Temporal Reasoning on Incomplete Paths

Dana Fisman[1] and Hillel Kugler[2]

[1] Ben-Gurion University, Be'er-Sheva, Israel
[2] Bar-Ilan University, Ramat-Gan, Israel

Abstract. Semantics of temporal logic over truncated paths (i.e. finite paths that correspond to prefixes of computations of the system at hand) have been found useful in incomplete verification methods (such as bounded model checking and dynamic verification), in modeling hardware resets, and clock shifts and in online and offline monitoring of cyber-physical systems. In this paper we explore providing semantics for temporal logics on other types of incomplete paths, namely incomplete ultimately periodic paths, segmentally broken paths and combinations thereof. We review usages of temporal logic reasoning in systems biology, and explore whether systems biology can benefit from the suggested extensions.

1 Introduction

In 1977, in his seminal paper [48], Pnueli suggested to use temporal logic for reasoning about programs or systems. The term *reactive systems* [49, 36] was later coined for the variety of systems considered. A reactive system is a system interacting with an environment in an ongoing manner. The resulting computation can thus be captured by an infinite path, and hence temporal logic formulas were originally defined over infinite paths.

To cope with systems that may exhibit terminating behaviors, semantics of temporal logic over finite *maximal* paths was proposed [45]. The extension of the semantics of infinite paths to finite paths concentrates on the temporal operator *next*. On an infinite path there is always a next time point while the same does not hold on a finite path. Thus, two versions of the *next* operator were suggested, the strong version $\underline{\mathsf{X}}\,\varphi$ demands that there exists a next time point and that φ holds on that point, while the weak version $\underset{\sim}{\mathsf{X}}\,\varphi$ stipulates that if there exists a next time point that φ holds on that point. The *until* operator already comes with a strong ($\underline{\mathsf{U}}$) and a weak ($\underset{\sim}{\mathsf{W}}$) version, and as expected $p\,\underline{\mathsf{U}}\,q$ holds on a finite path if at some point on the path (before it ended) q holds and up until then p holds.

In 2003, it was observed that in many cases it is required to evaluate temporal logic formulas over finite truncated paths [26]. For instance, in incomplete methods of verification, such as bounded model checking and dynamic verification the considered paths are prefixes of computations of the system at hand. Thus, they are not *maximal*, but *truncated*. Other early motivations for reasoning over truncated paths are operators such as hardware resets and clock shifts, that on an abstract level may be seen as truncating a path. On a truncated path not only we may not have a next time point, we may also not have a current time point, since the path can be empty (e.g. if truncated by a reset occurring on the very first cycle). This can easily be solved by introducing also weak and strong versions of atomic propositions or more generally [32], by introducing the operators $\underset{\sim}{\mathsf{X}}^i\varphi$ and $\underline{\mathsf{X}}^i\varphi$ for $i \geq 0$. It is less obvious how to handle, or what semantics to give the *until* operator on a truncated path. Consider the formula $\varphi = (p\,\underline{\mathsf{U}}\,q)$ and a truncated path w where p holds all along and q never does. In some cases, e.g. on a test w that was designed to be long enough for q to hold, we would like to say that φ does not hold, but in other cases. e.g. when we have no knowledge on the truncation point, we would like to say that φ holds (supporting the intuition that the fault of dis-satisfaction of q is in the test, not the system).

The approach advocated in [26] was to interpret a formula not just with regard to a path but also with regard to a *view*. The semantics considers three possible views: *weak*, *neutral* and *strong*. The views differ in their evaluation in cases where there is *doubt* regarding the satisfaction of the formula on an extension of the path. The *weak view* interprets the formula under the assumption that the test *is not* long enough to exhibit all the requested eventualities, and "forgives" unsatisfied eventualities, thus taking an *optimistic* view regarding the possible extension of the path to a complete infinite path of the system. The *strong view* interprets the formula under the assumption that the test *is* long enough to exhibit all the requested eventualities, thus taking a *pessimistic* view regarding the suffix. The *neutral* view interprets the truncated path as if it is a maximal computation path, thus, it will require q to occur before the path ends in order for φ above to hold. The three views approach enables the system specifier to use a single set of temporal logic formulas regardless of the verification approach/tool that will be used and altering only the view (if needed) when switching between verification approaches/tools. The three views approach is part of the IEEE standards PSL [38, 23] and SVA [39, 12], and it is implemented in supporting verification tools (c.f. IBM, RuleBase SixthSense Edition [37] and Synopsys, VC Formal Tools [50]).

One of the properties of the truncated semantics is the *strength relation theorem* stating that if a formula holds on a given path under the strong view then it also holds under the neutral view, and that if it holds under the neutral view, then it also holds under the weak view. This theorem allows one to interpret satisfaction in a multi-valued fashion and some tools provide satisfaction answers as one of the four possibilities: *holds strongly*, *holds neutrally*, *holds weakly* also called *pending*, and *fails*.

The truncated semantics also enjoys the *prefix-extension theorem* on which we elaborate later on, topological relations reminiscent to the safety-liveness dichotomy [24, 25], and relation to classification of safety formulas [42, 26].

In this paper we explore extending the semantics of temporal logic to other types of incomplete paths. For instance, since most considered systems are finite state, their computation paths are ultimately periodic. That is, their paths are of the form uv^ω for some finite paths u and v. It is conceivable that one may know a prefix u' of such an ultimately periodic path, and a prefix v' of its periodic part. In Section 4 we provide semantics for such *incomplete ultimately periodic paths*. Another conceivable incomplete path is a path obtained by composing several finite paths, where the time elapsing from one part to another, and the behavior of the system in the unspecified times is unknown. We refer to such paths as *segmentally broken paths* and in Section 5 provide semantics of temporal logic on segmentally broken paths. More generally, given segments of an observed path we might want to consider the case where the complete path is ultimately periodic and the first segments correspond to fragments of the transient part and the rest of the segments correspond to fragments of the periodic part. We provide semantics for such *segmentally broken ultimately periodic paths* in Section 6 and show that it generalizes the semantics on truncated paths, on incomplete ultimately periodic paths, and on segmentally broken paths. The truncated semantics was studied extensively in offline and online monitoring of cyber-physical systems [19]. Cyber-physical systems exhibit continuous (as opposed to discrete) computation paths, and a prominent logic for reasoning about CPS is Signal Temporal Logic (STL) [44, 19]. In Section 3 we provide a semantics of STL on truncated paths, and in the sections on incomplete ultimately periodic paths, segmentally broken paths and their combination we consider both LTL and STL. Finallly, in Section 7 we review applications of temporal logic reasoning in system biology, and identify cases in which the basic truncated semantics can be considered, and other cases where the extensions for handling ultimately periodic paths and segmentally broken paths may be useful.

2 LTL over Truncated Paths (ELTL)

Let AP be a set of atomic propositions and $\Sigma = 2^{AP}$. We use Σ^∞ for $\Sigma^* \cup \Sigma^\omega$. We use $|w|$ for the number of letters in w. For $1 \leq i \leq |w|$ we use $w[i]$ for the i-th letter of w. For overflow cases, i.e. when $i < 1$ or $i > |w|$ we regard $w[i]$ as ϵ, where ϵ is the empty word. We use $w[..i]$ for the prefix of w ending in $w[i]$, $w[i..]$ for the suffix of w starting with $w[i]$ and $w[i..j]$ for the infix of w starting in $w[i]$ and ending in $w[j]$, for $1 \leq i \leq j \leq |w|$. We use $u \preceq v$ (resp. $w \succeq v$) to denote that u is a prefix of v (resp. w is an extension of v, i.e. that there exists v' such that $w = vv'$).

The logic ELTL, the LTL core of PSL, extends the syntax of LTL by (i) having two versions of the next operator: $\underline{\mathsf{X}}\,\varphi$ (strong next) and $\underset{\sim}{\mathsf{X}}\,\varphi$ (weak next) and (ii) adding a truncation operator $[\varphi \,\underset{\sim}{\mathsf{T}}\, b]$ that is used to evaluate φ on a path truncated on the first satisfaction of b. All the temporal operators of ELTL have weak and strong versions, and the syntax conveniently decorates an operator op by its strength so that $\underline{\mathsf{op}}$ is a strong operator and $\underset{\sim}{\mathsf{op}}$ is a weak operator. Formulas of ELTL are defined as follows

$$\varphi ::= \underline{\mathsf{X}}^{\,i}\, p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid [\varphi_1 \,\underline{\mathsf{U}}\, \varphi_2] \mid [\varphi \,\underset{\sim}{\mathsf{T}}\, b]$$

for $p \in AP$, b a Boolean expression over AP and $i \in \mathbb{N}$ (we regard $\mathbb{N}$ as the set of non-negative integers, so that $0 \in \mathbb{N}$).

Additional derived operators are defined as follows:

$$
\begin{array}{lll}
\utilde{\mathsf{X}}^{\,i}\,\varphi \overset{\text{def}}{=} \neg \underline{\mathsf{X}}^{\,i}\,\neg\varphi & \varphi! \overset{\text{def}}{=} \underline{\mathsf{X}}^{\,0}\,\varphi & \underline{\mathsf{F}}_{\,[i..k]}\,\varphi \overset{\text{def}}{=} \vee_{i\leq j\leq k}\underline{\mathsf{X}}^{\,i} \\
\utilde{\mathsf{X}}\,\varphi \overset{\text{def}}{=} \utilde{\mathsf{X}}^{\,1}\,\varphi & \varphi_1 \wedge \varphi_2 \overset{\text{def}}{=} \neg(\neg\varphi_1 \vee \neg\varphi_2) & \utilde{\mathsf{G}}_{\,[i..k]}\,\varphi \overset{\text{def}}{=} \wedge_{i\leq j\leq k}\utilde{\mathsf{X}}^{\,i} \\
\underline{\mathsf{X}}\,\varphi \overset{\text{def}}{=} \underline{\mathsf{X}}^{\,1}\,\varphi & \utilde{\mathsf{G}}\,\varphi \overset{\text{def}}{=} \neg\,\underline{\mathsf{F}}\,\neg\varphi & [\varphi_1\,\utilde{\mathsf{W}}\,\varphi_2] \overset{\text{def}}{=} [\varphi_1\,\underline{\mathsf{U}}\,\varphi_2] \vee\ \utilde{\mathsf{G}}\,\varphi_1 \\
\varphi \overset{\text{def}}{=} \utilde{\mathsf{X}}^{\,0}\,\varphi & \underline{\mathsf{F}}\,\varphi \overset{\text{def}}{=} [\mathit{true}\,\underline{\mathsf{U}}\,\varphi] & [\varphi\,\underline{\mathsf{T}}\,b] \overset{\text{def}}{=} \neg[\neg\varphi\,\utilde{\mathsf{T}}\,b]
\end{array}
$$

2.1 Semantics under the three views

The semantics of ELTL over truncated paths is given with respect to a path, a word $w \in \Sigma^\infty$ and a view $\mathrm{v} \in \{-, \cdot, +\}$, where $-$ denote the *weak view*, $\cdot$ denotes the *the neutral view* and $+$ denotes the *strong view*. For a letter $\sigma \in \Sigma$ and a proposition $p \in AP$ we define $\sigma \Vdash p$ iff $\sigma \in 2^{AP}$ and $p \in \sigma$. We use $\langle w, \mathrm{v}\rangle \models \varphi$ to denote that φ is satisfied on the finite or infinite word w under the view v. If $\langle w, -\rangle \models \varphi$ we say that φ *holds weakly* on w, if $\langle w, \cdot\rangle \models \varphi$ we say that φ *holds neutrally* on w, and if $\langle w, +\rangle \models \varphi$ we say that φ *holds strongly* on w. For a view $\mathrm{v} \in \{-, \cdot, +\}$ we define $\overline{\mathrm{v}}$ as follows:

$$
\overline{\mathrm{v}} = \begin{cases} + & \text{if } \mathrm{v} = - \\ \cdot & \text{if } \mathrm{v} = \cdot \\ - & \text{if } \mathrm{v} = + \end{cases}
$$

The truncated semantics of ELTL is defined as follows [26]:

$$
\begin{array}{lcl}
\langle w, \mathrm{v}\rangle \models \underline{\mathsf{X}}^{\,i}\,p & \iff & \text{either } |w| > i \text{ and } w[i+1] \Vdash p \\
 & & \text{or } \mathrm{v} = - \text{ and if } |w| > i \text{ then } w[i+1] \Vdash p \\
\langle w, \mathrm{v}\rangle \models \neg\varphi & \iff & \langle w, \overline{\mathrm{v}}\rangle \not\models \varphi \\
\langle w, \mathrm{v}\rangle \models \varphi_1 \vee \varphi_2 & \iff & \langle w, \mathrm{v}\rangle \models \varphi_1 \text{ or } \langle w, \mathrm{v}\rangle \models \varphi_2 \\
\langle w, \mathrm{v}\rangle \models [\varphi_1\,\underline{\mathsf{U}}\,\varphi_2] & \iff & \exists k.\ \langle w[k..], \mathrm{v}\rangle \models \varphi_2 \text{ and } \forall j < k.\ \langle w[j..], \mathrm{v}\rangle \models \varphi_1 \\
\langle w, \mathrm{v}\rangle \models [\varphi\,\utilde{\mathsf{T}}\,b] & \iff & \text{either } \langle w, \mathrm{v}\rangle \models \varphi \\
 & & \text{or } \exists j.\ \langle w[j], \mathrm{v}\rangle \models b \text{ and } \langle w[..j-1], -\rangle \models \varphi
\end{array}
$$

Note that in the statement $\exists k$ used in the semantics of $\underline{\mathsf{U}}$, it may be that $k > |w|$ in which case $w[k..] = \epsilon$. To see that the semantics captures the intuition consider the formula $\underline{\mathsf{X}}^{\,5}\,p$. It holds weakly on any path of length 4 or less, but it doesn't hold strongly or neutrally on such paths. The formula $\utilde{\mathsf{X}}^{\,5}\,p$, on the other hand, holds on such paths under all views. But clearly $\utilde{\mathsf{X}}^{\,5}\,p$ does not hold, not even weakly, on a path of length 6 where p does not hold on the sixth letter. The formula $\utilde{\mathsf{G}}\,p$ holds weakly and neutrally on every finite path where p holds all along, but it does not hold strongly on any finite path. The formula $\underline{\mathsf{F}}\,p$ holds weakly on all finite paths, and it holds neutrally and strongly on every path where p holds at some point. The formula $[p\,\underline{\mathsf{U}}\,q]$ holds weakly on a finite path where p holds all along and q never holds, but it does not hold neutrally or strongly on such paths, whereas the formula $[p\,\utilde{\mathsf{W}}\,q]$ holds on such paths under all views.

The $\underset{\sim}{\mathsf{T}}$ operator gives the ability to truncate a path, and evaluate the given formula under the weak view (its less often used dual $\underline{\mathsf{T}}$ gives the ability to truncate a path and evaluate the formula under the strong view). The formula $[\varphi \, \underset{\sim}{\mathsf{T}} \, b]$ stipulates that either φ holds on the given path or b is satisfied somewhere along the path and φ holds weakly on the path truncated at the point where b first holds.

2.2 The $\top, \bot$ approach to defining the three views semantics

There are various ways to define the semantics of temporal logic over truncated paths under the *weak*, *neutral*, and *strong* views. The definition in Section 2.1 follows the original formulation in [26] and provides a direct definition for each of the views. It was shown in [27] that an equivalent definition can be given by augmenting the alphabet $\Sigma = 2^{AP}$ with two special symbols $\top$ and $\bot$ such that $\top$ satisfies everything, even *false*, and $\bot$ satisfies nothing, not even *true*. We now present this definition and follow it for the rest of the paper since it provides the basis for the extensions suggested herein.

Recall that we define $\Sigma = 2^{AP}$ for a given set of atomic propositions AP. Let $\hat{\Sigma} = \Sigma \cup \{\top, \bot\}$. Let $w = a_1 a_2 a_3 \dots$ be a word in Σ^∞. We use $\overline{w}$ for the word obtained from w by switching $\top$ with $\bot$ and vice versa. For a letter $\sigma \in \hat{\Sigma}$ and a proposition $p \in AP$ we define $\sigma \mathrel{\dot{\Vdash}} p$ iff $\sigma = \top$ or $\sigma \in 2^{AP}$ and $p \in \sigma$.

The concatenation of two words $u, v \in \Sigma^\infty$ is denoted $u \cdot v$ or simply uv. If $u \in \Sigma^*$ then the first $|u|$ letters of uv are u and the following $|v|$ letters are v. If $u \in \Sigma^\omega$ then uv is simply u. Given two sets $U, V \subseteq \Sigma^\infty$, their concatenation is denoted $U \cdot V$ or simply UV and it is the set $\{uv \mid u \in U \text{ and } v \in V\}$. For $k > 1$ we use U^k to denote $U \cdot U^{k-1}$, where U^1 and U^0 denote U and $\{\epsilon\}$, respectively. We use U^* for $\cup_{k \geq 0} U^k$ and U^+ for $\cup_{k \geq 1} U^k$. For a set U we use U^ω for the set $\{w \in \Sigma^\omega \mid \exists u_1, u_2, u_3, u_4, \ldots \in U \text{ s.t. } w = u_1 u_2 u_3 u_4 \cdots\}$.

Under the $\top, \bot$ approach we first provide the neutral view semantics, which we denote $\mathrel{\dot{\models}}$. It is defined very similar to the direct semantics, with the exception of using $\mathrel{\dot{\Vdash}}$ for evaluating propositions, thus taking into account the special letters $\top$ and $\bot$. The weak and strong views are then defined using concatenations with $\top^\omega$ and $\bot^\omega$ and reverting to the neutral view. The formal definition follows.

$$
\begin{array}{lcl}
w \mathrel{\dot{\models}} \underline{\mathsf{X}}^{\,i}\, p & \iff & |w| > i \text{ and } w[i+1] \mathrel{\dot{\Vdash}} p \\
w \mathrel{\dot{\models}} \neg\varphi & \iff & \overline{w} \mathrel{\not\dot{\models}} \varphi \\
w \mathrel{\dot{\models}} \varphi_1 \vee \varphi_2 & \iff & w \mathrel{\dot{\models}} \varphi_1 \text{ or } w \mathrel{\dot{\models}} \varphi_2 \\
w \mathrel{\dot{\models}} [\varphi_1 \,\underline{\mathsf{U}}\, \varphi_2] & \iff & \exists k.\ w[k..] \mathrel{\dot{\models}} \varphi_2 \text{ and } \forall j < k.\ w[j..] \mathrel{\dot{\models}} \varphi_1 \\
w \mathrel{\dot{\models}} [\varphi \,\mathsf{T}\, b] & \iff & \text{either } w \mathrel{\dot{\models}} \varphi \\
 & & \text{or } \exists j.\ w[j] \mathrel{\dot{\models}} b \text{ and } w[..j{-}1] \mathrel{\dot{\models}} \varphi \\
w \mathrel{\overset{-}{\models}} \varphi & \iff & w\top^\omega \mathrel{\dot{\models}} \varphi \\
w \mathrel{\overset{+}{\models}} \varphi & \iff & w\bot^\omega \mathrel{\dot{\models}} \varphi
\end{array}
$$

Note that we still have three views and only one semantics — the notation $w \mathrel{\overset{-}{\models}} \varphi$ abbreviates $\langle w, - \rangle \models \varphi$, the notation $w \mathrel{\overset{+}{\models}} \varphi$ abbreviates $\langle w, + \rangle \models \varphi$ and the notation $w \mathrel{\dot{\models}} \varphi$ abbreviates $\langle w, \cdot \rangle \models \varphi$.

Since $\top$ satisfies every Boolean expression, including *false*, the induction definition gives us that $\top^\omega$ satisfies every formula. Likewise, since $\bot$ satisfies no Boolean expression, including *true*, the induction definition gives us that $\bot^\omega$ satisfies no formula. In particular, as shown in [27] we get the exact same semantics as the one defined in Section 2.1.

In general the following relations hold:

Theorem 1 (The strength relation theorem [26]). *Let φ be a formula in ELTL, and w a finite or infinite word.*

$$w \models^{+} \varphi \quad \Longrightarrow \quad w \models^{\cdot} \varphi \quad \Longrightarrow \quad w \models^{-} \varphi$$

Theorem 2 (The prefix/extension theorem [26]). *Let φ be a formula in ELTL, and w a finite or infinite word.*

- $w \models^{-} \varphi \quad \Longleftrightarrow \quad \forall w' \preceq w.\ \ w' \models^{-} \varphi$
- $w \models^{+} \varphi \quad \Longleftrightarrow \quad \forall w'' \succeq w.\ w'' \models^{+} \varphi$

3 Signal Temporal Logic on Truncated Paths

Signal Temporal Logic (STL) [44, 19] is a logic extending LTL to reason over continuous time that is being used for online and offline monitoring of cyber-physical systems [19]. The considered models (signals) build on the idea of timed state sequences, as proposed in [3] in order to obtain a temporal logic for reasoning over real time system, that carries with it decidable verification algorithms.

3.1 Signal Temporal Logic (STL)

STL is defined over a set of variable $\mathbf{x} = \{x_1, x_2, \ldots, x_k\}$ ranging over $\mathbb{R}$. Predicates over these variables, i.e. functions from $\mathbf{x}$ to $\{\mathtt{tt}, \mathtt{ff}\}$, act as the set of atomic propositions. For instance, linear predicates are of the form $p ::= d_1 x_1 + \ldots + d_k x_k \bowtie d$ where $\bowtie \in \{<, \leq\}$, $d_1, \ldots, d_k, d \in \mathbb{R}$ and $x \in \mathbf{x}$. The syntax of STL is defined as follows:

$$\varphi ::= p(\mathbf{x}) \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \, \underline{\mathsf{U}}_I \, \varphi_2$$

where $p(\mathbf{x})$ is a predicate and I is an interval of $\mathbb{R}_{\geq 0}$. Given a vector $\mathbf{c} = (c_1, \ldots, c_k) \in \mathbb{R}^k$ we use $\mathbf{c} \models p(\mathbf{x})$ if the predicate p holds when the variables $x_1, \ldots, x_k$ are assigned with the real values $c_1, \ldots, c_k$.

Formulas of STL are defined over signals, which are a mapping $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}^k$ so that, $\alpha(t)$ provides the values of all variables $x_1, \ldots, x_k$ at any time point $t \in \mathbb{R}_{\geq 0}$. It is assumed signals are non-Zeno [35], that is they cannot change infinitely many times in any bounded interval (this property is also referred to as *finite variability*).[3] The semantics of STL is defined as follows:[4]

$$
\begin{array}{lcl}
\alpha, t \models p(\mathbf{x}) & \iff & \alpha(t) \Vdash p(\mathbf{x}) \\
\alpha, t \models \neg\varphi & \iff & \alpha, t \not\models \varphi \\
\alpha, t \models \varphi_1 \vee \varphi_2 & \iff & \alpha, t \models \varphi_1 \text{ or } \alpha, t \models \varphi_2 \\
\alpha, t \models [\varphi_1 \,\underline{\mathsf{U}}_I\, \varphi_2] & \iff & \exists t'' \in t + I.\ \alpha, t'' \models \varphi_2 \text{ and } \forall t' \in [t, t'').\ \alpha, t' \models \varphi_1
\end{array}
$$

The notation $\alpha \models \varphi$ abbreviates $\alpha, 0 \models \varphi$. The operator $\underline{\mathsf{U}}$ abbreviates $\underline{\mathsf{U}}_{[0,\infty)}$. Derived operators $\underline{\mathsf{F}}_I$ and $\underset{\sim}{\mathsf{G}}_I$ are defined similarly to LTL. That is, $\underline{\mathsf{F}}_I\,\varphi$ abbreviate $[\mathit{true}\,\underline{\mathsf{U}}_I\,\varphi]$ and $\underset{\sim}{\mathsf{G}}_I\,\varphi$ abbreviates $\neg\,\underline{\mathsf{F}}_I\,\neg\varphi$. Note that when $0 \in I$ then $\underline{\mathsf{U}}_I$ satisfies the following equivalence $[\varphi_1\,\underline{\mathsf{U}}_I\,\varphi_2] \equiv \varphi_2 \vee (\varphi_1 \wedge [\varphi_1\,\underline{\mathsf{U}}_I\,\varphi_2])$, which resembles the until fix-point characterization in LTL.

3.2 Defining STL on Truncated Paths

To define the semantics of STL over finite paths, we consider *finite signals* — these are signals which are mappings from an interval of $\mathbb{R}$ of the form $[0, t)$ to $\mathbb{R}^k$ (recall that k is the number of variables). We call the domain interval $[0, t)$ the *time duration* of α. The value $\alpha(t')$ for $t' \geq t$ is undefined. Note that the empty signal, denoted ε is the signal with time duration $[0, 0)$ which is undefined for every t. The semantics works as is and will give, for instance, that $[p\,\underline{\mathsf{U}}_{[0,7]}\,q]$ does not hold on a signal α with time duration $[0, 6)$ where $p(\alpha(t)) = \mathit{true}$ and $q(\alpha(t)) = \mathit{false}$ for all $t \in [0, 6)$.

To extend the semantics of STL to truncated paths, we need to be able to extend a given signal α with time duration $[0, t)$ to a signal with time duration $[0, \infty)$ so that all predicates p, even *false* evaluate to `tt` on any time point $t' > t$ to get the weak view, and likewise that any predicate, even *true*, evaluates to `ff` under the strong view.

We thus extend the definitions of signals to be a mapping from $\mathbb{R}_{\geq 0}$ to $\mathbb{R}^k \cup \{\top, \bot\}$, and extend the definition of $\Vdash$ to take elements in $\mathbb{R}^k \cup \{\top, \bot\}$ as the left hand side, so that $\top \mathrel{\dot{\Vdash}} p(\mathbf{x})$, $\bot \mathrel{\dot{\not\Vdash}} p(\mathbf{x})$ for any predicate p and for $\mathbf{c} \in \mathbb{R}^k$ the semantics is as before, i.e. $\mathbf{c} \mathrel{\dot{\Vdash}} p(\mathbf{x})$ if the predicate p holds when the variables $x_1, \ldots, x_k$ are assigned with the real values $c_1, \ldots, c_k$.

Next, given a finite signal $\alpha : [0, t) \to \mathbb{R}^k$ we use $\alpha^\top : \mathbb{R}_{\geq 0} \to \mathbb{R}^k \cup \{\top, \bot\}$ to denote its extensions to the signal returning $\alpha(t')$ for every $t' \in [0, t)$ and $\top$ for every $t' \geq t$. Likewise $\alpha^\bot : \mathbb{R}_{\geq 0} \to \mathbb{R}^k \cup \{\top, \bot\}$ denotes the extension of α to the signal returning $\alpha(t')$ for every $t' \in [0, t)$ and $\bot$ for every $t' \geq t$.

The semantics of STL over truncated semantics follows the semantics given in Section 3 but using $\mathrel{\dot{\Vdash}}$ instead of $\Vdash$ and switching evaluations to $\bot$ with

[3] See [4] for a discussion on the difference between time-event sequences and signals, and for their algebraic representation.

[4] There are various ways to define the semantics of the until operator in STL or MTL [40], differing in the type of closedness of the interval in which φ_1 is required to hold. We follow the so called *non-strict* and *non-matching* variant since it is closest to the semantics of LTL. It is shown in [34, 33] that this variant is as expressive as the strict variant.

evaluations to $\top$ and vice versa when negation is evaluated. Formally, given a signal $\alpha : D \to \mathbb{R}^k \cup \{\top, \bot\}$ we use $\overline{\alpha}$ to denote the signal that returns $\alpha(t)$ for every $t \in D$ such that $\alpha(t) \in \mathbb{R}^k$, returns $\top$ for every $t \in D$ such that $\alpha(t) = \bot$ and returns $\bot$ otherwise. The truncated semantics for STL is defined as follows.

$$
\begin{array}{lcl}
\alpha, t \overset{\cdot}{\models} p(\mathbf{x}) & \iff & \alpha(t) \Vdash p(\mathbf{x}) \\
\alpha, t \overset{\cdot}{\models} \neg\varphi & \iff & \overline{\alpha}, t \overset{\cdot}{\not\models} \varphi \\
\alpha, t \overset{\cdot}{\models} \varphi_1 \vee \varphi_2 & \iff & \alpha, t \overset{\cdot}{\models} \varphi_1 \text{ or } \alpha, t \overset{\cdot}{\models} \varphi_2 \\
\alpha, t \overset{\cdot}{\models} [\varphi_2 \, \underline{\mathsf{U}}_I \, \varphi_2] & \iff & \exists t'' \in t + I.\ \alpha, t'' \overset{\cdot}{\models} \varphi_2 \text{ and } \forall t' \in [t, t'').\ \alpha, t' \overset{\cdot}{\models} \varphi_1 \\
\alpha, t \overset{-}{\models} \varphi & \iff & \alpha^\top, t \overset{\cdot}{\models} \varphi \\
\alpha, t \overset{+}{\models} \varphi & \iff & \alpha^\bot, t \overset{\cdot}{\models} \varphi
\end{array}
$$

The notations $\alpha \overset{-}{\models} \varphi$ and $\alpha \overset{+}{\models} \varphi$ abbreviate $\alpha, 0 \overset{-}{\models} \varphi$ and $\alpha, 0 \overset{+}{\models} \varphi$, respectively.

We show that the strength relation theorem (Thm. 1) holds also on STL extended to reason over truncated paths.

Theorem 3 (The strength relation theorem for STL). *Let α be a finite or infinite signal, and φ an STL formula.*

$$\alpha \overset{+}{\models} \varphi \quad \Longrightarrow \quad \alpha \overset{\cdot}{\models} \varphi \quad \Longrightarrow \quad \alpha \overset{-}{\models} \varphi$$

Proof (Sketch). When φ is a predicate the claim holds by the semantics of $\overset{\cdot}{\Vdash}$. The complete proof follows by induction on the structure of the formula. $\square$

Given two finite signals $\alpha_1 : [0, t_1) \to \mathbb{R}^k \cup \{\top, \bot\}$ and $\alpha_2 : [0, t_2) \to \mathbb{R}^k \cup \{\top, \bot\}$ we say that α_1 is a prefix of α_2 denoted $\alpha_1 \preceq \alpha_2$ iff $t_1 \leq t_2$ and for every $t \in [0, t_1)$ it holds that $\alpha_1(t) = \alpha_2(t)$. When $\alpha_1 \preceq \alpha_2$ we also say that α_2 is an extension of α_1 and denote it $\alpha_2 \succeq \alpha_1$.

Theorem 4 (The prefix/extension theorem for STL).

- $\alpha \overset{-}{\models} \varphi \quad \iff \quad \forall \alpha' \preceq \alpha.\ \alpha' \overset{-}{\models} \varphi$
- $\alpha \overset{+}{\models} \varphi \quad \iff \quad \forall \alpha'' \succeq \alpha.\ \alpha'' \overset{+}{\models} \varphi$

Proof (Sketch). The $\Leftarrow$ directions follow trivially since $\alpha \preceq \alpha$ (for the first item) and $\alpha \succeq \alpha$ (for the second item). The proof of the $\Rightarrow$ directions is based on Proposition 10 intuitively saying that replacing a sub-interval with $\top$ helps satisfaction, and vice versa for replacing a $\bot$ sub-interval. $\square$

4 Reasoning over incomplete ultimately periodic paths

A *truncated path* is a special case of an *incomplete path* where only a prefix of the real path is observed. One can argue that other types of incomplete paths are to be considered. For instance, most systems we reason about are finite state and deterministic. As such their paths are ultimately periodic, that is they are of the form uv^ω for some finite paths u and v. It is conceivable that one may know a prefix of such a path and a prefix of the periodic part of the path. How can we reason about such incomplete paths?

4.1 Extending ELTL to reason over incomplete ultimately periodic paths

We suggest extending the weak and strong views to such incomplete paths as well. Assume we are given two finite words u and v such that u is a prefix of the real path and v a prefix of the periodic part. In the weak view we would like to allow some unbounded number of $\top$'s between u and v and some unbounded number of $\top$'s between v and the next time v repeats. We can define the weak view semantics of such path as satisfaction of some path in $(u\top^*)(v\top^*)^\omega$ and the strong view semantics of such a path as satisfaction of all paths in $(u\bot^*)(v\bot^*)^\omega$.

Formally, given $u, v \in \Sigma^*$ we define

$$
\begin{array}{lcl}
(u,v) \models^{-}_{\circlearrowleft} \varphi & \iff & \exists w \in (u\top^*)(v\top^*)^\omega.\ \ w \models \varphi \\
(u,v) \models^{\cdot}_{\circlearrowleft} \varphi & \iff & uv^\omega \models \varphi \\
(u,v) \models^{+}_{\circlearrowleft} \varphi & \iff & \forall w \in (u\bot^*)(v\bot^*)^\omega.\ \ w \models \varphi
\end{array}
$$

We claim that the strength relation theorem holds for the extension to incomplete ultimately periodic paths as well.

Theorem 5 (The strength relation theorem for incomplete ultimately periodic paths).

$$(u,v) \models^{+}_{\circlearrowleft} \varphi \quad \Longrightarrow \quad (u,v) \models^{\cdot}_{\circlearrowleft} \varphi \quad \Longrightarrow \quad (u,v) \models^{-}_{\circlearrowleft} \varphi$$

Proof. Assume $(u,v) \models^{+}_{\circlearrowleft} \varphi$. Then by definition $\forall w \in (u\bot^*)(v\bot^*)^\omega.\ \ w \models \varphi$. This is true in particular for $w = uv^\omega$. Thus $(u,v) \models^{\cdot}_{\circlearrowleft} \varphi$.

Assume $(u,v) \models^{\cdot}_{\circlearrowleft} \varphi$. Then since $uv^\omega \in (u\top^*)(v\top^*)^\omega$ by definition of $\models^{-}_{\circlearrowleft}$, we get that $(u,v) \models^{-}_{\circlearrowleft} \varphi$. $\square$

For pairs of words (u,v) and (u',v') in $\Sigma^* \times \Sigma^*$ we use $(u',v') \preceq (u,v)$ if $u' \preceq u$ and $v' \preceq v$. Likewise we use $(u'',v'') \succeq (u,v)$ if $u'' \succeq u$ and $v'' \succeq v$. The proof of the prefix/extension theorem for incomplete ultimately periodic paths makes use of the following claim.

Proposition 6. *Let $w \in \hat{\Sigma}^\infty$ and $\varphi \in$ ELTL.*

- *Let w' be the word obtained from w by replacing one or more letters with $\top$. Then $w \models \varphi$ implies $w' \models \varphi$.*
- *Let w'' be the word obtained from w by replacing one or more letters with $\bot$. Then $w'' \models \varphi$ implies $w \models \varphi$.*

The proposition clearly holds when w consists of a single letter. The complete proof follows the inductive definition of $\models$ for ELTL.

Note that this doesn't contradict that sometime a formula may not hold on a given path but it may hold after inserting some $\bot$'s in between. For instance $\underline{\mathsf{X}}\, p$ doesn't hold on a path u of length one where p holds on the first cycle, but it does hold on the path $\bot u$.

Theorem 7 (The prefix/extension theorem for incomplete ultimately periodic paths).

- $(u,v) \models^{-}_{\circlearrowleft} \varphi \quad \iff \quad \forall (u',v') \preceq (u,v).\ \ (u',v') \models^{-}_{\circlearrowleft} \varphi$
- $(u,v) \models^{+}_{\circlearrowleft} \varphi \quad \iff \quad \forall (u'',v'') \succeq (u,v).\ \ (u'',v'') \models^{+}_{\circlearrowleft} \varphi$

Proof. The $\Leftarrow$ directions follow trivially since $(u,v) \preceq (u,v)$ (for the first item) and $(u,v) \succeq (u,v)$ for the second item.

For the $\Rightarrow$ directions, assume $(u,v) \models^{-}_{\circlearrowleft} \varphi$. Then by definition $\exists w \in (u\top^*)(v\top^*)^\omega$. $w \models^{\cdot} \varphi$. Assume $w = (u\top^i)(v\top^j)^\omega$. Let $w' = (u'\top^{i+|u|-|u'|})(v'\top^{j+|v|-|v'|})^\omega$. It follows from Proposition 6 that $w \models^{\cdot} \varphi$ implies $w' \models^{\cdot} \varphi$ for any φ. Since $w' \in (u'\top^*)(v'\top^*)^\omega$ if follows that $(u',v') \models^{-}_{\circlearrowleft} \varphi$.

Assume $(u,v) \models^{+}_{\circlearrowleft} \varphi$. We have to show that $(u'',v'') \models^{+} \varphi$ for any $(u'',v'') \succeq (u,v)$. That is we have to show that $w'' \models^{\cdot} \varphi$ for any $w'' \in (u''\bot^*)(v''\bot^*)^\omega$. Let $w'' = (u''\bot^i)(v''\bot^j)^\omega$. Let $w = (u\bot^{i+|u''|-|u|})(v\bot^{j+|v''|-|v|})$. From $(u,v) \models^{+}_{\circlearrowleft} \varphi$ we get that $w \models^{\cdot} \varphi$. It follows from Proposition 6 that $w'' \models^{\cdot} \varphi$. ☐

We note that the semantics over incomplete ultimately periodic paths generalizes the semantics over truncated paths, in the following sense.

Proposition 8. *Let φ be an ELTL formula, and $u \in \Sigma^*$.*

- $u \models^{-} \varphi \iff (u,\epsilon) \models^{-}_{\circlearrowleft} \varphi$
- $u \models^{\cdot} \varphi \iff (u,\epsilon) \models^{\cdot}_{\circlearrowleft} \varphi$
- $u \models^{+} \varphi \iff (u,\epsilon) \models^{+}_{\circlearrowleft} \varphi$

The proof is quite immediate, noting that there is a single word in $(u\top^*)(\epsilon\top)^\omega$, which is $u\top^\omega$ (and likewise for $\bot$), and that $\epsilon^\omega = \epsilon$.

4.2 Extending STL to reason over incomplete ultimately periodic signals

To extend STL to reason over incomplete ultimately periodic signals we define for an interval D of the reals of the form $[0,t)$ the signals $D^\top$ and $D^\bot$ as follows $D^\top(t) = \top$ for any $t \in D$ and $D^\bot(t) = \bot$ for any $t \in D$. Given two signals α and β with time durations $[0,t_\alpha)$ and $[0,t_\beta)$ we define their concatenation, denoted $\alpha\beta$, to be the signal satisfying $\alpha\beta(t) = \alpha(t)$ if $t < t_\alpha$ and $\alpha\beta(t) = \beta(t - t_\alpha)$ otherwise. We extend the notion of concatenation to sets of signals Γ and Δ, so that $\Gamma\Delta = \{\gamma\delta \mid \gamma \in \Gamma \text{ and } \delta \in \Delta\}$.

We use $\mathbb{D}^\top$ and $\mathbb{D}^\bot$ to denote the set of all signals of the form $D^\top$ and $D^\bot$ respectively. For a finite signal α with time duration $[0,t_\alpha)$ we use α^ω for the infinite concatenation of α to itself. That is, $\alpha^\omega(t) = \alpha(t \bmod t_\alpha)$.

Given two signals α and β, the *weak*, *neutral*, and *strong* semantics for ultimately periodic signals is defined as follows:

$$
\begin{array}{lcl}
(\alpha,\beta) \models^{-}_{\circlearrowleft} \varphi & \iff & \exists \gamma \in \alpha\mathbb{D}^\top(\beta\mathbb{D}^\top)^\omega.\ \ \gamma \models^{\cdot} \varphi \\
(\alpha,\beta) \models^{\cdot}_{\circlearrowleft} \varphi & \iff & \alpha\beta^\omega \models^{\cdot} \varphi \\
(\alpha,\beta) \models^{+}_{\circlearrowleft} \varphi & \iff & \forall \gamma \in \alpha\mathbb{D}^\bot(\beta\mathbb{D}^\bot)^\omega.\ \ \gamma \models^{\cdot} \varphi
\end{array}
$$

Theorem 9 (The strength relation theorem for incomplete ultimately periodic signals).

$$(\alpha, \beta) \models^{+}_{\circlearrowright} \varphi \quad \Longrightarrow \quad (\alpha, \beta) \models^{\cdot}_{\circlearrowright} \varphi \quad \Longrightarrow \quad (\alpha, \beta) \models^{-}_{\circlearrowright} \varphi$$

Proof. Assume $(\alpha, \beta) \models^{+}_{\circlearrowright} \varphi$ then $\forall \gamma \in \alpha\mathbb{D}^{\perp}(\beta\mathbb{D}^{\perp})^{\omega}$ we have that $\gamma \dot{\models} \varphi$. In particular it holds for $\gamma = \alpha[0,0)^{\perp}(\beta[0,0)^{\perp})^{\omega} = \alpha\beta^{\omega}$. Thus $(\alpha, \beta) \models^{\cdot}_{\circlearrowright} \varphi$.

Assume $(\alpha, \beta) \models^{\cdot}_{\circlearrowright} \varphi$ then $\alpha\beta^{\omega} = \alpha[0,0)^{\top}(\beta[0,0)^{\top})^{\omega} \models \varphi$. Thus, there exists $\gamma \in \alpha\mathbb{D}^{\top}(\beta\mathbb{D}^{\top})^{\omega}$ such that $\gamma \dot{\models} \varphi$. Hence $(\alpha, \beta) \models^{-}_{\circlearrowright} \varphi$. □

For pairs of finite signals (α, β) and (α', β') we use $(\alpha', \beta') \preceq (\alpha, \beta)$ if $\alpha' \preceq \alpha$ and $\beta' \preceq \beta$. Recall the definition of $\preceq$ between signals as given in Section 3.2. Likewise we use $(\alpha'', \beta'') \succeq (\alpha, \beta)$ if $\alpha'' \succeq \alpha$ and $\beta'' \succeq \beta$.

Proposition 10. *Let α be a signal over $\mathbb{R}^k \cup \{\top, \perp\}$ and $\varphi \in STL$. Assume $\alpha = \beta\gamma\delta$ for some signals β, γ, δ.*

- *Let $\alpha' = \beta D^{\top}_{\gamma}\delta$ be the signal obtained from α by replacing the middle signal γ with the signal $D^{\top}_{\gamma}$ which has the same time duration $[0, t_\gamma)$ as γ. Then $\alpha \models \varphi$ implies $\alpha' \models \varphi$.*
- *Let $\alpha'' = \beta D^{\perp}_{\gamma}\delta$ be the signal obtained from α by replacing the middle signal γ with the signal $D^{\perp}_{\gamma}$ which has the same time duration $[0, t_\gamma)$ as γ. Then $\alpha'' \models \varphi$ implies $\alpha \models \varphi$.*

A corollary of Proposition 10 is that $\alpha \models \varphi$ implies that $\alpha' \models \varphi$ for any α' obtained from α by replacing any number of sub-intervals of α with $\top$ intervals of the same duration, and similarly $\alpha'' \models \varphi$ implies that $\alpha \models \varphi$ for any α obtained from α'' by replacing any number of $\perp$ sub-intervals of α with arbitrary signals of the same duration.

Theorem 11 (The prefix/extension theorem for incomplete ultimately periodic signals).

- $(\alpha, \beta) \models^{-}_{\circlearrowright} \varphi \quad \Longleftrightarrow \quad \forall (\alpha', \beta') \preceq (\alpha, \beta).\ \ (\alpha', \beta') \models^{-}_{\circlearrowright} \varphi$
- $(\alpha, \beta) \models^{+}_{\circlearrowright} \varphi \quad \Longleftrightarrow \quad \forall (\alpha'', \beta'') \succeq (\alpha, \beta).\ \ (\alpha'', \beta'') \models^{+}_{\circlearrowright} \varphi$

Proof. The $\Leftarrow$ direction follows trivially. We prove the $\Rightarrow$ direction. We assume here that the time duration of a signal γ is $[0, t_\gamma)$.

Assume $(\alpha, \beta) \models^{-}_{\circlearrowright} \varphi$. And let α' and β' be such that $\alpha' \preceq \alpha$ and $\beta' \preceq \beta$. Then by definition $\exists \gamma \in \alpha\mathbb{D}^{\top}(\beta\mathbb{D}^{\top})^{\omega}$. $\gamma \dot{\models} \varphi$. Assume $\gamma = (\alpha[0, t_i)^{\top})(\beta[0, t_i)^{\top})^{\omega}$. Let $\gamma' = (\alpha'[0, t'_i)^{\top})(\beta'[0, t'_j)^{\top})^{\omega}$ where $t'_i = t_i + (t_\alpha - t_{\alpha'})$ and $t'_j = t_j + (t_\beta - t_{\beta'})$. It follows from Proposition 10 that $\gamma \dot{\models} \varphi$ implies $\gamma' \dot{\models} \varphi$ for any φ. Since $\gamma' \in \alpha'\mathbb{D}^{\top}(\beta'\mathbb{D}^{\top})^{\omega}$ if follows that $(\alpha', \beta') \models^{-}_{\circlearrowright} \varphi$.

Assume $(\alpha, \beta) \models^{+}_{\circlearrowright} \varphi$. We have to show that $(\alpha'', \beta'') \models^{+}_{\circlearrowright} \varphi$ for any $(\alpha'', \beta'') \succeq (\alpha, \beta)$. That is, we have to show that $\gamma'' \dot{\models} \varphi$ for any $\gamma'' \in \alpha\mathbb{D}^{\perp}(\beta\mathbb{D}^{\perp})^{\omega}$. Let $\gamma'' = (\alpha''[0, t_i)^{\perp})(\beta''[0, t_j)^{\perp})^{\omega}$. Let $\gamma = (\alpha[0, t_i + t_{\alpha''} - t_\alpha)^{\perp})(\beta[t_j + t_{\beta''} - t_\beta)^{\perp})$. From $(\alpha, \beta) \models^{+}_{\circlearrowright} \varphi$ we get that $\gamma \dot{\models} \varphi$. It follows from Proposition 10 that $\gamma'' \dot{\models} \varphi$. □

We note that the semantics over incomplete ultimately periodic signals generalizes the semantics over truncated signals, in the following sense.

Proposition 12. *Let φ be an STL formula, and α a finite signal.*

- $\alpha \models^{-} \varphi \iff (\alpha, \varepsilon) \models^{-}_{\circlearrowleft} \varphi$
- $\alpha \models^{\cdot} \varphi \iff (\alpha, \varepsilon) \models^{\cdot}_{\circlearrowleft} \varphi$
- $\alpha \models^{+} \varphi \iff (\alpha, \varepsilon) \models^{+}_{\circlearrowleft} \varphi$

The proof here is also quite immediate, noting that there is a single signal in $(\alpha\mathbb{D}^{\top})(\varepsilon\mathbb{D}\top)^{\omega}$, which is $\alpha^{\top}$ (and likewise for $\bot$), and that $\varepsilon^{\omega} = \varepsilon$.

5 Reasoning over segmentally broken paths

5.1 Discrete time

The idea of reasoning over incomplete paths can be extended also for the case that one knows segments of the paths $u_1, u_2, \ldots, u_k$, such that the complete path is of the form $u_1 \cdot x_1 \cdot u_2 \cdot x_2 \cdots u_k \cdot x_k$ for some $x_k \in \Sigma^{\omega}$ and some x_i's in Σ^* for $1 \leq i < k$. This might be the case in the setting where the trace is obtained by sampling or when the trace collection method is not totally reliable. In such cases we would say that the segmentally broken path holds weakly iff it holds on some replacements of the missing parts with $\top$'s and similarly that it holds strongly iff it holds on all replacement of the missing parts with $\bot$'s. Formally,

$$
\begin{array}{lcl}
\langle u_1, u_2, \ldots, u_k \rangle \models^{-}_{|||} \varphi & \iff & \exists \gamma \in u_1 \top^* u_2 \top^* \ldots \top^* u_k \top^{\omega}.\ \ \gamma \models^{-} \varphi \\
\langle u_1, u_2, \ldots, u_k \rangle \models^{\cdot}_{|||} \varphi & \iff & u_1 u_2 \ldots u_k \models^{\cdot} \varphi \\
\langle u_1, u_2, \ldots, u_k \rangle \models^{+}_{|||} \varphi & \iff & \forall \gamma \in u_1 \bot^* u_2 \bot^* \ldots \bot^* u_k \bot^{\omega}.\ \ \gamma \models^{+} \varphi
\end{array}
$$

It is immediate from the definition that the semantics over segmentally broken paths generalizes the semantics over truncated paths:

Proposition 13. *Let φ be an ELTL formula, and u a finite word.*

- $u \models^{-} \varphi \iff \langle u \rangle \models^{-}_{|||} \varphi$
- $u \models^{\cdot} \varphi \iff \langle u \rangle \models^{\cdot}_{|||} \varphi$
- $u \models^{+} \varphi \iff \langle u \rangle \models^{+}_{|||} \varphi$

It is not hard to see that the extension of the strength relation theorem and the prefix relation theorem hold as well.

Theorem 14 (The strength relation theorem for segmentally broken paths).

$$\langle u_1, u_2, \ldots, u_k \rangle \models^{+}_{|||} \varphi \implies \langle u_1, u_2, \ldots, u_k \rangle \models^{\cdot}_{|||} \varphi \implies \langle u_1, u_2, \ldots, u_k \rangle \models^{-}_{|||} \varphi$$

For tuples of words $\langle u_1, \ldots, u_k \rangle$ and $\langle u_1', \ldots, u_k' \rangle$ in $(\Sigma^*)^k$ we use $\langle u_1', \ldots, u_k' \rangle \preceq \langle u_1, \ldots, u_k \rangle$ if $u_i' \preceq u_i$.

Theorem 15 (The prefix/extension theorem for segmentally broken paths).

- $\langle u_1, \ldots, u_k \rangle \models^{-}_{|||} \varphi \iff \forall \langle u_1', \ldots, u_k' \rangle \preceq \langle u_1, \ldots, u_k \rangle.\ \ \langle u_1', \ldots, u_k' \rangle \models^{-}_{|||} \varphi$
- $\langle u_1, \ldots, u_k \rangle \models^{+}_{|||} \varphi \iff \forall \langle u_1'', \ldots, u_k'' \rangle \succeq \langle u_1, \ldots, u_k \rangle.\ \ \langle u_1'', \ldots, u_k'' \rangle \models^{+}_{|||} \varphi$

5.2 Continuous time

Similarly we can define STL over segmentally broken signals. Let $\langle \alpha_1, \ldots, \alpha_k \rangle$ be a tuple of finite signals. We define

$$\begin{array}{lcl}
\langle \alpha_1, \ldots, \alpha_k \rangle \overset{-}{\models}_{|||} \varphi & \iff & \exists \gamma \in \alpha_1 \mathbb{D}^\top \alpha_2 \mathbb{D}^\top \ldots \alpha_k [0, \infty)^\top. \;\; \gamma \overset{-}{\models} \varphi \\
\langle \alpha_1, \ldots, \alpha_k \rangle \overset{\cdot}{\models}_{|||} \varphi & \iff & \alpha_1 \ldots \alpha_k \overset{\cdot}{\models} \varphi \\
\langle \alpha_1, \ldots, \alpha_k \rangle \overset{+}{\models}_{|||} \varphi & \iff & \forall \gamma \in \alpha_1 \mathbb{D}^\perp \alpha_2 \mathbb{D}^\perp \ldots \alpha_k [0, \infty)^\perp. \;\; \gamma \overset{\cdot}{\models} \varphi
\end{array}$$

It is immediate from the definition that the semantics over segmentally broken paths generalizes the semantics over truncated paths:

Proposition 16. *Let φ be an STL formula, and α a finite signal.*

- $\alpha \overset{-}{\models} \varphi \iff \langle \alpha \rangle \overset{-}{\models}_{|||} \varphi$
- $\alpha \overset{\cdot}{\models} \varphi \iff \langle \alpha \rangle \overset{\cdot}{\models}_{|||} \varphi$
- $\alpha \overset{+}{\models} \varphi \iff \langle \alpha \rangle \overset{+}{\models}_{|||} \varphi$

It is not hard to see that the extension of the strength relation theorem and the prefix relation theorem hold as well.

Theorem 17 (The strength relation theorem for segmentally broken paths).

$$\langle \alpha_1, \ldots, \alpha_k \rangle \overset{+}{\models}_{|||} \varphi \quad \Longrightarrow \quad \langle \alpha_1, \ldots, \alpha_k \rangle \overset{\cdot}{\models}_{|||} \varphi \quad \Longrightarrow \quad \langle \alpha_1, \ldots, \alpha_k \rangle \overset{-}{\models}_{|||} \varphi$$

For tuples of finite signals $\langle \alpha_1, \ldots, \alpha_k \rangle$ and $\langle \alpha'_1, \ldots, \alpha'_k \rangle$ we use $\langle \alpha'_1, \ldots, \alpha'_k \rangle \preceq \langle \alpha_1, \ldots, \alpha_k \rangle$ if $\alpha'_i \preceq \alpha_i$.

Theorem 18 (The prefix/extension theorem for segmentally broken paths).

- $\langle \alpha_1, \ldots, \alpha_k \rangle \overset{-}{\models}_{|||} \varphi \quad \iff \quad \forall \langle \alpha'_1, \ldots, \alpha'_k \rangle \preceq \langle \alpha_1, \ldots, \alpha_k \rangle. \; \langle \alpha'_1, \ldots, \alpha'_k \rangle \overset{-}{\models}_{|||} \varphi$
- $\langle \alpha_1, \ldots, \alpha_k \rangle \overset{+}{\models}_{|||} \varphi \quad \iff \quad \forall \langle \alpha''_1, \ldots, \alpha''_k \rangle \succeq \langle \alpha_1, \ldots, \alpha_k \rangle. \; \langle \alpha''_1, \ldots, \alpha''_k \rangle \overset{+}{\models}_{|||} \varphi$

6 Reasoning over segmentally broken ultimately periodic paths

It could be that we are observing segments of an ultimately priodic path where both the transient part and the periodic part are broken into segments with uncertainties on the time elapsing between them and the value of the propositions in these segments. We can combine the semantics of Sections 4 and 5 to reason about such incomplete paths as follows.

6.1 Discrete time

Let $u_1, u_2, \ldots, u_k \in \Sigma^*$ and $v_1, v_2, \ldots, v_\ell \in \Sigma^*$.

$$(\langle u_1, \ldots, u_k \rangle, \langle v_1, \ldots, v_\ell \rangle) \models^{-}_{\substack{|||\\ \circlearrowleft}} \varphi \iff \exists \gamma \in (u_1 \top^* \ldots u_k \top^*)(v_1 \top^* \ldots v_\ell \top^*)^\omega . \gamma \models^{\cdot} \varphi$$

$$(\langle u_1, \ldots, u_k \rangle, \langle v_1, \ldots, v_\ell \rangle) \models^{\cdot}_{\substack{|||\\ \circlearrowleft}} \varphi \iff (u_1 u_2 \ldots u_k)(v_1 v_2 \ldots v_\ell)^\omega \models^{\cdot} \varphi$$

$$(\langle u_1, \ldots, u_k \rangle, \langle v_1, \ldots, v_\ell \rangle) \models^{+}_{\substack{|||\\ \circlearrowleft}} \varphi \iff \forall \gamma \in (u_1 \bot^* \ldots u_k \bot^*)(v_1 \bot^* \ldots v_\ell \bot^*)^\omega . \ \gamma \models^{\cdot} \varphi$$

It is immediate from the definition that this semantics generalizes the semantics over (i) truncated paths, (ii) ultimately periodic paths, and (iii) segmentally broken paths.

Proposition 19. *Let φ be an ELTL formula, $u, v, u_1, \ldots, u_k$ be finite or infinite words.*

(I)
- $u \models^{-} \varphi \iff (\langle u \rangle, \langle \epsilon \rangle) \models^{-}_{\substack{|||\\ \circlearrowleft}} \varphi$
- $u \models^{\cdot} \varphi \iff (\langle u \rangle, \langle \epsilon \rangle) \models^{\cdot}_{\substack{|||\\ \circlearrowleft}} \varphi$
- $u \models^{+} \varphi \iff (\langle u \rangle, \langle \epsilon \rangle) \models^{+}_{\substack{|||\\ \circlearrowleft}} \varphi$

(II)
- $(u, v) \models^{-}_{\circlearrowleft} \varphi \iff (\langle u \rangle, \langle v \rangle) \models^{-}_{\substack{|||\\ \circlearrowleft}} \varphi$
- $(u, v) \models^{\cdot}_{\circlearrowleft} \varphi \iff (\langle u \rangle, \langle v \rangle) \models^{\cdot}_{\substack{|||\\ \circlearrowleft}} \varphi$
- $(u, v) \models^{+}_{\circlearrowleft} \varphi \iff (\langle u \rangle, \langle v \rangle) \models^{+}_{\substack{|||\\ \circlearrowleft}} \varphi$

(III)
- $\langle u_1, \ldots, u_k \rangle \models^{-}_{|||} \varphi \iff (\langle u_1, \ldots, u_k \rangle, \langle \epsilon \rangle) \models^{-}_{\substack{|||\\ \circlearrowleft}} \varphi$
- $\langle u_1, \ldots, u_k \rangle \models^{\cdot}_{|||} \varphi \iff (\langle u_1, \ldots, u_k \rangle, \langle \epsilon \rangle) \models^{\cdot}_{\substack{|||\\ \circlearrowleft}} \varphi$
- $\langle u_1, \ldots, u_k \rangle \models^{+}_{|||} \varphi \iff (\langle u_1, \ldots, u_k \rangle, \langle \epsilon \rangle) \models^{+}_{\substack{|||\\ \circlearrowleft}} \varphi$

It is not hard to see that the extension of the strength relation theorem and the prefix relation theorem hold as well.

Theorem 20 (The strength relation theorem for segmentally broken ultimately periodic paths).

$$\begin{aligned}(\langle u_1, \ldots, u_k \rangle, \langle v_1, \ldots, v_\ell \rangle) \models^{+}_{\substack{|||\\ \circlearrowleft}} \varphi \quad &\Longrightarrow \quad (\langle u_1, \ldots, u_k \rangle, \langle v_1, \ldots, v_\ell \rangle) \models^{\cdot}_{\substack{|||\\ \circlearrowleft}} \varphi \\ &\Longrightarrow \quad (\langle u_1, \ldots, u_k \rangle, \langle v_1, \ldots, v_\ell \rangle) \models^{-}_{\substack{|||\\ \circlearrowleft}} \varphi\end{aligned}$$

Let $\mathbf{u} = \langle u_1, \ldots, u_k \rangle$, $\mathbf{u}' = \langle u'_1, \ldots, u'_k \rangle$, $\mathbf{v} = \langle v_1, \ldots, v_\ell \rangle$ and $\mathbf{v}' = \langle v'_1, \ldots, v'_\ell \rangle$ be tuples of words. We say that $(\mathbf{u}, \mathbf{v}) \preceq (\mathbf{u}', \mathbf{v}')$ if $\mathbf{u} \preceq \mathbf{u}'$ and $\mathbf{v} \preceq \mathbf{v}'$.

Theorem 21 (The prefix/extension theorem for segmentally broken ultimately periodic paths).

- $(\boldsymbol{u}, \boldsymbol{v}) \models^{-}_{\substack{|||\\ \circlearrowleft}} \varphi \iff \forall (\boldsymbol{u}', \boldsymbol{v}') \preceq (\boldsymbol{u}, \boldsymbol{v}). \ (\boldsymbol{u}', \boldsymbol{v}') \models^{-}_{\substack{|||\\ \circlearrowleft}} \varphi$
- $(\boldsymbol{u}, \boldsymbol{v}) \models^{+}_{\substack{|||\\ \circlearrowleft}} \varphi \iff \forall (\boldsymbol{u}'', \boldsymbol{v}'') \succeq (\boldsymbol{u}, \boldsymbol{v}). \ (\boldsymbol{u}'', \boldsymbol{v}'') \models^{+}_{\substack{|||\\ \circlearrowleft}} \varphi$

6.2 Continuous time

Similarly we can define STL over segmentally broken ultimately periodic signals. Let $\langle \alpha_1, \ldots, \alpha_k \rangle$ and $\langle \beta_1, \ldots, \beta_\ell \rangle$ be tuples of finite signals. We define

$$(\langle \alpha_1, \ldots, \alpha_k \rangle, \langle \beta_1, \ldots, \beta_k \rangle) \stackrel{-}{\models}_{\substack{\Vvert \\ \circlearrowleft}} \varphi \iff \exists \gamma \in (\alpha_1 \mathbb{D}^\top \ldots \alpha_k \mathbb{D}^\top)(\beta_1 \mathbb{D}^\top \ldots \beta_\ell \mathbb{D}^\top)^\omega .\ \gamma \stackrel{\cdot}{\models} \varphi$$

$$(\langle \alpha_1, \ldots, \alpha_k \rangle, \langle \beta_1, \ldots, \beta_k \rangle) \stackrel{\cdot}{\models}_{\substack{\Vvert \\ \circlearrowleft}} \varphi \iff (\alpha_1 \ldots \alpha_k)(\beta_1 \ldots \beta_\ell)^\omega \stackrel{\cdot}{\models} \varphi$$

$$(\langle \alpha_1, \ldots, \alpha_k \rangle, \langle \beta_1, \ldots, \beta_k \rangle) \stackrel{\pm}{\models}_{\substack{\Vvert \\ \circlearrowleft}} \varphi \iff \forall \gamma \in (\alpha_1 \mathbb{D}^\perp \ldots \alpha_k \mathbb{D}^\perp)(\beta_1 \mathbb{D}^\perp \ldots \beta_\ell \mathbb{D}^\perp)^\omega \ \ \gamma \stackrel{\cdot}{\models} \varphi$$

It is immediate from the definition that the semantics over segmentally broken ultimately periodic paths generalizes the semantics over (i) truncated signals, (ii) incomplete ultimately periodic signals, and (iii) segmentally broken signals.

Proposition 22. *Let φ be an STL formula, and $\alpha, \alpha_1, \ldots, \alpha_k, \beta, \beta_1, \ldots, \beta_\ell$ finite signals.*

(I)
- $\alpha \stackrel{-}{\models} \varphi \iff (\langle \alpha \rangle, \langle \varepsilon \rangle) \stackrel{-}{\models}_{\substack{\Vvert \\ \circlearrowleft}} \varphi$
- $\alpha \stackrel{\cdot}{\models} \varphi \iff (\langle \alpha \rangle, \langle \varepsilon \rangle) \stackrel{\cdot}{\models}_{\substack{\Vvert \\ \circlearrowleft}} \varphi$
- $\alpha \stackrel{\pm}{\models} \varphi \iff (\langle \alpha \rangle, \langle \varepsilon \rangle) \stackrel{\pm}{\models}_{\substack{\Vvert \\ \circlearrowleft}} \varphi$

(II)
- $(\alpha, \beta) \stackrel{-}{\models}_{\circlearrowleft} \varphi \iff (\langle \alpha \rangle, \langle \beta \rangle) \stackrel{-}{\models}_{\substack{\Vvert \\ \circlearrowleft}} \varphi$
- $(\alpha, \beta) \stackrel{\cdot}{\models}_{\circlearrowleft} \varphi \iff (\langle \alpha \rangle, \langle \beta \rangle) \stackrel{\cdot}{\models}_{\substack{\Vvert \\ \circlearrowleft}} \varphi$
- $(\alpha, \beta) \stackrel{\pm}{\models}_{\circlearrowleft} \varphi \iff (\langle \alpha \rangle, \langle \beta \rangle) \stackrel{\pm}{\models}_{\substack{\Vvert \\ \circlearrowleft}} \varphi$

(III)
- $\langle \alpha_1, \ldots, \alpha_k \rangle \stackrel{-}{\models}_{\Vvert} \varphi \iff (\langle \alpha_1, \ldots, \alpha_k \rangle, \langle \varepsilon \rangle) \stackrel{-}{\models}_{\substack{\Vvert \\ \circlearrowleft}} \varphi$
- $\langle \alpha_1, \ldots, \alpha_k \rangle \stackrel{\cdot}{\models}_{\Vvert} \varphi \iff (\langle \alpha_1, \ldots, \alpha_k \rangle, \langle \varepsilon \rangle) \stackrel{\cdot}{\models}_{\substack{\Vvert \\ \circlearrowleft}} \varphi$
- $\langle \alpha_1, \ldots, \alpha_k \rangle \stackrel{\pm}{\models}_{\Vvert} \varphi \iff (\langle \alpha_1, \ldots, \alpha_k \rangle, \langle \varepsilon \rangle) \stackrel{\pm}{\models}_{\substack{\Vvert \\ \circlearrowleft}} \varphi$

It is not hard to see that the extension of the strength relation theorem and the prefix relation theorem hold as well. Let $\boldsymbol{\alpha} = \langle \alpha_1, \ldots, \alpha_k \rangle$ and $\boldsymbol{\beta} = \langle \beta_1, \ldots, \beta_\ell \rangle$ be tuples of finite signals.

Theorem 23 (The strength relation theorem for segmentally broken ultimately periodic paths).

$$(\boldsymbol{\alpha}, \boldsymbol{\beta}) \stackrel{\pm}{\models}_{\substack{\Vvert \\ \circlearrowleft}} \varphi \quad \Longrightarrow \quad (\boldsymbol{\alpha}, \boldsymbol{\beta}) \stackrel{\cdot}{\models}_{\substack{\Vvert \\ \circlearrowleft}} \varphi \quad \Longrightarrow \quad (\boldsymbol{\alpha}, \boldsymbol{\beta}) \stackrel{-}{\models}_{\substack{\Vvert \\ \circlearrowleft}} \varphi$$

Let $\boldsymbol{\alpha} = \langle \alpha_1, \ldots, \alpha_k \rangle, \boldsymbol{\alpha}' = \langle \alpha'_1, \ldots, \alpha'_k \rangle, \boldsymbol{\beta} = \langle \beta_1, \ldots, \beta_\ell \rangle$ and $\boldsymbol{\beta}' = \langle \beta'_1, \ldots, \beta'_\ell \rangle$ be tuples of finite signals. We say that $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \preceq (\boldsymbol{\alpha}', \boldsymbol{\beta}')$ if $\boldsymbol{\alpha} \preceq \boldsymbol{\alpha}'$ and $\boldsymbol{\beta} \preceq \boldsymbol{\beta}'$.

Theorem 24 (The prefix/extension theorem for segmentally broken ultimately periodic paths).

- $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \stackrel{-}{\models}_{\substack{\Vvert \\ \circlearrowleft}} \varphi \quad \iff \quad \forall (\boldsymbol{\alpha}', \boldsymbol{\beta}') \preceq (\boldsymbol{\alpha}, \boldsymbol{\beta}).\ \ (\boldsymbol{\alpha}', \boldsymbol{\beta}') \stackrel{-}{\models}_{\substack{\Vvert \\ \circlearrowleft}} \varphi$
- $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \stackrel{\pm}{\models}_{\substack{\Vvert \\ \circlearrowleft}} \varphi \quad \iff \quad \forall (\boldsymbol{\alpha}'', \boldsymbol{\beta}'') \succeq (\boldsymbol{\alpha}, \boldsymbol{\beta}).\ \ (\boldsymbol{\alpha}'', \boldsymbol{\beta}'') \stackrel{\pm}{\models}_{\substack{\Vvert \\ \circlearrowleft}} \varphi$

7 Temporal Reasoning in Systems Biology

Systems Biology takes an integrated and *holistic* view to study the complexity of biological systems and emphasizes the value of computational models and methods as a complementary approach to experimental work [2, 11, 43, 47]. A key aspect is building computational models of the biological system studied, that can typically integrate several aspects of the system behavior which according to the classical *reductionist* approach are studied in isolation. There are several advantages that can be gained by constructing computational models: 1) The process of model construction helps identify gaps in our understanding of the system; 2) Complex behavior can arise due to the interaction between different components in the system, and as the scope of the studied systems expands, it becomes hard to reason about the system behavior without computational models and supporting tools; 3) A model should in principle be able to reproduce all known experimental results. Keeping track of the relevant experiments and ensuring a mechanistic model can indeed reproduce them may become challenging as the size of the model and number of experiments grow. Using formal specification languages, in particular temporal logic, can enable a clear and accurate specification of experimental results, and the application of automated formal verification approaches to ensure the model indeed satisfies the specification; 4) The most exciting aspect of modeling is the ability to use the derived models to make new predictions that can then be tested experimentally and lead to new biological insights. Temporal logic can be used gainfully to formulate queries that form the basis for making new predictions.

While temporal logic is now being actively used in the biological context (see e.g. [28, 13, 14, 8, 51, 20, 7] and references therein), important open questions remain regarding the appropriate type of temporal logic needed, how to provide intuitive interfaces and tools that can facilitate use by biologists, and the complexity of key verification and synthesis algorithms in the biological context. We next provide an overview of the use of temporal logic in biological modeling, emphasizing the subsets of temporal logic that have been identified as particularly useful and are currently applied in biological modeling. We suggest that considering truncated semantics for temporal logic is of relevance for biological applications, and identify cases in which the basic truncated semantics can be considered, and other cases where the extensions for handling ultimately periodic paths and segmentally broken paths may be useful.

7.1 Truncated Path Semantics

Early work on using temporal logic in systems biology was based on the observation that specifying experiments and hypotheses in temporal logic can open the way to applying powerful model checking algorithms and tools that have been originally developed for verification of engineered systems. In [28, 29] the Pathway Logic tool is presented, which allows to use rewrite logic systems and the Maude toolset [16, 30] to describe how biological signalling pathways operate in cells. Such signalling pathways are composed of molecular components that regulate each others activity and allow a cell to interact with its environment and identify required nutrients or harmful toxins, and receive signals from neighbouring cells. Properties describing biological questions of interest were specified using linear temporal logic (LTL), an example of such an LTL property described in [28] is:[5]

$$\underset{\sim}{\mathsf{G}}\,(\,\underset{\sim}{\mathsf{G}}\,(p1 \land \neg p2) \rightarrow \neg\,\underline{\mathsf{F}}\,p3)$$

An additional question studied in [29] is concerned with the necessity of eventually reaching a state in which two signals $p1$ and $p2$ are activated from some initial state $q1$.

$$q1 \rightarrow\ \underline{\mathsf{F}}\,(p1 \land p2)$$

Instantiations of this formula are then studied in which certain signals are removed from state $q1$, to obtain a different initial state $q1'$. The Maude tool also provides powerful simulation capabilities, which were used to study simpler queries related to forward or backward reachability. We propose that using the truncated semantics presented in this paper could allow a unified view between the verification and simulation capabilities of the Maude tool. For large models in which verification becomes more challenging, using truncated semantics could potentially allow to use the Maude simulation capabilities to refute LTL properties or increase the confidence that they hold. More generally, for biological modeling tools that utilize temporal logic or equivalent specification languages, we suggest that the dynamic properties are specified once in a uniform manner, and the appropriate view is configured according to the tool and methods used to analyze the properties.

One of the challenges in using temporal logic for specifying biological models is making the process easy and intuitive for experimental biologists. In [46] the use of *patterns* for frequently used types of biological specification is suggested, and a set of patterns based on a survey of previous biological modeling efforts is proposed. The underlying temporal logic used is CTL. While some patterns lie in the common fragment of LTL and CTL, this does not hold for all. For instance one type of *sequence patterns* is given by the CTL formula $\mathsf{A}\underset{\sim}{\mathsf{G}}\,(p \rightarrow \mathsf{E}\underline{\mathsf{F}}\,q)$ which does not have an equivalent in LTL. *Stabilization properties*, stating a system is stabilizing if there exists a unique state that is eventually reached in all executions may be expressed using the CTL formula $\varphi = \mathsf{A}\underline{\mathsf{F}}\,(\mathsf{A}\underset{\sim}{\mathsf{G}}\,(s))$. Such formulas appear in the works [14] for modeling regulatory networks and are supported by the BMA tool [17].[6] The formula φ is different than the LTL formula $\varphi' = \underline{\mathsf{F}}\,\underset{\sim}{\mathsf{G}}\,(s)$. These two formulas represent a well-known example for properties that are not at the intersection of LTL and CTL, where φ' cannot be expressed in CTL, and φ cannot be expressed in LTL. It will be of interest to gain a better understanding of which property may be more appropriate in various biological contexts.

[5] These works use the traditional box and diamond notations for LTL.

[6] Formulas supported by the tool [17] use also quantification on variables and value domains. While this is not part of LTL it is part of PSL and its support is orthogonal to the use of truncated views.

The models developed in [17, 1] are logical models (finite state variables with deterministic logical conditions determining the transition relation) and their temporal logic semantics are described over infinite runs. A special graphic interface allows to compose the above operators into more complex formulas, ensuring that there are no problems with mismatched parenthesis, and aiming to support users that have less experience in logic or programming. While specifying temporal logic properties the users also supply an integer value that is used as a *bound* while applying bounded model checking to verify or refute the property. The supported view is the neutral view. Extending the tool to support the weak and strong views will allow the user to indicate whether unsatisfied eventualities should be forgiven or not. It would be of interest to explore support for the three views and gain experience on how they may be utilized in the process of developing biological models.

7.2 Ultimately Periodic Paths

In [52] a computational model of the mammalian cell cycle is studied using model-checking methods. The cell cycle can be viewed as a periodic mechanistic behavior that controls the process of cell division and is governed by molecular and genetic interactions, leading to the next cell division and the repetition of the cycle in the daughter cells. A logical model of some of the main genetic components is constructed in [52], extending an earlier Boolean model [31]. A translation of the model into NuSMV [15] allows using formal verification to study the dynamics of the model. One type of property that has been used is a formula of the form $[S_1, S_2 ... S_n]$ stating that there is no sequence of states that starts from an initial state satisfying some assertion S_1 and visiting a path that goes through $S_2, S_3 \cdots$ until reaching S_n. If this property does not hold, then a counterexample gives us the desired behavior of a path through states satisfying $S_1, S_2 \cdots S_n$. This property can be stated using the following LTL formula:

$$\neg(S_1 \wedge \underline{\mathsf{F}}\,(S_2 \wedge \underline{\mathsf{F}}\,(S_3 \wedge \cdots (S_{n-1} \wedge \underline{\mathsf{F}}\,S_n) \cdots)$$

To ensure the periodic behavior of the cell cycle, the sequence can be required to form a loop or a lasso, where $S_n = S_i$ for some $i < n$. We suggest that explicitly considering the ultimately periodic paths extension is of interest, allowing to decide for given $u, v \in \Sigma^*$ if $(u, v) \models^{\equiv}_{\circlearrowleft} \varphi$. Understanding what is the transient behavior (the word u) and what is the periodic behavior (the word v) may be of particular interest in biological modeling. Note that stabilizing dynamics ([17, 14] described above) is a special case of ultimately periodic behavior with the periodic part consisting of the stable state $v \in \Sigma$.

Other properties that may benefit the extension to incomplete ultimately periodic paths are oscillation properties (c.f. [14]), e.g. stating that a molecular product p oscillates ($\underset{\sim}{\mathsf{G}}\,((p \rightarrow \underline{\mathsf{F}}\,\neg p) \wedge (\neg p \rightarrow \underline{\mathsf{F}}\,p))$ in LTL).

7.3 Segmentally Broken Paths

We have shown that when reasoning about the periodic behavior of the cell cycle it is useful to identify sequences of the form $[S_1, S_2, \ldots, S_n]$ where the sequence specifies the periodic behavior. Similar sequences are also useful for time course measurements of the system that are not periodic. In particular, sometimes it is important to specify that the events occur in a sequence as above, but each S_i should not occur before its place or after its place, which also eliminates the periodic behavior. This can be specified by the formula:

$$(S_1 \wedge \underline{\mathsf{X}}\,(N\,\underline{\mathsf{U}}\,(S_2 \wedge \underline{\mathsf{X}}\,(N\,\underline{\mathsf{U}}\,(S_3 \wedge \cdots (N\,\underline{\mathsf{U}}\,(S_{n-1} \wedge \underline{\mathsf{X}}\,(N\,\underline{\mathsf{U}}\,S_n)\cdots)$$

Where $N = \neg S_1 \wedge \neg S_2 \wedge \cdots \neg S_n$ specifies the requirement that none of the propositions $S_1, S_2, \ldots, S_n$ hold. Such patterns are naturally specified in live sequence charts using existential charts [18, 41]. We suggest that the segmentally broken paths semantics presented here could allow to use the different views to reason in a uniform way about such types of formulas, where for $S_1, S_2, \ldots, S_k$ $\langle S_1, S_2, \ldots, S_k\rangle \models^{-}_{\shortparallel} \varphi$ for the weak view, and similarly for the neutral and strong views.

In [21, 53] formal reasoning and verification methods are used to synthesize Boolean Network models satisfying a set of experimental observations. The language for specifying observations currently supports constraints on bounded paths. For each experiment, a logical proposition on the value of the components of the network can be specified, where a component can be either 1 (active) or 0 (inactive). Assuming a discrete time model, the language allows specifying the timepoint at which an assertion should hold. Abstract Boolean Networks (ABNs) are introduced as a modeling formalism that incorporates uncertainty about the precise network topology and regulation mechanisms of a biological system. ABNs can be made concrete by selecting a subset of the possible interactions and making a precise choice of the regulation condition (logical template determining the transition relation give choice of possible and definite interactions) of each component.

The specification language considers reachability properties over the states of various components at different steps during executions of the system. The goal is to formalize observations obtained from different *experiments*, denoted by the set E, where each experiment $e \in E$ represents a different execution of the system. Observations are constructed using the terms (e, n, c, v), where

- $e \in E$ is the experiment label,
- $n \in 0 \ldots K$ denotes a specific time step, and
- $c \in C$ denotes a component of the ABN,
- $v \in \mathbb{B}$ represents the observed state of component c.

Let $t = q_0, \ldots, q_K$ denote a trajectory of the transition system $\mathcal{T}$ for one of the concrete models represented by an ABN $\mathcal{A}$. Trajectory t satisfies the term (e, n, c, v) if and only if $q_n(c) = v$ and we require that, for all experiments $e \in E$, there exists a trajectory t_e that satisfies all the terms labeled by e.

This can be formalized in LTL as follows:

$$\underline{\mathsf{X}}^{\,i_1}\,(S_1 \wedge \underline{\mathsf{X}}^{\,i_2-i_1}\,(S_2 \wedge \underline{\mathsf{X}}^{\,i_3-i_2}\,S_3 \wedge \cdots \underline{\mathsf{X}}^{\,i_n-i_{n-1}}\,(S_n)\cdots)$$

where we order the terms for experiment e by time step and for the i-th term (e,j,c,v) S_i is the conjunction of assertions of the form $q_j(c) = v$.

In addition, the terms $\mathit{KO}(e,c,v)$ and $\mathit{FE}(e,c,v)$ allow to define knockout and forced expression perturbations, which are assigned to a given experiment and component but are not time dependent. These perturbations modify the dynamics of the system along trajectory t_e, where component c is always active (forced expression) or inactive (knockout) when $v = \mathtt{tt}$, regardless of the regulation conditions for c or the state of each of its regulators (the update rules are applied as before when $v = \mathtt{ff}$). Finally, the constraint $\mathit{Fixpoint}(e,n)$ is used to indicate that the trajectory t_e, satisfying all constraints labeled by e, must reach a fixed point at step n. In other words, the only possible transition from the state q_n of t_e (reached at time step n) is a self-loop. Different terms (e,n,c,v), $\mathit{KO}(e,c,v)$, $\mathit{FE}(e,c,v)$ and $\mathit{Fixpoint}(e,n)$ are combined into logical expressions using the operators $\{\wedge, \vee, \Rightarrow, \Leftrightarrow, \neg\}$, which allows us to formalize various experimental observations. We suggest that sometimes the exact time points specifying when certain assertions are not precisely known, and then investigating the use of segmentally broken paths semantics in the framework described in [53] may be of interest.

8 Related Work

Other approaches for reasoning about truncated paths use multi-valued logics. For instance, Ref [9] suggests LTL_3, a 3-valued logics to reason with LTL on truncated path, as well as TLTL_3, a 3-valued logic to reason on a continuous time extension of LTL on truncated signals, and Ref [10] suggests LTL_4, a 4-valued logics to reason with LTL on truncated paths. In both the approach is that a formula φ should be evaluated to $\mathtt{tt}$ on a finite path u if all extensions of u satisfy φ and to $\mathtt{ff}$ if all extensions falsify φ. That is, a formula evaluates to $\mathtt{tt}$ under LTL_3 iff it strongly evaluates to true, and it evaluates to $\mathtt{ff}$ under LTL_3 iff its negation strongly evaluates to true. In other cases evaluation under LTL_3 results in the value $\mathtt{??}$ and under LTL_4 it results in either $\mathtt{tt}_p$ or $\mathtt{ff}_p$ according to whether the formula neutrally evaluates to $\mathtt{tt}$ or $\mathtt{ff}$.

A disadvantages of both approaches is that their definition is not inductive. Apart from that, they do not achieve the desired intuition of providing a way for the specifier to use the same set of formulas whichever verification method she is using, and only indicate for each verification method in what way the formula should be evaluated. To illustrate this, consider the case that the formula to be verified is $[p\,\underline{\mathsf{U}}\,q]$. The user would like to evaluate it neutrally when using model checking; to evaluate it weakly when using monitoring; and to evaluate it strongly on a set of test cases designed to be long enough for q to hold.

In the truncated semantics, asking to provide an answer under the weak view provides a way to *weaken the formula*, but there is no value of LTL_3 or LTL_4

that can be understood as the value of the weak view. Consider for instance the formula $\varphi = [p\,\underline{\mathsf{U}}\,(q \wedge \neg q)]$, its weak version $\varphi' = [p\,\utilde{\mathsf{W}}\,(q \wedge \neg q)]$ holds on a finite word w where all letters satisfy p, and indeed φ itself is evaluated to `tt` under the weak view, however, the semantics of φ on w under LTL_3 or LTL_4 is `ff`.[7]

Another disadvantage of LTL_3 and LTL_4 is that their monitors are of size doubly exponential in the size of the formula [10]. However, monitors for formulas under the weak view (the view usually chosen for monitoring) are only exponential in the size of the formula, since $w \not\models\!\!\!\!\!\!\!\!\!\approx\;\; \varphi$ iff w is an *informative prefix* for φ [22] and there exist an NFA recognizing all the informative prefixes of formula φ in exponential time [42]. (Loosely speaking, an *informative prefix* for the dis-satisfaction of φ on w is a prefix that can explain why φ fails on w. Some formulas, e.g. $\utilde{\mathsf{G}}\;\underline{\mathsf{F}}\,\varphi \wedge \underline{\mathsf{F}}\;\utilde{\mathsf{G}}\,\neg\varphi$ do not have informative prefixes. See [42] for precise definition and classification of safety formulas and [22] for the relation of the truncated semantics to the classification of safety formulas). [8]

A recent work [6] suggests adding a counting semantics for evaluating LTL on finite paths. The motivation has to do with formulas involving both the $\underline{\mathsf{F}}$ and $\utilde{\mathsf{G}}$ operators. Consider, for instance, the formula $\utilde{\mathsf{G}}\,(\mathit{req} \rightarrow \underline{\mathsf{F}}\,\mathit{gnt})$ and a finite trace where *req* holds on cycles 3 and 10, and *gnt* holds on cycle 8. Under LTL_3 the result will be `??`, under the weak view `tt` and under the strong view `ff`. In all cases the result disregards the fact that the first *req* was eventually followed by *gnt*, and the time elapsed between that *req* and that *gnt*. The semantics provided in [6] aims to take into account the occurrences of *req* and *gnt* in the observed path and the time elapsing between them, so that if the considered path ends less than 5 cycles after the second *req* then the result would be $\mathtt{tt}_p$ corresponding to *presumably true* and meeting the intuition that the next *gnt* is likely to occur on cycle 15 (since 5 cycles is the time elapsed between the first *req* and the following *gnt*) whereas if the considered path ends more than 5 cycles after the last *req*, then the result would be $\mathtt{ff}_p$ corresponding to *presumably false*, meeting the intuition that we should have already observed the *gnt* following the second *req*. We disagree that the result $\mathtt{ff}_p$ is always the expected result in the second case, and that $\mathtt{tt}_p$ is always the expected result in the first case. Indeed, it can be that the periodic part is longer than a single sequence involving one *req* and one *gnt* (e.g. the periodic part may involve three *req*s with different times elapsing between their corresponding *gnt*s) or it could be that we have not yet observed the periodic part. The semantics proposed in Section 4 (and its generalization presented in Section 6) lets the user determine which part of the observed path correspond to the transient behavior and which to the periodic part. And she

[7] Approaches that extend the set of atomic propositions with a dedicated *end* symbol, and require formulas to explicitily reason about it (e.g. the approach implemented in the tool RuleR [5]) suffer similar drawbacks.

[8] Ref [10] claims some disadvantages of the three views approach to the truncated semantics, but the analysis distinguishes the weak and strong views as if they are different semantics while they are not (they are part of the same semantics with negation switching roles) and requires the *next* operator to be a dual of itself, while the semantics is designed so that the *strong next* is dual to the *weak next*, and strong propositions are dual to weak propositions.

can do that using a combination of her knowledge on the system, her intuitions and her desire for strong/weak guarantees. For instance, in the given example, she can determine that the path starting at cycle 3 and ending in cycle 8 is the periodic part, and using the neutral ultimately periodic approach get the same result as in [6]. Or, as another example, she can determine that the periodic part started at cycle 3 and still continues, thus using the weak ultimately periodic approach she will get a positive result even if the path is longer than 15. Another drawback of the semantics proposed in [6], as observed therein, is that it cannot semantically characterize tautologies and contradiction (and, for instance, the evaluation of the formula $(p \wedge \neg p)$ on the empty word results in `??` whereas it should have result in `ff`, since the formula is a contradiction).

We refer the reader to [6] for a discussion on other related work.

9 Discussion

We have proposed an extension of the three views approach to reason over truncated paths from LTL to STL, and generalized reasoning over truncated paths to reasoning over incomplete ultimately periodic paths, segmentally broken paths and combinations thereof. The extensions are provided in both the discrete dimension (LTL) and the continuous dimension (STL).

The standard temporal logics PSL and SVA have regular expressions based formulas, and most formulas are written using regular expressions. The regular expression-based operators increase the expressive power of the logic from star-free ω-regular languages to the full ω-regular languages. But the main reason they are popular is because they are easier to use. The ease of use consideration is important for answering the demand for the specification to be intuitive for experimental biologists. The three views approach is defined for PSL, and thus for the regular expression-based operators as well. Timed regular expressions were defined in [4]. It would be interesting to extend timed regular expressions to consider truncated paths and more generally segmentally broken ultimately periodic paths.

We have reviewed use of temporal reasoning in system biology, and conclude that it is plausible that the proposed extensions may be useful in modeling and analyzing biological systems. We foresee that it will be useful in many other applications as well.

References

1. Zara Ahmed, David Benque, Sergey Berezin, Anna Caroline E Dahl, Jasmin Fisher, Benjamin A Hall, Samin Ishtiaq, Jay Nanavati, Nir Piterman, Maik Riechert, et al. Bringing ltl model checking to biologists. In *International Conference on Verification, Model Checking, and Abstract Interpretation*, pages 1–13. Springer, 2017.
2. Uri Alon. *An introduction to systems biology: design principles of biological circuits*. CRC press, 2006.
3. Rajeev Alur, Tomás Feder, and Thomas A. Henzinger. The benefits of relaxing punctuality. *J. ACM*, 43(1):116–146, 1996.

4. Eugene Asarin, Paul Caspi, and Oded Maler. Timed regular expressions. *J. ACM*, 49(2):172–206, 2002.
5. Howard Barringer, David Rydeheard, and Klaus Havelund. Rule systems for runtime monitoring: From eagle to ruler. In Oleg Sokolsky and Serdar Taşıran, editors, *Runtime Verification*, pages 111–125, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
6. E. Bartocci, R. Bloem, D. Nickovic, and F. Roeck.
7. Ezio Bartocci and Pietro Lió. Computational modeling, formal analysis, and tools for systems biology. *PLoS computational biology*, 12(1):e1004591, 2016.
8. G. Batt, D. Ropers, H.de Jong, J. Geiselmann, R. Mateescu, M. Page, and D. Schneider. Validation of qualitative models of genetic regulatory networks by model checking: analysis of the nutritional stress response in *escherichia coli*. *Bioinformatics*, 21:19–28, 2005.
9. Andreas Bauer, Martin Leucker, and Christian Schallhart. Monitoring of real-time properties. In *FSTTCS 2006: Foundations of Software Technology and Theoretical Computer Science, 26th International Conference, Kolkata, India, December 13-15, 2006, Proceedings*, pages 260–272, 2006.
10. Andreas Bauer, Martin Leucker, and Christian Schallhart. Comparing LTL semantics for runtime verification. *J. Log. Comput.*, 20(3):651–674, 2010.
11. S. Brenner. Sequences and consequences. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365(1537):207–212, 2010.
12. Eduard Cerny, Surrendra Dudani, John Havlicek, and Dmitry Korchemny. *SVA: The Power of Assertions in SystemVerilog.* Series on Integrated Circuits and Systems. Springer, 2012.
13. N. Chabrier and F. Fages. Symbolic model checking of biochemical networks. In *Proceedings of the First Intl. Workshop on Computational Methods in Systems Biology*, Lecture Notes in Computer Science, pages 149–162. Springer, 2003.
14. N. Chabrier-Rivier, M. Chiaverini, V. Danos, F. Fages, and V. Schächter. Modeling and querying biomolecular interaction networks. *Theoretical Computer Science*, 325(1):25–44, 2004.
15. Alessandro Cimatti, Edmund Clarke, Fausto Giunchiglia, and Marco Roveri. Nusmv: a new symbolic model checker. *International Journal on Software Tools for Technology Transfer*, 2(4):410–425, 2000.
16. Manuel Clavel, Francisco Durán, Steven Eker, Patrick Lincoln, Narciso Martí-Oliet, José Meseguer, and Carolyn Talcott. *All about maude-a high-performance logical framework: how to specify, program and verify systems in rewriting logic.* Springer-Verlag, 2007.
17. B. Cook, J. Fisher, E. Krepska, and N. Piterman. Proving stabilization of biological systems. In *Proceedings of the 12th international conference on Verification, model checking, and abstract interpretation*, pages 134–149. Springer, 2011.
18. W. Damm and D. Harel. LSCs: Breathing life into message sequence charts. *Formal Methods in System Design*, 19(1):45–80, 2001. Preliminary version appeared in Proc. 3rd IFIP Int. Conf. on Formal Methods for Open Object-Based Distributed Systems (FMOODS'99).
19. Jyotirmoy V. Deshmukh, Alexandre Donzé, Shromona Ghosh, Xiaoqing Jin, Garvit Juniwal, and Sanjit A. Seshia. Robust online monitoring of signal temporal logic. *Formal Methods in System Design*, 51(1):5–30, Aug 2017.
20. E. Dubrova, M. Teslenko, and L. Ming. Finding attractors in synchronous multiple-valued networks using SAT-based bounded model checking. In *40th IEEE Intl. Symposium on Multiple-Valued Logic (ISMVL)*, pages 144–149, 2010.

21. Sara-Jane Dunn, G. Martello, B. Yordanov, S. Emmott, and A.G. Smith. Defining an essential transcription factor program for naïve pluripotency. *Science*, 344(6188):1156–1160, 2014.
22. C. Eisner, D. Fisman, J. Havlicek, Y. Lustig, A. McIsaac, and D. Van Campenhout. Reasoning with temporal logic on truncated paths. In *W. Hunt Jr and F. Somenzi, editors* Proc. 15^{th} Intl. Conference on Computer Aided Verification (CAV'03), Boulder, CO, USA, July 2003. To appear.
23. Cindy Eisner and Dana Fisman. *A Practical Introduction to PSL*. Series on Integrated Circuits and Systems. Springer, 2006.
24. Cindy Eisner, Dana Fisman, and John Havlicek. A topological characterization of weakness. In *Proceedings of the Twenty-Fourth Annual ACM Symposium on Principles of Distributed Computing, PODC 2005, Las Vegas, NV, USA, July 17-20, 2005*, pages 1–8, 2005.
25. Cindy Eisner, Dana Fisman, and John Havlicek. Safety and liveness, weakness and strength, and the underlying topological relations. *ACM Trans. Comput. Log.*, 15(2):13:1–13:44, 2014.
26. Cindy Eisner, Dana Fisman, John Havlicek, Yoad Lustig, Anthony McIsaac, and David Van Campenhout. Reasoning with temporal logic on truncated paths. In *Computer Aided Verification, 15th International Conference, CAV 2003, Boulder, CO, USA, July 8-12, 2003, Proceedings*, pages 27–39, 2003.
27. Cindy Eisner, Dana Fisman, John Havlicek, and Johan Mårtensson. The $\top, \bot$ approach for truncated semantics.
28. Steven Eker, Merrill Knapp, Keith Laderoute, Patrick Lincoln, José Meseguer, and Kemal Sonmez. Pathway logic: Symbolic analysis of biological signaling. In *Biocomputing 2002*, pages 400–412. World Scientific, 2001.
29. Steven Eker, Merrill Knapp, Keith Laderoute, Patrick Lincoln, and Carolyn Talcott. Pathway logic: Executable models of biological networks. *Electronic Notes in Theoretical Computer Science*, 71:144–161, 2004.
30. Steven Eker, José Meseguer, and Ambarish Sridharanarayanan. The maude ltl model checker. *Electronic Notes in Theoretical Computer Science*, 71:162–187, 2004.
31. A. Fauré, A. Naldi, C. Chaouiya, and D. Thieffry. Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. *Bioinformatics*, 22(14):124–131, 2006.
32. Dana Fisman. On the characterization of until as a fixed point under clocked semantics. In *Hardware and Software: Verification and Testing, Third International Haifa Verification Conference, HVC 2007, Haifa, Israel, October 23-25, 2007, Proceedings*, pages 19–33, 2007.
33. Carlo A. Furia and Matteo Rossi. No need to be strict: On the expressiveness of metric temporal logics with (non-)strict operators. *Bulletin of the EATCS*, 92:150–160, 2007.
34. Carlo A. Furia and Matteo Rossi. On the expressiveness of MTL variants over dense time. In *Formal Modeling and Analysis of Timed Systems, 5th International Conference, FORMATS 2007, Salzburg, Austria, October 3-5, 2007, Proceedings*, pages 163–178, 2007.
35. Angelo Gargantini and Angelo Morzenti. Automated deductive requirements analysis of critical systems. *ACM Trans. Softw. Eng. Methodol.*, 10(3):255–307, July 2001.
36. David Harel, Hagi Lachover, Amnon Naamad, Amir Pnueli, Michal Politi, Rivi Sherman, and Aharon Shtull-Trauring. Statemate; A working environment for

the development of complex reactive systems. In *Proceedings, 10th International Conference on Software Engineering, Singapore, Singapore, April 11-15, 1988.*, pages 396–406, 1988.
37. IBM, RuleBase SixthSense Edition. https://www.research.ibm.com/haifa/projects/verification/Formal_Methods-Home/index.shtml.
38. IEEE Standard for Property Specification Language (PSL), Annex B. IEEE Std 1850$^{\mathrm{TM}}$.
39. IEEE Standard for SystemVerilog – Unified Hardware Design, Specification, and Verification Language, Annex F. IEEE Std 1800$^{\mathrm{TM}}$.
40. Ron Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2(4):255–299, Nov 1990.
41. H. Kugler, D. Harel, A. Pnueli, Y. Lu, and Y. Bontemps. Temporal Logic for Scenario-Based Specifications. In *TACAS*, volume 3440 of *Lecture Notes in Computer Science*, pages 445–460. Springer, 2005.
42. Orna Kupferman and Moshe Y. Vardi. Model checking of safety properties. In *Computer Aided Verification, 11th International Conference, CAV '99, Trento, Italy, July 6-10, 1999, Proceedings*, pages 172–183, 1999.
43. N. Le Novère. Quantitative and logic modelling of molecular and gene networks. *Nature Reviews Genetics*, 16(3):146–158, 2015.
44. Oded Maler and Dejan Nickovic. Monitoring properties of analog and mixed-signal circuits. *STTT*, 15(3):247–268, 2013.
45. Zohar Manna and Amir Pnueli. *The temporal logic of reactive and concurrent systems - specification.* Springer, 1992.
46. Pedro T Monteiro, Delphine Ropers, Radu Mateescu, Ana T Freitas, and Hidde De Jong. Temporal logic patterns for querying dynamic models of cellular interaction networks. *Bioinformatics*, 24(16):i227–i233, 2008.
47. B. Palsson. *Systems Biology: Constraint-based Reconstruction and Analysis.* Cambridge University Press, 2015.
48. Amir Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977*, pages 46–57, 1977.
49. Amir Pnueli. Linear and branching structures in the semantics and logics of reactive systems. In *Automata, Languages and Programming, 12th Colloquium, Nafplion, Greece, July 15-19, 1985, Proceedings*, pages 15–32, 1985.
50. Synopsys, VC Formal Tools. https://www.synopsys.com/verification/static-and-formal-verification/vc-formal.html.
51. A. Tiwari, C. Talcott, M. Knapp, P. Lincoln, and K. Laderoute. Analyzing pathways using SAT-based approaches. In *Proceedings of the 2nd international conference on Algebraic biology*, Lecture Notes in Computer Science, pages 155–169. Springer, 2007.
52. Pauline Traynard, Adrien Fauré, François Fages, and Denis Thieffry. Logical model specification aided by model-checking techniques: application to the mammalian cell cycle regulation. *Bioinformatics*, 32(17):i772–i780, 2016.
53. B. Yordanov, Sara-Jane Dunn, H. Kugler, A. Smith, G. Martello, and S. Emmott. A method to identify and analyze biological programs through automated reasoning. *npj Systems Biology and Applications*, 2(16010), 2016.