How to Say ’You Are Tired’ in Temporal Logic

Kristina Gogoladze*
UU, UMCU
k.gogoladze@uu.nl

We outline a proposal to support surgeons during robot-assisted surgery. Our approach is based on run-time monitoring for properties that may indicate that the surgeon is tired or experiencing other difficulties, as well as checking that the distance to anatomical structures, speed etc. variable values are within safe limits.

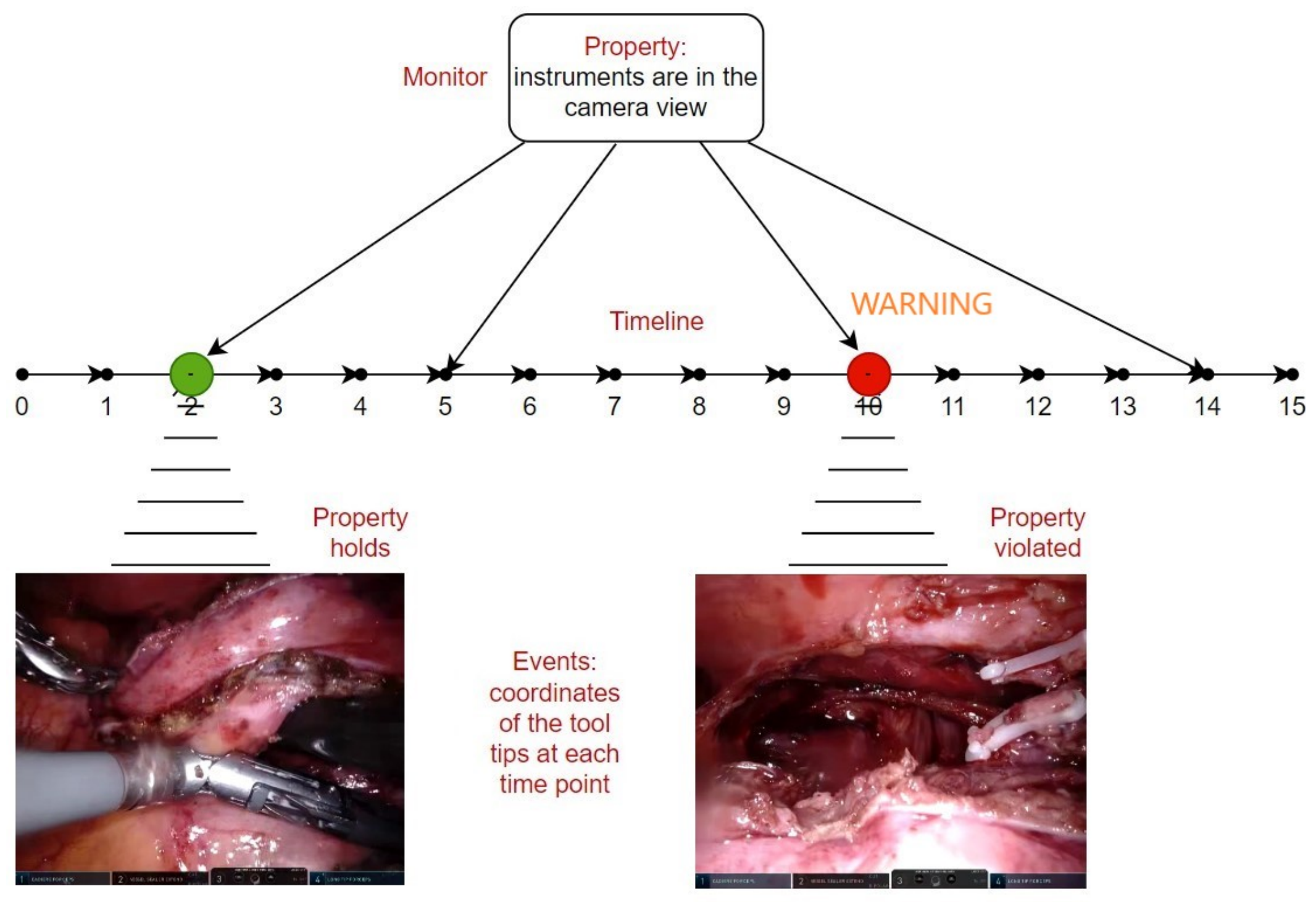
Robot-assisted surgery has several advantages over traditional, open, surgery, including improved dexterity, tremor filtering, three-dimensional magnified vision, and smaller incisions. However, robot-assisted surgery is complex and requires considerable training to become proficient. Even trained surgeons may still make mistakes or come close to making a mistake, such as touching or damaging anatomical structures not involved in the operation.

This is the first step in a project to improve patient outcomes by supporting surgeons during robot-assisted surgery. We aim to develop a run-time verification system that warns a surgeon if the robot manipulators are getting too close to, e.g., the aorta, or that the pressure exerted is too high, or the movement too fast or uneven.

Run-time Verification

Run-time verification (RV) is an approach to ensuring the correct operation of software, hardware, or cyber-physical systems. Correctness is checked with respect to properties stated in a formal language. These properties are typically safety properties (something ”bad” should not happen). RV uses a computer program called a monitor that receives a stream of events from the system being monitored, and processes them to check whether a property is violated. The monitor is usually implemented as an automaton, and safety properties are specified in a temporal logic: a formal language to describe patterns on flows of time corresponding to traces of executions of the system.

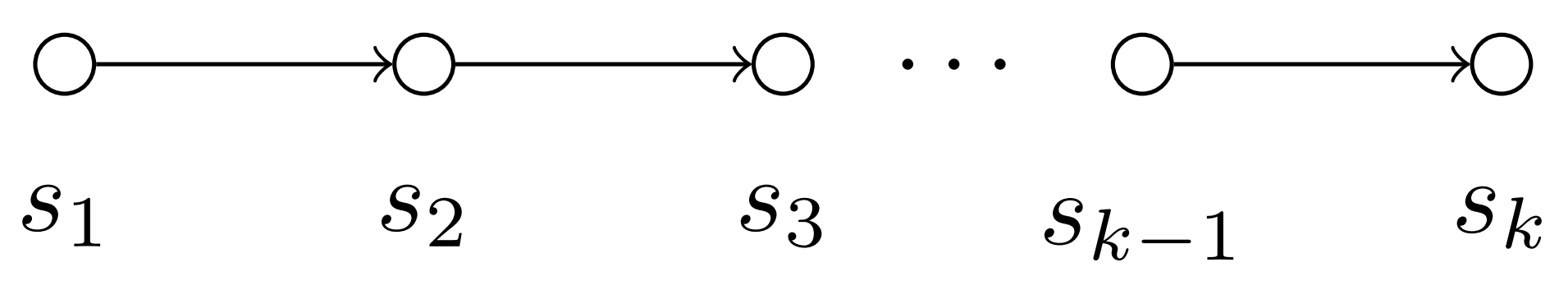
RV has been successfully used in a wide range of applications, including software systems, avionics, and autonomous space robotics. However, run-time verification of robot-assisted surgery has not previously been attempted, and involves significant challenges, including: the complexity of the behaviour (of the robot manipulated by a human); the complexity of the environment (soft tissues in a human body, which move with every breath and heartbeat, and change during surgical dissection); and the need to come up with a specification language that is sufficiently expressive to allow the statement of medically relevant properties while at the same time admitting efficient verification. We focus on the case study of Robot-Assisted Minimally Invasive Esophagectomy (RAMIE), but the approach will be applicable to other surgical procedures.



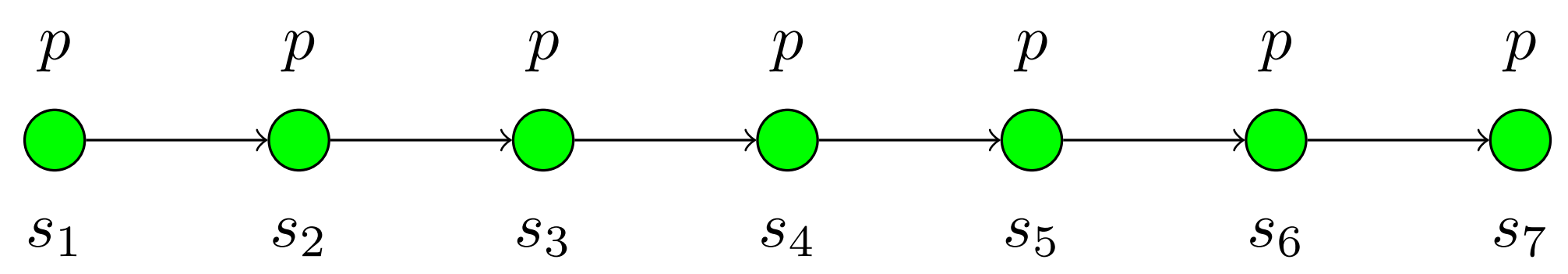
Mission-time Linear Temporal Logic

In run-time verification, properties to monitor are often expressed in Linear Temporal Logic (LTL). This formal language is intended for describing constraints on runs of the system (sequences of states of the system). It can say that some statement φ holds in the Next state, that φ holds Globally (in every state) and that some statement ψ holds until φ becomes true (ψ Until φ).

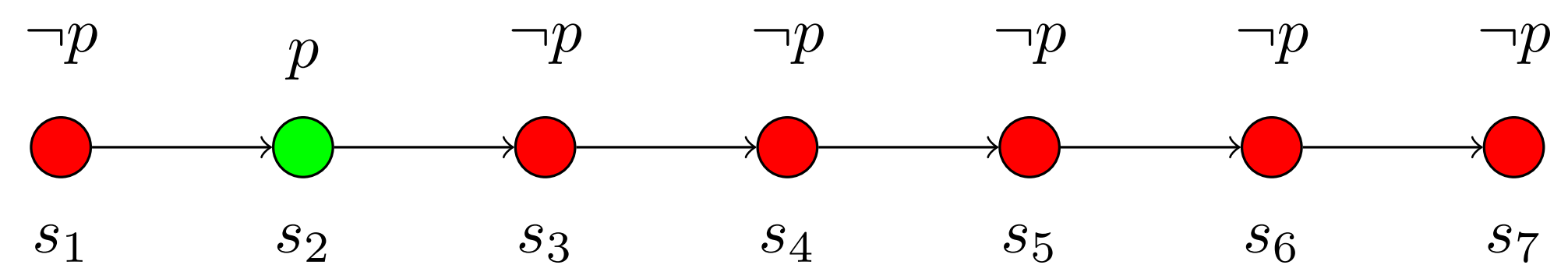
A system execution is formally a simple *linear trace* — a sequence of events that capture a behaviour of the monitored system we are interested in to analyse. This is a finite trace where S_i is the system's state at the execution step i.



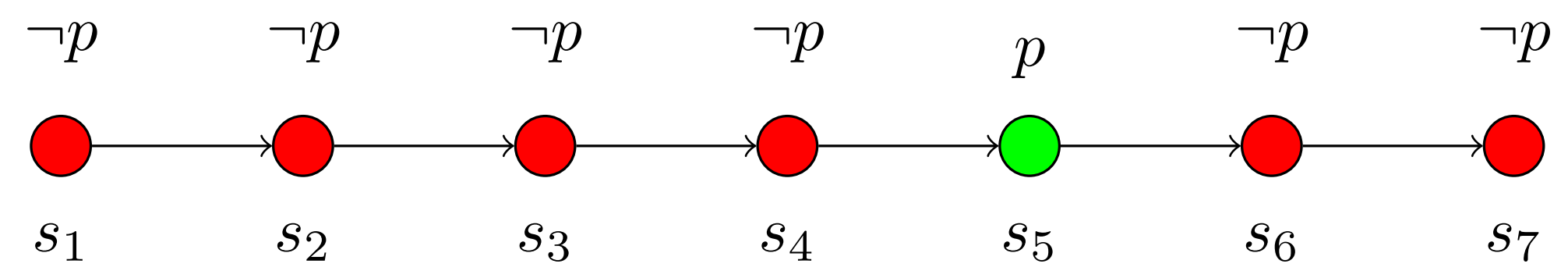
Here are some examples of the meanings of LTL operators. This figure shows the interpretation of $\Box p$ (Globally p), which means ’property p holds at every state on the path’.



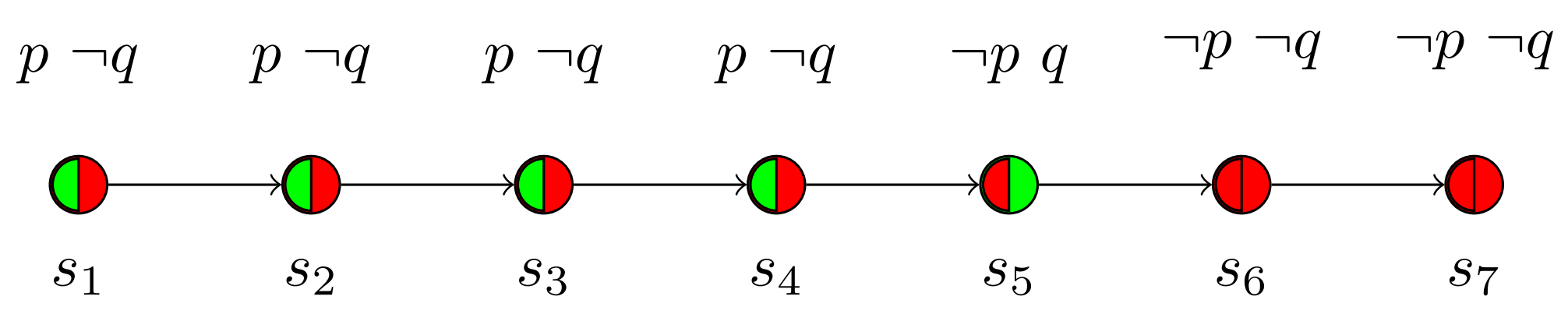
This is an illustration of the interpretation of $\mathcal{X}p$ (Next p), which means ’property p holds at the second state on the path’ (if we are at state 1).



This figure depicts the interpretation of $\Diamond p$ (Eventually p), which means ’property p holds at some state on the path’ (after a finite number of states).



And, finally, an illustration of the interpretation of $p\mathcal{U}q$ (p Until q), this formula holds if eventually q holds, and at every state before that p holds.



Mission-time linear temporal logic(MLTL) is formally defined as follows. A (closed) interval over the naturals $I = [a, b]$ (where $0 \leq a \leq b$ are natural numbers) is a set of naturals $\{i \mid a \leq i \leq b\}$. I is called *bounded* iff $b < +\infty$; otherwise I is *unbounded*. MLTL is defined using bounded intervals. Unlike Metric Temporal Logic, it is not necessary to introduce open or half-open intervals over the natural domain, as every open or half-open bounded interval is reducible to an equivalent closed bounded interval, e.g., $(1, 2) = \emptyset$, $(1, 3) = [2, 2]$, $(1, 3] = [2, 3]$, etc. Let $\mathcal{AP}$ be a set of atomic propositions, then the syntax of a formula in MLTL is

$$\phi ::= true \mid false \mid p \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi\mathcal{U}_I\psi$$

where I is a bounded interval, $p \in \mathcal{AP}$ is an *atom*, and ϕ and ψ are subformulas.

The semantics of MLTL formulas is interpreted over finite traces bounded by intervals. Let π be a finite trace in which every position $\pi[i]$ ($i \geq 0$) is over $2^{\mathcal{AP}}$, and $|\pi|$ denotes the length of π ($|\pi| < +\infty$ when π is a finite trace). Then π_i ($|\pi| > i \geq 0$) represents the suffix of π starting from position i (including i).

For $a, b \in \mathbb{I}$, $a \leq b$, the notion of a trace π satisfies an MLTL formula ϕ, denoted as $\pi \models \phi$, is defined as follows:

$\pi \models p$ iff $p \in \pi[0]$;

$\pi \models \neg\phi$ iff $\pi \not\models \phi$;

$\pi \models \phi \wedge \psi$ iff $\pi \models \phi$ and $\pi \models \psi$;

$\pi \models \phi\mathcal{U}_{[a,b]}\psi$ iff $|\pi| > a$ and, there exists $i \in [a, b]$, $i < |\pi|$ such that $\pi_i \models \psi$ and for every $j \in [a, b]$, $j < i$ it holds that $\pi_j \models \phi$;

Expressing Properties in MLTL

An example of a “safe distance to the anatomy” property is ‘the distance to aorta should always be at least d’. This can be expressed in MLTL as:

$$\Box_{[0,t]}(distance > d)$$

where t is the maximal duration of the surgery (note that natural numbers in MLTL can be interpreted over different units of time in the same formula, so t could, for example, be 10 hours). Restricting to the maximal duration of the surgery can be expressed in standard LTL. However it may be useful to restrict safe distance properties to different phases in the surgery, defined either using temporal bounds or qualitatively.

A similar property is monitoring for too high speed of the tool tip:

$$\Box_{[0,t]}(speedToolTip < s)$$

This property can also be expressed using LTL, but it is useful to be able to specify the intervals when different speed restrictions apply.

An additional challenge is in formally defining properties which are to do with human behaviour rather than the values of physical characteristics of the robotic tools (speed, position, etc.). For example, properties such as the surgeon's movements being too hesitant, or movements indicating that the surgeon is tired. These types of warnings are considered by surgeons to be very useful, and their timing is less critical than the ‘distance to aorta’ type of properties. However as far as we know, there is no work on formal run-time monitoring for this kind of property. For example, one way of specifying formally the property that ‘the surgeon's movement is hesitant’, could be stating that within a specified interval, the speed of the tool tip should not be above s, below s, and then above it again:

$$\Box_{[0,t]}\neg((speedTool > s) \wedge \mathcal{X}(speedTool < s) \wedge \mathcal{X}\mathcal{X}(speedTool > s))$$

Similarly properties suggesting that the surgeon is slowing down or is making more aggressive movements can be specified. The specification of such properties may be specific to a particular surgeon.

Which Properties?

- closeness to anatomical structures (distance between the instrument and the anatomical structure should never drop below the safety distance)
- hesitant movements (tool tip should not, be moving, then stop/pause moving and start moving again within 1 second)
- too slow or too fast movements (speed of the tool tip must always be within specified limits)
- suturing in the wrong direction etc.
- instruments not in camera view (coordinates of the tool tip must be within the coordinates of the camera view rectangle)
- instruments touching (getting signals from the capacitors)

References

[1] Pnueli, A.: The temporal logic of programs. In: Proceedings of the 18th Annual Symposium on Foundations of Computer Science (SFCS-77). pp. 46–57. IEEE (1977)

[2] Reinbacher, T., Rozier, K.Y., Schumann, J.: Temporal-logic based runtime observer pairs for system health management of real-time systems. In: Ábrahám, E., Havelund, K. (eds.) Tools and Algorithms for the Construction and Analysis of Systems. pp. 357–372. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)

[3] Intuitive Surgical Inc.: Robotic Surgical Systems — Da Vinci — Ion — Intuitive `https://www.intuitive.com/en-us/patients/da-vinci-robotic-surgery` (Accessed on 4/9/2023)

[4] Ruurda, J.P., van der Sluis, P.C., van der Horst, S., van Hilllegersberg, R.: Robot-assisted minimally invasive esophagectomy for esophageal cancer: A systematic review. J Surg Oncol. 112, 257–265 (2015).

[5] Macrander, Y.: Monitoring da Vinci, Safety Managed by a Robot. A study into Runtime Verification and Monitoring Systems for Robot-Assisted Surgery. Master's thesis, Utrecht University (2022)

*Supervisors: Natasha Alechina and Jelle Ruurda

Funded by NWO project OCENW.M.21.377