^{j..} \models^c f \\
&\Leftrightarrow w \models^c \mathbf{X}^{m+n} f
\end{aligned}$$

B Proofs of Propositions of Section 3

Proposition 1. *For any LTL[@] formula f with no clock operator, a boolean expression c and an infinite, finite, or empty word w , the following holds:*

$$w \models^c f \quad \text{if and only if} \quad w|_c \models f$$

Proof. The proof is by induction on the structure of the formula:

1. $w \models^c b$
 - \iff [clocked semantics]
 - if $j < |w|$ s.t. $w^{0..j}$ is a clock tick of c , $w^j \models b$
 - \iff [definition of $w|_c$]
 - $|w|_c| = 0$ or $(w|_c)^0 \models b$
 - \iff [unclocked semantics]
 - $w|_c \models b$
2. $w \models^c b!$
 - \iff [clocked semantics]
 - there exists natural number $j < |w|$ s.t. $w^{0..j}$ is a clock tick of c and $w^j \models b$
 - \iff [definition of $w|_c$]
 - $|w|_c| > 0$ and $(w|_c)^0 \models b$
 - \iff [unclocked semantics]
 - $w|_c \models b!$
3. $w \models^c \neg g$
 - \iff [clocked semantics]
 - $w \not\models^c g$
 - \iff [induction]
 - $w|_c \not\models g$
 - \iff [unclocked semantics]
 - $w|_c \models \neg g$
4. $w \models^c g \wedge h$
 - \iff [clocked semantics]
 - $w \models^c g$ and $w \models^c h$
 - \iff [induction]
 - $w|_c \models g$ and $w|_c \models h$
 - \iff [unclocked semantics]
 - $w|_c \models g \wedge h$
5. $w \models^c X!^m g$
 - \iff [clocked semantics]
 - there exist $j < |w|$ such that $w^{0..j}$ are $m + 1$ clock ticks of c and $w^{j..} \models^c g$
 - \iff [by definition of $w|_c$, induction]
 - $|w|_c| > m$ and $(w|_c)^{m..} \models g$
 - \iff [unclocked semantics]
 - $w|_c \models X!^m g$
6. $w \models^c [g \cup h]$

\iff [clocked semantics]
 there exists natural number $k < |w|$ s.t. $w^k \models c$ and $w^{k..} \models^c h$ and for every $j < k$ s.t. $w^j \models c$, $w^{j..} \models^c g$.
 \iff [induction]
 there exists natural number $k < |w|$ s.t. $w^k \models c$ and $w^{k..}|_c \models h$ and for every $j < k$ s.t. $w^j \models c$, $w^{j..}|_c \models g$.
 \iff [by definition of $w|_c$]
 there exists natural number $k' < |w|_c$ s.t. $(w|_c)^{k'} \models h$ and for every $j' < k'$ $(w|_c)^{j'} \models g$.
 \iff [unclocked semantics]
 $w|_c \models [g \text{ U } h]$

□

Proposition 2. For any LTL° formula f and boolean expressions c_1 and c_2 the following holds:

$$f @_{c_1} @_{c_2} \stackrel{\circ}{=} f @_{c_1}$$

Proof. Let c be any boolean expression.

$$\begin{aligned}
 & w \models^c f @_{c_1} @_{c_2} \\
 \iff & w \models^{c_2} f @_{c_1} \\
 \iff & w \models^{c_1} f \\
 \iff & w \models^c f @_{c_1}
 \end{aligned}$$

Proposition 3. For any LTL° formula f and boolean expression b the following holds:

$$(\neg f) @ b \stackrel{\circ}{=} \neg(f @ b)$$

Proof. Let c be any boolean expression.

$$\begin{aligned}
 & w \models^c (\neg f) @ b \\
 \iff & w \models^b \neg f \\
 \iff & w \not\models^b f \\
 \iff & w \models^c f @ b \\
 \iff & w \models^c \neg(f @ b)
 \end{aligned}$$

Proposition 4. For boolean expressions b , clk and c , a finite word w , and an infinite or finite word w' , the following holds:

$$w \models^c (F b) @ clk \implies ww' \models^c (F b) @ clk$$

Proof. $w \models^c (F b) @ c$
 \iff [definition of F]
 $w \models^c [\text{T U } b] @ c$

\iff [clocked semantics]
 $w \models^c [\text{T U } b]$
 \iff [clocked semantics]
 there exists $k < |w|$ s.t. $w^k \models c$ and $w^{k..} \models^c b$, and for every $j < k$ s.t.
 $w^j \models c, w^{j..} \models^c \text{T}$
 \implies for every word w' there exists $k < |ww'|$ s.t. $(ww')^k \models c$ and $(ww')^{k..} \models^c b$,
 and for every $j < k$ s.t. $(ww')^j \models c, (ww')^{j..} \models^c \text{T}$
 \iff [clocked semantics]
 for every word $w', ww' \models^c [\text{T U } b]$
 \iff [clocked semantics]
 for every word $w', ww' \models^\kappa [\text{T U } b] \textcircled{c}$
 \iff [definition of F]
 for every word $w', ww' \models^\kappa (\text{F } b) \textcircled{c}$

□

Proposition 5. For $\text{LTL}^\textcircled{\circ}$ formulas f and g with no clock operators, and a boolean expression b , the following holds:

$$f \equiv g \implies f \textcircled{b} \equiv g \textcircled{b}$$

Proof. Assume by way of contradiction two unclocked equivalent formulas f and g for which $f \textcircled{c}$ is not clocked equivalent to $g \textcircled{c}$. That is, there exists a path w and context κ such that (without loss of generality) $w \models^\kappa f \textcircled{c}$ and $w \not\models^\kappa g \textcircled{c}$. By the semantics of $\textcircled{\circ}$ and Proposition 1, $w|_c \models f$ and $w|_c \not\models g$ in contradiction to f and g being unclocked equivalent. □

Proposition 6. If g is a sub-formula of f , and $g' \equiv g$, then the following holds:

$$f \equiv f[g \leftarrow g']$$

where $\varphi[\psi \leftarrow \psi']$ denotes the formula obtained from φ by replacing sub-formula ψ with ψ' .

Lemma 1. Let $f, f', g,$ and g' be $\text{LTL}^\textcircled{\circ}$ formulas, let c be a boolean expression, and assume that $f \equiv f'$ and $g \equiv g'$. Then

1. $\neg f \equiv \neg f'$
2. $f \wedge g \equiv f' \wedge g'$
3. $X^m f \equiv X^m f'$
4. $[f \text{ U } g] \equiv [f' \text{ U } g']$
5. $f \textcircled{c} \equiv f' \textcircled{c}$

Proof. (of the lemma)

1. $w \models^\kappa \neg f$
 $\iff w \not\models^\kappa f$
 $\iff [f \equiv^\circledast f']$
 $\iff w \not\models^\kappa f'$
 $\iff w \models^\kappa \neg f'$
2. $w \models^\kappa f \wedge g$
 $\iff w \models^\kappa f$ and $w \models^\kappa g$
 $\iff [f \equiv^\circledast f']$ and $[g \equiv^\circledast g']$
 $\iff w \models^\kappa f'$ and $w \models^\kappa g'$
 $\iff w \models^\kappa f' \wedge g'$
3. $w \models^\kappa X!^m f$
 \iff there exist $j < |w|$ such that $w^{0..j}$ are $m + 1$ clock ticks of κ and $w^{j..} \models^\kappa f$
 $\iff [f \equiv^\circledast f']$
 \iff there exist $j < |w|$ such that $w^{0..j}$ are $m + 1$ clock ticks of κ and $w^{j..} \models^\kappa f'$
 $\iff w \models^\kappa X!^m f'$
4. $w \models^\kappa f \cup g$
 \iff there exists a natural number $k < |w|$ such that $w^k \models \kappa$ and $w^{k..} \models^\kappa g$, and
for every natural number $j < k$ such that $w^j \models \kappa$, $w^{j..} \models^\kappa f$
 $\iff [f \equiv^\circledast f', g \equiv^\circledast g']$
 \iff there exists a natural number $k < |w|$ such that $w^k \models \kappa$ and $w^{k..} \models^\kappa g'$, and
for every natural number $j < k$ such that $w^j \models \kappa$, $w^{j..} \models^\kappa f'$
 $\iff w \models^\kappa f' \cup g'$
5. $w \models^\kappa f @c$
 $\iff w \models^c f$
 $\iff [f \equiv^\circledast f']$
 $\iff w \models^c f'$
 $\iff w \models^\kappa f' @c$

□

Proof. (of the proposition)

By induction. If $g = f$, then $f[g \leftarrow g'] = g'$, and so $g' \equiv^\circledast g$ implies that $f \equiv^\circledast f[g \leftarrow g']$. Assume now that g is a proper sub-formula. We consider cases for the top level structure of f .

1. $f = \neg f_1$. Then g is a subformula of f_1 . By induction, $f_1 \equiv^\circledast f_1[g \leftarrow g']$, and by Lemma 1, $f \equiv^\circledast \neg(f_1[g \leftarrow g']) = f[g \leftarrow g']$.
2. $f = f_1 \wedge f_2$. Without loss of generality, g is a subformula of f_1 . By induction, $f_1 \equiv^\circledast f_1[g \leftarrow g']$, and by Lemma 1, $f \equiv^\circledast (f_1[g \leftarrow g']) \wedge f_2 = f[g \leftarrow g']$.
3. $f = X!^m f_1$. Then g is a subformula of f_1 . By induction, $f_1 \equiv^\circledast f_1[g \leftarrow g']$, and by Lemma 1, $f \equiv^\circledast X!^m (f_1[g \leftarrow g']) = f[g \leftarrow g']$.
4. $f = f_1 \cup f_2$. Without loss of generality, g is a subformula of f_1 . By induction, $f_1 \equiv^\circledast f_1[g \leftarrow g']$, and by Lemma 1, $f \equiv^\circledast (f_1[g \leftarrow g']) \cup f_2 = f[g \leftarrow g']$.

5. $f = f_1 @c$. Then g is a subformula of f_1 . By induction, $f_1 \stackrel{\circ}{=} f_1[g \leftarrow g']$, and by Lemma 1, $f \stackrel{\circ}{=} (f_1[g \leftarrow g'])@c = f[g \leftarrow g']$. □

Proposition 7. *For every word w ,*

$$w \stackrel{T}{=} (G(p \rightarrow Xq))@clka \iff w \models G((clka \wedge p) \rightarrow X[\neg clka \ W (clka \wedge q)])$$

For convenience, the proof uses rewrite rules for derived formulas, in addition to the rewrite rules presented in Section 3.3. These rules can be obtained by applying the original set of rewrite rules to the additional operators presented in Section 2.1. The derived rewrite rules are:

- $\mathcal{T}^c(f \vee g) \equiv \mathcal{T}^c(f) \vee \mathcal{T}^c(g)$
- $\mathcal{T}^c(f \rightarrow g) \equiv \mathcal{T}^c(f) \rightarrow \mathcal{T}^c(g)$
- $\mathcal{T}^c(X f) \equiv [\neg c \ W (c \wedge X(\neg c \ W (c \wedge \mathcal{T}^c(f))))]$
- $\mathcal{T}^c(F f) \equiv F(c \wedge \mathcal{T}^c(f))$
- $\mathcal{T}^c(G f) \equiv G(c \rightarrow \mathcal{T}^c(f))$
- $\mathcal{T}^c([f \ W \ g]) \equiv [(c \rightarrow \mathcal{T}^c(f)) \ W (c \wedge \mathcal{T}^c(g))]$

Proof. $\mathcal{T}^T((G(p \rightarrow Xq))@c)$
 $\equiv \mathcal{T}^c(G(p \rightarrow Xq))$
 $\equiv G(c \rightarrow \mathcal{T}^c(p \rightarrow Xq))$
 $\equiv G(c \rightarrow (\mathcal{T}^c(p) \rightarrow \mathcal{T}^c(Xq)))$
 $\equiv G(c \rightarrow ([\neg c \ W (c \wedge p)] \rightarrow [\neg c \ W (c \wedge X[\neg c \ W (c \wedge \mathcal{T}^c(q))])]))$
 $\equiv G(c \rightarrow ((c \wedge p) \rightarrow [\neg c \ W (c \wedge X[\neg c \ W (c \wedge \mathcal{T}^c(q))])]))$
 $\equiv G((c \wedge p) \rightarrow [\neg c \ W (c \wedge X[\neg c \ W (c \wedge \mathcal{T}^c(q))])])$
 $\equiv G((c \wedge p) \rightarrow X[\neg c \ W (c \wedge \mathcal{T}^c(q))])$
 $\equiv G((c \wedge p) \rightarrow X[\neg c \ W (c \wedge [\neg c \ W (c \wedge q)])])$
 $\equiv G((c \wedge p) \rightarrow X[\neg c \ W (c \wedge q)])$

□

Proposition 8. *Let f be any LTL[@] formula, c a boolean expression, and w a word.*

$$w \stackrel{c}{=} f \quad \text{if and only if} \quad w \models \mathcal{T}^c(f)$$

Proof. By induction on formula structure. Throughout, j and k are understood to be natural numbers. Note that by Claim 1 (last two items) we can use in the induction X and $X!$ instead of X^m and $X!^m$.

1. $w \models \mathcal{T}^c(b)$
 \iff [definition of $\mathcal{T}^c()$]
 $w \models [\neg c \ W (c \wedge b)]$
 \iff either there exists $k < |w|$ such that $w^{k..} \models c \wedge b$ and for every $j < k$, $w^{j..} \models \neg c$, or for every $j < |w|$, $w^{j..} \models \neg c$
 \iff if there exists $k < |w|$ such that $w^{0..k}$ is a clock tick of c , then $w^k \models b$
 $\iff w \stackrel{c}{=} b$

2. $w \models \mathcal{T}^c(b!)$
 - \iff [definition of $\mathcal{T}^c()$]
 - $w \models [\neg c \cup (c \wedge b)]$
 - \iff there exists $k < |w|$ such that $w^{k..} \models c \wedge b$ and for every $j < k$, $w^{j..} \models \neg c$
 - \iff there exists $k < |w|$ such that $w^{0..k}$ is a clock tick of c and $w^k \models b$
 - $\iff w \stackrel{c}{\models} b!$
3. $w \models \mathcal{T}^c(\mathbf{X}!^0 f)$
 - \iff [definition of $\mathcal{T}^c()$]
 - $w \models [\neg c \cup (c \wedge f)]$
 - \iff there exists $k < |w|$ such that $w^{k..} \models c \wedge f$ and for every $j < k$, $w^{j..} \models \neg c$
 - \iff there exists $k < |w|$ such that $w^{0..k}$ is a clock tick of c and $w^k \models f$
 - $\iff w \stackrel{c}{\models} \mathbf{X}!^0 f$
4. $w \models \mathcal{T}^c(\neg g)$
 - \iff [definition of $\mathcal{T}^c()$]
 - $w \models \neg \mathcal{T}^c(g)$
 - $\iff w \not\models \mathcal{T}^c(g)$
 - \iff [induction]
 - $w \not\stackrel{c}{\models} g$
 - $\iff w \stackrel{c}{\models} \neg g$
5. $w \models \mathcal{T}^c(g \wedge h)$
 - \iff [definition of $\mathcal{T}^c()$]
 - $w \models \mathcal{T}^c(g) \wedge \mathcal{T}^c(h)$
 - $\iff w \models \mathcal{T}^c(g)$ and $w \models \mathcal{T}^c(h)$
 - \iff [induction]
 - $w \stackrel{c}{\models} g$ and $w \stackrel{c}{\models} h$
 - $\iff w \stackrel{c}{\models} g \wedge h$
6. $w \models \mathcal{T}^c(\mathbf{X}!g)$
 - \iff [definition of $\mathcal{T}^c()$]
 - $w \models [\neg c \cup (c \wedge \mathbf{X}![\neg c \cup (c \wedge \mathcal{T}^c(g))])]$
 - \iff there exists $k < |w|$ such that $w^{k..} \models c \wedge \mathbf{X}![\neg c \cup (c \wedge \mathcal{T}^c(g))]$ and for every $j < k$, $w^{j..} \models \neg c$
 - \iff there exists $k < |w|$ such that $w^{0..k}$ is a clock tick of c and $w^{k..} \models \mathbf{X}![\neg c \cup (c \wedge \mathcal{T}^c(g))]$
 - \iff there exists $k < |w|$ such that $w^{0..k}$ is a clock tick of c and $|w^{k..}| > 1$ and $(w^{k..})^{1..} \models [\neg c \cup (c \wedge \mathcal{T}^c(g))]$
 - \iff $[(w^{k..})^{1..} = w^{k+1..}]$
 - \iff there exists $k < |w|$ such that $w^{0..k}$ is a clock tick of c and $|w^{k+1..}| > 0$ and $w^{k+1..} \models [\neg c \cup (c \wedge \mathcal{T}^c(g))]$
 - \iff there exists $k < |w|$ such that $w^{0..k}$ is a clock tick of c and there exists $m < |w^{k+1..}|$ such that $(w^{k+1..})^{m..} \models c \wedge \mathcal{T}^c(g)$ and for every $j < m$, $(w^{k+1..})^{j..} \models \neg c$
 - \iff there exists $k < |w|$ such that $w^{0..k}$ is a clock tick of c and there exists $m < |w^{k+1..}|$ such that $(w^{k+1..})^{0..m}$ is a clock tick of c and $(w^{k+1..})^{m..} \models \mathcal{T}^c(g)$

- \iff [induction, $(w^{k+1..})^{m..} = w^{k+1+m..}$, $(w^{k+1..})^{0..m} = w^{k+1..k+1+m}$
 there exists $k < |w|$ such that $w^{0..k}$ is a clock tick of c and there exists
 $m < |w^{k+1..}|$ such that $w^{k+1..k+1+m}$ is a clock tick of c and $w^{k+1+m..} \models^c g$
- \iff [let $j = k + 1 + m$
 there exists $k < j < |w|$ such that $w^{0..k}$ is a clock tick of c and
 $w^{k+1..j}$ is a clock tick of c and $w^{j..} \models^c g$
- $\iff w \models^c \mathbf{X}!g$
7. $w \models \mathcal{T}^c([g \mathbf{U} h])$
- \iff [definition of $\mathcal{T}^c()$
 $w \models [(c \rightarrow \mathcal{T}^c(g)) \mathbf{U} (c \wedge \mathcal{T}^c(h))]$
- \iff there exists $k < |w|$ such that $w^{k..} \models c \wedge \mathcal{T}^c(h)$ and for every $j < k$,
 $w^{j..} \models c \rightarrow \mathcal{T}^c(g)$
- \iff there exists $k < |w|$ such that $w^k \models c$ and $w^{k..} \models \mathcal{T}^c(h)$ and for every $j < k$
 such that $w^j \models c$, $w^{j..} \models \mathcal{T}^c(g)$
- \iff [induction]
 there exists $k < |w|$ such that $w^k \models c$ and $w^{k..} \models^c h$ and for every $j < k$
 such that $w^j \models c$, $w^{j..} \models^c g$
- $\iff w \models^c [g \mathbf{U} h]$
8. $w \models \mathcal{T}^c(g \mathbf{O} d)$
- \iff [definition of $\mathcal{T}^c()$
 $w \models \mathcal{T}^d(g)$
- \iff [induction]
 $w \models^d g$
- $\iff w \models^c g \mathbf{O} d$

□