Acta Informatica manuscript No.
(will be inserted by the editor)

Antonín Kučera · Jan Strejček

The Stuttering Principle Revisited

Received: date / Revised version: date

Abstract It is known that LTL formulae without the 'next' operator are invariant under the so-called *stutter equivalence* of words. In this paper we extend this principle to general LTL formulae with given nesting depths of both 'next' and 'until' operators. This allows us to prove the semantical strictness of three natural hierarchies of LTL formulae, which are parametrized either by the nesting depth of just one of the two operators, or by both of them. Further, we provide an effective characterization of languages definable by LTL formulae with a bounded nesting depth of the 'next' operator.

Keywords Linear Temporal Logic (LTL), Stuttering

1 Introduction

Linear temporal logic (LTL) [12] is a popular formalism for specifying properties of (concurrent) programs. The syntax of LTL is given by the following abstract syntax equation:

$$\varphi \ ::= \ \mathtt{tt} \ \mid \ p \ \mid \ \neg\varphi \ \mid \ \varphi_1 \wedge \varphi_2 \ \mid \ \mathsf{X}\varphi \ \mid \ \varphi_1 \,\mathsf{U}\, \varphi_2$$

Here p ranges over a countable set $\Lambda = \{o, p, q, \ldots\}$ of *letters*. We also use $\mathsf{F}\varphi$ to abbreviate $\mathtt{tt}\,\mathsf{U}\,\varphi$, and $\mathsf{G}\varphi$ to abbreviate $\neg\mathsf{F}\neg\varphi$.

The first author is supported by the research centre "Institute for Theoretical Computer Science (ITI)", project No. 1M0021620808. The second author is supported by grant No. 1ET-408050503 and by GAČR, grant No. 201/03/1161.

This paper is a revised and extended version of [6].

Antonín Kučera · Jan Strejček
Faculty of Informatics, Masaryk University in Brno
Botanická 68a, CZ-60200 Brno, Czech Republic
Tel.: ++420-549 494 476, Fax: ++420-549 491 820
E-mail: {kucera,strejcek}@fi.muni.cz

We define the semantics of LTL in terms of languages over infinite words (all of our results carry over to finite words immediately). An *alphabet* is a finite set $\Sigma \subseteq \Lambda$. An *ω-word* over Σ is an infinite sequence[1] $\alpha = \alpha(0)\alpha(1)\alpha(2)\ldots$ of letters from Σ. The set of all ω-words over Σ is denoted Σ^ω. For every $i \geq 0$ we denote by α_i the i^{th} suffix of α, i.e., the word $\alpha(i)\alpha(i+1)\ldots$ (we use this notation for finite words as well). Moreover, for all $i \geq 0$ and $j > 0$, the symbol $\alpha(i,j)$ denotes the subword of α of length j starting with $\alpha(i)$.
Let φ be an LTL formula. The *validity* of φ for a given $\alpha \in \Sigma^\omega$ is defined inductively as follows:

$$
\begin{array}{lll}
\alpha \models \mathtt{tt} & & \\
\alpha \models p & \text{iff} & p = \alpha(0) \\
\alpha \models \neg\varphi & \text{iff} & \alpha \not\models \varphi \\
\alpha \models \varphi_1 \wedge \varphi_2 & \text{iff} & \alpha \models \varphi_1 \wedge \alpha \models \varphi_2 \\
\alpha \models \mathsf{X}\varphi & \text{iff} & \alpha_1 \models \varphi \\
\alpha \models \varphi_1 \,\mathsf{U}\, \varphi_2 & \text{iff} & \exists i \in \mathbb{N}_0 : \alpha_i \models \varphi_2 \wedge \forall\, 0 \leq j < i : \alpha_j \models \varphi_1
\end{array}
$$

Let us note that the results presented in this paper remain valid if the logic LTL is built over atomic propositions rather than over letters.

For every alphabet Σ and every LTL formula φ we define the language $L^\Sigma_\varphi = \{\alpha \in \Sigma^\omega \mid \alpha \models \varphi\}$. If Σ is understood from the context, we write just L_φ.

It is well-known that languages definable by LTL formulae form a proper subclass of ω-regular languages (see, e.g., [15]). More precisely, LTL languages are exactly the languages definable in first-order logic with the signature $\{suc, <\} \cup \Lambda'$, where *suc* and $<$ are binary predicates standing for *successor* and *less than*, respectively, and Λ' is a set of unary predicates corresponding to the set of letters Λ. See [5,4] for more details regarding the relationship between LTL and first-order logic.

Since LTL contains just two modal connectives, a natural question is how they influence the expressive power of LTL. First, let us (inductively) define the *nesting depth* of the X and the U modality in a given LTL formula φ, denoted $X(\varphi)$ and $U(\varphi)$, respectively.

$$
\begin{array}{ll}
U(\mathtt{tt}) = 0 & X(\mathtt{tt}) = 0 \\
U(p) = 0 & X(p) = 0 \\
U(\varphi \wedge \psi) = \max\{U(\varphi), U(\psi)\} & X(\varphi \wedge \psi) = \max\{X(\varphi), X(\psi)\} \\
U(\neg\varphi) = U(\varphi) & X(\neg\varphi) = X(\varphi) \\
U(\mathsf{X}\varphi) = U(\varphi) & X(\mathsf{X}\varphi) = X(\varphi) + 1 \\
U(\varphi \,\mathsf{U}\, \psi) = \max\{U(\varphi), U(\psi)\} + 1 & X(\varphi \,\mathsf{U}\, \psi) = \max\{X(\varphi), X(\psi)\}
\end{array}
$$

Now we introduce three natural hierarchies of LTL formulae. For all $m, n \in \mathbb{N}_0$ we define

$$
\begin{aligned}
\mathrm{LTL}(\mathsf{U}^m, \mathsf{X}^n) &= \{\varphi \in \mathrm{LTL} \mid U(\varphi) \leq m \;\wedge\; X(\varphi) \leq n\} \\
\mathrm{LTL}(\mathsf{U}^m, \mathsf{X}) &= \textstyle\bigcup_{i=0}^{\infty} \mathrm{LTL}(\mathsf{U}^m, \mathsf{X}^i) \\
\mathrm{LTL}(\mathsf{U}, \mathsf{X}^n) &= \textstyle\bigcup_{i=0}^{\infty} \mathrm{LTL}(\mathsf{U}^i, \mathsf{X}^n)
\end{aligned}
$$

[1] We use $o, p, q, \ldots$ to range over Σ, $u, v, \ldots$ to range over Σ^*, and $\alpha, \beta, \ldots$ to range over Σ^ω.

Hence, the LTL($\mathsf{U}^m, \mathsf{X}^n$) hierarchy takes into account the nesting depths of both modalities, while the LTL(U^m, X) and LTL(U, X^n) hierarchies 'count' just the nesting depth of U and X, respectively. The languages definable by formulae of LTL($\mathsf{U}^m, \mathsf{X}^n$) are called LTL($\mathsf{U}^m, \mathsf{X}^n$) *languages*.

Our work is motivated by basic questions about the presented hierarchies; in particular, the following problems seem to be among the most natural ones:

Question 1. Are those hierarchies semantically strict? That is, if we increase m or n just by one, do we always obtain a strictly more expressive fragment of LTL?

Question 2. If we take two classes A, B in the above hierarchies which are syntactically incomparable (for example, we can consider LTL($\mathsf{U}^4, \mathsf{X}^3$) and LTL($\mathsf{U}^2, \mathsf{X}^5$), or LTL($\mathsf{U}^3, \mathsf{X}^0$) and LTL($\mathsf{U}^2, \mathsf{X}$)), are they also semantically incomparable? That is, are there formulae $\varphi_A \in A$ and $\varphi_B \in B$ such that φ_A is not expressible in B and φ_B is not expressible in A?

Question 3. In the case of LTL($\mathsf{U}^m, \mathsf{X}^n$) hierarchy, what is the semantical intersection of LTL($\mathsf{U}^{m_1}, \mathsf{X}^{n_1}$) and LTL($\mathsf{U}^{m_2}, \mathsf{X}^{n_2}$)? That is, what languages are expressible in both fragments?

We provide (positive) answers to Question 1 and Question 2. Here, the results about the LTL($\mathsf{U}^m, \mathsf{X}^n$) hierarchy seem to be particularly interesting. As for Question 3, one is tempted to expect the following answer: The semantical intersection of LTL($\mathsf{U}^{m_1}, \mathsf{X}^{n_1}$) and LTL($\mathsf{U}^{m_2}, \mathsf{X}^{n_2}$) are exactly the languages expressible in LTL($\mathsf{U}^m, \mathsf{X}^n$), where $m = \min\{m_1, m_2\}$ and $n = \min\{n_1, n_2\}$. Surprisingly, this answer turns out to be *incorrect*. For all $m \geq 1$, $n \geq 0$ we give an example of a language L which is definable both in LTL($\mathsf{U}^{m+1}, \mathsf{X}^n$) and LTL($\mathsf{U}^m, \mathsf{X}^{n+1}$), but not in LTL($\mathsf{U}^m, \mathsf{X}^n$). This shows that the answer to Question 3 is not as easy as one might expect. In fact, Question 3 is left open as an interesting challenge directing our future work.

The results on Question 1 are closely related to the work of Etessami and Wilke [3] (see also [17] for an overview of related results). They consider an until hierarchy of LTL formulae which is similar to our LTL(U^m, X) hierarchy. The difference is that they treat the F operator 'explicitly', i.e., their U-depth counts just the nesting of the U operator and ignores all occurrences of X and F (in our approach, $\mathsf{F}\varphi$ is just an abbreviation for $\mathtt{tt}\,\mathsf{U}\,\varphi$, and hence 'our' U-depth of $\mathsf{F}p$ is one and not zero). They prove the strictness of their until hierarchy in the following way: First, they design an appropriate Ehrenfeucht-Fraïssé (EF) game for LTL (the game is played on a pair of words) which in a sense characterizes those pairs of words which can be distinguished by LTL formulae where the temporal operators are nested only to a certain depth. Then, for every k they construct a formula Fair_k with until depth k and prove that this particular formula cannot be equivalently expressed by any other formula with U-depth equal to $k-1$. Here the previous results about the designed EF game are used. Since the formula Fair_k contains just one F operator and many nested X and U operators, this proof carries over to our LTL(U^m, X) hierarchy. In fact, [3] presents a 'stronger' result in the sense that one additional nesting level of U cannot be 'compensated' by arbitrarily-deep nesting of X and F. On the other hand, the proof does not allow to conclude that, e.g., LTL($\mathsf{U}^3, \mathsf{X}^0$) contains a formula which is not expressible in LTL(U^2, X) (because Fair_k contains the nested X modalities).

Our method for solving Questions 1 and 2 is different. Instead of designing appropriate Ehrenfeucht-Fraïssé games which could (possibly) characterize the membership to LTL($\mathsf{U}^m, \mathsf{X}^n$), we formulate a general 'stuttering theorem' for LTL($\mathsf{U}^m, \mathsf{X}^n$) languages. Roughly speaking, the theorem says that under certain 'local-periodicity' conditions (which depend on m and n) one can remove a given subword u from a given word α without influencing the (in)validity of LTL($\mathsf{U}^m, \mathsf{X}^n$) formulae (we say that u is (m, n)-*redundant* in α). This result can be seen as a generalization of the well-known form of stutter invariance admitted by LTL(U, X^0) formulae (a detailed discussion is postponed to Section 2). Thus, we obtain a simple (but surprisingly powerful) tool allowing to prove that a certain formula φ is *not* definable in LTL($\mathsf{U}^m, \mathsf{X}^n$). The theorem is applied as follows: we choose a suitable alphabet Σ, consider the language L_φ, and find an appropriate $\alpha \in L_\varphi$ and its subword u such that

- u is (m, n)-redundant in α;
- $\alpha' \not\models \varphi$ where α' is obtained from α by deleting the subword u.

If we manage to do that, we can conclude that φ is not expressible in LTL($\mathsf{U}^m, \mathsf{X}^n$).

We use our stuttering theorem to answer Questions 1 and 2. Proofs are remarkably short though it took us some time to find appropriate formulae which witness the presented claims. It is worth noting that some of the known results about LTL (like, e.g., the formula '$\mathsf{G}_2 p$' is not definable in LTL) admit a one-line proof if our general stuttering theorem is applied. We also obtain an alternative characterization of LTL languages which are exactly the ω-regular languages closed under the generalized stutter equivalence of words (see Section 3). These results are still valid when interpreting LTL over finite words.

The paper is organized as follows. In Section 2 we formulate and prove a general stuttering theorem for LTL($\mathsf{U}^m, \mathsf{X}^n$) languages together with some related results. Using this theorem, we answer Questions 1–3 in Section 4. In Section 3, we examine the question whether the considered forms of stutter invariance fully characterize the corresponding LTL fragments. Finally, in Section 5 we draw our conclusions and identify directions of future research.

2 A General Stuttering Theorem

In this section we formulate and prove the promised general stuttering theorem for LTL($\mathsf{U}^m, \mathsf{X}^n$) languages. General stuttering combines and extends two independent principles of *letter stuttering* (n-stuttering) and *subword stuttering*, which are applicable to the LTL(U, X^n) and LTL($\mathsf{U}^m, \mathsf{X}^0$) fragments of LTL, respectively. We start by explaining these two principles in Section 2.1 and Section 2.2. This material has been included for two reasons. First, the two simplified principles are interesting on their own. In Section 3.1 we present special results about letter stuttering which do not hold for general stuttering. Secondly, the remarks and proof sketches given in Section 2.1 and Section 2.2 should help the reader in gaining some intuition about the functionality and underlying principles of general stuttering.

2.1 Letter stuttering (n-stuttering)

Letter stuttering is a simple generalization of the well-known principle of *stutter invariance* of LTL(U, X^0) formulae [8] saying that LTL(U, X^0) formulae cannot distinguish between one and more adjacent occurrences of the same letter in a given word. Formally, a letter $\alpha(i)$ of an ω-word α is called *redundant* iff $\alpha(i) = \alpha(i+1)$ and there is $j > i$ such that $\alpha(i) \neq \alpha(j)$. The *canonical form* of α is the ω-word obtained by deleting all redundant letters from α. Two ω-words α, β are *stutter equivalent* iff they have the same canonical form.

Theorem 1 ([8]) *Every* LTL(U, X^0) *language is closed under stutter equivalence.*

Intuitively, it is not very surprising that this principle can be extended to LTL(U, X^n) formulae (where $n \in \mathbb{N}_0$). The so-called *n-stuttering* is based on a simple observation that LTL(U, X^n) formulae cannot distinguish between $n+1$ and more adjacent occurrences of the same letter in a given ω-word. Formally, a letter $\alpha(i)$ is *n-redundant* iff $\alpha(i) = \alpha(i+1) = \cdots = \alpha(i+n+1)$ and there is some $j > i$ such that $\alpha(i) \neq \alpha(j)$. The *n-canonical form* and *n-stutter equivalence* are defined in the same way as above.

Theorem 2 (n-stuttering) *Every* LTL(U, X^n) *language is closed under n-stutter equivalence.*

Proof The theorem can be proven directly by induction on n. Since it is a consequence of Theorem 9, we do not give an explicit proof here[2]. $\square$

Theorem 2 can be used to show that a given property is *not* expressible in LTL(U, X^n) (or even in LTL) in the following way.

Example 3 A standard example of an ω-regular language which is not definable in LTL is '$\mathsf{G}_2 p$' [18]. The language consists of all $\alpha \in \Sigma^\omega$ such that $\alpha(i) = p$ for every even $i \in \mathbb{N}_0$. With the help of Theorem 2 we can easily prove that $\mathsf{G}_2 p$ is not an LTL(U, X^n) language for any $n \in \mathbb{N}_0$ (assuming $|\Sigma| \geq 2$) and hence it is not an LTL language. Suppose the converse, i.e., there are $n \in \mathbb{N}_0$ and $\varphi \in$ LTL(U, X^n) such that $L_\varphi = \mathsf{G}_2 p$. Now consider the ω-words $\alpha = p^{2n+2} q p^\omega$ and $\beta = p^{2n+1} q p^\omega$, where $q \in \Sigma \smallsetminus \{p\}$. Clearly α and β are n-stutter equivalent, and $\alpha \notin L_\varphi$ while $\beta \in L_\varphi$. Hence, L_φ is not n-stutter closed which contradicts Theorem 2.

2.2 Subword stuttering

Since letter stuttering takes into account just the X-depth of LTL formulae, a natural question is whether there is another form of stutter-like invariance determined by the U-depth of a given LTL formula. We provide a (positive) answer to this question by formulating the principle of *subword stuttering*,

[2] A direct proof of Theorem 2 is of course simpler than the proof of Theorem 9. It can be found in [7].

which is applicable to LTL($\mathsf{U}^m, \mathsf{X}^0$) formulae (where $m \geq 1$). The term 'subword stuttering' reflects the fact that we do not necessarily delete/pump just individual letters, but whole subwords. The essence of the idea is formulated in the following claim:

Claim 4 *Let $\varphi \in$ LTL($\mathsf{U}^m, \mathsf{X}^0$) where $m \geq 1$. For all $v, u \in \Sigma^*$ and $\alpha \in \Sigma^\omega$ we have that $vu^{m+1}\alpha \models \varphi$ iff $vu^m\alpha \models \varphi$.*

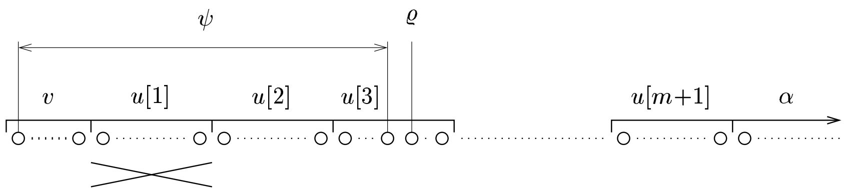
In other words, LTL($\mathsf{U}^m, \mathsf{X}^0$) cannot distinguish between m and more adjacent occurrences of the same subword u in a given word. Note that there are no assumptions about the length of u.

Claim 4 can be easily proven by induction on m. We just sketch the crucial part of the argument (a full proof is in fact contained in the proof of Theorem 9). Let us suppose that $\varphi = \psi \,\mathsf{U}\, \varrho$, where $\psi, \varrho \in$ LTL($\mathsf{U}^{m-1}, \mathsf{X}^0$). We want to show that $vu^{m+1}\alpha \models \varphi$ iff $vu^m\alpha \models \varphi$. We concentrate just on the induction step (i.e., $m \geq 2$) of the '$\Longrightarrow$' part (the other direction is similar). By induction hypothesis, the following equivalences hold for all $0 \leq \ell < |vu|$:

$$(vu)_\ell \, u^m \, \alpha \models \psi \;\text{ iff }\; (vu)_\ell \, u^{m-1} \, \alpha \models \psi \tag{1}$$

$$(vu)_\ell \, u^m \, \alpha \models \varrho \;\text{ iff }\; (vu)_\ell \, u^{m-1} \, \alpha \models \varrho \tag{2}$$

Let $vu^{m+1}\alpha \models \psi \,\mathsf{U}\, \varrho$. Then there is $j \in \mathbb{N}_0$ such that $(vu^{m+1}\alpha)_j \models \varrho$ and $(vu^{m+1}\alpha)_i \models \psi$ for all $0 \leq i < j$. If $j < |vu|$, we immediately obtain $vu^m\alpha \models \psi \,\mathsf{U}\, \varrho$ by applying (1) and (2) above. If $j \geq |vu|$, we can imagine that the word $vu^m\alpha$ was obtained from $vu^{m+1}\alpha$ by deleting the *first* copy of u (from now on, we denote the k^{th} copy of u in $vu^{m+1}\alpha$ by $u[k]$). The situation can be pictured as follows:



Realize that the (in)validity of ψ and ϱ for any suffix of $u[2]\,u[3]\,\cdots\,u[m+1]\,\alpha$ is not influenced by deleting the $u[1]$ subword (LTL is future-only in our settings). That is, it suffices to show that for each suffix v' of v we have that $v'u^{m+1}\alpha \models \psi$ implies $v'u^m\alpha \models \psi$. However, this follows from (1) above.

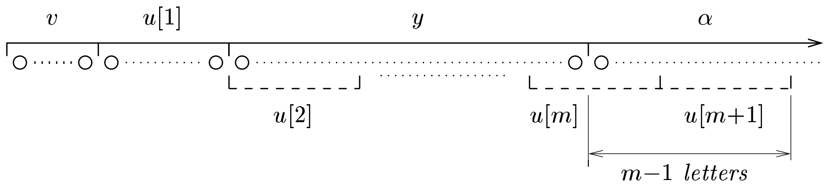
The principle of subword stuttering, as formulated in Claim 4, is quite simple and intuitively clear. Now we refine this principle into a stronger form.

Claim 5 *Let $\varphi \in$ LTL($\mathsf{U}^m, \mathsf{X}^0$) where $m \geq 0$. For all $v, y \in \Sigma^*$, $u \in \Sigma^+$, and $\alpha \in \Sigma^\omega$ such that*

- $|y| = |u| \cdot m - m + 1$,
- *y is a prefix of u^ω*

we have that $vuy\alpha \models \varphi$ iff $vy\alpha \models \varphi$.

The structure of $vuy\alpha$ can be illustrated as follows:



In other words, the u subword has to be repeated 'basically' $m+1$ times as in Claim 4, but now we can ignore the last $m-1$ letters of $u[1]\cdots u[m+1]$. Note that there is no assumption about the length of u; if u is 'short' and m is 'large', it can happen that the last $m-1$ letters actually 'subsume' several trailing copies of u.

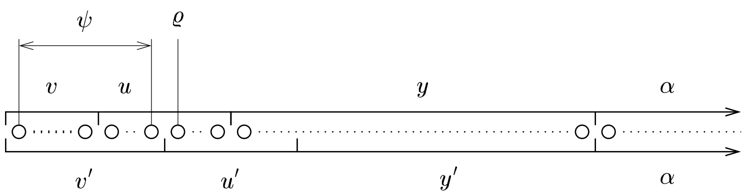
Claim 5 can also be proven by induction on m. Again, we concentrate just on the crucial step when $\varphi = \psi \mathsf{U} \varrho$ and $\psi, \varrho \in \mathrm{LTL}(\mathsf{U}^{m-1}, \mathsf{X}^0)$. We only show the '$\Longrightarrow$' part (the other direction is similar). So, let $vuy\alpha \models \psi \mathsf{U} \varrho$. Then there is $j \in \mathbb{N}_0$ such that $(vuy\alpha)_j \models \varrho$ and $(vuy\alpha)_i \models \psi$ for all $0 \le i < j$. We distinguish three possibilities (the first two of them are handled in the same way as in Claim 4):

(i) $j < |v|$. To prove that $vy\alpha \models \psi \mathsf{U} \varrho$, it suffices to show that for every suffix v' of v we have that
   - $v'uy\alpha \models \psi$ implies $v'y\alpha \models \psi$,
   - $v'uy\alpha \models \varrho$ implies $v'y\alpha \models \varrho$.

   However, this follows directly from induction hypothesis.

(ii) $j \ge |vu|$. First, realize that the (in)validity of ψ and ϱ for any suffix of $y\alpha$ is not influenced by deleting the u subword. Hence, it suffices to show that $v'uy\alpha \models \psi$ implies $v'y\alpha \models \psi$ for each suffix v' of v. This follows from the induction hypothesis in the same way as in (i).

(iii) $|v| \le j < |vu|$. This requires more care. A key observation is that the word $vuy\alpha$ can be seen as $v'u'y'\alpha = vuy\alpha$, where $|v'| = j$, $|u'| = |u|$, and $|y'| = |y| + |v| - |v'|$.



Due to the periodicity of y we have that $vy\alpha = v'y'\alpha$. Hence, it suffices to show that $y'\alpha \models \varrho$ and $v''y'\alpha \models \psi$ for every nonempty suffix v'' of v'. We know that $u'y'\alpha \models \varrho$ and $v''u'y'\alpha \models \psi$; so, if y' is 'sufficiently long', we can use induction hypothesis to finish the proof. That is, we need to verify that $|y'| \ge |u'| \cdot (m-1) - (m-1) + 1$, but this follows immediately from the known (in)equalities $|y'| = |y| + |v| - |v'|$, $|u'| = |u|$, and $|v| > |v'| - |u|$.

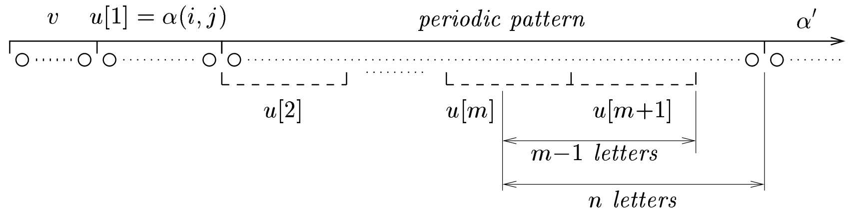
2.3 General stuttering

In this section we combine the previously discussed principles of letter stuttering and subword stuttering into a single 'general stuttering theorem' which is applicable to LTL($\mathsf{U}^m, \mathsf{X}^n$) formulae.

Definition 6 Let Σ be an alphabet and $m, n \in \mathbb{N}_0$.

- A subword $\alpha(i, j)$ of a given $\alpha \in \Sigma^\omega$ is (m, n)-*redundant* if the word $\alpha(i + j, m \cdot j - m + 1 + n)$ is a prefix of $\alpha(i, j)^\omega$.
- The relation $\succ_{m,n} \subseteq \Sigma^\omega \times \Sigma^\omega$ is defined as follows: $\alpha \succ_{m,n} \beta$ iff β can be obtained from α by deleting some (possibly infinitely many) non-overlapping (m, n)-redundant subwords. The (m, n)-*stutter equivalence* is the least equivalence over Σ^ω subsuming the relation $\succ_{m,n}$.
- A language $L \subseteq \Sigma^\omega$ is (m, n)-*stutter closed* if it is closed under (m, n)-stutter equivalence.

The structure of an ω-word α with an (m, n)-redundant subword $\alpha(i, j)$ can be illustrated as follows:



Hence, the $\alpha(i, j)$ subword has to be repeated 'basically' $m + 1$ times but we can ignore the last $(m - 1) - n$ letters (if $(m - 1) - n$ is negative, we must actually prolong the repetition 'beyond' the $m + 1$ copies of $\alpha(i, j)$—see the figure above). Note that there is no assumption about the size of m, n, and j.

Our goal is to prove that the (in)validity of LTL($\mathsf{U}^m, \mathsf{X}^n$) formulae is not influenced by deleting/pumping (m, n)-redundant subwords. First, let us realize that this result is a proper generalization of both Theorem 2 and Claim 5. If we compare the 'periodicity assumptions' of Theorem 2, Claim 5, and Definition 6, we can observe that

- a letter $\alpha(i)$ is n-redundant iff it is consecutively repeated at least $n + 1$ times. That is, $\alpha(i)$ is n-redundant iff $\alpha(i+1, n+1)$ is a prefix of $\alpha(i, 1)^\omega$. For every $m \in \mathbb{N}_0$ we get that $\alpha(i)$ is n-redundant iff $\alpha(i, 1)$ is (m, n)-redundant as $\alpha(i+1, n+1) = \alpha(i+1, m \cdot 1 - m + 1 + n)$. In other words, the notion of n-redundancy coincides with (m, n)-redundancy for subwords of length 1.
- the condition of Claim 5 matches exactly the definition of $(m, 0)$-redundancy.

Before formulating and proving the general stuttering theorem, we need to state two auxiliary lemmas.

Lemma 7 *Let Σ be an alphabet, $m, n \in \mathbb{N}_0$, and $\alpha \in \Sigma^\omega$. If a subword $\alpha(i, j)$ is*

(*i*) (m,n)*-redundant then it is also* (m',n')*-redundant for all* $0 \le m' \le m$ *and* $0 \le n' \le n$.
(*ii*) $(m,n+1)$*-redundant then* $\alpha(i+1,j)$ *is* (m,n)*-redundant.*
(*iii*) $(m+1,n)$*-redundant then* $\alpha(i+k,j)$ *is* (m,n)*-redundant for every* k *satisfying* $0 \le k < j$.

Proof (*i*) follows immediately as $j > 0$ implies

$$m' \cdot j - m' + 1 + n' \le m \cdot j - m + 1 + n$$

(*ii*) is also simple—due to the $(m,n{+}1)$-redundancy of $\alpha(i,j)$ we know that the subword is repeated at least on the next $m \cdot j - m + 2 + n$ letters. Hence, the subword $\alpha(i{+}1,j)$ is repeated at least on the next $m \cdot j - m + 1 + n$ letters and thus it is (m,n)-redundant. A proof of (*iii*) is similar; if $\alpha(i,j)$ is repeated on the next $(m{+}1) \cdot j - m + n$ letters, then the subword $\alpha(i{+}k,j)$ (where $0 \le k < j$) is repeated on the next $(m{+}1) \cdot j - m + n - k = m \cdot j - m + n + j - k$ letters, i.e., $\alpha(i{+}k,j)$ is $(m,n+j-k-1)$-redundant. The (m,n)-redundancy of $\alpha(i{+}k,j)$ follows from (*i*) and $k < j$. □

Lemma 8 *For all* $m \ge 1$, $n \ge 0$, *and all* $\alpha, \beta \in \Sigma^\omega$ *such that* $\alpha \succ_{m,n} \beta$ *there exists a surjective function* $g : \mathbb{N}_0 \longrightarrow \mathbb{N}_0$ *such that*

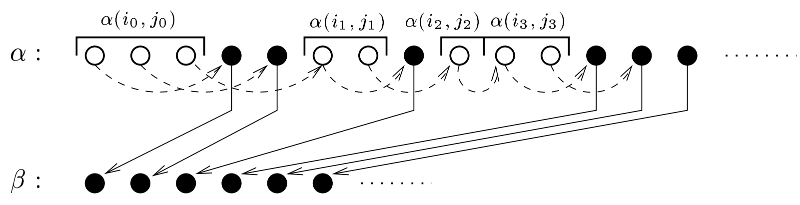
(*i*) *for all* $\ell, x \in \mathbb{N}_0$, *where* $0 \le x < g(\ell)$, *there exists* $0 \le \ell' < \ell$ *such that* $g(\ell') = x$,
(*ii*) *for each* $\ell \in \mathbb{N}_0$ *we have that* $\alpha_\ell \succ_{m-1,n} \beta_{g(\ell)}$.

Proof Let $m \ge 1$, $n \ge 0$ and $\alpha, \beta \in \Sigma^\omega$ such that $\alpha \succ_{m,n} \beta$. Let $D = \alpha(i_0,j_0), \alpha(i_1,j_1), \ldots$ be the (finite or infinite) sequence of non-overlapping (m,n)-redundant subwords which were deleted from α to obtain β (we assume that $i_0 < i_1 < \cdots$). We say that a given $\ell \in \mathbb{N}_0$ is *covered* by a subword $\alpha(i_q,j_q)$ of D if $i_q \le \ell \le i_q{+}j_q{-}1$. For each such ℓ we further define $jump(\ell) = \ell + j_q$ and $pos(\ell) = \ell - i_q + 1$. If ℓ is not covered by any subword of D, we put $pos(\ell) = 0$ and $jump(\ell) = \ell$. The set of all ℓ's that are covered by the subwords of D is denoted $cov(D)$. For each $\ell \notin cov(D)$, the symbol $length(\ell)$ denotes the total length of all subwords of D which cover some $k \le \ell$.

The function g is defined as follows:

$$g(\ell) = \begin{cases} \ell - length(\ell) & \text{if } \ell \notin cov(D), \\ g(jump(\ell)) & \text{otherwise.} \end{cases}$$

The structure of g can be illustrated as follows:



In particular, note that uncovered letters of α are projected to the "same" letters in β, and covered letters are in fact mapped to uncovered ones by performing one or more *jumps* of possibly different length. Also note that g is not monotonic in general.

First we show that g is well-defined, i.e., for each $\ell \in cov(D)$ there is $k \in \mathbb{N}$ such that $jump^k(\ell) \notin cov(D)$ (here $jump^k$ denotes $jump$ applied k-times). This is an immediate consequence of the following observation:

For each $\ell \in cov(D)$ there is $k \in \mathbb{N}$ such that $pos(jump^k(\ell)) < pos(\ell)$.

Proof of the observation: First, let us realize that $pos(\ell) \geq pos(jump(\ell))$ for every $\ell \in cov(D)$. Now assume that the observation does not hold. Then there is $\ell \in cov(D)$ such that $pos(jump^k(\ell)) = pos(\ell)$ for every $k \in \mathbb{N}$. Let $\alpha(i_q, j_q)$ be the subword of D covering ℓ, and let D_q be the sequence obtained from D by removing the first q elements. Since $pos(jump^k(\ell)) = pos(\ell)$ for every $k \in \mathbb{N}$, all subwords of D_q are adjacent and the length of each of them is at least $pos(\ell)$. Hence, each $\ell' \geq \ell$ is covered by some subword of D_q, which contradicts the assumption that β is infinite.

Proof of (i): First we show that for every $\ell \in \mathbb{N}_0$ we have that $g(\ell+1) \leq g(\ell)+1$. Let us assume that there is some $\ell' \in \mathbb{N}_0$ such that $g(\ell'+1) > g(\ell')+1$, and let $k \in \mathbb{N}_0$ be the least number such that $\ell = jump^k(\ell')$ is either uncovered or satisfies $g(jump(\ell)+1) \leq g(jump(\ell))+1$. Observe that such a k must exist, and that ℓ satisfies $g(\ell+1) > g(\ell)+1$ (otherwise we get a contradiction with the minimality of k). Now we distinguish two possibilities:

- $pos(\ell+1) \leq 1$. Let ℓ'' be the least uncovered index greater or equal to $\ell+1$. It follows easily from the definition of g that $g(\ell+1) = g(\ell'')$. Hence, $g(\ell)$ is either equal to $g(\ell+1)-1$ (if $\ell \notin cov(D)$), or greater or equal to $g(\ell+1)$ (if $\ell \in cov(D)$). Again, this contradicts the assumption that $g(\ell+1) > g(\ell)+1$.
- $pos(\ell{+}1) \geq 2$. Then $\ell, \ell{+}1$ are covered by the same subword of D. By applying the definition of g we obtain $g(\ell) = g(jump(\ell))$ and $g(\ell{+}1) = g(jump(\ell{+}1))$. Moreover, $jump(\ell{+}1) = jump(\ell)+1$ because $\ell, \ell{+}1$ are covered by the same subword of D. If $pos(jump(\ell){+}1)$ is equal to 0 or 1, we derive a contradiction using the arguments of previous cases. If $pos(jump(\ell){+}1) \geq 2$, we have that $jump(\ell) \in cov(D)$, hence $g(jump(\ell){+}1) \leq g(jump(\ell)){+}1$ due to the assumption adopted above. Altogether, we derived a contradiction with $g(\ell{+}1) > g(\ell){+}1$.

Now we are ready to finish the proof of (i). Let us assume that (i) does not hold, and let $\ell \in \mathbb{N}_0$ be the least number such that (i) is violated for ℓ and some $0 \leq x < g(\ell)$. Clearly $\ell > 0$, because $g(0) = 0$. Further, $g(\ell-1) \geq g(\ell)-1$ due to the claim just proved. This means that either $g(\ell-1) = x$, or $\ell - 1$ also violates (i). In both cases we have a contradiction with our choice of ℓ.

Proof of (ii): We show that $\alpha_\ell \succ_{m-1,n} \beta_{g(\ell)}$ for each $\ell \in \mathbb{N}_0$. We proceed by induction on $pos(\ell)$.

Basis. $pos(\ell) = 0$. This means that $\ell \notin cov(D)$. Clearly $\alpha_\ell \succ_{m,n} \beta_{g(\ell)}$ because $\beta_{g(\ell)}$ is obtained from α_ℓ by deleting all those subwords $\alpha(i_q, j_q)$ of D such that $i_q > \ell$. Hence, we also have $\alpha_\ell \succ_{m-1,n} \beta_{g(\ell)}$ by applying Lemma 7 (i).

Induction step. Let $pos(\ell) > 0$ and let $k \in \mathbb{N}$ be the least number such that $pos(jump^k(\ell)) < pos(\ell)$. To simplify our notation, we put $\ell' = jump^k(\ell)$. Clearly $g(\ell) = g(\ell')$ by definition of g. By induction hypothesis we have that $\alpha_{\ell'} \succ_{m-1,n} \beta_{g(\ell')}$. Hence, it suffices to show that $\alpha(\ell, \ell' - \ell)$ is a sequence of $(m-1, n)$-redundant subwords. Let us assume that ℓ is covered by $\alpha(i_q, j_q)$. Consider the sequence of subwords

$$\alpha(i_q, j_q), \ldots, \alpha(i_{q+k-1}, j_{q+k-1})$$

From the minimality of k we obtain that these subwords are adjacent and the length of each of them is at least $pos(\ell)$. Hence, $\alpha(\ell, \ell' - \ell)$ can be seen as a sequence of words

$$\alpha(i_q + pos(\ell) - 1, j_q), \ldots, \alpha(i_{q+k-1} + pos(\ell) - 1, j_{q+k-1})$$

Moreover, each of these words is $(m-1, n)$-redundant by Lemma 7 (iii). □

Theorem 9 (general stuttering) *Every* $\mathrm{LTL}(\mathsf{U}^m, \mathsf{X}^n)$ *language is closed under* (m, n)*-stutter equivalence.*

Proof Let $m, n \in \mathbb{N}_0$ and $\varphi \in \mathrm{LTL}(\mathsf{U}^m, \mathsf{X}^n)$. It suffices to prove that for all $\alpha, \beta \in \Sigma^\omega$ such that $\alpha \succ_{m,n} \beta$ we have that $\alpha \models \varphi \iff \beta \models \varphi$. We proceed by a simultaneous induction on m and n (we write $(m', n') < (m, n)$ iff $m' \leq m$ and $n' < n$, or $m' < m$ and $n' \leq n$).

Basis. $m = 0$ and $n = 0$. Let $\alpha, \beta \in \Sigma^\omega$ be ω-words such that $\alpha \succ_{0,0} \beta$. Let D denote the sequence of non-overlapping $(0, 0)$-redundant subwords $D = \alpha(i_0, j_0), \alpha(i_1, j_1), \ldots$ which were deleted from α to obtain β (we assume that $i_0 < i_1 < \ldots$). Since $\mathrm{LTL}(\mathsf{U}^0, \mathsf{X}^0)$ formulae are just 'Boolean combinations' of letters and $\mathtt{tt}$, it suffices to show that $\alpha(0) = \beta(0)$. If $i_0 > 0$ then it is clearly the case. Now let $i_0 = 0$, and let $k \in \mathbb{N}_0$ be the least number such that the subwords $\alpha(i_k, j_k)$ and $\alpha(i_{k+1}, j_{k+1})$ are not adjacent (i.e., $i_{k+1} > i_k + j_k$). Hence, $\beta(0) = \alpha(i_k + j_k)$ and $(0, 0)$-redundancy of the subwords in D implies that

$$\alpha(0) = \alpha(i_0) = \alpha(i_1) = \alpha(i_2) = \ldots = \alpha(i_k) = \alpha(i_k + j_k) = \beta(0).$$

Induction step. Let $m, n \in \mathbb{N}_0$, and let us assume that the theorem holds for all m', n' such that $(m', n') < (m, n)$. Let $\alpha, \beta \in \Sigma^\omega$ be ω-words such that $\alpha \succ_{m,n} \beta$, and let $D = \alpha(i_0, j_0), \alpha(i_1, j_1), \ldots$ $(i_0 < i_1 < \ldots)$ be the sequence of non-overlapping (m, n)-redundant subwords which were deleted from α to obtain β. We distinguish four possibilities:

- $\varphi \in \mathrm{LTL}(\mathsf{U}^{m'}, \mathsf{X}^{n'})$ for some $(m', n') < (m, n)$. Since every $\alpha(i, j)$ from D is (m', n')-redundant by Lemma 7 (i), we just apply induction hypothesis.
- $\varphi = \mathsf{X}\psi$. We need to prove that $\alpha_1 \models \psi \iff \beta_1 \models \psi$. By induction hypothesis, ψ cannot distinguish between $(m, n-1)$-stutter equivalent ω-words. Hence, it suffices to show that $\alpha_1 \succ_{m,n-1} \beta_1$. If $i_0 > 0$, then $\alpha_1(i_0 - 1, j_0), \alpha_1(i_1 - 1, j_1), \alpha_1(i_2 - 1, j_2), \ldots$ are (m, n)-redundant and due to Lemma 7 (i) they are also $(m, n-1)$-redundant. Moreover, β_1 can be obtained from α_1 by deleting these subwords.

If $i_0 = 0$, then let $k \in \mathbb{N}_0$ be the least number such that the subwords $\alpha(i_k, j_k)$ and $\alpha(i_{k+1}, j_{k+1})$ are not adjacent. The ω-word β_1 can be obtained from α_1 by deleting the subwords

$$\alpha_1(i_0, j_0), \ldots, \alpha_1(i_k, j_k), \alpha_1(i_{k+1}-1, j_{k+1}), \alpha_1(i_{k+2}-1, j_{k+2}), \ldots$$

The subwords $\alpha_1(i_0, j_0), \alpha_1(i_1, j_1), \ldots, \alpha_1(i_k, j_k)$ are $(m, n-1)$-redundant by Lemma 7 (ii), and the other subwords are $(m, n-1)$-redundant by applying Lemma 7 (i).

- $\varphi = \psi \mathsf{U} \varrho$. By induction hypothesis, ψ, ϱ cannot distinguish between $(m-1, n)$-stutter equivalent ω-words. Let g be the function of Lemma 8 constructed for the considered m, n, α, β (i.e., $\alpha_\ell \succ_{m-1,n} \beta_{g(\ell)}$ for every $\ell \in \mathbb{N}_0$).
  Now we show that if $\alpha \models \psi \mathsf{U} \varrho$ then also $\beta \models \psi \mathsf{U} \varrho$. If $\alpha \models \psi \mathsf{U} \varrho$, there is $c \geq 0$ such that $\alpha_c \models \varrho$ and for every $d < c$ we have that $\alpha_d \models \psi$. By induction hypothesis we get $\beta_{g(c)} \models \varrho$. Further, for every $d' < g(c)$ there is $d < c$ such that $g(d) = d'$. By Lemma 8, for every $d' < g(c)$ there is $d < c$ such that $\alpha_d \succ_{m-1,n} \beta_{g(d)} = \beta_{d'}$ and hence $\beta_{d'} \models \psi$. Altogether, we obtain that $\beta \models \psi \mathsf{U} \varrho$.
  Similarly, we also show that if $\beta \models \psi \mathsf{U} \varrho$ then $\alpha \models \psi \mathsf{U} \varrho$. If $\beta \models \psi \mathsf{U} \varrho$, there is $c \geq 0$ such that $\beta_c \models \varrho$ and for every $d < c$ we have that $\beta_d \models \psi$. Let c' be the least number satisfying $g(c') = c$ (there is such a c' because g is surjective). Then $\alpha_{c'} \models \varrho$ by induction hypothesis. From the definition of g we get that for every $d' < c'$ it holds that $g(d') < g(c') = c$ (otherwise we would obtain a contradiction with our choice of c'). Thus, $\alpha_{d'} \models \psi$ and hence $\alpha \models \psi \mathsf{U} \varrho$.
- φ is a 'Boolean combination' of formulae of the previous cases. Formally, this case is handled by an 'embedded' induction on the structure of φ. The basic step (when φ is *not* of the form $\neg\psi$ or $\psi \wedge \varrho$) is covered by the previous cases. The induction step ($\varphi = \neg\psi$ or $\varphi = \psi \wedge \varrho$ where we assume that our theorem holds for ψ, ϱ) follows immediately. $\square$

3 Stuttering as a Sufficient Condition

In Section 2 we have shown that formulae of certain LTL fragments are invariant under certain forms of stutter equivalence of ω-words. These results (Theorem 2, Claim 4, Claim 5, and Theorem 9) were formulated as "pumping lemmas", i.e., necessary conditions which must be satisfied by languages of the respective LTL fragments. In this section we show that certain forms of stutter invariance together with some additional assumptions in fact *characterize* certain LTL fragments.

3.1 Letter stuttering

It has been proved by Peled and Wilke [9] that every LTL language closed under stuttering is definable in $\mathrm{LTL}(\mathsf{U}, \mathsf{X}^0)$. This proof can be straightforwardly generalized to n-stuttering. Hence, every n-stutter closed LTL property is definable in $\mathrm{LTL}(\mathsf{U}, \mathsf{X}^n)$. For the sake of completeness, we present this proof explicitly. (Later we formulate further observations which refer to technical details of this proof.)

Theorem 10 *Let $L \subseteq \Sigma^\omega$. The following conditions are equivalent:*

(a) L is definable in $\mathrm{LTL}(\mathsf{U}, \mathsf{X}^n)$.
(b) L is an n-stutter closed LTL language.

Proof The $(a) \Longrightarrow (b)$ direction follows from Theorem 2. We prove the other direction. Let φ be an LTL formula such that L_φ is n-stutter closed. We translate φ into an equivalent formula $\tau_n(\varphi) \in \mathrm{LTL}(\mathsf{U}, \mathsf{X}^n)$.

Let Θ be the set of letters occurring in φ, and let $\theta = \bigvee_{p\in\Theta} p$. For all $p \in \Theta$ and $i > 0$ we define formulae σ_{p^i}, $\sigma_{p^i\neg p}$, $\sigma_{\neg\Theta^i}$, and $\sigma_{\neg\Theta^i\Theta}$ as follows:

$$
\begin{array}{ll}
\sigma_{p^1} = p & \sigma_{p^{i+1}} = p \wedge \mathsf{X}\sigma_{p^i} \\
\sigma_{p^0\neg p} = \neg p & \sigma_{p^i\neg p} = p \wedge \mathsf{X}\sigma_{p^{i-1}\neg p} \\
\sigma_{\neg\Theta^1} = \neg\theta & \sigma_{\neg\Theta^{i+1}} = \neg\theta \wedge \mathsf{X}\sigma_{\neg\Theta^i} \\
\sigma_{\neg\Theta^0\Theta} = \theta & \sigma_{\neg\Theta^{i+1}\Theta} = \neg\theta \wedge \mathsf{X}\sigma_{\neg\Theta^i\Theta}
\end{array}
$$

Observe that $X(\sigma_{p^{i+1}}) = X(\sigma_{p^i\neg p}) = X(\sigma_{\neg\Theta^{i+1}}) = X(\sigma_{\neg\Theta^i\Theta}) = i$.

The translation $\tau_n(\varphi)$ is defined inductively on the structure of φ.

- $\tau_n(p) = p$
- $\tau_n(\neg\psi) = \neg\tau_n(\psi)$
- $\tau_n(\psi \wedge \varrho) = \tau_n(\psi) \wedge \tau_n(\varrho)$
- $\tau_n(\psi \,\mathsf{U}\, \varrho) = \tau_n(\psi) \,\mathsf{U}\, \tau_n(\varrho)$
- $\tau_n(\mathsf{X}\psi) = \Phi(\psi) \vee \Gamma(\psi)$ where

$$\Phi(\psi) = (\mathsf{G}\neg\theta \vee \bigvee_{p\in\Theta} \mathsf{G}p) \wedge \tau_n(\psi)$$

and

$$\Gamma(\psi) = \bigvee_{1\le i\le n+1} (\,\xi(\psi, \neg\Theta, i) \vee \bigvee_{p\in\Theta} \xi(\psi, p, i)\,).$$

The subformulae $\xi(\psi, \neg\Theta, i)$ and $\xi(\psi, p, i)$ of $\Gamma(\psi)$ are constructed as follows:

$$\xi(\psi, p, i) = \begin{cases} \sigma_{p^i\neg p} \ \wedge \ p \,\mathsf{U}\, (\sigma_{p^{i-1}\neg p} \wedge \tau_n(\psi)) & \text{if } i \le n \\ \sigma_{p^{n+1}} \ \wedge \ p \,\mathsf{U}\, (\sigma_{p^n\neg p} \wedge \tau_n(\psi)) & \text{if } i = n{+}1 \end{cases}$$

$$\xi(\psi, \neg\Theta, i) = \begin{cases} \sigma_{\neg\Theta^i\Theta} \ \wedge \ \neg\theta \,\mathsf{U}\, (\sigma_{\neg\Theta^{i-1}\Theta} \wedge \tau_n(\psi)) & \text{if } i \le n \\ \sigma_{\neg\Theta^{n+1}} \ \wedge \ \neg\theta \,\mathsf{U}\, (\sigma_{\neg\Theta^n\Theta} \wedge \tau_n(\psi)) & \text{if } i = n{+}1 \end{cases}$$

One can readily confirm that the X-depth of $\tau_n(\varphi)$ is n. We need to prove that if L_φ^Σ is n-stutter closed, then φ is equivalent to $\tau_n(\varphi)$. Since φ and $\tau_n(\varphi)$ cannot distinguish between letters which do not belong to Θ, we can assume that $\Sigma \subseteq \Theta \cup \{o\}$, where $o \notin \Theta$ represents all letters not occurring in φ.

As both L_φ and $L_{\tau_n(\varphi)}$ are n-stutter closed (in the case of $L_{\tau_n(\varphi)}$ we apply Theorem 2), it actually suffices to prove that φ and $\tau_n(\varphi)$ cannot be distinguished by any *n-stutter free* ω-word $\alpha \in \Sigma^\omega$ (an ω-word α is n-stutter free if α has no n-redundant letters). That is, for every n-stutter free $\alpha \in \Sigma^\omega$ we show that $\alpha \models \varphi$ iff $\alpha \models \tau_n(\varphi)$. We proceed by induction on the structure of φ. All subcases except for $\varphi = \mathsf{X}\psi$ are trivial. Here we distinguish two possibilities:

- $\alpha = p^\omega$ for some $p \in \Sigma$. Then $\alpha_1 = \alpha$ and thus we get $\alpha \models \mathsf{X}\psi$ iff $\alpha_1 \models \psi$ iff $\alpha_1 \models \tau_n(\psi)$ (by induction hypothesis) iff $\alpha \models \tau_n(\psi)$. Hence, this subcase is 'covered' by the formula $\Phi(\psi)$ saying that α is of the form p^ω and that $\tau_n(\psi)$ holds (the particular case when $\alpha = o^\omega$ corresponds to $\mathsf{G}\neg\theta$).
- $\alpha = p^i q\beta$ where $p, q \in \Sigma$, $p \neq q$, $1 \leq i \leq n+1$, and $\beta \in \Sigma^\omega$.
  Let us first consider the case when $p = o$. Then $p^i q\beta \models \mathsf{X}\psi$ iff $p^{i-1}q\beta \models \psi$ iff $p^{i-1}q\beta \models \tau_n(\psi)$ (we use induction hypothesis). If $i \leq n$, then the last condition is equivalent to

  $$p^i q\beta \models \sigma_{\neg\Theta^i\Theta} \wedge \neg\theta \,\mathsf{U}\, (\sigma_{\neg\Theta^{i-1}\Theta} \wedge \tau_n(\psi))$$

  If $i = n{+}1$, then the condition is equivalent to

  $$p^{n+1}q\beta \models \sigma_{\neg\Theta^{n+1}} \wedge \neg\theta \,\mathsf{U}\, (\sigma_{\neg\Theta^n\Theta} \wedge \tau_n(\psi))$$

  In both cases, the resulting formula corresponds to $\xi(\psi, \neg\Theta, i)$.
  The case when $p \in \Theta$ is handled similarly; we have that $p^i q\beta \models \mathsf{X}\psi$ iff $p^{i-1}q\beta \models \psi$ iff $p^{i-1}q\beta \models \tau_n(\psi)$ (by induction hypothesis). If $i \leq n$ then the last condition is equivalent to

  $$p^i q\beta \models \sigma_{p^i\neg p} \wedge p \,\mathsf{U}\, (\sigma_{p^{i-1}\neg p} \wedge \tau_n(\psi))$$

  If $i = n{+}1$ then the condition is equivalent to

  $$p^{n+1}q\beta \models \sigma_{p^{n+1}} \;\wedge\; p \,\mathsf{U}\, (\sigma_{p^n\neg p} \wedge \tau_n(\psi))$$

  In both cases, the resulting formula corresponds to $\xi(\psi, p, i)$.
  To sum up, the case when $\alpha = p^i q\beta$ is 'covered' by the formula $\Gamma(\psi)$. $\square$

In general, the size of $\tau_n(\varphi)$ is exponential in $X(\varphi)$. However, the size of the *circuit*[3] representing $\tau_n(\varphi)$ is only $\mathcal{O}((n+1)\cdot|\varphi|^2)$. To see this, realize the following:

(1) The total size of all circuits representing the formulae $\sigma_{p^n\neg p}, \sigma_{p^{n+1}}$ (for all $p \in \Theta$) and $\sigma_{\neg\Theta^n\Theta}, \sigma_{\neg\Theta^{n+1}}$ is $\mathcal{O}((n{+}1)\cdot|\varphi|)$. Moreover, all circuits representing the formulae $\sigma_{p^i\neg p}$ and $\sigma_{\neg\Theta^i\Theta}$ (for all $0 \leq i \leq n$) are contained in the circuits representing $\sigma_{p^n\neg p}$ or $\sigma_{\neg\Theta^n\Theta}$, respectively.
(2) Assuming that the circuits of (1) and the circuit representing $\tau_n(\psi)$ are at our disposal, we only need to add a constant number of new nodes to represent the formulae $\xi(\psi, \neg\Theta, i)$ and $\xi(\psi, p, i)$ for given $p \in \Theta$ and $1 \leq i \leq n{+}1$. This means that we need to add $\mathcal{O}((n+1)\cdot|\varphi|)$ new nodes when constructing the circuit for $\tau_n(\mathsf{X}\psi)$.
(3) Since φ contains $\mathcal{O}(|\varphi|)$ subformulae of the form $\mathsf{X}\psi$, the circuit representing φ has $\mathcal{O}((n+1)\cdot|\varphi|^2)$ nodes in total.

[3] A circuit (or DAG) representing a given LTL formula φ is obtained from the syntax tree of φ by identifying all nodes which correspond to the same subformula.

Theorem 11 *Let φ be an LTL formula and $n \in \mathbb{N}_0$. The problem whether there is a formula $\psi \in \mathrm{LTL}(\mathsf{U}, \mathsf{X}^n)$ equivalent to φ is* PSPACE*-complete (assuming unary encoding of n).*

Proof It suffices to show that the problem whether a given LTL formula φ defines an n-stutter closed language is PSPACE-complete. The proof for $n = 0$ has been presented in [10].

Similarly as in [10], the PSPACE-lower bound is obtained by reducing the validity problem for LTL formulae, which is known to be PSPACE-complete [13]. For every LTL formula ϱ we define a formula

$$\pi(\varrho) = p \wedge \mathsf{X}p \wedge \mathsf{XX}p \wedge \ldots \wedge \overbrace{\mathsf{XX}\ldots\mathsf{X}}^{n}(p \wedge \mathsf{X}q \wedge \mathsf{XX}\neg\varrho).$$

The language $L_{\pi(\varrho)} = p^{n+1}qL_{\neg\varrho}$ is n-stutter closed iff $L_{\neg\varrho}$ is empty. That is, $L_{\pi(\varrho)}$ is n-stutter closed iff ϱ is valid.

The matching PSPACE-upper bound is obtained by applying a similar argument as in [2]—due to the (proof of) Theorem 10 we have that L_φ is n-stutter closed iff φ is equivalent to $\tau_n(\varphi)$. First, we construct the circuit representing $\tau_n(\varphi)$ (its size is $\mathcal{O}((n+1) \cdot |\varphi|^2)$ as shown above). Then we check the validity of the formula $\varphi \Leftrightarrow \tau_n(\varphi)$ (represented as a circuit), which can be also done in polynomial space [13]. □

Finally, let us note that the condition (b) of Theorem 10 cannot be weakened to "L is an n-stutter closed ω-regular language", because there are ω-regular languages which are n-stutter closed for all $n \in \mathbb{N}_0$, yet not definable in LTL. A concrete example of such a language is $L = \{(p^+q^+)^{2i}r^\omega \mid i \in \mathbb{N}\}$ which is clearly n-stutter closed for every $n \in \mathbb{N}_0$, but not (m,n)-stutter closed for any $m,n \in \mathbb{N}_0$ (and hence not definable in LTL).

3.2 General stuttering

In Section 3.1 we have shown that $\mathrm{LTL}(\mathsf{U}, \mathsf{X}^n)$ languages are exactly n-stutter closed LTL languages. A natural question is whether $\mathrm{LTL}(\mathsf{U}^m, \mathsf{X}^n)$ languages are fully characterized by the closure property induced by (m,n)-stuttering. In this section we show that this is *not* the case. Nevertheless, regular (m,n)-stutter closed languages are inevitably *noncounting*, and hence expressible in LTL. This means that if L is ω-regular and (m,n)-stutter closed, then $L \in \mathrm{LTL}(\mathsf{U}^{m'}, \mathsf{X}^{n'})$ for some m', n'. In this section we also show that there is no functional relationship between (m', n') and (m,n).

Definition 12 A language $L \subseteq \Sigma^\omega$ is *noncounting* if there is $k \in \mathbb{N}_0$ such that for all $n \geq k$ and $x, y, z, u \in \Sigma^*$ we have the following:

- $xu^nyz^\omega \in L \iff xu^{n+1}yz^\omega \in L$,
- $x(yu^nz)^\omega \in L \iff x(yu^{n+1}z)^\omega \in L$.

Theorem 13 *Let $L \subseteq \Sigma^\omega$. The following conditions are equivalent:*

(a) L is definable in LTL,

(b) L is ω-regular and noncounting,
(c) L is ω-regular and (m,n)-stutter closed for some $m, n \in \mathbb{N}_0$.

Proof The equivalence of (a) and (b) is a consequence of several results; Kamp [5] proved that languages (of infinite words) definable in LTL are exactly the languages expressible in first-order logic. Using the results presented in [14] and [1], Perrin [11] showed that a language is definable in first-order logic iff it is ω-regular and noncounting.

The implication (a) $\Longrightarrow$ (c) is given by Theorem 9. The implication (c) $\Longrightarrow$ (b) follows from a straightforward observation that a language violating noncounting property is not (m,n)-stutter closed for any $m, n \in \mathbb{N}_0$. $\square$

A natural question is whether the condition (c) of Theorem 13 can be weakened to "L is (m,n)-stutter closed for some $m, n \in \mathbb{N}_0$". The answer is given in our next theorem.

Theorem 14 *For all $m \geq 2$ and $n \geq 1$ there is an (m,n)-stutter closed language $L \subseteq \{o,p,q,r\}^\omega$ which is not definable in LTL.*

Proof Due to Lemma 7 (i), we just need to consider the case when $m = 2$ and $n = 1$. We say that a word $w \in \Sigma^*$ is *square-free* if it does not contain a subword of the form uu, where $|u| \geq 1$. It is known that there are infinitely many square-free words[4] $w_0, w_1, \ldots$ over the alphabet $\{o,p,q\}$ [16]. Now observe that for each of these w_i there is no other word $v \in \{o,p,q\}^*$ such that $w_i r^\omega \succ_{(2,1)} v r^\omega$ or $v r^\omega \succ_{(2,1)} w_i r^\omega$. This means that $L = \{w_i r^\omega \mid i \in \mathbb{N}_0\}$ is $(2,1)$-stutter closed. Obviously, L is not ω-regular by using standard arguments (pumping lemma for ω-regular languages). Thus, L is not definable in LTL. $\square$

Due to Theorem 13, we know that if L is ω-regular and (m,n)-stutter closed, then L is definable in LTL, i.e., there are $m', n' \in \mathbb{N}$ such that L is definable in LTL($\mathsf{U}^{m'}, \mathsf{X}^{n'}$). However, it is not clear what is the relationship between m, n and m', n'. One might be tempted to think that m', n' can be expressed (or at least bounded) by some simple functions in m, n, for example $m' = m$ and $n' = n$. Our next theorem says that there is no such relationship.

Theorem 15 *Let $m \geq 2$ and $n \geq 1$. For all $m', n' \in \mathbb{N}_0$ there is an (m,n)-stutter closed LTL language $L \subseteq \{o,p,q,r\}^\omega$ which is not definable in* LTL($\mathsf{U}^{m'}, \mathsf{X}^{n'}$).

Proof First, realize that for all $m', n' \in \mathbb{N}_0$ there are only finitely many pairwise non-equivalent LTL($\mathsf{U}^{m'}, \mathsf{X}^{n'}$) formulae over the alphabet $\{o,p,q,r\}$. Hence, it suffices to show that for all $m \geq 2$ and $n \geq 1$ there are infinitely many (m,n)-stutter closed LTL languages over the alphabet $\{o,p,q,r\}$. Due to Lemma 7 (i), we just need to consider the case when $m = 2$ and $n = 1$. Let L be the language constructed in the proof of Theorem 14. Now realize that each of the infinitely many finite subsets of L is a $(2,1)$-stutter closed LTL language. $\square$

[4] The sequence $w_0, w_1, \cdots$ is defined inductively by $w_0 = o$ and $w_{i+1} = f(w_i)$, where f is a word homomorphism given by $f(o) = opqop$, $f(p) = oqopqp$, $f(q) = oqpqoqp$. The proof in [16] reveals that if w is square-free, then so is $f(w)$.

Finally, let us note that possible generalizations of Theorem 14 and Theorem 15 cannot cross certain limits—they do not hold for all $m, n \in \mathbb{N}_0$ and every alphabet Σ. For example, every $(1,0)$-stutter closed language over the alphabet $\{p,q\}$ is definable in $\mathrm{LTL}(\mathsf{U}^2, \mathsf{X}^0)$. To see this, realize that the quotient of $\{p,q\}^\omega$ under $(1,0)$-stutter equivalence has exactly eight equivalence classes represented by words $(pq)^\omega$, $(qp)^\omega$, p^ω, q^ω, pq^ω, qp^ω, pqp^ω, and qpq^ω. Hence, there are exactly $2^8 = 256$ languages over $\{p,q\}$ which are $(1,0)$-stutter closed. Since each equivalence class of the quotient is a language definable in $\mathrm{LTL}(\mathsf{U}^2, \mathsf{X}^0)$, we can conclude that each of these 256 languages is definable in $\mathrm{LTL}(\mathsf{U}^2, \mathsf{X}^0)$.

4 Answers to Questions 1, 2, and 3

Now we are ready to provide answers to Questions 1, 2, and 3 which were stated in Section 1 (though Question 3 will be left open in fact). We start with a simple observation.

Lemma 16 *For each $n \geq 1$ there is a formula $\varphi \in \mathrm{LTL}(\mathsf{U}^0, \mathsf{X}^n)$ which cannot be expressed in* $\mathrm{LTL}(\mathsf{U}, \mathsf{X}^{n-1})$.

Proof Let $\Sigma = \{p,q\}$ and $n \geq 1$. Consider the formula

$$\varphi = \overbrace{\mathsf{X}\mathsf{X}\cdots\mathsf{X}}^{n} p.$$

We show that L_φ is not closed under $(n-1)$-stutter equivalence (which suffices due to Theorem 2). This is easy; realize that $p^{n+1}q^\omega \in L_\varphi$ and the first occurrence of p in this word is $(n-1)$-redundant. Since $p^n q^\omega \notin L_\varphi$, we are done. □

A 'dual' fact is proven below (this is already non-trivial).

Lemma 17 *For each $m \geq 1$ there is a formula $\varphi \in \mathrm{LTL}(\mathsf{U}^m, \mathsf{X}^0)$ which cannot be expressed in* $\mathrm{LTL}(\mathsf{U}^{m-1}, \mathsf{X})$.

Proof Let $m \geq 1$ and let $\Sigma = \{q, p_1, \ldots, p_m\}$. We define a formula $\varphi \in \mathrm{LTL}(\mathsf{U}^m, \mathsf{X}^0)$ as follows:

$$\varphi = \mathsf{F}(p_1 \wedge \mathsf{F}(p_2 \wedge \ldots \wedge \mathsf{F}(p_{m-1} \wedge \mathsf{F}p_m)\ldots))$$

Let us fix an arbitrary $n \in \mathbb{N}_0$, and define a word $\alpha \in \Sigma^\omega$ by

$$\alpha = (q^{n+1}\, p_m\, p_{m-1} \cdots p_1)^m\, q^\omega$$

Clearly $\alpha \models \varphi$ and the subword $\alpha(0, n{+}1{+}m)$ is $(m-1, n)$-redundant. As the word β obtained from α by removing $\alpha(0, n{+}1{+}m)$ does not model φ, the language L_φ is not $(m-1, n)$-stutter closed. As this holds for every $n \in \mathbb{N}_0$, the formula φ is not expressible in $\mathrm{LTL}(\mathsf{U}^{m-1}, \mathsf{X})$. □

The last technical lemma which is needed to formulate answers to Questions 1 and 2 follows.

Lemma 18 *For all* $m, n \in \mathbb{N}_0$ *there is a formula* $\varphi \in \mathrm{LTL}(\mathsf{U}^m, \mathsf{X}^n)$ *which is expressible neither in* $\mathrm{LTL}(\mathsf{U}^{m-1}, \mathsf{X}^n)$ *(assuming* $m \geq 1$*), nor in* $\mathrm{LTL}(\mathsf{U}^m, \mathsf{X}^{n-1})$ *(assuming* $n \geq 1$*).*

Proof If $m = 0$ or $n = 0$, we can apply Lemma 16 or Lemma 17, respectively. Now let $m, n \geq 1$, and let $\Sigma = \{p_1, \ldots, p_k, q\}$ where $k = \max\{m, n{+}1\}$. We define formulae ψ and φ as follows:

$$\psi = \begin{cases} p_m \wedge \mathsf{X}^n p_{m-n} & \text{if } m > n \\ p_m \wedge \mathsf{X}^n p_{m+1} & \text{if } m \leq n \end{cases}$$

$$\varphi = \begin{cases} \mathsf{F}\psi & \text{if } m = 1 \\ \mathsf{F}(p_1 \wedge \mathsf{F}(p_2 \wedge \mathsf{F}(p_3 \wedge \ldots \wedge \mathsf{F}(p_{m-1} \wedge \mathsf{F}\psi) \ldots))) & \text{if } m > 1 \end{cases}$$

where X^l abbreviates $\overbrace{\mathsf{X}\mathsf{X}\ldots\mathsf{X}}^{l}$. The formula φ belongs to $\mathrm{LTL}(\mathsf{U}^m, \mathsf{X}^n)$. Let us consider the ω-word α defined by

$$\alpha = \begin{cases} (p_m\, p_{m-1} \ldots p_1)^m p_m\, p_{m-1} \ldots p_{m-n+1} q^\omega & \text{if } m > n \\ (p_{n+1}\, p_n \ldots p_1)^{m+1} q^\omega & \text{if } m = n \\ (p_{n+1}\, p_n \ldots p_1)^{m+1} p_{n+1}\, p_n \ldots p_{m+2} q^\omega & \text{if } m < n \end{cases}$$

It is easy to check that $\alpha \in L_\varphi$ and that the subword $\alpha(0, k)$ (where $k = \max\{m, n{+}1\}$) is $(m, n{-}1)$-redundant as well as $(m{-}1, n)$-redundant. As the word β obtained from α by removing $\alpha(0, k)$ does not satisfy φ, the language L_φ is neither $(m, n{-}1)$-stutter closed, nor $(m{-}1, n)$-stutter closed. $\square$

The knowledge presented in the three lemmata above allows to conclude the following:

Corollary 19 (Answer to Question 1) *The* $\mathrm{LTL}(\mathsf{U}^m, \mathsf{X}^n)$, $\mathrm{LTL}(\mathsf{U}^m, \mathsf{X})$, *and* $\mathrm{LTL}(\mathsf{U}, \mathsf{X}^n)$ *hierarchies are strict.*

Corollary 20 (Answer to Question 2) *Let* A *and* B *be classes of* $\mathrm{LTL}(\mathsf{U}^m, \mathsf{X}^n)$, $\mathrm{LTL}(\mathsf{U}^m, \mathsf{X})$, *or* $\mathrm{LTL}(\mathsf{U}, \mathsf{X}^n)$ *hierarchy (not necessarily of the same one) such that* A *is syntactically not included in* B*. Then there is a formula* $\varphi \in A$ *which cannot be expressed in* B*.*

Although we cannot provide a full answer to Question 3, we can at least reject the aforementioned 'natural' hypotheses (see Section 1).

Lemma 21 (About Question 3) *For all* $m, n \in \mathbb{N}_0$ *there is a language definable in* $\mathrm{LTL}(\mathsf{U}^{m+2}, \mathsf{X}^n)$ *as well as in* $\mathrm{LTL}(\mathsf{U}^{m+1}, \mathsf{X}^{n+1})$ *which is not definable in* $\mathrm{LTL}(\mathsf{U}^{m+1}, \mathsf{X}^n)$.

Proof We start with the case when $m = n = 0$. Let $\Sigma \supseteq \{p, q\}$, and let $\psi_1 = \mathsf{F}(q \wedge (q \,\mathsf{U} \neg q))$ and $\psi_2 = \mathsf{F}(q \wedge \mathsf{X}\neg q)$. Note that $\psi_1 \in \mathrm{LTL}(\mathsf{U}^2, \mathsf{X}^0)$ and $\psi_2 \in \mathrm{LTL}(\mathsf{U}^1, \mathsf{X}^1)$. Moreover, ψ_1 and ψ_2 are equivalent as they define the same language $L = \Sigma^* q(\Sigma \smallsetminus \{q\})\Sigma^\omega$. This language is not definable in $\mathrm{LTL}(\mathsf{U}^1, \mathsf{X}^0)$ as it is not $(1, 0)$-stutter closed; for example, the ω-word $\alpha = pqpq^\omega \in L$ contains a $(1, 0)$-redundant subword $\alpha(0, 2)$ but $\alpha_2 = pq^\omega \notin L$.

The above example can be generalized to arbitrary m, n (using the designed formulae ψ_1, ψ_2). For given m, n we define formulae $\varphi_1 \in \mathrm{LTL}(\mathsf{U}^{m+2}, \mathsf{X}^n)$ and $\varphi_2 \in \mathrm{LTL}(\mathsf{U}^{m+1}, \mathsf{X}^{n+1})$, both defining the same language L over $\Sigma = \{q, p, p_1, \ldots, p_{m+1}\}$, and we give an example of an ω-word $\alpha \in L$ with an $(m+1, n)$-redundant subword such that α without this subword is not from L. We distinguish three cases.

- $m = n > 0$. For $i \in \{1, 2\}$ we define

$$\varphi_i = \overbrace{\mathsf{XF}(p \wedge \mathsf{XF}(p \wedge \mathsf{XF}(p \wedge \ldots \wedge \mathsf{XF}(p \wedge}^{m\text{-times}} \psi_i) \ldots)))$$

  The ω-word $\alpha = (pq)^{m+2} q^\omega \in L$, $\alpha(0, 2)$ is $(m+1, n)$-redundant, and $\alpha_2 = (pq)^{m+1} q^\omega \notin L$.
- $m > n$. For $i \in \{1, 2\}$ we define

$$\varphi_i = \overbrace{\mathsf{XF}(q \wedge \mathsf{XF}(q \wedge \ldots \wedge \mathsf{XF}(q \wedge}^{n\text{-times}} \varphi_i') \ldots))$$

  where

$$\varphi_i' = \overbrace{\mathsf{F}(p_1 \wedge \mathsf{F}(p_2 \wedge \ldots \wedge \mathsf{F}(p_{m-n} \wedge}^{(m-n)\text{-times}} \psi_i) \ldots))$$

  The ω-word $\alpha = (q p_{m-n} p_{m-n-1} \ldots p_1)^{m+1} q^\omega \in L$, $\alpha(0, m-n+1)$ is $(m+1, n)$-redundant, and $\alpha_{m-n+1} \notin L$.
- $m < n$. For $i \in \{1, 2\}$ we define

$$\varphi_i = \overbrace{\mathsf{F}(p_1 \wedge \mathsf{F}(p_2 \wedge \ldots \wedge \mathsf{F}(p_m \wedge}^{m\text{-times}} \overbrace{\mathsf{XX} \ldots \mathsf{X}}^{n} \psi_i) \ldots))$$

  The ω-word $\alpha = (q^{n-m} p_{m+1} p_m \ldots p_1)^{m+2} q^\omega \in L$, $\alpha(0, n+1)$ is $(m+1, n)$-redundant, and $\alpha_{n+1} \notin L$. $\square$

In fact, the previous lemma says that if we take two classes $\mathrm{LTL}(\mathsf{U}^{m_1}, \mathsf{X}^{n_1})$ and $\mathrm{LTL}(\mathsf{U}^{m_2}, \mathsf{X}^{n_2})$ which are syntactically incomparable and where $m_1, m_2 \geq 1$, then their semantical intersection (i.e., the intersection of the corresponding classes of languages) is strictly greater than the class of languages definable in $\mathrm{LTL}(\mathsf{U}^m, \mathsf{X}^n)$ where $m = \min\{m_1, m_2\}$ and $n = \min\{n_1, n_2\}$. Another consequence of Lemma 21 is that there is generally no "best" way how to minimize the nesting depths of X and U modalities in a given LTL formula.

5 Conclusions

The main technical contributions of this paper are the theorems about n-stuttering and general stuttering presented in Section 2. With their help we were able to construct (short) proofs of other results. In particular, we gave an alternative characterization of $\mathrm{LTL}(\mathsf{U}, \mathsf{X}^n)$ languages (which are exactly n-stutter closed languages), proved the strictness of the three hierarchies of LTL formulae introduced in Section 1, and we also showed several related facts about the relationship among the classes in the three hierarchies. All of the presented results carry over to LTL interpreted over finite words.

Some problems are left open. For example, the exact characterization of the semantical intersection of LTL($\mathsf{U}^{m_1}, \mathsf{X}^{n_1}$) and LTL($\mathsf{U}^{m_2}, \mathsf{X}^{n_2}$) classes (in the case when they are syntactically incomparable) surely deserves further attention. Another interesting question is whether Theorem 9 can serve as a basis for new state-space reduction methods in the model-checking area.

References

1. Arnold, A.: A syntactical congruence for rational ω-languages. Theoretical Computer Science **39**, 333–335 (1985)
2. Etessami, K.: A note on a question of Peled and Wilke on stutter-invariant LTL. Information Processing Letters **75**(6), 261–263 (2000)
3. Etessami, K., Wilke, Th.: An until hierarchy and other applications of an Ehrenfeucht-Fraïssé game for temporal logic. Information and Computation **160**, 88–108 (2000)
4. Gabbay, D., Pnueli, A., Shelah, S., Stavi, J.: On the temporal analysis of fairness. In: Conference Record of the 7th ACM Symposium on Principles of Programming Languages, pp. 163–173. ACM Press (1980)
5. Kamp, J.A.W.: Tense logic and the theory of linear order. Ph.D. thesis, University of California, Los Angeles (1968)
6. Kučera, A., Strejček, J.: The stuttering principle revisited: On the expressiveness of nested X and U operators in the logic LTL. In: J. Bradfield (ed.) CSL '02: 11th Annual Conference of the European Association for Computer Science Logic, *Lecture Notes in Computer Science*, vol. 2471, pp. 276–291. Springer-Verlag (2002)
7. Kučera, A., Strejček, J.: An effective characterization of properties definable by LTL formulae with a bounded nesting depth of the next-time operator. Tech. Rep. FIMU-RS-2004-4, Faculty of Informatics, Masaryk University Brno (2004)
8. Lamport, L.: What good is temporal logic? In: R. E. A. Mason (ed.) Proceedings of the IFIP Congress on Information Processing, pp. 657–667. North-Holland, Amsterdam (1983)
9. Peled, D., Wilke, Th.: Stutter-invariant temporal properties are expressible without the next-time operator. Information Processing Letters **63**(5), 243–246 (1997)
10. Peled, D., Wilke, Th., Wolper, P.: An algorithmic approach for checking closure properties of ω-regular languages. Theoretical Computer Science **195**(2), 183–203 (1998)
11. Perrin, D.: Recent results on automata and infinite words. In: M.P. Chytil, V. Koubek (eds.) Proceedings of the 11th Symposium on Mathematical Foundations of Computer Science, *Lecture Notes in Computer Science*, vol. 176, pp. 134–148. Springer (1984)
12. Pnueli, A.: The temporal logic of programs. In: Proceedings of the 18th IEEE Symposium on the Foundations of Computer Science, pp. 46–57. IEEE Computer Society Press (1977)
13. Sistla, A., Clarke, E.: The complexity of propositional linear temporal logics. Journal of the ACM **32**, 733–749 (1985)
14. Thomas, W.: Star-free regular sets of ω-sequences. Information and Control **42**(2), 148–156 (1979)
15. Thomas, W.: Automata on infinite objects. In: J. van Leeuwen (ed.) Handbook of Theoretical Computer Science, vol. B, Formal models and semantics, pp. 133–191. Elsevier (1990)
16. Thue, A.: Über unendliche Zeichenreihen. Kra. Vidensk. Selsk. Skrifter, I. Mat. Nat. Kl. **1906**(7), 1–22 (1906)

17. Wilke, Th.: Classifying discrete temporal properties. In: C. Meinel, S. Tison (eds.) STACS '99: Annual Symposium on Theoretical Aspects of Computer Science, *Lecture Notes in Computer Science*, vol. 1563, pp. 32–46. Springer-Verlag (1999)
18. Wolper, P.: Temporal logic can be more expressive. Information and Control **56**, 72–99 (1983)