Journal of Applied and Computational Topology (2025) 9:17
https://doi.org/10.1007/s41468-025-00217-9

Topology-aware planning under linear temporal logic constraints

Dan P. Guralnik[1] · **Yu Wang**[1] · **Warren E. Dixon**[1]

Received: 12 February 2025 / Revised: 14 June 2025 / Accepted: 17 June 2025

Abstract

Symbolic planning methods for continuous state spaces have traditionally relied on model-checking techniques being applied to a discrete model of the space in question. Such models are usually obtained as dual graphs to tilings of the state space by contractible regions (finite polytopes, usually), converting the planning problem into a graph search problem. The inherently high computational complexity of these methods motivates considering discretizations that are more frugally constructed, while retaining all the pertinent topological information about the state space. Moreover, Farber's theory of topological complexity of continuous planning favors the requirement that the homotopy types of the state space and its models coincide. The Nerve Lemma indicates it may be possible to obtain the desired models of a state space as particular sub-complexes of the barycentric subdivision of the nerve $\mathcal{N}(\mathbb{U})$ of a good open cover $\mathbb{U}$ indexed by the symbols. This article develops the basic theory required for conducting symbolic planning over models obtained in this way. The obstructions to deploying the model-checking paradigm for path-planning over an open cover $\mathbb{U}$ are identified and characterized, resulting in a model $\mathcal{N}_{red}(\mathbb{U})$ called the reduced nerve of the cover, and an algorithm for solving LTL-planning problems is presented. Furthermore, in the case of a good cover, it is shown that all the vertices of the complement of $\mathcal{N}_{red}(\mathbb{U})$ may be deleted from the subdivided nerve without altering its homotopy type.

Keywords Temporal logic based planning · Continuous planning · Homotopy type · Nerve complex · Triangulated space · Discretization · Transition system

Dan P. Guralnik, Yu Wang, and Warren E. Dixon all authors contributed equally to this work.

✉ Dan P. Guralnik
danguralnik@ufl.edu

Yu Wang
yuwang1@ufl.edu

Warren E. Dixon
wdixon@ufl.edu

1 Department of Mechanical & Aerospace Engineering, University of Florida, P.O.Box 116250, Gainesville, Florida 32611, United States

Published online: 23 June 2025

1 Introduction

Planning for a broad range of tasks, rather than specific ones, has long been a goal in robotics and machine intelligence (Koditschek 1992, 2021; Kress Gazit et al. 2018; Lamnabhi Lagarrigue et al. 2017). The implicit, though crucial, role played by topology in determining the complexities of robotic planning and control problems has been moving to the forefront of research in recent years. For example, a homotopy invariant $TC(X) \in \mathbb{Z}_{\geq 0}$ of a topological space X called topological complexity was introduced in Farber (2003) to serve as a measure of the complexity of point-to-point planning. The space X is of minimal complexity, $TC(X) = 0$, if and only if a continuous planner exists for X, if and only if X is contractible. Similarly, topology of the state space impacts the quality of guarantees for reactive navigation (Koditschek and Rimon 1990; Rimon and Koditschek 1992), and, more generally state-feedback control, beginning with seminal results such as Brockett's obstruction (Brockett 1983), and culminating with the recent works (Baryshnikov 2023; Kvalheim 2023), including modern hybrid approaches (Montgomery and Sanfelice 2024). Topological insights inform many computational tools for geometric planning. For instance, topological abstractions are used to decipher and simplify the shapes of configuration spaces while preserving essential properties (Pokorny et al. 2014; Marinakis and Dudek 2010). Furthermore, topological invariants are used to develop algorithms that classify and compute optimal solutions across diverse topologies in configuration spaces (Bhattacharya et al. 2013; Mavrogiannis et al. 2023).

At the same time, topological considerations in the management of composite tasks expressed through logical formulae remain less studied despite multiple recent examples of real-world applications in areas such as unmanned aerial vehicle coordination (Karaman and Frazzoli 2011), traffic regulation (Maierhofer et al. 2022), sequential robotic manipulation (Wells et al. 2021), and others. Planning problems for tasks expressed in temporal logic can be complex, requiring policies that consider historical actions. Existing literature often necessitates user intervention to decompose these tasks into sequences of simpler tasks, which can be computationally expensive and, in general, not scalable.

Among temporal logic formalisms, Linear Temporal Logic (LTL) is more extensively applied due to its expressive capabilities and relatively low computational demands. LTL extends Boolean logic by incorporating the temporal connectives "next" ($\bigcirc$) and "until" (U). These connectives extend the foundational principles of Boolean logic to articulate changes in system state over discrete linear time. The inclusion of these temporal connectives enables the expression of diverse time-related planning objectives. For example, given atomic propositions α and β representing regions in configuration space, the simple expression $\alpha \mathsf{U} \beta$ states the requirement that the agent confine its state to the region α until the region β is reached. Infinite-horizon objectives are also expressible in LTL, such as ensuring the recurrence of a desired set of semantic states.

LTL was introduced as a high-level language for expressing general planning objectives for robots (Fainekos et al. 2005b, 2009; Kress Gazit et al. 2009) based on a discrete abstraction of the configuration space. Planning methods were then generalized from discrete to continuous spaces using hybrid approaches combining high-level discrete planning with low-level continuous planning (Fainekos et al. 2005a), or by directly operating in the continuous configuration space (Karaman et al. 2008; Belta and Sadraddini 2019). More recently, planning objectives have been extended from LTL to other temporal logics, such as cLTL (Sahin et al. 2019) and HyperLTL (Wang and Pajic 2020). However, all the aforementioned works assume the configuration space is a regular Euclidean domain, and do not account for its topology. Thus, planning for LTL in more general topological spaces remains an open problem, which we address in this paper.

1.1 Generalizing the LTL planning paradigm

A particularly attractive feature of LTL is the representability of LTL formulae φ by finite automata $\mathcal{A}_\varphi$, which enables the solution of discrete planning problems stated in terms of LTL-based constraints. The states of the automata $\mathcal{A}_\varphi$ represent historical information, while their transitions represent reasoning over that information given incoming observations. The sensory inputs of the system are modeled as Boolean labels $L = \{l_\alpha : X \to \{\top, \bot\}\}_{\alpha \in \mathrm{AP}}$ represented by a finite set of atomic propositions AP, so a transition of $\mathcal{A}_\varphi$ occurring at a state $x \in X$ is labeled with an observation

$$\varsigma(x) \triangleq \{\alpha \in \mathrm{AP}\colon l_\alpha(x) = \top\}\,.$$

When the system state X is discrete, the state-of-the-art solution to the problem of planning a path satisfying φ is to search the extension of X by $\mathcal{A}_\varphi$, often referred to as the product automaton, for a valid directed path consisting of an initial segment ending at an accepting state (transient behavior), followed by a loop (recurrent behavior). It is natural to ask whether such searches could be supplanted by simpler computations in the presence of additional structural information about the system. In particular, in the case of X being a tame enough topological space, we ask what properties one should minimally require of the sets $\mathbb{U}(\alpha) \triangleq l_\alpha^{-1}(\top)$ to make use of available information about topological invariants of X, for the purpose of avoiding a fruitless search?

Our paradigm captures a general class of planning problems. The symbols $\alpha \in \mathrm{AP}$ may be used to express a wide variety of semantic requirements and constraints and the space X itself does not need to be an instance of some specific physical environment. For example, the dynamical states of a mechanical system $\mathcal{S}$ are often described as lying in the tangent bundle $\pi : TE \twoheadrightarrow E$ over a manifold E which is itself a principal $\mathbb{R}^N$-bundle $p : E \twoheadrightarrow B$ over a "shape space" B, which may have non-trivial topology.[1] If X is such a space (i.e., $X = TE$ in the example), a symbol α may correspond to a collection of shape configurations of $\mathcal{S}$ expressing a mechanical constraint on the interaction between components of $\mathcal{S}$; or a dynamic state relating the velocities of different system components to their relative positions and orientations. In the context of a particular task, any such symbol or a Boolean combination thereof may be specified as acceptable ("safe"), desirable (a goal to be reached), or to be avoided ("unsafe"), where each of these roles may depend on historical context within

[1] E.g., for an n-link planar arm, B is a domain in the n-torus; if the arm is not anchored, then E is an $\mathbb{R}^2$-principal bundle over the n-torus B.

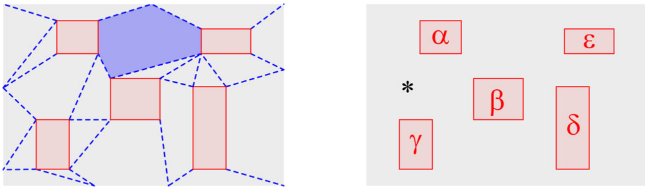

Fig. 1 The two prevalent methods of modeling continuous state spaces in the LTL-based planning literature. The tiling model (left), where each atomic proposition corresponds to the system state being contained in a specific tile (blue/red/grey); and the hub model (right), which is a simplified setup with a single 'hub' tile (grey) connecting all the others (red). Note how the tiling model may be used to refine the hub model

the current mission. Here, the word "safe" is most often associated in the literature with system states which, given the current history and state, must be avoided for fear of damaging the system or incurring some other unacceptable cost, such as hurting a person.

A crucial characteristic of the LTL context is, therefore, that robotic tasks in continuous spaces are formulated in terms of discrete sensory inputs (e.g., is a certain landmark nearby? Are any of the robot's legs touching the ground? Is the robot walking on concrete or on sand?), whereas the configuration spaces of robotic systems are described in terms of the complex interactions between robot morphology and environment geometry. Bringing these two different viewpoints of a mission—the discrete and the continuous—to a common denominator is a necessary step, but the approaches to doing so differ. Direct computational methods such as mixed integer programming (Shoukry et al. 2017) are known to be computationally expensive. To mitigate this complexity, the configuration space is often abstracted into a discrete graph (c.f. Kress Gazit et al. 2009; Fainekos et al. 2009), where nodes represent system states and edges represent transitions between states. Some efforts have relied on heuristic partitioning methods using zonotopes (Ren et al. 2021) or more general polytopes (Tokuda et al. 2021). However, the specific partitioning methods were primarily geared towards simplifying local computations rather than towards accounting for the global topology of the configuration space, resulting in only semi-decidable algorithms: in other words, these algorithms may not always terminate to give feasible paths.

Two ideas about partitioning prevail in the literature, whose idealized descriptions (the actual implementations vary) may be summarized as follows (see Figure 1).

- **"Tiling model".** The connected configuration space X is expressed as being tiled by the $\{\mathbb{U}(\alpha)\}_{\alpha\in\mathrm{AP}}$ (e.g. in Fainekos et al. (2009), Kress Gazit et al. (2009)). That is, $\mathbb{U}(\alpha)$ is open and non-empty for all $\alpha \in \mathrm{AP}$, and $\mathbb{U}(\alpha) \cap \mathbb{U}(\beta) = \varnothing$ whenever $\alpha \neq \beta$ (tiles are interiorly disjoint), but, at the same time, $X = \bigcup_{\alpha\in\mathrm{AP}} \overline{\mathbb{U}(\alpha)}$ (the closures of the tiles cover everything). The individual tiles $\mathbb{U}(\alpha)$ are required to be path-connected (usually contractible, e.g., convex polytopes). Each pair with $\overline{\mathbb{U}(\alpha)} \cap \overline{\mathbb{U}(\beta)} \neq \varnothing$ corresponds to an edge of a "connectivity graph" with vertex set AP. The resulting graph is used to construct a discrete transition system, whose transitions correspond to controllers effecting motions between neighboring tiles 'along' the edges of the graph. Notably, the underlying semantics are highly simplified: at any time, the system state either satisfies at most one atomic proposition, or a small number thereof (depending on the actual implementation and combinatorics of the tiling), the latter case only occurring instantaneously (or nearly so) during transitions, with the origin and goal tiles being known in advance.

- **"Hub model".** This is a special case of a tiling model, where one has $* \notin \mathrm{AP}$ as an additional symbol, and open sets $\mathbb{U}(\alpha)$, $\alpha \in \mathrm{AP} \cup \{*\}$ forming a tiling in the preceding sense, but with the additional property that the edges in the connectivity graph of the tiling are precisely those of the form $\{\alpha, *\}$. The hub model often arises as a simplification of the tiling model, where less consequential tiles are unified into a single region labeled by $*$. In fact, works like (Fainekos et al. 2009; Kress Gazit et al. 2009) combine the use of the tiling model for navigation with the use of the simplified hub model for planning. Note that the set $\mathbb{U}(*)$ both separates between any two of the $\mathbb{U}(\alpha)$, $\alpha \in \mathrm{AP}$, and serves as a conduit between them, while not being contractible in general (Fig. 1, right, for example). A majority of TL-based planning applications, both in discrete and continuous settings, are of this form, necessitating a dedicated solution for the navigation problem in $\mathbb{U}(*)$.

Since a mission specification may not be provided in terms of labels conforming to either model, the first step in the TL-based planning pipeline is to refine the labels (further subdivide X) so that the original semantic labels of the mission could be expressed in terms of a tiling—or hub—model. Any LTL formula specifying the original mission is then rewritten in terms of the new, much larger, alphabet, incurring major added computational costs stemming from the generally doubly exponential complexity of subsequent planning, regarded as a function of the size of the alphabet. Nevertheless, interactions between the refined labels are kept to a minimum, which is expressible completely in term of the known connectivity graph of the tiling. If the tiles are simple enough (in practice they are often convex polytopes, with the exception of $\mathbb{U}(*)$), then controllers/planners may be constructed to execute motion within and between tiles, reducing LTL-based planning over X to LTL-based planning over the connectivity graph of the tiling. Thus, avoiding the computational costs of seemingly convenient ad-hoc expansions of the mission specification alphabet becomes a priority.

The main contribution of the tiling approaches lies with the encoding of the continuous space X as the dual graph of the tiling. Therefore, one expects that more compact—yet equally informative—discrete representations of X could be obtained if the tiling is replaced with an appropriate cover of X representing a coarser subdivision of the original alphabet (and therefore incurring a lower computational cost). If possible, such a construction would have the added benefit of making use of topological information, which is obtainable at a much lower cost in the pre-processing stage of the mission (e.g., the presence of known cut-sets in X that could be inferred from homological computations), for optimizations in the planning stage, such as enabling faster rejection of infeasible mission specifications possessing topological obstructions. The central question becomes:

Question 1.1 *Which covers represent the connectivity properties of X with sufficient precision to address the complexities of path planning? Given an appropriate cover, how should a discretization of X be constructed?*

Another motivating principle for the present work is that continuous behaviors are desirable in robotic contexts. For example, if the task is to reach a particular configuration $x^* \in X$, then one would like the path taken by the robot to x^* to depend continuously on the initial condition $x_0 \in X$, as well as on x^*. This preference was considered by Farber in his seminal work (Farber 2003), where the topological complexity of path planning was introduced. Recall that if $\mathbf{P}(X)$ is the space of continuous paths $c\colon [0,1] \to X$ with the uniform topology and $\pi : \mathbf{P}(X) \to X \times X$ is the end-point map $c \mapsto (c(0), c(1))$, then a continuous path planner is a continuous section $s : X \times X \to \mathbf{P}(X)$ of π, that is: $\pi \circ s = \mathrm{id}_{X \times X}$. Farber's work departs from the observation that X admits a continuous path planner if and only if X is contractible—a result that may be seen as analogous to the realization that a point attractor of a continuous vector field has a contractible basin. The topological complexity $\mathrm{TC}(X)$ is then defined as the smallest $m \in \mathbb{Z}_{\geq 0}$ such that $X \times X$ admits an open cover $\{U_0, \ldots, U_m\}$ with continuous partial sections $s_i : U_i \to \mathbf{P}(X)$ such that $\pi \circ s_i = \mathrm{id}_{U_i}$ for all $i = 0, \ldots, m$, and is shown to be a homotopy invariant of X.

Absent a direct linkage between topological complexity and LTL-based planning, but seeking to retain information about the homotopy type of X, it seems sensible to require that the 'patches' $\mathbb{U}(\alpha)$ corresponding to the labels $l_\alpha \in L$ be open and contractible. For sufficiently tame spaces, where the inclusion map of any point is a cofibration[2], the task of reaching a prescribed point may then be regarded as being equivalent to that of reaching a prescribed contractible neighborhood of that point. While this consideration resonates well with the tiling model (especially when the tiles happen to be convex polytopes), here it is also proposed to take advantage of possible intersections of the $\mathbb{U}(\alpha)$ rather than refine them to a tiling, in the interest of avoiding the increase in complexity incurred by refinement, and trying to obtain complexity savings from computable topological invariants. If the open sets $\mathbb{U}(\alpha)$, $\alpha \in \mathrm{AP}$ are allowed to form a cover, dropping the more restrictive requirement that they form a tiling, then a classical result called the nerve lemma (Kozlov 2008) may be invoked to conclude that the abstract simplicial complex $\mathcal{N}(\mathbb{U}) = \{\sigma \subset \mathrm{AP}\colon \bigcap_{\alpha\in\sigma} \mathbb{U}(\alpha) \neq \varnothing\}$, known as the nerve of the indexed cover $\mathbb{U}$, is homotopy-equivalent to X (and therefore has the same topological complexity), provided all non-empty intersections of the $\mathbb{U}(\alpha)$ are contractible. As a result, $\mathcal{N}(\mathbb{U})$ may seem acceptable as a discretization of X over which the LTL-based planning paradigm described above could be run (instead of over the connectivity graph of a tiling), but in a "topology-aware" way, with a potential of forming new insights into the connections between topological complexity of continuous planning in X and the computational complexity of LTL-based planning.

It must be stressed at this point that this work does not come to address the question of how to obtain good covers of a space X, but, rather, on how to make use of such covers when an LTL-based task specification over X is given in terms of such a cover. Problems, such as constructing a minimal good cover from a given cellular decomposition of a space, lie outside the scope of this paper.

[2] Such as manifolds and simplicial complexes, for example, or, more generally, locally finite cellular complexes, and counting out local pathologies exhibited by spaces such as the Hawaiian earring.

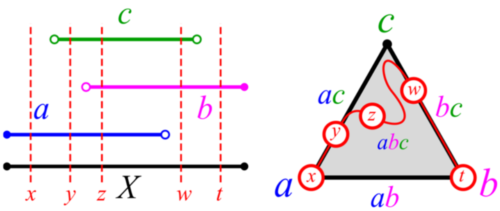

Fig. 2 The interval $X = [0, 1]$ covered by three intervals indexed by the set $\mathrm{AP} = \{a, b, c\}$ (left) yields the 2-simplex as its nerve, however, only the simplices $\{a\}$, $\{b\}$, $\{a, c\}$, $\{b, c\}$ and $\{a, b, c\}$ are realized by the cover (right). Any partition of unity $\{\psi_\alpha\}_{\alpha\in\mathrm{AP}}$ satisfying $\psi_\alpha^{-1}(0) = X \smallsetminus \mathbb{U}(\alpha)$ results in a map $\Psi : X \to |\mathcal{N}(\mathbb{U})|$, $\Psi(x) = \sum_\alpha \psi_\alpha(x)\mathbf{e}_\alpha$, such as the one depicted here in red, with its image missing the unrealized simplices $\{c\}$ and $\{a, b\}$

1.2 Results

The first contribution of this work is the observation that $\mathcal{N}(\mathbb{U})$ is almost always an inadequate candidate for the discretization of X sought in Question 1.1. A key reason for this is that many simplices of $\mathcal{N}(\mathbb{U})$ may be unrealized, in the following sense:

Definition 1.2.1 A simplex $\sigma \in \mathcal{N}(\mathbb{U})$ is $\mathbb{U}$-realized if $\sigma = \varsigma(x)$ for some $x \in X$. □

Considering unrealized simplices shows that $\mathcal{N}(\mathbb{U})$ is rarely an adequate model for path planning. The following example will be the running example throughout the discussion of unrealized simplices in this article, presenting a simple instance of having unrealized simplices.

Example 1.2.2 Let $X = [0, 1]$, covered by three intervals, $\mathbb{U}(a) \triangleq [0, 1-2\varepsilon)$, $\mathbb{U}(b) = (2\varepsilon, 1]$, and $\mathbb{U}(c) = (\varepsilon, 1-\varepsilon)$, for some $\varepsilon \in (0, \frac{1}{5})$. Then $\mathbb{U}$ is a good cover of X, and its nerve is the full 2-simplex with vertices a, b, c (to simplify notation in examples, we will identify simplices, e.g. $\{a, b, c\}$, with strings, e.g. resp., abc). Note that not all the simplices in $\mathcal{N}(\mathbb{U})$ are $\mathbb{U}$-realized, as discussed in detail in Figure 2.

The upshot of Example 1.2.2 is that unrealized simplices form an obstruction to path-planning in X using the nerve $\mathcal{N}(\mathbb{U})$. Example 1.2.2 demonstrates additional obstructions: the edge ab, regarded as a path from the vertex a to the vertex b, does not correspond to any continuous path in the space X in the sense that there exists no continuous path in $[0, 1]$ from a point satisfying $a \wedge \neg b \wedge \neg c$ to a point satisfying $\neg a \wedge b \wedge \neg c$ via points satisfying $a \wedge b \wedge \neg c$. Even subtler obstructions are present. For example, note how no path in X induces a direct transition between the vertex a and the simplex $\{a, b, c\}$. In summary, we conclude it is necessary to replace $\mathcal{N}(\mathbb{U})$ with a more refined model space, from which the $\mathbb{U}$-unrealized simplices of $\mathcal{N}(\mathbb{U})$ have been removed, and whose combinatorial paths represent paths in X. A natural environment within which to attempt such a removal meaningfully is the barycentric subdivision $\mathrm{sd}(\mathcal{N}(\mathbb{U}))$ of the nerve.

The second contribution of this article is the characterization of a sub-complex $\mathcal{N}_{red}(\mathbb{U})$ of the barycentric subdivision $\mathrm{sd}(\mathcal{N}(\mathbb{U}))$ of $\mathcal{N}(\mathbb{U})$, which, under appropriate conditions on the cover $\mathbb{U}$, has the property that any combinatorial path in the 1-skeleton of $\mathcal{N}_{red}(\mathbb{U})$ is induced from a path in X. The relevant conditions are hinted at in the following example.

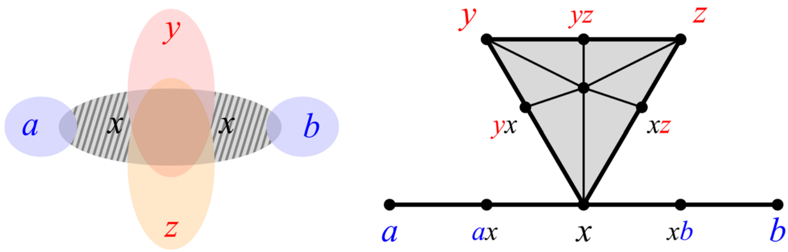

Fig. 3 A disconnected witness set. A topological disk X is covered by convex open ellipsoidal regions marked a, b, y, z, x, which together constitute a good cover $\mathbb{U}$ of X (left). The 0-simplex $\{x\}$ (ellipsoid at center) is a realized simplex of $\mathcal{N}(\mathbb{U})$, but the set of points $\varsigma^{-1}(\{x\})$ realizing it in X is disconnected (shaded). Though a combinatorial path exists in $\mathrm{sd}(\mathcal{N}(\mathbb{U}))$ (right) joining $\{a\}$ to $\{b\}$ via $\{x\}$, while avoiding any simplices containing y or z, there is no path in X capable of avoiding the region $y \cup z$

Example 1.2.3 Figure 3 depicts a good open cover $\mathbb{U}$ of a topological disk X, where all simplices of the nerve are realized, but, at the same time, paths exist in $\mathrm{sd}(\mathcal{N}(\mathbb{U}))$ that are not induced from paths in X. This problem is precisely analogous to the one identified in Pappas et al. (2000) as the obstruction to the deployment of hierarchical controllers. The path of simplices (a, ax, x, xb, b) in $\mathrm{sd}(\mathcal{N}(\mathbb{U}))$ does not describe the behavior of a path in X, because the connected component of $\varsigma^{-1}(\{x\})$ reachable from $\varsigma^{-1}(ax)$ is not reachable from $\varsigma^{-1}(xb)$, as is plainly visible in Fig. 3.

A precise criterion for avoiding the problem presented in Example 1.2.3 is stated in combinatorial terms in Theorem 6.3.4. As a corollary, one obtains that the path-connectedness of all witness sets (of realized simplices of the nerve) is sufficient for the path correspondence to hold (Corollary 6.3.5). The final contribution of this paper is the following partial result:

Theorem 1.2.4 *Suppose $\mathbb{U}$ is a good indexed open cover of a space X and let $K = \mathcal{N}(\mathbb{U})$. Then, the sub-complex $N[0]$ of $N = sd(K)$ obtained by deleting all the vertices corresponding to unrealized simplices[3] is a strong deformation retract of N. In particular, $N[0]$ has the homotopy type of X.*

Note how the reduced nerve $\mathcal{N}_{red}(\mathbb{U})$ may be obtained from $N = \mathrm{sd}(\mathcal{N}(\mathbb{U}))$ by successive deletions, first of the unrealized vertices, resulting in $N[0]$, and then of higher-order unrealized simplices, in order of increasing dimension. Hence, Theorem 1.2.4 provides an intermediate construction interpolating between the reduced nerve, for which a planning result is available, and the nerve, for which a result on homotopy type reconstruction is available. This intermediate space, $N[0]$, is a solution to the basic realization problem, though, unlike the reduced nerve, it falls short of solving its higher-order extension. Further work is required to obtain a strong deformation retraction of $\mathrm{sd}(\mathcal{N}(\mathbb{U}))$ onto the reduced nerve under reasonable conditions on $\mathbb{U}$ that could be guaranteed in the design stage.

To close, the results of this paper imply that execution of the standard LTL-based planning paradigm over the 1-skeleton of the reduced nerve is equivalent to LTL-based path planning in X without requiring a refining of the atomic labels into a tiling,

[3] When deleting a vertex, all the simplices containing that vertex are deleted as well.

while possibly retaining information about higher homotopy-theoretic properties of X (which are fully retained in the intermediate discrete model provided by Theorem 1.2.4). This capability comes at a computational price: it is necessary to both ensure in the design stage that the cover $\mathbb{U}$ is good and that the conditions for the path correspondence between X and $\mathcal{N}_{red}(\mathbb{U})$ are met, and to compute the reduced nerve in preparation for planning. Nevertheless, the required computations consume no more resources than the process of refining the planning problem into a tiling with special geometric properties, while being of minimal complexity given the open cover $\mathbb{U}$.

This article is organized as follows. Section 2 recalls the necessary preliminaries from topology, such as simplicial complexes, maps and their geometric realizations, nerves of covers, and some homotopy-theoretic methods such as simplicial collapse, with the aim of introducing some new useful computations and generalized collapses in Sects. 2.4.2 and 2.5.2. Section 3 formally discusses the problem of LTL-based planning using the nerve of a good open cover, introducing the appropriate semantics and presenting the formal desiderata for the $\mathcal{N}_{red}(\mathbb{U})$ construction. A product automaton method for LTL-based planning over the reduced nerve is presented in Sect. 4, and the technical work required to characterize $\mathcal{N}_{red}(\mathbb{U})$ and derive its properties, such as proving Theorem 1.2.4, is presented in Sect. 5. The proof of the path correspondence is discussed in Sect. 6.

2 Topology preliminaries

This section is intended as a reference for the rest of this article, recalling several of the classical constructions underpinning combinatorial topology and establishing notational conventions that will be used throughout. In addition, two technical results are presented for which we did not find satisfactory accounts in the literature: a detailed description of links in barycentric subdivisions (Sect. 2.4.2), and the preservation of homotopy type under generalized collapses (Sect. 2.5.2). The reader familiar with topological techniques may want to skip to the next section, revisiting this one as needed.

2.1 Nice topological spaces

All topological spaces, referred to as $X, Y, Z, \ldots$, are assumed to be II-countable, normal, Hausdorff, locally compact, locally connected, and locally contractible, to avoid well-known pathologies. Note that a locally contractible space is locally path-connected. Recall that two maps $f, g\colon X \to Y$ are said to be homotopic (denoted by $f \dot{\simeq} g$), if there exists a map $H : X \times [0, 1] \to X$, called a homotopy from f to g, with the property that $H(x, 0) = f(x)$ and $H(x, 1) = g(x)$ for all $x \in X$. The space X is said to be contractible if id_X is homotopic to a constant map.

Roughly, the local connectivity assumptions on a space X provide that the connected components of an open subspace U are open[4] in X, and coincide with the path components of U. The countability and separation assumptions, on the other hand, guarantee that X has many continuous real-valued functions on it: every open set U may be written as $X \smallsetminus Z(f)$, where $Z(f)$ is the set of zeroes of a continuous function $f : X \to [0, 1]$.

The word ‘space’ shall henceforth refer to spaces satisfying all the above assumptions. Primary examples are compact manifolds and finite simplicial complexes.

2.2 Simplicial complexes

Recall that an (abstract) simplicial complex with vertex set $V \neq \varnothing$ is a collection K of finite subsets of V that is closed under taking subsets, that is: if $\sigma \in K$ and $\tau \subseteq \sigma$, then $\tau \in K$. The sets $\sigma \in K$ are referred to as the (abstract) simplices of K. An abstract simplex $\sigma \in K$ is said to be of dimension $d \in \mathbb{Z}$, or a d-simplex, if $|\sigma| = d + 1$. The set of simplices of dimension less than or equal to d is referred to as the d-skeleton of K and denoted by $\mathrm{sk}_d(K)$. Note that $\mathrm{sk}_0(K) = V \cup \{\varnothing\}$, $\mathrm{sk}_{-1}(K) = \{\varnothing\}$, and $\mathrm{sk}_d(K) = \varnothing$ for all $d < -1$. The 0-simplices of K (i.e., the elements of V) are also referred to as the vertices of K. For any collection of simplices $A \subset K$, we denote by $\downarrow(A)$ the collection of all $\tau \subseteq K$ contained in some element of A. A subset L of a simplicial complex K is said to be a sub-complex of K, if L is itself a simplicial complex.

Example 2.2.1 Let V be a non-empty finite set. The simplex with vertex set V, denoted by $\mathfrak{B}^V$ and referred to as the V-simplex, for short, is the simplicial complex containing all the subsets of V. The sub-complex of $\mathfrak{B}^V$ obtained by removing the simplex V is denoted by $\mathfrak{S}^V$. It will be referred to as the hollow V-simplex.

2.2.1 Maps, special sub-complexes, constructions

If K_1 and K_2 are simplicial complexes on vertex sets V_1 and V_2, then a function $f : V_1 \to V_2$ is said to be a simplicial map of K_1 to K_2 if $f(\sigma) \in K_2$ holds for all $\sigma \in K_1$. An isomorphism of simplicial complexes is a bijective simplicial map whose inverse is simplicial.

For a simplex σ of a simplicial complex K, the (closed) star $\mathrm{st}_K(\sigma)$ of σ in K is the sub-complex of all simplices $\tau \in K$ such that $\sigma \cup \tau \in K$. The link $\mathrm{lk}_K(\sigma)$ of σ in K is the sub-complex of all τ such that $\tau \cap \sigma = \varnothing$ and $\tau \cup \sigma \in K$. The open star $\mathrm{st}^\circ_K(\sigma)$ is defined as the set of all $\tau \in K$ with $\sigma \subseteq \tau$ (it is not a sub-complex). Thus, the operation $\tau \mapsto \tau \smallsetminus \sigma$ is a bijective and order-preserving[5] map of $\mathrm{st}^\circ_K(\sigma)$ onto $\mathrm{lk}_K(\sigma)$. The inverse of this map will henceforth be denoted by $\tau \mapsto \sigma + \tau$. For subsets $A \subset \mathrm{lk}_K(\sigma)$, the notation $\sigma + A$ refers to the set $\{\sigma + \tau : \tau \in A\}$.

[4] In fact, since the connected components of a subspace are always relatively closed, those of U are clopen in U. Here, it is the conclusion regarding the path components of U coinciding with its components that matters most in the context of this work.

[5] Unless otherwise stated, all mention of order in simplicial complexes refer to their ordering with respect to inclusion, also known as the face poset order.

The join $K_1 * K_2$ of two simplicial complexes is the simplicial complex consisting of all sets[6] $\sigma_1 \sqcup \sigma_2$ such that $\sigma_1 \in K_1$ and $\sigma_2 \in K_2$. For example, $\mathfrak{B}^{V_1} * \mathfrak{B}^{V_2} = \mathfrak{B}^{V_1 \sqcup V_2}$. For another example, if K is a simplicial complex and σ is any simplex of K, then the closed star $\mathrm{st}_K(\sigma)$ is isomorphic to the join $\mathrm{lk}_K(\sigma) * \mathfrak{B}^\sigma$ via the simplicial map taking any vertex v of $\mathrm{st}_K(\sigma)$ to itself, noting that any vertex of $\mathrm{st}_K(\sigma)$ does not lie in σ if and only if it is a vertex of $\mathrm{lk}_K(\sigma)$.

2.2.2 Geometric realization

The geometric realization of a simplicial complex K on a vertex set V is the topological space $|K| \subset \mathbb{P}(V)$,

$$\mathbb{P}(V) \triangleq \left\{\xi \in [0, \infty)^V : \sum_{v \in V} \xi(v) = 1\right\}, \tag{1}$$

constructed as

$$|K| \triangleq \bigcup_{\sigma \in K} \dot{\Delta}^\sigma(V) = \bigcup_{\sigma \in K} \Delta^\sigma(V), \tag{2}$$

where

$$\dot{\Delta}^\sigma(V) \triangleq \{\xi \in \mathbb{P}(V) : (\forall_{v \in V})(v \in \sigma \leftrightarrow \xi(v) > 0)\}, \tag{3}$$

$$\Delta^\sigma(V) \triangleq \{\xi \in \mathbb{P}(V) : (\forall_{v \in V})(v \notin \sigma \rightarrow \xi(v) = 0)\}, \tag{4}$$

are the open and closed (geometric) simplices of $|K|$, respectively. We shall generally omit any mention of V, using $\dot{\Delta}^\sigma$ and Δ^σ instead of $\dot{\Delta}^\sigma(V)$ and $\Delta^\sigma(V)$, respectively, when there is little risk of ambiguity.

Example 2.2.2 Let V be a finite set. Then, $\left|\mathfrak{B}^V\right| = \Delta^V$ is the standard Euclidean simplex in $\mathbb{R}^V$, and $\left|\mathfrak{S}^V\right| = \Delta^V \setminus \dot{\Delta}^V$ is the standard hollow simplex.

Note how K is in one-to-one correspondence with the open simplices $\dot{\Delta}^\sigma$ of $|K|$: every point $\xi \in |K|$ lies in exactly one open simplex, and distinct open simplices are pairwise disjoint. For each $v \in V$, the open simplex $\dot{\Delta}^{\{v\}}$ contains only a single point, the vector $\mathbf{e}_v \in \mathbb{P}(V)$ defined by

$$\mathbf{e}_v(w) \triangleq \begin{cases} 1, & \text{if } w = v, \\ 0, & \text{if } w \neq v. \end{cases} \tag{5}$$

The simplex Δ^σ is then the convex hull of its vertices $\{\mathbf{e}_v : v \in \sigma\}$. The barycenter of $\dot{\Delta}^\sigma$ is a convenient representative of σ in the disjoint decomposition of $|K|$ into open simplices:

$$\zeta_\sigma \triangleq \tfrac{1}{|V|} \sum_{v \in \sigma} \mathbf{e}_v. \tag{6}$$

We shall abuse notation by denoting $|A| \triangleq \bigcup_{\sigma \in A} \dot{\Delta}^\sigma$ for any subset $A \subset K$. For example, observe that the set $\left|\mathrm{st}_K^\circ(\sigma)\right| \subseteq |K|$ constitutes a connected open neighborhood of ζ_σ in $|K|$ (hence the name, "open star").

[6] The symbol $\sqcup$ shall henceforth denote the disjoint union (co-product) operator on sets.

Geometric realization is a functor, in the sense that any simplicial map $f : K \to L$ gives rise to a continuous map $|f| : |K| \to |L|$ given by

$$|f|\left(\sum_{v\in\sigma} \xi_v \mathbf{e}_v\right) \triangleq \sum_{v\in\sigma} \xi_v \mathbf{e}_{f(v)}, \tag{7}$$

for all $\sigma \in K$ and $\xi \in \mathbb{P}(\sigma)$. The assignment $f \mapsto |f|$ is functorial, that is: $|\mathrm{id}_K| = \mathrm{id}_{|K|}$ for any simplicial complex K, and $|g \circ f| = |g| \circ |f|$ for any pair of composable simplicial maps f, g.

2.3 Indexed covers and the nerve construction

The prime example of a simplicial complex in this paper is that of the nerve of an open cover. This section considers discretizations of a topological space $(X, \mathscr{T})$ arising from locally finite open covers.

Definition 2.3.1 (Admissible Cover). Let AP be a nonempty set. An admissible cover of $(X, \mathscr{T})$ over AP is a map $\mathbb{U} : \mathrm{AP} \to \mathscr{T}$ such that the sets

$$\varsigma(x) \triangleq \{\alpha \in \mathrm{AP} \colon x \in \mathbb{U}(\alpha)\} \tag{8}$$

are non-empty and finite for all $x \in X$. Dually, for any $\sigma \subseteq \mathrm{AP}$, the witness set of σ in X is defined as

$$\widetilde{\mathbb{U}}(\sigma) \triangleq \bigcap_{\alpha\in\sigma} \mathbb{U}(\alpha). \tag{9}$$

The set σ is said to be consistent, if it has a witness—that is, if $\widetilde{\mathbb{U}}(\sigma)$ is non-empty. □

Note that the collection $\{\mathbb{U}(\alpha)\}_{\alpha\in\mathrm{AP}}$ is a locally finite open cover of X if $\mathbb{U}$ is an admissible cover of X over AP. Many of the ideas developed in this paper apply to infinite covers, but we will restrict attention to finite sets AP henceforth.

Definition 2.3.2 (Good Cover). An admissible cover $\mathbb{U}$ of X over AP is said to be good, if the witness set of each consistent $\sigma \subseteq \mathrm{AP}$ is contractible. □

Using the language just established, the nerve $\mathcal{N}(\mathbb{U})$ is defined as the simplicial complex of all consistent subsets of AP:

$$\mathcal{N}(\mathbb{U}) \triangleq \left\{\sigma \subseteq \mathrm{AP} \colon \widetilde{\mathbb{U}}(\sigma) \neq \varnothing\right\}. \tag{10}$$

The following result about $\mathcal{N}(\mathbb{U})$ is classical.

Lemma 2.3.3 (Nerve Lemma, Kozlov (2008, Theorem 15.21)). *If $\mathbb{U}$ is a good cover of X, then $|\mathcal{N}(\mathbb{U})|$ is homotopy-equivalent to X.* □

Proofs of the nerve lemma depend on the notion of a partition of unity to construct a map from X to the geometric realization of $\mathcal{N}(\mathbb{U})$. Under the admissibility assumptions considered in this article the following notion is natural.

Definition 2.3.4 (Partition of Unity Representing a Cover). Let $\mathbb{U}$ be an indexed open cover of X over AP. A partition of unity representing $\mathbb{U}$ is a collection $\Psi \triangleq \{\psi_\alpha : X \to [0,1]\}_{\alpha \in \mathrm{AP}}$ of continuous functions such that $\mathbb{U}(\alpha) = X \smallsetminus Z(\psi_\alpha)$ for all $\alpha \in \mathrm{AP}$, and $\sum_{\alpha \in \mathrm{AP}} \psi_\alpha(x) = 1$ for all $x \in X$. □

Since $\mathbb{U}$ is locally finite, a partition of unity representing $\mathbb{U}$ may be obtained from an arbitrary collection of continuous functions $\varphi_\alpha : X \to [0, \infty)$ satisfying $\mathbb{U}(\alpha) = X \smallsetminus Z(\varphi_\alpha)$ for all α by setting $\psi_\alpha \triangleq \varphi_\alpha \left(\sum_{\alpha \in \mathrm{AP}} \varphi_\alpha\right)^{-1}$. Every partition of unity Ψ representing $\mathbb{U}$ gives rise to a continuous map of X to the geometric realization of $\mathcal{N}(\mathbb{U})$:

Lemma 2.3.5 *Let $\mathbb{U}$ be an indexed open cover of X over AP and let Ψ be a partition of unity representing $\mathbb{U}$. Then, the map $\boldsymbol{\rho}_\Psi : X \to \Delta^{AP}$ defined by $\boldsymbol{\rho}_\Psi(x) \triangleq \sum_{\alpha \in AP} \psi_\alpha(x)\boldsymbol{e}_\alpha$ is continuous and satisfies*

$$\boldsymbol{\rho}_\Psi(x) \in \dot{\Delta}^{\varsigma(x)} \tag{11}$$

for all $x \in X$. In particular, $\boldsymbol{\rho}_\Psi(X) \subseteq |\mathcal{N}(\mathbb{U})|$. □

Proofs of the nerve lemma rely, in essence, on showing that the map $\boldsymbol{\rho}_\Psi$ is a homotopy equivalence.

2.4 Barycentric subdivision

For any simplicial complex K with vertex set V, the barycentric subdivision $\mathrm{sd}(K)$ is the simplicial complex of all finite chains of the poset $(K, \subseteq)$. In other words, $A \in \mathrm{sd}(K)$ if and only if $A \subset K$, A is finite, and, for any $\sigma, \tau \in A$, either $\sigma \subseteq \tau$ or $\tau \subseteq \sigma$. In particular, the vertices of $\mathrm{sd}(K)$ are the elements of K.

At the level of geometric realizations, $|\mathrm{sd}(K)| \subset \mathbb{P}(K) \subset \mathbb{R}^K$ is canonically homeomorphic to $|K| \subset \mathbb{P}(V) \subset \mathbb{R}^V$ via the map $|\mathrm{sd}|_K : |\mathrm{sd}(K)| \to |K|$ obtained by linear extension from the assignment $\mathbf{e}_\sigma \mapsto \zeta_\sigma$.

2.4.1 Functorial properties

Barycentric subdivision is also a functor, as follows. A simplicial map $f : K \to L$ gives rise to a simplicial map $\mathrm{sd}(f) : \mathrm{sd}(K) \to \mathrm{sd}(L)$ defined by $\mathrm{sd}(f)(\sigma) \triangleq f(\sigma)$ for all $\sigma \in \mathrm{sd}(K)^0 = K$. It is then clear that $\mathrm{sd}(\mathrm{id}_K) = \mathrm{id}_{\mathrm{sd}(K)}$ for any simplicial complex K. Furthermore, $\mathrm{sd}(f \circ g) = \mathrm{sd}(f) \circ \mathrm{sd}(g)$ holds for any pair of composable simplicial maps f, g.

As a simple example, consider a sub-complex $L < K$. Then, the inclusion map $\iota : L \hookrightarrow K$ induces embeddings $|\iota| : |L| \hookrightarrow |K|$ and $|\mathrm{sd}(\iota)| : |\mathrm{sd}(L)| \hookrightarrow |\mathrm{sd}(K)|$, respectively. We will henceforth identify the images of these embeddings in $|K|$ and $|\mathrm{sd}(K)|$ with $|L|$ and $|\mathrm{sd}(L)|$, respectively. Finally, observe that $|\mathrm{sd}|_K (|\mathrm{sd}(L)|) = |L|$ for all $L < K$.

2.4.2 Computing links in the barycentric subdivision.

The following is a computation of links in a barycentric subdivision of a complex K, instrumental in the derivation of Theorem 1.2.4.

Lemma 2.4.1 *Let K be a simplicial complex on a vertex set V and let $S = \{\sigma_0 \subset \ldots \subset \sigma_d\}$, $d \geq 0$ be a simplex of $sd(K)$. For $i = 0, \ldots, d$, let $\tau_i = \sigma_i \smallsetminus \sigma_{i-1}$, where $\sigma_{-1} \triangleq \varnothing$ and $\sigma_{d+1} \triangleq V$. Then, the map*

$$g_S \colon lk_{sd(K)}(S) \to sd(lk_K(\sigma_d)) * sd(\mathfrak{S}^{\tau_0}) * \cdots * sd(\mathfrak{S}^{\tau_d}) \tag{12}$$

defined on the vertices of $lk_{sd(K)}(S)$ by setting $g_S(\sigma) \triangleq \sigma \smallsetminus \sigma_{i-1}$ if and only if $\sigma_{i-1} \subset \sigma \subset \sigma_i$ (for $i = 0, \ldots, d+1$) is a simplicial isomorphism.

Proof First, recall that $\sigma \in K$ is a vertex of $L \triangleq \mathrm{lk}_{\mathrm{sd}(K)}(S)$ if and only if $\sigma \notin S$ and $S \cup \{\sigma\}$ is a chain. Therefore, the map g_S is well-defined as a map of vertex sets of its domain and range complexes. A chain $T \in \mathrm{sd}(K)$ belongs in L if and only if $T \cap S = \varnothing$ and $T \cup S$ is a chain. This requirement is equivalent to T being the disjoint union of chains T_i, $i = 0, \ldots, d+1$, such that each $\tau \in T_i$ satisfies $\sigma_{i-1} \subset \tau \subset \sigma_i$, for every i. Equivalently, the chain $g_S(T_i)$ must lie in $\mathrm{sd}(\mathfrak{S}^{\tau_i})$ for all $i = 0, \ldots, d$, and the chain $g_S(T_{d+1})$ must lie in $\mathrm{sd}(\mathrm{lk}_K(\sigma_d))$. Thus, the map g_S is indeed simplicial with simplicial inverse. □

Corollary 2.4.2 *Let K be a simplicial complex. For all $\sigma \in K$,*

$$lk_{sd(K)}(\{\sigma\}) \cong sd(lk_K(\sigma)) * sd(\mathfrak{S}^{\sigma}).$$

Proof Simply apply Lemma 2.4.1 with $d = 0$, $\tau_0 = \sigma_0 = \sigma$. □

2.5 Deletions and collapses

Recall that a continuous map $f : X \to Y$ is a homotopy equivalence if there exists $g : Y \to X$ such that $g \circ f \dot{\simeq} \mathrm{id}_X$ and $f \circ g \dot{\simeq} \mathrm{id}_Y$. The spaces X and Y are then said to have the same homotopy type. A closed subspace A is said to be a deformation retract of X if there is a retraction $r : X \to A$ (that is, $r^2 = r$ and $r(X) = A$) and $r \dot{\simeq} \mathrm{id}_X$ via a homotopy $H : X \times [0, 1] \to X$ that fixes A pointwise at all times.[7] In this situation, the inclusion map $\iota : A \to X$ is a homotopy equivalence (with homotopy inverse r). It is natural to ask which sub-complexes of a simplicial complex K obtained through the application of successive combinatorial deletions are deformation retracts. Specifically, the sub-complex $\mathrm{del}_K(\sigma) \triangleq K \smallsetminus \mathrm{st}^{\circ}_K(\sigma)$ is referred to as the complex obtained from K by a deletion Kozlov (2008, Definition 2.12). The effect of a deletion on the geometric realization of K is that of removing from $|K|$ all the open simplices $\dot{\Delta}^{\tau}$ whose closures contain $\dot{\Delta}^{\sigma}$. One type of deletion always resulting in a deformation retract (of the parent complex) is a simplicial collapse.

[7] That is, $H(x, t) = x$ for all $t \in [0, 1]$ and $x \in A$.

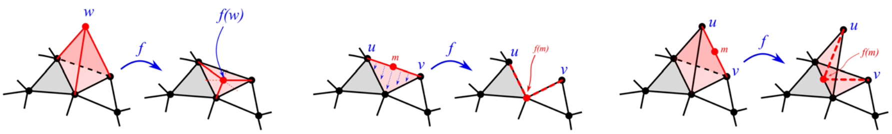

Fig. 4 A vertex collapse (left) and edge collapses in dimensions two (center) and three (right), together with the corresponding retractions

2.5.1 Simplicial collapse

A special and well-studied class of strong deformation retractions arising in combinatorial topology is that of simplicial collapses.

Definition 2.5.1 (Free Simplex, Simplicial Collapse). A simplex $\sigma \in K$ is said to be free, if it is contained in exactly one maximal simplex, that will be denoted by σ^*. If $\sigma \in K$ is a free simplex, then the deleted complex $\mathrm{del}_K(\sigma)$ is said be obtained from K by a simplicial collapse (of σ).

Intuitively, the space $|\mathrm{del}_K(\sigma)|$ may be seen as a deformation retract of $|K|$, via the map gradually "pressing in" the faces $\dot{\Delta}^\tau$ of Δ^{σ^*} for $\tau \supseteq \sigma$, until there is nothing left of $\dot{\Delta}^{\sigma^*}$ or any of the facets $\dot{\Delta}^\tau$, see Fig. 4.

More generally, the sub-complex $L \subset K$ is said to have been obtained from K by simplicial collapse, if L is the result of a finite sequence of simplicial collapses having been applied to K. New simplicial collapses may become available after a preceding one, and collapses of distinct free simplices may not commute, except when supported on pairwise non-interacting maximal simplices, hence the following definition.

Definition 2.5.2 (Independent Simplices). A family of simplices $\Sigma \subset K$ is said to be independent if $\mathrm{st}^\circ_K(\sigma) \cap \mathrm{st}^\circ_K(\tau) = \varnothing$ whenever $\sigma, \tau \in \Sigma, \sigma \neq \tau$. □

If Σ is a family of free simplices, then it is independent if and only if $\sigma^* \neq \tau^*$ whenever $\sigma, \tau \in \Sigma$ and $\sigma \neq \tau$. Deletions of simplices from an independent family may be carried out in an arbitrary order, always leading to the same result,

$$\mathrm{del}_K(\Sigma) \triangleq K \smallsetminus \bigcup_{\sigma \in \Sigma} \mathrm{st}^\circ_K(\sigma) = \bigcap_{\sigma \in \Sigma} \mathrm{del}_K(\sigma) = \left\{\tau \in K : \forall_{\sigma \in \Sigma}\ \sigma \nsubseteq \tau\right\}. \quad (13)$$

These "wholesale deletions", together with the generalized collapses of the next section, become instrumental in the proof of Theorem 1.2.4.

2.5.2 Generalized collapse

Suppose K is a simplicial complex and $\sigma \in K$ has the property that its link, $L \triangleq \mathrm{lk}_K(\sigma)$ is contractible. It is not clear how to obtain the deleted complex $\mathrm{del}_K(\sigma)$ in a step-by-step fashion, using simplicial collapses. Nevertheless, one has the following result, which is central to the argument establishing Theorem 1.2.4.

Lemma 2.5.3 *If $\sigma \in K$ is a simplex whose link in K is contractible, then $|del_K(\sigma)|$ is a deformation retract of $|K|$. In particular, $|K| \doteq |del_K(\sigma)|$.*

Proof Recalling that $U \triangleq \left|\mathrm{st}_K^\circ(\sigma)\right|$ is an open subset of $|K|$ containing $\dot{\Delta}^\sigma$ and denoting $N \triangleq \mathrm{st}_K(\sigma)$, observe that $\partial U = |N| \smallsetminus U = |\mathrm{del}_N(\sigma)|$. Thus, $\partial U = |\mathrm{del}_N(\sigma)|$ separates U from $|K| \smallsetminus \mathtt{cl}_{|K|}(U) = |K| \smallsetminus |N|$, implying that $|\mathrm{del}_K(\sigma)|$ is a strong deformation retract of $|K|$ if and only if $|\mathrm{del}_N(\sigma)|$ is a strong deformation retract of $|N| = |L * \mathfrak{B}^\sigma|$, which coincides with the geometric join of $|L|$ with Δ^σ.

To obtain a strong deformation retraction of $|N|$ onto $|\mathrm{del}_N(\sigma)|$, it suffices to prove, by Kozlov (2008, Corollary 7.15), that the inclusion of $|\mathrm{del}_N(\sigma)|$ into $|N|$ is a homotopy equivalence. Observe that $|N| = |L * \mathfrak{B}^\sigma|$ is contractible, because $\Delta^\sigma = |\mathfrak{B}^\sigma|$ is. Also, $|\mathrm{del}_N(\sigma)| = |L * \mathfrak{S}^\sigma|$ is contractible, because L is. Thus, the inclusion of the latter in the former is a homotopy equivalence (with homotopy inverse any constant map from Δ^σ to $|L|$). □

The preservation of homotopy type in the transition from K to $\mathrm{del}_K(\sigma)$ under the conditions of the preceding lemma motivates the following definition.

Definition 2.5.4 (Removal, Generalized Collapse). A simplex $\sigma \in K$ is said to be removable, if $|\mathrm{lk}_K(\sigma)|$ is contractible. In this situation, the complex $\mathrm{del}_K(\sigma)$ is said to be obtained from K by a removal. More generally, a sub-complex K' of K is said to be the result of a generalized collapse of K, if K' may be obtained from K by a finite sequence of removals. □

Corollary 2.5.5 (Generalized Collapse Yields Deformation Retracts). *If Σ is a finite independent family of removable simplices in K, then $del_K(\Sigma)$ may be obtained from K by generalized collapse, and $|del_K(\Sigma)|$ is a deformation retract of $|K|$.* □

3 Problem formulation

We consider the LTL-based path planning problem on a space X over a given set AP of atomic propositions. Semantics is provided by realizing the elements of AP as Boolean functions, called labels, whose supports form a good open indexed cover of X. Formally, consider the following definition.

Definition 3.0.6 A Labeled Topological Space is a tuple $(X, \mathrm{AP}, \mathbb{U}, L)$, where

- X is a topological space with topology $\mathscr{T}$ satisfying the assumptions stated in Sect. 2.1;
- $\mathbb{U} : \mathrm{AP} \to \mathscr{T}$ is a good open cover of X (see Sect. 2.3).

The set of functions $l_\alpha : X \to \{\top, \bot\}$ defined by $l_\alpha^{-1}(\top) = \mathbb{U}(\alpha)$ for $\alpha \in \mathrm{AP}$ shall be referred to as the set L of labels corresponding to the indexed cover $\mathbb{U}$. □

For the rest of this section, let $(X, \mathrm{AP}, \mathbb{U}, L)$ be a given labeled topological space and let X_0 be a set of initial states definable in terms of L. For a system with state space X, the labels L should be thought of as assigning meaning to states $x \in X$. For example, a label $l_\alpha \in L$ may correspond to safety, with a state x deemed safe if $l_\alpha(x) = \top$ and unsafe otherwise.

Definition 3.0.7 (State Satisfaction). A Boolean state over AP is a subset $\sigma \subset \mathrm{AP}$. A state x in a labeled topological space $(X, \mathrm{AP}, \mathbb{U}, L)$ is said to satisfy a Boolean state $\sigma \subset \mathrm{AP}$, denoted by $x \models \sigma$, if $l_\alpha(x) = \top$ for all $\alpha \in \sigma$ and $l_\alpha(x) = \bot$ for all $\alpha \in \mathrm{AP} \setminus \sigma$ (equivalently, $\sigma = \varsigma(x)$ holds if and only if x satisfies σ). □

Remark 3.0.8 Recalling the notion of a consistent set $\sigma \subset \mathrm{AP}$ (Definition 2.3.1), note that $x \models \sigma$ if and only if x witnesses σ and σ is maximal with respect to this property.

3.1 LTL semantics over topological spaces and the path planning problem

LTL employs temporal and logical connectives to express how the Boolean values of labels change as the system state evolves over time. For that, one considers Boolean paths over the set AP of atomic propositions, as follows.

Definition 3.1.1 (Boolean Paths). A Boolean path over AP is a sequence (finite or infinite) $\sigma = (\sigma_n)_{n\geq 0}$ of Boolean states. The k-prefix $\mathrm{pref}_k(\sigma)$ of σ is defined as the path $(\sigma_n)_{n=0}^{k-1}$, and the k-tail (or suffix) of σ is defined as the path $\mathrm{suff}_k(\sigma) \triangleq (\sigma_{n+k})_{n\geq 0}$. □

Intuitively, a continuous path in the state space X gives rise to a Boolean path over AP, through the labels L, but one must be careful to avoid pathologies.

Definition 3.1.2 (Tame Paths). A tame path in a labeled topological space $(X, \mathrm{AP}, \mathbb{U}, L)$ is a pair $(c, \mathscr{J})$, where $c : J \to X$ is a continuous map of a non-empty interval $J \subset \mathbb{R}$ to X, and $\mathscr{J} = \{J_m\}_{m\in A}$ is a disjoint decomposition $J = \bigcup_m J_m$ of J into non-empty sub-intervals[8], with m ranging over an initial interval A of $\mathbb{N} \cup \{0\}$, satisfying the requirements:

1. For all $m \in A$, if $s, t \in J_m$ then $l_\alpha(c(s)) = l_\alpha(c(t))$ for all $\alpha \in \mathrm{AP}$;
2. For all $m, n \in A$ with $m < n$, if $s \in J_m$ and $t \in J_n$ then $s < t$.

The k-prefix $(\mathrm{pref}_k(c), \mathrm{pref}_k(\mathscr{J}))$ and k-tail $(\mathrm{suff}_k(c), \mathrm{suff}_k(\mathscr{J}))$ of a tame path $(c, \mathscr{J})$ as above are defined by restricting c to $\bigcup \mathrm{pref}_k(\mathscr{J})$ and $\bigcup \mathrm{suff}_k(\mathscr{J})$, respectively, where

$$\mathrm{pref}_k(\mathscr{J}) \triangleq \{J_m\}_{m\in A, m<k}, \tag{14}$$

$$\mathrm{suff}_k(\mathscr{J}) \triangleq \{J_{m+k}\}_{m\in \mathbb{N}\cup\{0\}, m+k\in A}. \tag{15}$$

Any tame path $(c, \mathscr{J})$ gives rise to a Boolean path via $\sigma_m(c, \mathscr{J}) := \varsigma(c(t_m))$, where $t_m \in J_m$ is chosen arbitrarily for each $m \in A$. □

Remark 3.1.3 Not all continuous paths in X are tame, as examples may be constructed where $(l_\alpha \circ c)^{-1}(\bot)$ is a Cantor set in J.

The restriction to tame paths allows us to apply LTL, whose semantics are originally defined for discrete-time paths, to continuous-time paths by tracking the changes in

[8] Some of which may be degenerate, e.g., J_m may be a singleton for some $m \in A$.

the truth value of atomic propositions along the path. In fact, the tame path condition is also commonly used in the syntax and semantics of Signal Temporal Logic (STL) or Metric Interval Temporal Logic (MITL), which can be viewed as counterparts of LTL, to ensure the decidability of their decision problems (Ouaknine and Worrell 2005; Maler and Nickovic 2004; Roohi and Viswanathan 2017). In this article, we use the semantics of LTL over tame paths instead of STL or MITL because we only care about the temporal order of changes in the truth value of the atomic propositions and ignore the timing.

Restricting reasoning to tame paths makes it possible to consider a wide range of behavioral specifications expressible through the labels, e.g., continuing the example where a label l corresponded to the subset of X where the system state must be confined out of safety considerations, the LTL formula $\Box l$ expresses the requirement that, beginning with the present initial state, the system should remain safe for all time. Thus, $\Box$ is the temporal connective standing for "henceforth". More formally, the syntax of temporal and logical connectives in LTL is inductively generated by AP, the unary operators $\neg$ and $\bigcirc$, and the binary operators $\wedge$ and U, subject to the requirements that:

1. Each $\alpha \in$ AP is an LTL formula;
2. If φ is an LTL formula then $\neg\varphi$ and $\bigcirc\varphi$ are LTL formulae;
3. If φ, ψ are LTL formulae then $\varphi \wedge \psi$ and $\varphi \mathsf{U} \psi$ are LTL formulae.

The logical connectives $\neg$ and $\wedge$ stand for negation and conjunction, whereas the temporal connectives $\bigcirc$ and U stand for "next" and "until", respectively. Disjunction is implicitly introduced through $\varphi_1 \vee \varphi_2 := \neg(\neg\varphi_1 \wedge \neg\varphi_2)$. Other common temporal logic connectives may be derived from this basic syntax, such as $\Diamond\varphi := \top \mathsf{U} \varphi$ ("eventually φ") and $\Box\varphi := \neg(\Diamond\neg\varphi)$ ("henceforth φ")[9]. The semantics of LTL formulae over a labelled topological space are defined formally as follows.

Definition 3.1.4 (LTL Satisfaction Relation over a Labeled Space). Let $(X, \mathrm{AP}, \mathbb{U}, L)$ be a labeled topological space and let $(c, \mathscr{J})$ be a tame path, which will be referred to as c, for brevity. The satisfaction relation $c \models \varphi$, extending Definition 3.0.7, is defined for LTL formulae over AP inductively, as follows:

$$
\begin{array}{lcl}
c \models \alpha & \Leftrightarrow & c(t_0) \models \alpha \text{ for some, and hence any, } t_0 \in J_0 \\
c \models \neg\varphi & \Leftrightarrow & c \not\models \varphi \\
c \models \varphi_1 \wedge \varphi_2 & \Leftrightarrow & c \models \varphi_1 \text{ and } c \models \varphi_2 \\
c \models \bigcirc\varphi & \Leftrightarrow & \mathrm{suff}_1(c) \models \varphi \\
c \models \varphi_1 \mathsf{U} \varphi_2 & \Leftrightarrow & \exists t.\ (\mathrm{suff}_t(c) \models \varphi_2 \text{ and } \forall s < t.\ \mathrm{suff}_s(c) \models \varphi_1)\,.
\end{array}
$$

□

Remark 3.1.5 (Only the Temporal Order Matters for Semantics). On one hand, note that the interval $J = \bigcup \mathscr{J}$ and the decomposition $\mathscr{J} = \{J_m\}_{m \in A}$ in Definition 3.1.4 are

[9] Some authors prefer the use of "always" to "henceforth", but note there is a difference between these notions if one allows past observations into the semantics. We pick the latter interpretation to avoid any ambiguities.

part and parcel of the semantics: formally speaking, the family $\mathscr{J}$ must be prescribed in order for the semantics to be specified. On the other hand, a tame path may be reparametrized: if $g : I \to J$ is any order-preserving homeomorphism of an interval $I \subseteq \mathbb{R}$ onto J, then the pair $(c \circ g, \mathscr{I})$ with $\mathscr{I} := \{g^{-1}(J_m)\}_{m \in A}$ is also a tame path in the same labeled space, satisfying the same set of LTL formulae. Thus, the semantics defined here may be regarded as being sensitive only to the ordering of events, but not to their particular timing, allowing the domain of definition of a path to be one of four types, namely: a single point (e.g., $J = \{0\}$), a non-degenerate closed interval (resp., $J = [0, 1]$), an open interval (resp., $J = \mathbb{R}$), or a semi-open interval (resp., $J = [0, \infty)$).

This article is motivated by two fundamental questions about the feasibility of planning in topological spaces, the first of which is formulated below.

Problem 3.1 (Path Planning to LTL Specification). For any LTL formula φ over AP, decide whether or not a tame path c exists emanating from X_0 and satisfying $c \models \varphi$, and produce such a path. □

Some restrictions on the labels—or, equivalently, on the sets $\mathbb{U}(\alpha)$—are necessary for Problem 3.1 to become tractable. In addition to the mildly restrictive requirement that the $\mathbb{U}(\alpha)$ be open sets (which contributes to a certain lack of symmetry between atomic propositions and their negations), the requirement that $\mathbb{U}$ form a good indexed cover comes explicitly to harness the nerve $\mathcal{N}(\mathbb{U})$ as a potential discrete model of X of high topological fidelity, over which one might hope to reason when planning.

3.2 Motivation: constructing flows to an LTL specification

Another problem motivating the current work is, intuitively, of higher complexity, and relates to the generation of behaviors conforming to an LTL specification, generalizing the idea of an attractor (e.g., a limit cycle) in dynamical systems.

Problem 3.2 (Construction of LTL-specified Behaviors). Given an LTL formula φ over AP, determine whether there exists a continuous flow $f : X \times [0, \infty) \to X$ such that every trajectory $c : [0, \infty) \to X$ of f with $c(0) \in X_0$ is a tame path satisfying $c \models \varphi$. If the flow f exists, construct it. □

Remark 3.2.1 (Motivation for Problem 3.2). It is worth spending some time discussing the difference between a plan and a behavior. In the controls and robotics literature, a plan is a designated path $x_d : J \to X$ for the system to execute up to a prescribed error tolerance or a desired level of stability: given the plan and the system dynamics, one closes the control loop by designing a control input for which, respectively, the position error $\|x_d(t) - x(t)\|$ could be bounded by a user-provided constant, or by a function of t converging to zero at a prescribed pace. In contrast, by a behavior one generally means designing a controller to produce closed-loop dynamics with a prescribed attractor (or several) having user-defined stability properties. Consequently, one could regard path-planning based control as focusing on a single instance of a behavior. For example, the navigation task in X from a path planning perspective is that of generating, for each x_0 and x^*, a path $x_d : [0, 1] \to X$ with $x_d(0) = x_0$ and $x_d(1) = x^*$ (and then tracking that path) could be generalized to that of specifying a vector field g on X having x^* as an asymptotically stable attractor with a large enough basin (and then designing a controller that enforces the closed-loop dynamics given by g). In the case of navigation functions, the basin is required to be the whole of X with the possible exception of a null set of cuts (Rimon and Koditschek 1992). More complex behaviors, such as robot gaits (Holmes et al. 2006; Baryshnikov and Shapiro 2014) arise as flows of vector fields with attractive limit cycles. Potentially even more complex behaviors may be generated using hybrid dynamics as envisioned, e.g., in Burridge et al. (1999), by switching between different continuous controllers depending on the system's logical state, but topological constraints are imposed (on basins of attraction) by the choice of desired attractor(s). Thus, Problem 3.2 presents an extension of the notion of a behavior using LTL-based specifications.

3.3 The path correspondence problem and reduced nerves

Both Problem 3.1 and Problem 3.2 are construction problems requiring *a-priori* a resolution in terms of an adequate discrete model of X accounting for all the possible Boolean states as dictated by the labels L. On first inspection, the nerve $\mathcal{N}(\mathbb{U})$ seems to be just such a model: $\mathcal{N}(\mathbb{U})$ both contains all the Boolean states of the labeled topological space $(X, \text{AP}, \mathbb{U}, L)$ (as simplices) and constitutes a good candidate model of X because it has the same homotopy type as X. Thus, a tame path $(c, \mathscr{J})$ in X gives rise to a sequence of faces of $\mathcal{N}(\mathbb{U})$, or, equivalently, to a sequence of vertices of $\text{sd}(\mathcal{N}(\mathbb{U}))$, by mapping each $c(t_m)$, $t_m \in J_m$ to the Boolean state $\sigma_m = \varsigma(c(t_m))$, and one expects each σ_m and σ_{m+1} to be related in a way that reflects the continuity of c (as will be discussed in Sect. 5.1).

These properties, however, are insufficient for planning purposes, because not every walk in the 1-skeleton of $\text{sd}(\mathcal{N}(\mathbb{U}))$ is obtainable from a tame path in X by the above procedure. In fact, a vertex $\sigma \in \text{sd}(\mathcal{N}(\mathbb{U}))$ may exist for which no $x \in X$ satisfies σ. These difficulties motivate the study of the following problem, addressed in this paper.

Problem 3.3 Characterize a simplicial sub-complex $\mathcal{N}_{red}(\mathbb{U})$ of $\text{sd}(\mathcal{N}(\mathbb{U}))$ such that:

1. $|\mathcal{N}_{red}(\mathbb{U})|$ is a deformation retract of $|\text{sd}(\mathcal{N}(\mathbb{U}))|$;
2. $\sigma \in \mathcal{N}(\mathbb{U})$ is a vertex in $\mathcal{N}_{red}(\mathbb{U})$ if and only if $x \models \sigma$ for some $x \in X$;
3. Every path in the 1-skeleton of $\mathcal{N}_{red}(\mathbb{U})$ is induced by a tame path in X. □

While Requirements 2. and 3. of $\mathcal{N}_{red}(\mathbb{U})$ stated in Problem 3.3 are geared directly towards making $\mathcal{N}_{red}(\mathbb{U})$ into a discrete model of X suitable for solving Problem 3.1, the implications of the first requirement are more circumspect. Since $\mathbb{U}$ is a good cover, if $|\mathcal{N}_{red}(\mathbb{U})|$ is a deformation retract of $|\text{sd}(\mathcal{N}(\mathbb{U}))|$, then $|\mathcal{N}_{red}(\mathbb{U})|$ has the homotopy type of X, retaining the homotopy invariants of the latter. In particular, the topological complexity of continuous path planning in X and in $|\mathcal{N}_{red}(\mathbb{U})|$ are the same. Moreover, if an explicit deformation retraction of $|\text{sd}(\mathcal{N}(\mathbb{U}))|$ onto $|\mathcal{N}_{red}(\mathbb{U})|$ is available (in some computable combinatorial form), then continuous path planners in the sense of Farber over X may be converted into such planners over $|\mathcal{N}_{red}(\mathbb{U})|$ and *vice-versa*, opening the door for a discussion of the topological complexity of LTL-based path planning. The next section details a method for solving Problem 3.1 over $\mathcal{N}_{red}(\mathbb{U})$ under the assumption that a construction of $\mathcal{N}_{red}(\mathbb{U})$ satisfying Requirements 2. and 3. is available.

4 Planning space construction

This section is dedicated to solving Problem 3.1 assuming a solution, $K \triangleq \mathcal{N}_{red}(\mathbb{U})$ of Problem 3.3 is known for the labeled space $(X, \mathrm{AP}, \mathbb{U}, L)$. The technique is agnostic to the specific structure of K, based on the fact that a finite state transition system $\mathcal{A}_\varphi$, an ω-regular automaton, can monitor the satisfaction of an LTL formula φ along a discrete path through a discrete labeled transition system $\mathcal{T}$by constructing a "product automaton". However, it is necessary to precisely determine what transition structures (Sect. 4.1) may serve legitimately in the role of discretizations of continuous motions in the continuous space X (Sect. 4.3). Once this goal is achieved, the automaton $\mathcal{A}_\varphi$ (whose structure is reviewed in Sect. 4.4) may be used for a generalized construction that solves the associated Planning problem (Sect. 4.5).

4.1 Transition systems

Treating topological spaces and discrete automata in a unified framework is made conceptually easier using a category-theoretical viewpoint of non-deterministic transition systems. The simplification arises from maps of transition systems being used as a means for comparing the systems. One very general way of encoding a transition system with multiple transitions at each vertex is provided in the following definition.

Definition 4.1.1 (Transition System). A transition system is a tuple $\mathcal{T} = (V, E, \mathfrak{s}, \mathfrak{t})$, where V is a non-empty set of states and E is a set of non-deterministic transitions. The map $\mathfrak{s} : E \to V$ specifies for each $e \in E$ its initial state $\mathfrak{s}e$, and $\mathfrak{t} : E \to \mathbf{2}^V$ specifies the set of possible terminal states $\mathfrak{t}e \subseteq V$. □

Definition 4.1.2 (Maps of Transition Systems). A map $\psi : \mathcal{T}_1 \to \mathcal{T}_2$ of transition systems $\mathcal{T}_i = (V_i, E_i, \mathfrak{s}_i, \mathfrak{t}_i)$, $i \in \{1, 2\}$, is a pair of maps $\psi = (\psi_0, \psi_1)$ such that $\psi_0 : V_1 \to V_2$ and $\psi_1 : E_1 \to E_2$ satisfying $\psi_0(\mathfrak{s}_1 e) = \mathfrak{s}_2 \psi_1(e)$ and $\psi_0(\mathfrak{t}_1 e) \subseteq \mathfrak{t}_2 \psi_1(e)$ for all $e \in E_1$. □

We shall, henceforth, use the boundary map symbols $\mathfrak{s}$ and $\mathfrak{t}$ while omitting any additional decorations indicating the transition system to which they belong. For consistency, any $\mathcal{T} = (V, E, \mathfrak{s}, \mathfrak{t})$ will be referred to simply as $\mathcal{T} = (V, E)$, also denoting $V_\mathcal{T} = V$ and $E_\mathcal{T} = E$. The simplest example of a transition system is as follows.

Example 4.1.3 An interval is a transition system Γ where V_Γ is an initial interval of $\mathbb{N} \cup \{0\}$, $E_\Gamma = V_\Gamma \setminus \{0\}$, and the boundary maps are defined as $\mathfrak{s}e = e - 1$ and $\mathfrak{t}e = \{e\}$ for all $e \in E$.

A fundamental notion in discrete dynamics is that of a trajectory. Intuitively, a trajectory of $\mathcal{T} = (V, E)$ ought to be a sequence $(v_0, v_1, \ldots, v_n, \ldots)$—finite or infinite—satisfying the requirement that for each $i \geq 0$ for which v_{i+1} is defined there exists $e_i \in E$ such that $v_i = \mathfrak{s}e_i$ and $v_{i+1} \in \mathfrak{t}e_i$. However, this definition puts too much emphasis on states, and leaves no record of the transitions applied between the states. Therefore, instead, we introduce the following more informative definition.

Definition 4.1.4 (Execution). An execution of a transition system $\mathcal{T}$ is a map of transition systems $\gamma : \Gamma \to \mathcal{T}$, where Γ is an interval. The trace $\mathtt{tr}(\gamma)$ of γ is defined as the sequence $(\gamma_0(v))_{v \in V_\Gamma}$, with the indices v in increasing order.

In other words, the intuitive notion of a trajectory coincides with the trace of an execution. Another use for the notion of a map of transition systems is the following notion of a refinement order on transition systems.

Definition 4.1.5 (Refinement). Let $\mathcal{T}_i = (V, E_i)$ be transition systems with the same state set V. The system $\mathcal{T}_1$ is said to refine $\mathcal{T}_2$, if there exists a map $\psi : \mathcal{T}_1 \to \mathcal{T}_2$ with $\psi_0 = \mathrm{id}_V$. □

Observe that if $\mathcal{T}_1$ refines $\mathcal{T}_2$, then every execution of $\mathcal{T}_1$ is also an execution of $\mathcal{T}_2$, though the latter may afford executions that are invalid for the former.

4.2 Labeled transition systems and control

Labeled transition systems may be seen as having control inputs. The formal construction is as follows.

Definition 4.2.1 (Labeled Transition System). A transition system labeled by a set Λ is a pair $(\mathcal{T}, \lambda)$ where $\lambda : E_\mathcal{T} \to \Lambda$ is a function satisfying the requirement that, for all $e, f \in E_\mathcal{T}$, if $\mathfrak{s}e = \mathfrak{s}f$ and $\lambda(e) = \lambda(f)$ then $e = f$. □

In a labeled transition system, each label $\ell \in \Lambda$ selects (at most) a single transition for each vertex. It is then possible to think of Λ as a set of control inputs for generating executions/trajectories inductively, for each $v_0 \in V_\mathcal{T}$ and $\ell_\omega \triangleq (\ell_m)_{m=1}^{\infty} \in \Lambda^\omega$, as follows:

- *Induction base.* The desired execution of $\mathcal{T}$ may be initiated by selecting $v_1 \in \mathfrak{t}e_1$ if there is an $e_1 \in E_\mathcal{T}$ such that $\mathfrak{s}(e_1) = v_0$ and $\lambda(e_1) = \ell_1$, noting that ℓ_1 determines e_1. If there is no such e_1, then the only execution compatible with the control input ℓ is $\gamma : \Gamma \to \mathcal{T}$ with $\Gamma = (\{0\}, \varnothing)$.
- *Induction step.* Suppose Γ is a finite interval, $V_\Gamma = \{0, \ldots, m\}$, $m \in \mathbb{N}$, and $\gamma : \Gamma \to \mathcal{T}$ is an execution satisfying $\gamma_0(0) = v_0$ and $\lambda(\gamma_1(i)) = \ell_i$ for all $i = 1, \ldots, m$. We ask whether an extension $\gamma^+ : \Gamma^+ \to \mathcal{T}$ exists, where Γ^+ is the interval of length $m + 1$ and the labeling constraint is satisfied. Clearly, this happens if and only if there exists a transition $e_{m+1} \in E_\mathcal{T}$ such that $\mathfrak{s}e_{m+1} = \gamma_0(m)$ and $\lambda(e_{m+1}) = \ell_{m+1}$. If not, then the execution γ terminates.

Therefore, each sequence of labels ℓ_ω determines a collection of compatible executions $\mathtt{ex}(\ell_\omega)$ and the resulting traces $\mathtt{tr}(\ell_\omega)$, some of which may terminate in finite time. Those $\gamma \in \mathtt{ex}(\ell_\omega)$ which terminate in finite time satisfy the labeling constraints associated with later times vacuously.

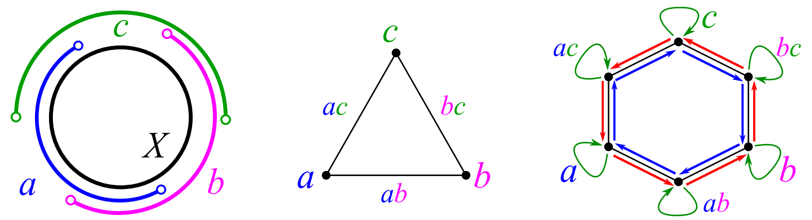

Fig. 5 A planning problem on the circle (Example 4.3.2). $X = \mathbb{S}^1$ is covered by three sets (left), indexed by $\mathrm{AP} = \{a, b, c\}$, giving rise to $\mathcal{N}(\mathbb{U}) = \mathfrak{S}^{\mathrm{AP}} \cong \mathfrak{S}^2$ (center). This example was selected to satisfy $\mathcal{N}_{red}(\mathbb{U}) = \mathrm{sd}(\mathcal{N}(\mathbb{U}))$, for simplicity. The induced transition system $\mathcal{T}_{\mathbb{U}}$ is shown (right)

4.3 From labeled topological spaces to finite transition systems

This section considers ways for a labeled topological space to support transition systems that are mindful of its topology. A labeled topological space gives rise to the following coarse transition system reflecting the preference for tame paths.

Definition 4.3.1 (Induced transition system). Let $(X, \mathrm{AP}, \mathbb{U}, L)$ be a labeled topological space and let $K = \mathcal{N}_{red}(\mathbb{U})$. The induced transition system $\mathcal{T}_{\mathbb{U}}$ has vertex and edge sets $V = E = \mathrm{sk}_0(K)$, coinciding with the set of all $\mathbb{U}$-realized simplices $\sigma \in \mathcal{N}(\mathbb{U})$, and, for each such σ, the initial and terminal maps are defined as

$$\begin{aligned} \mathfrak{s}\sigma &\triangleq \sigma, \\ \mathfrak{t}\sigma &\triangleq \{\tau \in \mathcal{N}(\mathbb{U}) \colon \{\sigma, \tau\} \in K\}, \end{aligned} \tag{16}$$

which is the 1-neighborhood of σ in the 1-skeleton of K. □

Therefore, a solution $\mathcal{N}_{red}(\mathbb{U})$ to Problem 3.3 produces a discrete transition system whose trajectories in $\mathcal{T}_{\mathbb{U}}$ are precisely the Boolean paths over AP induced by tame paths in X according to Definition 3.1.2. It remains to connect this construction to actual planning problems.

Example 4.3.2 Consider the labeled space from Fig. 5, together with a robot with state in $X = \mathbb{S}^1$ (left, black), endowed with controllers capable of executing either one of two commands, labeled `cw` and `ccw`, and corresponding, respectively, to clockwise motion or counter-clockwise motion along the circle. The atoms of the alphabet $\mathrm{AP} = \{a, b, c\}$ are realized as depicted (left), and $\mathcal{N}_{red}(\mathbb{U}) = \mathrm{sd}(\mathcal{N}(\mathbb{U}))$ holds by visual inspection. The induced transition system $\mathcal{T}_{\mathbb{U}}$ is depicted on the right of Fig. 5, with the blue arrows clearly corresponding to `cw`-induced transitions and the red arrows—to the `ccw`-induced ones. The loops depicted in green result from the possibility that a `cw` or `ccw` command over a short enough period of time might not result in a state transition over the vertex set $V = \{a, b, c, ab, ac, bc\}$ of $\mathcal{N}_{red}(\mathbb{U})$. As a result, in every discrete state, the actions associated with `cw` and `ccw` determine, respectively, *a pair of arrows*: a blue/red arrow and a green loop. Given an LTL formula φ over AP, the goal is to plan a sequence of `cw`/`ccw` commands that would satisfy[10] φ.

[10] In the sense of the semantics of tame paths (Sect. 3.1).

Thus, practical concerns require the consideration of labeled transition systems, and the planning problem becomes that of computing an element (or elements) of Λ^ω guaranteeing that every trajectory emanating from a specified initial set X_0 in the vertex set of K and generated by the control input will satisfy φ.

Definition 4.3.3 (Admissible Transition System). Let $(X, \mathrm{AP}, \mathbb{U}, L)$ be a labeled topological space. A labeled transition system $(\mathcal{T}, \lambda)$ on the state space $V_{\mathcal{T}_{\mathbb{U}}}$ is said to be admissible for $\mathbb{U}$, if $\mathcal{T}$ refines $\mathcal{T}_{\mathbb{U}}$. □

The labeling itself does not matter for admissibility. The labeling is an external element having to do with the way input is encoded—not with what transitions are allowed to happen in principle. Since any trajectory of $\mathcal{T}$ is also a trajectory of $\mathcal{T}_{\mathbb{U}}$, we have the automatic guarantee that any input sequence $(\ell_m)_{m=1}^{\infty}$ only generates trajectories induced from tame paths.

4.4 Limiting deterministic Büchi automata (LDBA)

There are many varieties of ω-regular automata, one of which, following (Sickert et al. 2016), was adopted in this article.

Definition 4.4.1 (LDBA). An LDBA is a tuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, B)$ where Q is a finite set of automaton states; Σ is a finite alphabet of input symbols; $\delta : Q \times (\Sigma \cup \{\epsilon\}) \to \mathbf{2}^Q$ is a non-deterministic transition function [11]; q_0 is an initial state; and B is a set of accepting states. In addition,

1. δ is deterministic except for ϵ-moves, i.e., $|\delta(q, \alpha)| = 1$ for all $q \in Q, a \in \Sigma$;
2. there exists a bipartition $Q = Q_{ini} \cup Q_{acc}$ such that:
   - Q_{acc} contains all accepting states;
   - Q_{acc} is invariant: $\delta(q, v) \subseteq Q_{acc}$ for all $q \in Q_{acc}$ and $v \in \Sigma$;
   - ϵ-moves are not allowed in Q_{acc}: $\delta(q, \epsilon) = \varnothing$ for all $q \in Q_{acc}$.

A sequence of states $\mathbf{q} = (q_i)_{i=0}^{\infty} \in Q^{\mathbb{N}\cup\{0\}}$ beginning with the initial state q_0 is said to be a *valid* path in $\mathcal{A}$, if there are $v_i \in \Sigma \cup \{\epsilon\}, i \in \mathbb{N} \cup \{0\}$, such that $\delta(q_i, v_i) = q_{i+1}$ for all i. A valid path $\mathbf{q}$ is *accepted* by $\mathcal{A}$ if and only if B is visited infinitely many times[12] by q (the Büchi condition). A path that is invalid or not accepted is said to be rejected by $\mathcal{A}$. □

Note that the presence of ϵ-moves renders an LDBA non-deterministic. Intuitively, such moves represent state updates which take no time, motivating the concept of ϵ-closure of a state, or, more generally, of a set of states $A \subset Q$, defined as

$$[A]_\epsilon \triangleq \{q \in Q : q \text{ is reachable from } A \text{ by a finite sequence of } \epsilon \text{ transitions}\} \supseteq A. \tag{17}$$

[11] Hereafter; ϵ denotes the empty input.

[12] Since B is finite, this is equivalent to there being a $b \in B$ such that $q_i = b$ for infinitely many different values of i.

An LDBA $\mathcal{A}$ may be regarded as a transition system $\mathcal{T} = \mathcal{T}(\mathcal{A})$ labeled by Σ, with $V_{\mathcal{T}} = Q$, $E_{\mathcal{T}} = Q \times \Sigma$ and labeling function $\lambda : E_{\mathcal{T}} \to \Sigma$ defined by $\lambda(q, \sigma) = \sigma$, and where the boundary maps are given by $\mathfrak{s}(q, \sigma) = q$ and

$$\mathfrak{t}(q, \sigma) \triangleq \delta\left([q]_\epsilon, \sigma\right) = \left\{\delta(q', \sigma) \colon q' \in [q]_\epsilon\right\}. \tag{18}$$

Thus, a valid path $\mathbf{q}$ is precisely a trajectory of $\mathcal{T}(\mathcal{A})$ emanating from q_0.

Definition 4.4.2 (Language accepted by an LDBA). Let $\mathcal{A} = (Q, \Sigma, \delta, q_0, B)$ be an LDBA. A sequence[13] $\boldsymbol{\sigma} = (\sigma_i)_{i=0}^{\infty} \in \Sigma^{\mathbb{N}\cup\{0\}}$ is said to generate the path $\mathbf{q} = (q_i)_{i=0}^{\infty}$, if $q_{i+1} \in \delta([q_i]_\epsilon, \sigma_i)$ for all $i \in \mathbb{N}\cup\{0\}$. The set of all paths $\mathbf{q}$ generated by $\boldsymbol{\sigma}$ is denoted by $L(\boldsymbol{\sigma})$. The language $L(\mathcal{A})$ accepted by the LDBA $\mathcal{A}$ is defined by $\boldsymbol{\sigma} \in L(\mathcal{A})$ if and only if $L(\boldsymbol{\sigma})$ contains an accepted path. □

The following result is one of several in model checking enabling the use of finite-state automata (here, specifically, LDBAs) as state representations of LTL formulae.

Lemma 4.4.3 Sickert et al. (2016, Theorem 1). *For any LTL formula φ over AP, there exists an LDBA with input alphabet 2^{AP},*

$$\mathcal{A}_\varphi \triangleq (Q_\varphi, \mathbf{2}^{AP}, \delta_\varphi, q_0, B), \tag{19}$$

such that, for any Boolean path σ over AP, $\sigma \models \varphi$ holds if and only if $\sigma \in L(\mathcal{A}_\varphi)$. □

While converting an LTL formula φ into the LDBA $\mathcal{A}_\varphi$ is a non-trivial procedure, the use of LDBAs is preferable, in part due to the fact that not all LDBAs are of the form $\mathcal{A}_\varphi$. Thus, LDBAs represent an even broader class of temporal specifications than LTL does.

4.5 The planning space

We are now ready to provide a solution to Problem 3.1, in a slightly more general, practical form. For the rest of this section, let $(X, \mathrm{AP}, \mathbb{U}, L)$ be a labeled topological space and let X_0 be a set of vertices of $\mathcal{N}_{red}(\mathbb{U})$, regarded as the initial condition for the planning problem. Let $(\mathcal{T}, \lambda)$ be a labeled $\mathbb{U}$-admissible transition system and let φ be an LTL formula over AP: recall that the labels Λ are to be interpreted as control inputs to $\mathcal{T}$, while the formula φ is the LTL formula encoding the task specification and represented by the automaton $\mathcal{A}_\varphi = (Q_\varphi, \mathbf{2}^{\mathrm{AP}}, \delta_\varphi, q_0, B)$. Also, let Γ denote the infinite interval (see Example 4.1.3).

Definition 4.5.1 (Planning Space). Given the data above, the associated planning space is defined as the transition system $\widetilde{\mathcal{T}}_\varphi = (\widetilde{V}, \widetilde{E})$ with states $\widetilde{V} \triangleq V_{\mathcal{T}} \times Q_\varphi$, transitions

$$\widetilde{E} \triangleq \left\{(e, f) \in E_{\mathcal{T}} \times E_{\mathcal{T}(\mathcal{A}_\varphi)} \colon \lambda f = \mathfrak{s}e\right\}, \tag{20}$$

[13] Commonly referred to as an ω-word.

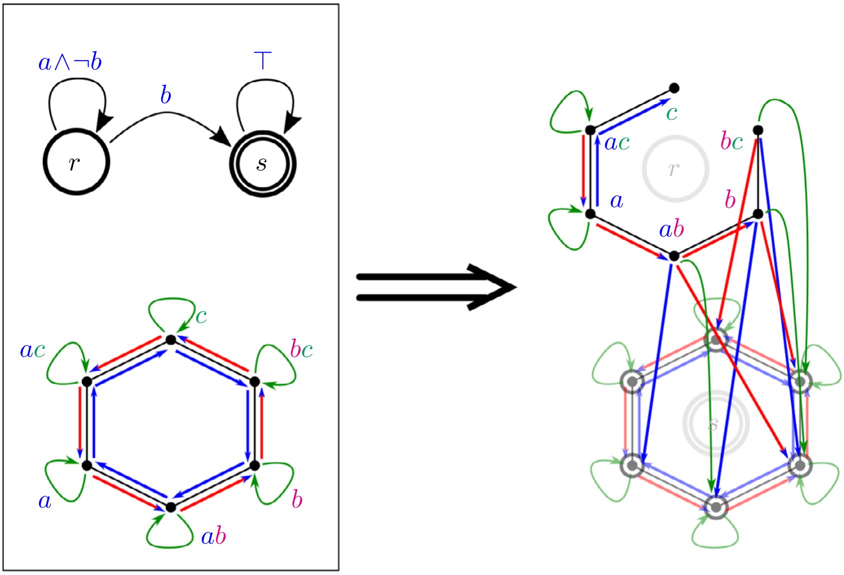

Fig. 6 Constructing the planning space for $\varphi = aUb$ and the cover $\mathbb{U}$ of Example 4.3.2 (also see Fig. 5). The automaton $\mathcal{A}_\varphi$ (left, top) and transition system $\mathcal{T}_\mathbb{U}$ (left, bottom) give rise to the planning space $\widetilde{\mathcal{T}}_\varphi$ (right), also known as the "product automaton", which may be used for planning, according to Lemma 4.5.3. By a simplifying convention, only transitions in $\mathcal{A}_\varphi$ along possible accepted paths are depicted, while any others are omitted and may be thought of as arriving at a designated sink state (also omitted). Furthermore, transitions in $\mathcal{A}_\varphi$ are labeled in an abbreviated fashion, by Boolean formula over AP $= \{a, b, c\}$ whose truth set coincides with the set of $\lambda \in \mathbf{2}^{\mathrm{AP}}$ appearing as labels for that transition. For example, the arrow from r to s labeled b (top left) means r transitions to s for any input σ such that $b \in \sigma$; the loop at s labeled $\top$ means the automaton stay in s for all inputs

and boundary maps

$$\begin{aligned} \mathfrak{s}(e, f) &\triangleq (\mathfrak{s}e, \mathfrak{s}f), \\ \mathfrak{t}(e, f) &\triangleq \mathfrak{t}e \times \mathfrak{t}f, \end{aligned} \tag{21}$$

with labeling $\lambda(e, f) \triangleq \lambda e$. □

A construction of the planning space for Example 4.3.2 is illustrated in Fig. 6. An elementary observation about the planning space follows.

Lemma 4.5.2 *Let maps* $\pi : \widetilde{\mathcal{T}}_\varphi \to \mathcal{T}$ *and* $\beta : \widetilde{\mathcal{T}}_\varphi \to \mathcal{T}(\mathcal{A}_\varphi)$ *be defined by* $\pi = (\pi_0, \pi_1)$ *and* $\beta = (\beta_0, \beta_1)$, *where*

$$\pi_0(\sigma, q) \triangleq \sigma, \quad \beta_0(\sigma, q) \triangleq q,$$

and

$$\pi_1(e, f) \triangleq e, \quad \beta_1(e, f) \triangleq f,$$

for all $(\sigma, q) \in \widetilde{V}$ *and* $(e, f) \in \widetilde{E}$. *Then, both* π *and* β *are maps of transition systems. Also,* π *is label-preserving.*

Proof We verify the conditions of Definition 4.1.2. For any $(e, f) \in \widetilde{E}$,

$$\pi_0(\mathfrak{s}(e, f)) = \pi_0(\mathfrak{s}e, \mathfrak{s}f) = \mathfrak{s}e, \qquad \mathfrak{s}\pi_1(e, f) = \mathfrak{s}e;$$
$$\pi_0(\mathfrak{t}(e, f)) = \pi_0(\mathfrak{t}e \times \mathfrak{t}f) = \mathfrak{t}e, \qquad \mathfrak{t}\pi_1(e, f) = \mathfrak{t}e.$$

Thus, π is a map of transition systems. The labels are preserved by π by definition, as $\lambda(e, f) = \lambda e$ by construction. An analogous computation holds for β. □

Any execution $\tilde{\gamma} : \Gamma \to \widetilde{\mathcal{T}}_\varphi$ gives rise to an execution $\pi \circ \tilde{\gamma} : \Gamma \to \mathcal{T}$. Conversely, given an execution $\gamma : \Gamma \to \mathcal{T}$ one says that an execution $\tilde{\gamma} : \Gamma \to \widetilde{\mathcal{T}}_\varphi$ is a *lift* of γ, if $\gamma = \pi \circ \tilde{\gamma}$. Lifts are key to LTL-based planning, because of the following satisfaction criterion.

Lemma 4.5.3 *Let Γ be the infinite interval and let $\gamma : \Gamma \to \mathcal{T}$ be an execution emanating from X_0. Then, $\mathtt{tr}(\gamma) \models \varphi$ if and only if γ has a lift $\tilde{\gamma}$ emanating from $X_0 \times \{q_0\}$ and visiting $\mathbf{2}^{AP} \times B$ infinitely many times.*

Proof Suppose the required lift $\tilde{\gamma}$ exists. Then $\mathtt{tr}(\gamma)$ is a sequence of label inputs to the transition system $\mathcal{T}(\mathcal{A})$ generating the execution $\beta \circ \tilde{\gamma}$, which emanates from q_0 and visits the set B of accept states infinitely many times. Thus, $\mathtt{tr}(\gamma) \models \varphi$, by Lemma 4.4.3.

Conversely, if $\mathtt{tr}(\gamma) \models \varphi$ then $\mathtt{tr}(\gamma) \in L(\mathcal{A}_\varphi)$, meaning that $L(\mathtt{tr}(\gamma))$ contains an accepted path $\mathbf{q} \triangleq (q_i)_{i=0}^{\infty}$ (see Definition 4.4.2). In particular, there is an execution $\mu : \Gamma \to \mathcal{T}(\mathcal{A}_\varphi)$ satisfying $\mathtt{tr}(\mu) = \mathbf{q}$. Construct an execution $\tilde{\gamma} : \Gamma \to \widetilde{\mathcal{T}}_\varphi$ by setting $\tilde{\gamma}_0(m) = (\gamma_0(m), \mu_0(m))$, $m \geq 0$ and $\tilde{\gamma}_1(m) = (\gamma_1(m), \mu_1(m))$, $m \geq 1$. To ensure that $\tilde{\gamma}_1(m) \in \widetilde{E}$, it is necessary to verify that $\lambda\mu_1(m) = \mathfrak{s}\gamma_1(m)$ for all $m \geq 1$, by (20). Now, $\lambda\mu_1(m) = \gamma_0(m-1)$ by the construction of $\mathcal{T}(\mathcal{A}_\varphi)$, and, at the same time, $\mathfrak{s}\gamma_1(m) = \gamma_0(\mathfrak{s}m) = \gamma_0(m-1)$, by the definition of a map of transition systems (Definition 4.1.2). Thus, $\tilde{\gamma}$ is an execution of $\widetilde{\mathcal{T}}_\varphi$ satisfying the desired properties. □

Corollary 4.5.4 *The system $\mathcal{T}$ has an infinite trajectory emanating from X_0 and satisfying φ if and only if $\widetilde{\mathcal{T}}_\varphi$ has a state (σ, q) such that $q \in B$ and:*

1. *(σ, q) is reachable from $X_0 \times \{q_0\}$;*
2. *$\widetilde{\mathcal{T}}_\varphi$ has a directed cycle based at (σ, q).* □

Remark 4.5.5 Note that AP is a finite set, implying that $\mathcal{N}_{red}(\mathbb{U})$ and $\widetilde{\mathcal{T}}_\varphi$ are finite. Therefore, the set $\widetilde{B}$ of states of $\widetilde{\mathcal{T}}_\varphi$ with Q_φ-component in B is visited infinitely many times by $\tilde{\gamma}$ if and only if $\tilde{\gamma}$ contains a loop based in $\widetilde{B}$.

Corollary 4.5.4 explains how the problem of planning for φ in the labeled space $(X, \mathrm{AP}, \mathbb{U}, L)$ may be solved by searching the discrete model $\widetilde{\mathcal{T}}_\varphi$, providing at least one avenue for an algorithmic solution of Problem 3.1 using existing formal methods approaches, but for the more general class of labels obtained as good covers, provided a solution of Problem 3.3, which is the topic of the next section.

5 The reduced nerve of an open cover

This section is dedicated to studying notions of realizability and to establishing results about the structure of the reduced nerve, recalling from Sect. 1.2 the various examples of obstructions to symbolic path planning over $\mathcal{N}(\mathbb{U})$ and the solution of Problem 3.3.

5.1 Realizability

A subtler phenomenon than the realizability introduced in Definition 1.2.1 is that of a realized simplex in $\mathrm{sd}(\mathcal{N}(\mathbb{U}))$:

Definition 5.1.1 ($\mathbb{U}$-Small Simplex). Let $T \subset \mathcal{N}(\mathbb{U})$. A $\mathbb{U}$-small singular T-simplex in X is a map $g : \Delta^T \to X$ such that $(\varsigma \circ g)(\Delta^S) = S$ for all $S \subseteq T$. If there exists a $\mathbb{U}$-small T-simplex, the collection T is said to be $\mathbb{U}$-realized. □

Note how a singleton $T = \{\sigma\} \subset \mathcal{N}(\mathbb{U})$ is $\mathbb{U}$-realized if and only if $\sigma \in \mathcal{N}(\mathbb{U})$ is $\mathbb{U}$-realized. Also, any sub-collection S of a $\mathbb{U}$-realized collection T is also $\mathbb{U}$-realized (indeed, if g is a $\mathbb{U}$-small T-simplex, then $g\big|_{\Delta^S}$ is a $\mathbb{U}$-small S-simplex). Thus, the set of $\mathbb{U}$-realized subsets of $\mathcal{N}(\mathbb{U})$ forms a simplicial complex, whose vertex set is the set of $\mathbb{U}$-realized simplices of $\mathcal{N}(\mathbb{U})$. Figure 7 illustrates the notion of a $\mathbb{U}$-small simplex on our running example, Example 1.2.2, and hints at the following result.

Lemma 5.1.2 *A $\mathbb{U}$-realized collection $T \subseteq \mathcal{N}(\mathbb{U})$ is a chain under inclusion. Hence, the set of $\mathbb{U}$-realized collections $T \subseteq \mathcal{N}(\mathbb{U})$ forms a sub-complex of the barycentric subdivision $sd(\mathcal{N}(\mathbb{U}))$ of $\mathcal{N}(\mathbb{U})$.*

Proof If not, then let T be a $\mathbb{U}$-realized collection containing a pair of incomparable simplices σ, τ of $\mathcal{N}(\mathbb{U})$, which together form a $\mathbb{U}$-realized pair $S \triangleq \{\sigma, \tau\}$. Find $\alpha, \beta \in \mathrm{AP}$ such that $\alpha \in \sigma \smallsetminus \tau$ and $\beta \in \tau \smallsetminus \sigma$. Let $g : \Delta^S \to X$ be a continuous map satisfying $\varsigma(g(\xi_0)) = \sigma$, $\varsigma(g(\xi_1)) = \tau$, and $\varsigma(g(\xi_t) \in S$ for all $t \in [0, 1]$, where $\xi_t \triangleq (1 - t)e_\sigma + te_\tau$. Consider the closed sets $A, B \subset [0, 1]$ defined as

$$A \triangleq \{t \in [0, 1] \colon g(\xi_t) \notin \mathbb{U}(\alpha)\}, \quad B \triangleq \{t \in [0, 1] \colon g(\xi_t) \notin \mathbb{U}(\beta)\}.$$

From $\alpha \notin \tau$ and $\tau = \varsigma(g(\xi_1))$ it follows that $1 \in A$. Similarly, $0 \in B$ since $\beta \notin \sigma$ and $\sigma = \varsigma(g(\xi_0))$.

We claim that $A \cup B = [0, 1]$. For any $t \in [0, 1]$, either $\varsigma(g(\xi_t)) = \sigma$, implying $g(\xi_t) \notin \mathbb{U}(\beta)$ and $t \in B$, or $\varsigma(g(\xi_t)) = \tau$, implying $g(\xi_t) \notin \mathbb{U}(\alpha)$ and $t \in A$.

Next, we claim that $A \cap B = \varnothing$. Indeed, for every $t \in [0, 1]$ one has either $\varsigma(g(\xi_t)) = \sigma$ and then $g(\xi_t) \in \widetilde{\mathbb{U}}(\sigma) \subset \mathbb{U}(\alpha)$, or $\varsigma(g(\xi_t)) = \tau$ implying $g(\xi_t) \in \widetilde{\mathbb{U}}(\tau) \subset \mathbb{U}(\beta)$. Either way, $g(\xi_t) \in \mathbb{U}(\alpha) \cup \mathbb{U}(\beta)$ for all $t \in [0, 1]$. However, any $t \in A \cap B$ must satisfy $g(\xi_t) \notin \mathbb{U}(\alpha) \cup \mathbb{U}(\beta)$—a contradiction.

Since neither of A, B is empty, we obtain a contradiction to $[0, 1]$ being connected, finishing the proof of the lemma. □

The result of the last lemma motivates the following definition.

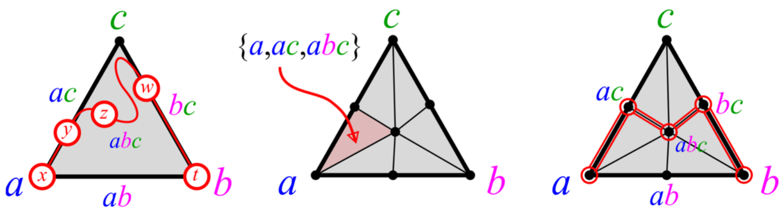

Fig. 7 Continuing Example 1.2.2 (left), observe how the simplices $\{a, abc\}, \{a, ac, abc\} \in \mathrm{sd}(\mathcal{N}(\mathbb{U}))$ are not $\mathbb{U}$-realized (center) because there is no path in X from $\widetilde{\mathbb{U}}(abc)$ to a point of $\mathbb{U}(a) \setminus \widetilde{\mathbb{U}}(bc)$ avoiding $\widetilde{\mathbb{U}}(ac)$. At the same time, $\{a, ac\}$ is a $\mathbb{U}$-realized simplex in X, as witnessed by the interval in X joining the points x and y. The simplices $\{a, ab\}, \{b, ab\} \in \mathrm{sd}(\mathcal{N}(\mathbb{U}))$, too, are unrealized, since $ab \in \mathcal{N}(\mathbb{U})$ is. Overall, the reduced nerve $\mathcal{N}_{red}(\mathbb{U})$ (right, red highlight) is a contractible 1-dimensional sub-complex of $\mathrm{sd}(\mathcal{N}(\mathbb{U}))$

Definition 5.1.3 (Reduced Nerve). Let $(X, \mathscr{T})$ be a space and let $\mathbb{U} \colon \mathrm{AP} \to \mathscr{T}$ be an indexed cover of X. The reduced nerve $\mathcal{N}_{red}(\mathbb{U})$ of $\mathbb{U}$ is the sub-complex of $\mathrm{sd}(\mathcal{N}(\mathbb{U}))$ whose simplices are the $\mathbb{U}$-realized collections in $\mathcal{N}(\mathbb{U})$. □

5.2 Basic properties of realized simplices

Realized simplices in $\mathcal{N}(\mathbb{U})$ have the following characterization.

Lemma 5.2.1 *A simplex $\sigma \in \mathcal{N}(\mathbb{U})$ is unrealized if and only if $\widetilde{\mathbb{U}}(\sigma) \subseteq \bigcup_{\beta \in AP \setminus \sigma} \mathbb{U}(\beta)$.*

Proof Let $\sigma \in \mathcal{N}(\mathbb{U})$. By the definition of ς, for any $x \in X$,

$$\sigma = \varsigma(x) \;\Leftrightarrow\; x \in \bigcap_{\alpha \in \sigma} \mathbb{U}(\alpha) \setminus \bigcup_{\beta \in \mathrm{AP} \setminus \sigma} \mathbb{U}(\beta). \tag{22}$$

Thus σ is $\mathbb{U}$-unrealized if and only if

$$\bigcap_{\alpha \in \sigma} \mathbb{U}(\alpha) \setminus \bigcup_{\beta \in \mathrm{AP} \setminus \sigma} \mathbb{U}(\beta) = \varnothing,$$

which proves the assertion of the lemma. □

Lemma 5.2.2 *If $\sigma \in \mathcal{N}(\mathbb{U})$ is maximal, then $\sigma = \varsigma(x)$ for all $x \in \bigcap_{\alpha \in \sigma} \mathbb{U}(\alpha)$.*

Proof Indeed, if $\sigma \in \mathcal{N}(\mathbb{U})$ is a maximal simplex and $\beta \in \mathrm{AP} \setminus \sigma$ then $\sigma \cup \{\beta\} \notin \mathcal{N}(\mathbb{U})$, implying that $\bigcap_{\alpha \in \sigma} \mathbb{U}(\alpha) \subseteq X \setminus \mathbb{U}(\beta)$. Since $\beta \in \mathrm{AP} \setminus \sigma$ was arbitrary, the criterion in (22) implies every $x \in \bigcap_{\alpha \in \sigma} \mathbb{U}(\alpha)$ satisfies $\sigma = \varsigma(x)$. □

The preceding lemma focuses attention on the differences between two presentations of $|\mathcal{N}(\mathbb{U})|$, highlighting the role of realized simplices in "keeping the nerve together".

Corollary 5.2.3 *Let $\mathbb{U}$ be an indexed open cover of X. Then the following equality holds:*

$$|\mathcal{N}(\mathbb{U})| = \bigcup_{x \in X} \Delta^{\varsigma(x)}. \tag{23}$$

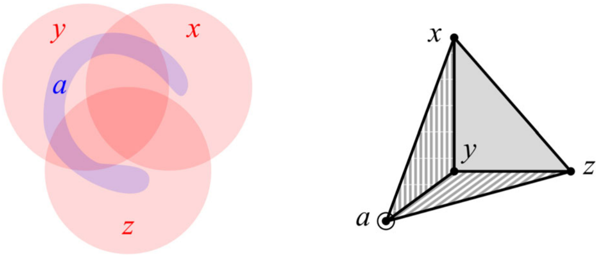

Fig. 8 A good cover $\mathbb{U}$ of a topological disk featuring an unrealized simplex of $\mathcal{N}(\mathbb{U})$ that is not free, as described in Example 5.2.4)

Proof The geometric realization of a simplicial complex is the union of geometric realizations of its closed maximal simplices, which, by the preceding lemma, are all $\mathbb{U}$-realized. □

Comparing the above presentation with the definition $|\mathcal{N}(\mathbb{U})| = \bigcup_{\sigma \in \mathcal{N}(\mathbb{U})} \dot{\Delta}^{\sigma}$, note that some of the open simplices $\dot{\Delta}^{\sigma}$ appear as proper faces of the closed simplices in (23). All $\mathbb{U}$-unrealized simplices are proper faces of maximal ones, and it is therefore reasonable to investigate their contribution to the homotopy type of the nerve.

Example 5.2.4 (A non-free unrealized simplex). We construct a cover $\mathbb{U}$ of a topological disk X, indexed by the set of symbols $\mathrm{AP} = \{x, y, z, a\}$, as depicted in Fig. 8, with $X = \mathbb{U}(x) \cup \mathbb{U}(y) \cup \mathbb{U}(z)$. Note that $\mathbb{U}(a)$ is a redundant covering element, rendering $\{a\}$ an unrealized simplex. The nerve $\mathcal{N}(\mathbb{U})$ may be described as the complex obtained from the 2-dimensional simplex with vertices x, y, z by appending the simplices $\{x, y, a\}$ and $\{y, z, a\}$ (and all their faces). The realization of $\mathcal{N}(\mathbb{U})$ is a disk. The realization of the link $L = \mathrm{lk}_{\mathcal{N}(\mathbb{U})}(a)$ is a 2-path embedded in the disk. Observe how $\{a\}$ is the only unrealized simplex of $\mathcal{N}(\mathbb{U})$, and that it is not free in $\mathcal{N}(\mathbb{U})$, being contained in the two maximal simplices $\{x, y, a\}$ and $\{y, z, a\}$. However, since L is contractible, $\{a\}$ is removable, hinting at the possibility of applying a generalized collapse (Lemma 2.5.3) to obtain a simplified model of the disk X.

5.3 Unrealized simplices of the subdivided nerve

Lemma 5.2.2 and Corollary 5.2.3 motivate the question of whether or not the unrealized simplices of $\mathcal{N}(\mathbb{U})$ could be removed in some way without altering its homotopy type, while Example 5.2.4 demonstrates that simplicial collapses fail to achieve this goal. However, the following Lemma shows that generalized collapses (Section 2.5.4) are the right tool for the job.

Lemma 5.3.1 *Suppose $\sigma \in \mathcal{N}(\mathbb{U})$ is any $\mathbb{U}$-unrealized simplex of a good cover. Then σ is a removable simplex of $\mathcal{N}(\mathbb{U})$, as well as a removable vertex of $sd(\mathcal{N}(\mathbb{U}))$.*

Proof Denote $K \triangleq \mathcal{N}(\mathbb{U})$ and suppose $\sigma \in K$ is $\mathbb{U}$-unrealized. Then, by Lemma 5.2.1, the non-empty subspace $\widetilde{\mathbb{U}}(\sigma)$ is covered by the open contractible sub-spaces $\mathbb{U}_{\sigma}(\alpha) \triangleq$

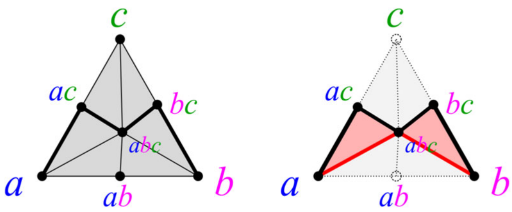

Fig. 9 In Figure 7, the reduced nerve of the running example (Example 1.2.2) was computed as a sub-complex of $\mathrm{sd}(\mathcal{N}(\mathbb{U}))$ (left, bold). The removal of the two unrealized vertices $\{c\}$, $\{ab\} \in \mathrm{sk}_0(\mathrm{sd}(\mathcal{N}(\mathbb{U})))$ results in a sub-complex containing the reduced nerve (right). Still, some $\mathbb{U}$-unrealized simplices remain in this complex (red), such as the edge $\{a, abc\}$ and the 2-simplex $\{a, ac, abc\}$

$\widetilde{\mathbb{U}}(\sigma \cup \{\alpha\})$, where α ranges over $\mathrm{AP} \setminus \sigma$. Of these sets, only the ones with $\sigma \cup \{\alpha\} \in K$ are non-empty, corresponding to the vertices of the link $L \triangleq \mathrm{lk}_K(\sigma)$, the set of which we denote by L_0.

We conclude that $\mathbb{U}_\sigma$ is a good cover of $\widetilde{\mathbb{U}}(\sigma)$ indexed by L_0. Also, the inclusion map $j : L_0 \to \mathrm{AP}$ induces a simplicial isomorphism of the nerve $\mathcal{N}(\mathbb{U}_\sigma)$ with L, since, for all $x \in X$, x witnesses a simplex $\tau \in \mathcal{N}(\mathbb{U}_\sigma)$ if and only if x witnesses the simplex $\sigma \cup \tau$ in K. Therefore, since $\widetilde{\mathbb{U}}(\sigma)$ is contractible, the Nerve Lemma applied to $\mathbb{U}_\sigma$ implies that $|L|$ is contractible and it follows that σ is a removable simplex of K.

Finally, to see that σ is a removable vertex of $\mathrm{sd}(K)$, apply Corollary 2.4.2 to conclude that $\mathrm{lk}_{\mathrm{sd}(K)}(\{\sigma\})$ is isomorphic to the join $\mathrm{sd}(L) * \mathrm{sd}(\mathfrak{S}^\sigma)$. Since L is contractible, so is $\mathrm{sd}(L)$, and consequently also $\mathrm{lk}_{\mathrm{sd}(K)}(\{\sigma\})$, proving that σ is a removable vertex of $\mathrm{sd}(K)$. □

Among other things, the proof implies that $\mathbb{U}$-unrealized simplices of co-dimension 1 in $\mathcal{N}(\mathbb{U})$ are always free (e.g., the simplex ab in Fig. 2), explaining the need for the slight subtlety of Example 5.2.4, where special care was needed to create a non-free unrealized simplex of $\mathcal{N}(\mathbb{U})$.

Note that performing generalized collapses on unrealized simplices of $\mathcal{N}(\mathbb{U})$ does not yield consistent results: in Fig. 2, both the unrealized simplices of $\mathcal{N}(\mathbb{U})$ are free, but collapsing the vertex $\{c\}$ yields the 1-simplex $\{a, b, ab\}$, while collapsing the edge $\{ab\}$ yields the 2-path $\{a, ac, c, bc, b\}$; both are contractible models of X, but there is not an obvious way of merging them into a common representation of X and $\mathbb{U}$. By contrast, generalized collapses of the unrealized vertices of $\mathrm{sd}(\mathcal{N}(\mathbb{U}))$ in this example may be carried out to obtain a sub-complex of $\mathrm{sd}(\mathcal{N}(\mathbb{U}))$ containing the reduced nerve, see Fig. 9(right). This difference between $\mathcal{N}(\mathbb{U})$ and $\mathrm{sd}(\mathcal{N}(\mathbb{U}))$ provided the original motivation for considering removals in $\mathrm{sd}(\mathcal{N}(\mathbb{U}))$ instead of in $\mathcal{N}(\mathbb{U})$. Unrealized simplices of higher dimension in $\mathrm{sd}(\mathcal{N}(\mathbb{U}))$ (as opposed to unrealized simplices of $\mathcal{N}(\mathbb{U})$, which amount to 0-dimensional simplices of $\mathrm{sd}(\mathcal{N}(\mathbb{U}))$) hold additional information about the homotopy type of X, as discussed in Fig. 9.

5.4 Organized removal of unrealized simplices

Throughout this section, let $\mathbb{U}$ be a good finite indexed cover of X, let $K \triangleq \mathcal{N}(\mathbb{U})$, $N \triangleq \mathrm{sd}(\mathcal{N}(\mathbb{U}))$, and let δ denote the maximum dimension of a $\mathbb{U}$-unrealized simplex in K. For every integer $d \geq 0$, let $\Sigma[d]$ denote the family of $\mathbb{U}$-unrealized simplices in N of dimension less than or equal to d, and let $N[d] \triangleq \mathrm{del}_N(\Sigma[d])$ be the sub-complex of N obtained by deleting all the simplices in $\Sigma[d]$. Note that $\mathcal{N}_{red}(\mathbb{U}) = N[d]$ for every $d \geq \delta$. We study the sub-complex $N[0]$.

Proposition 5.4.1 *$N[0]$ is obtainable from N by generalized collapse.*

Proof For integers $j \geq -1$, let $\Sigma[j, 0]$ denote the family of vertices $\{\sigma\} \in N$ where $\sigma \in K$ ranges over the $\mathbb{U}$-unrealized simplices of dimension less than or equal to j. Then $(\Sigma[j, 0])_{j\geq 0}$ is a monotone non-decreasing sequence with $\Sigma[j, 0] = \Sigma[\delta, 0]$ for all $j \geq \delta$. The sub-complexes $N[j, 0] \triangleq \mathrm{del}_N(\Sigma[j, 0])$ of N satisfy $N[0] = N[j, 0]$ for all $j \geq \delta$, and note that $N[-1, 0] = N$, as $\Sigma[-1, 0]$ is empty.

Setting $\Pi[j] \triangleq \Sigma[j, 0] \smallsetminus \Sigma[j-1, 0]$, observe that $N[j, 0] = \mathrm{del}_{N[j-1,0]}(\Pi[j])$ holds for all $j \geq 0$. Therefore, to obtain the result of the proposition, it suffices to verify that $\Pi[j]$ is an independent family of removable simplices of $N[j-1, 0]$ for all $j \geq 0$. By the construction of $N = \mathrm{sd}(K)$, no simplex of N may contain (as elements) two distinct simplices of K of equal dimensions. Thus, for all $j \geq 0$, $\Pi[j]$ is an independent family of simplices of N, and hence also of the sub-complex $N[j-1, 0]$.

It remains to check that every vertex $\{\sigma\} \in N[j-1, 0]$ is removable if it belongs to $\Pi[j]$, for all $j \geq 0$. Applying Lemma 2.4.1, we obtain the simplicial isomorphism $g_{\{\sigma\}} : \mathrm{lk}_N(\{\sigma\}) \to \mathrm{sd}(\mathrm{lk}_K(\sigma)) * \mathrm{sd}(\mathfrak{S}^\sigma)$, mapping any vertex $\{\tau\}$ of $\mathrm{lk}_N(\{\sigma\})$ to itself, if $\tau \subsetneq \sigma$ and to $\{\tau \smallsetminus \sigma\}$ if $\sigma \subsetneq \tau$. If $\sigma \in \Pi[j] \cap N[j-1, 0]$, then $\mathrm{lk}_{N[j-1,0]}(\{\sigma\})$ is obtained from $\mathrm{lk}_N(\{\sigma\})$ by removing all simplices (of N) which contain a vertex of the form $\{\sigma'\}$ for some $\sigma' \subsetneq \sigma$. In other words, $\mathrm{lk}_{N[j-1,0]}(\{\sigma\})$ is isomorphic to the join of $\mathrm{sd}(\mathrm{lk}_K(\sigma))$ with another simplicial complex. Since $\mathrm{lk}_K(\sigma)$ is contractible by Lemma 5.3.1, so is $\mathrm{lk}_{N[j-1,0]}(\{\sigma\})$. □

6 The connectivity trisp and the path correspondence

To address questions about the connectivity of $\mathcal{N}_{red}(\mathbb{U})$, a more refined combinatorial model of X is required, constructed from *all* the $\mathbb{U}$-small singular simplices in X. For motivation, we revisit Example 1.2.3, where the set $\varsigma^{-1}(\sigma)$ of a $\mathbb{U}$-realized simplex $\sigma \in \mathcal{N}(\mathbb{U})$ was disconnected, resulting in an obstruction to meaningful planning over $\mathcal{N}_{red}(\mathbb{U})$.

Example 6.0.2 The failure of the combinatorial path (a, ax, x, xb, b) to be induced by a tame path in X (Fig. 10, left) is explained by there being two path components x_1 and x_2 of $\varsigma^{-1}(\{x\})$, both of which share a connection to $y \cup z$ via, e.g., a pair of $\mathbb{U}$-small singular $T \triangleq \{yz, xyz\}$-simplices g_1, g_2 connecting $x \cap y \cap z$ to x_1 and x_2, respectively. While g_1 and g_2 are homotopic to each other, they cannot be joined by a homotopy restricting to the same $\mathbb{U}$-small singular T-simplex at every stage of the deformation. Therefore, it seems possible to distinguish between g_1 and g_2 (as well as

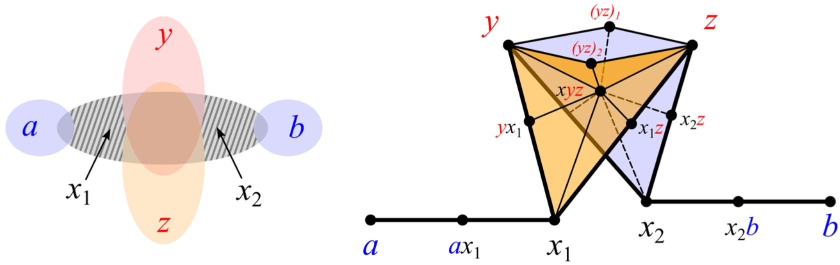

Fig. 10 A good cover of a topological disk requiring a refinement of its reduced nerve to enforce a correspondence between tame paths and combinatorial paths (Example 6.0.2). Observe that "splitting" the vertex $\{x\}$ from Fig. 3 into two vertices corresponding to the path components of $\varsigma^{-1}(\{x\})$ results in a more complete model for path planning that is capable of demonstrating directly the fact that no (tame) path exists in X from $\varsigma^{-1}(\{a\})$ to $\varsigma^{-1}(\{b\})$ avoiding $y \cup z$. Also note how the two path components of $\varsigma^{-1}(\{y, z\})$ give rise to the intricate structure at the top of the "doubled simplex" corresponding to xyz

other such pairs) using a more restricted notion of homotopy, and one expects this fact to give rise to an enriched diagram (Fig. 10, right), which folds onto the one produced by $\mathcal{N}_{red}(\mathbb{U})$ in Fig. 3(right).

The last observations hint at the necessity of constructing a discrete model of X large enough to distinguish between certain restricted homotopy classes of $\mathbb{U}$-small singular simplices. To execute the construction, the notion of a trisp is required, which will now be reviewed briefly, following Kozlov (2008, Section 2.3).

6.1 Preliminary: trisps

Triangulated spaces, also known as trisps, are a generalization of simplicial complexes using a more versatile encoding of the gluing instructions necessary for producing useful subdivision and geometric realization functors in the presence of loops, multiple edges, and their higher dimensional analogues. We recall the definition of the simplex category first.

Definition 6.1.1 (Simplex Category). $\mathbf{\Delta}$ is the category with Ob$\mathbf{\Delta} = \mathbb{Z}_{\geq 0}$ and the set of morphisms $\mathbf{\Delta}(m, n)$ from m to n, $m, n \in$ Ob$\mathbf{\Delta}$, defined as the set of all increasing maps $\alpha : [m + 1] \to [n + 1]$, endowed with the standard composition. □

One thinks of this category as describing face maps between standard simplices, as follows. Given integers $0 \leq m \leq n$, an m-face of the standard n-simplex $\Delta^{[n+1]}$ corresponds to an increasing map $\alpha : [m + 1] \to [n + 1]$, which may be regarded as a simplicial embedding $\alpha : \mathfrak{B}^{[m+1]} \hookrightarrow \mathfrak{B}^{[n+1]}$, giving rise to the piecewise-linear embedding $|\alpha| : \Delta^{[m+1]} \hookrightarrow \Delta^{[n+1]}$. Trisps make use of these maps by regarding them as "abstract face types" to be used in gluing together a collection of simplices of varying dimensions.

Definition 6.1.2 (Trisp). The gluing instructions for a trisp are a contra-variant functor $\Gamma : \mathbf{\Delta} \to \mathbf{Set}$ from the simplex category to the category of sets. Each $\Gamma(n)$, $n \in$ Ob$\mathbf{\Delta}$ is regarded as the set of n-simplices of Γ. Each $\Gamma(\alpha) : \Gamma(n) \to \Gamma(m)$, $\alpha \in \mathbf{\Delta}(m, n)$

is regarded as assigning to each n-simplex its α-face. To avoid a proliferation of parentheses, we adopt the notation $\Gamma(\alpha) : \sigma \mapsto \Gamma(\alpha)\sigma$ for all $\alpha \in \mathbf{\Delta}(m, n)$. □

The compactness of this definition is deceptive. Unpacking it, one observes that the composition requirement, the identity $\Gamma(\beta \circ \alpha) = \Gamma(\alpha) \circ \Gamma(\beta)$, means that a face of a face is a face, enabling consistent gluing instructions.

Definition 6.1.3 (Realization of a trisp). Let Γ be gluing instructions for a trisp. The geometric realization $|\Gamma|$ is the quotient space obtained as

$$|\Gamma| \triangleq \bigsqcup_{n\geq 0} \Gamma(n) \times \Delta^{[n+1]} \Big/ (\sim_\Gamma), \tag{24}$$

where the $\Gamma(n)$ are taken with the discrete topology, and the equivalence relation $(\sim_\Gamma)$ is generated by all expressions of the form

$$(\sigma, |\alpha|(\xi)) \sim_\Gamma (\Gamma(\alpha)\sigma, \xi) \tag{25}$$

for $\alpha \in \mathbf{\Delta}(m, n)$, $\xi \in \Delta^{[m+1]}$, and $\sigma \in \Gamma(n)$. □

Remark 6.1.4 (Irregular trisps). Self-gluings of a simplex σ occur when $\Gamma(\alpha)\sigma = \Gamma(\beta)\sigma$ for $\alpha, \beta \in \mathbf{\Delta}(m, n)$, $\alpha \neq \beta$. Also, for example, two edges $e, f \in \Gamma(1)$, $e \neq f$, may share their boundary vertices while remaining interiorly disjoint in $|\Gamma|$: this may be achieved by setting $\alpha, \beta : [1] \to [2]$ to have $\alpha(1) = 1$ and $\beta(1) = 2$, with $\Gamma(\alpha)e = \Gamma(\alpha)f$ and $\Gamma(\beta)e = \Gamma(\beta)f$.

For simplicity, we will henceforth refer to gluing data Γ (for a trisp) and to the resulting space $|\Gamma|$ interchangeably as 'trisps', same as we do for (abstract) simplicial complexes and their realizations.

Definition 6.1.5 (Regular Trisp). A trisp Γ is said to be regular if, for all $n \in \mathrm{Ob}\mathbf{\Delta}$, all $\sigma \in \Gamma(n)$ and all $\alpha, \beta : [1] \to [n + 1]$ with $\alpha \neq \beta$ one has $\Gamma(\alpha)\sigma \neq \Gamma(\beta)\sigma$.

In other words, Γ is regular if no simplex of Γ has two of its vertices identified to a single point in $|\Gamma|$. In what follows, all the trisps considered in this article will be regular trisps.

Another characteristic of trisps is that their edges, the elements of $\Gamma(1)$, come equipped with an orientation. Setting $\alpha, \beta : [1] \to [2]$ with $\alpha(1) = 1$ and $\beta(1) = 2$, the initial point $\Gamma(\alpha)e$ of an edge e will be denoted by $\partial_0 e$, and the terminal point $\Gamma(\beta)e$ will henceforth be denoted by $\partial_1 e$, coinciding with a more standard notation used for directed graphs. Since we are more interested in edges of Γ as arcs in the space $|\Gamma|$, a definition of an (undirected) edge-path is needed:

Definition 6.1.6 (Edge-Path In a Trisp). An edge-path in a trisp Γ is a sequence $\gamma = (f_k)_{k=1}^{m}$ in $\Gamma(1)$ for which there exists a sequence $(v_k)_{k=1}^{m+1}$ in $\Gamma(0)$ with the property that $\{v_k, v_{k+1}\} = \{\partial_0 f_k, \partial_1 f_k\}$. The initial and terminal vertices v_1, v_{m+1} of γ shall be denoted by $\partial_0\gamma$ and $\partial_1\gamma$, respectively.

6.2 Construction of the connectivity trisp

A formal definition of the restricted notion of homotopy alluded to in Example 6.0.2 is as follows.

Definition 6.2.1 ($\mathbb{U}$-homotopy). Let $T \in \mathrm{sd}(\mathcal{N}(\mathbb{U}))$ be any simplex. The space of all $\mathbb{U}$-small singular T-simplices, endowed with the sub-space topology from $\mathscr{C}(\Delta^T, X)$, is denoted by $\mathbf{C}_{\mathbb{U}}(T)$. For $f, g \in \mathbf{C}_{\mathbb{U}}(T)$, a homotopy $H\colon \Delta^T \times [0, 1] \to X$ from f to g is said to be a $\mathbb{U}$-homotopy, if the map $H_t : \xi \mapsto H(\xi, t)$ is a $\mathbb{U}$-small singular T-simplex for all $t \in [0, 1]$. The $\mathbb{U}$-homotopy class of $f \in \mathbf{C}_{\mathbb{U}}(T)$ will be denoted by $[f]_{\mathbb{U}}$, and the space of all such classes, taken with the discrete topology, will be denoted by $\mathbf{hC}_{\mathbb{U}}(T)$. □

Observe that every simplex $T = \{\sigma_1 \subsetneq \ldots \subsetneq \sigma_{n+1}\}$ in $\mathrm{sd}(\mathcal{N}(\mathbb{U}))$, $n \in \mathbb{Z}_{\geq 0}$ is uniquely represented by the increasing[14] map $[n+1] \to \mathcal{N}(\mathbb{U})$ sending i to σ_i. For the rest of this section, we identify any such T with the corresponding map, which will be referred to by the same symbol T.

The identification of simplices in $\mathrm{sd}(\mathcal{N}(\mathbb{U}))$ with increasing maps into $\mathcal{N}(\mathbb{U})$ makes it easier to keep track of face relations among singular simplices. Namely, for each $T \in \mathrm{sd}(\mathcal{N}(\mathbb{U}))$ as above, any $0 \leq m \leq n$, and any $\alpha \in \mathbf{\Delta}(m, n)$ give rise to the face $T \circ \alpha$ of T. Consequently, for any singular T-simplex $g : \Delta^T \to X$, the inclusion map $T \circ \alpha \hookrightarrow T$ (regarded as sets) induces the embedding of $\Delta^{T\circ\alpha}$ in Δ^T as a face, giving rise to the $\mathbb{U}$-small singular $T \circ \alpha$-simplex

$$\partial_\alpha g \triangleq g \circ |T \circ \alpha \hookrightarrow T|\,. \tag{26}$$

The following elementary fact enables the construction of the connectivity trisp.

Lemma 6.2.2 *Let $0 \leq m \leq n$ be integers and let $T \in \mathcal{N}_{red}(\mathbb{U})$ be an n-simplex. Then for any increasing map $\alpha : [m+1] \to [n+1]$ and any $f, g \in \mathbf{C}_{\mathbb{U}}(T)$, if $[f]_{\mathbb{U}} = [g]_{\mathbb{U}}$ then also $[\partial_\alpha f]_{\mathbb{U}} = [\partial_\alpha g]_{\mathbb{U}}$. In other words, $\partial_\alpha [f]_{\mathbb{U}} \triangleq [\partial_\alpha f]_{\mathbb{U}}$ is a well-defined operation on $\mathbf{hC}_{\mathbb{U}}(T)$.*

Proof Follows directly from Definitions 5.1.1 and 6.2.1. □

Two trisps emerge from the preceding observations. First, the reduced nerve $\mathcal{N}_{red}(\mathbb{U})$ itself may be seen as having a trisp structure, which we denote by $\mathbf{R}_{\mathbb{U}}$:

- For $n \in \mathrm{Ob}\mathbf{\Delta}$, the set $\mathbf{R}_{\mathbb{U}}(n)$ of n-simplices (of the reduced nerve) is the set of all $T \in \mathrm{sd}(\mathcal{N}(\mathbb{U}))$ of cardinality $n+1$, admitting a $\mathbb{U}$-small singular T-simplex.
- For $\alpha \in \mathbf{\Delta}(m, n)$, the map $\mathbf{R}_{\mathbb{U}}(\alpha)\colon \mathbf{R}_{\mathbb{U}}(n) \to \mathbf{R}_{\mathbb{U}}(m)$ is defined by $\mathbf{R}_{\mathbb{U}}(\alpha)T \triangleq T \circ \alpha$.

Note that this construction may be applied to any simplicial complex, resulting in an identical geometric realization. The second trisp is constructed from the homotopy classes of $\mathbb{U}$-small simplices, with Lemma 6.2.2 facilitating the gluing among the faces, as follows.

[14] Increasing, in the sense that $i < j$ implies $\sigma_i \subsetneq \sigma_j$ for all relevant i, j.

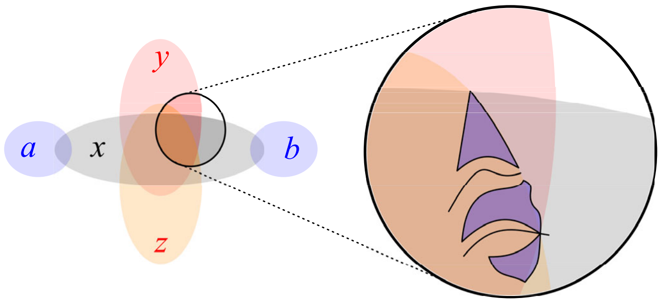

Fig. 11 Illustration of applying Definition 6.2.3 to Example 6.0.2, obtaining the geometric realization depicted in Fig. 10. In the zoomed-in inset (right), three 2-dimensional $\mathbb{U}$-small simplices are shown, top to bottom: an $\{y, xy, xyz\}$-simplex, an $\{x, xy, xyz\}$-simplex, and an $\{x, xz, xyz\}$ simplex. Up to $\mathbb{U}$-homotopy, these simplices share edges, causing them to form a 'gallery' connecting these simplices together so they fan out of the vertex corresponding to the single $\mathbb{U}$-homotopy class representing $\{xyz\}$. These shared edges are represented by the two singular 1-dimensional simplices—one is an $\{xy, xyz\}$-simplex and the other is an $\{x, xyz\}$-simplex, depicted as solid black curves—which are $\mathbb{U}$-homotopic on either side to the corresponding 1-faces of their 2-dimensional 'neighbors'

Definition 6.2.3 (Connectivity Trisp). For an admissible cover $\mathbb{U}$ of X, the connectivity trisp $\widetilde{\mathbf{R}}_{\mathbb{U}}$ is defined as

$$\widetilde{\mathbf{R}}_{\mathbb{U}}(n) \triangleq \bigcup_{T \in \mathbf{R}_{\mathbb{U}}(n)} \mathbf{hC}_{\mathbb{U}}(T) = \{[g]_{\mathbb{U}} \colon \exists\, T \in \mathbf{R}_{\mathbb{U}}(n).\ g \in \mathbf{C}_{\mathbb{U}}(T)\}\,, \tag{27}$$

for $n \in \mathrm{Ob}\boldsymbol{\Delta}$, and the face maps $\widetilde{\mathbf{R}}_{\mathbb{U}}(\alpha) \colon \widetilde{\mathbf{R}}_{\mathbb{U}}(n) \to \widetilde{\mathbf{R}}_{\mathbb{U}}(m)$ are defined as

$$\widetilde{\mathbf{R}}_{\mathbb{U}}(\alpha)[g]_{\mathbb{U}} \triangleq [\partial_\alpha g]_{\mathbb{U}} \tag{28}$$

for all $m, n \in \mathrm{Ob}\boldsymbol{\Delta}$ and $\alpha \in \boldsymbol{\Delta}(m, n)$. □

Note that the (contra-variant) functoriality of $\widetilde{\mathbf{R}}_{\mathbb{U}}$ follows from (26) and the functoriality of geometric realization, as applied to simplicial complexes (see Sect. 2.2.2). At the same time, the geometric realization $\left|\widetilde{\mathbf{R}}_{\mathbb{U}}\right|$ is precisely the construction realizing the motivation discussed in Example 6.0.2 and Fig. 10. Figure 11 provides an illustration of the gluing law (28) on the same example.

The geometric realizations $\left|\widetilde{\mathbf{R}}_{\mathbb{U}}\right|$ and $|\mathbf{R}_{\mathbb{U}}| = |\mathcal{N}_{red}(\mathbb{U})|$ are related by a map realizing the folding idea from Example 6.0.2. For each $n \in \mathrm{Ob}\boldsymbol{\Delta}$ there is a map $\pi(n) : \widetilde{\mathbf{R}}_{\mathbb{U}}(n) \to \mathbf{R}_{\mathbb{U}}(n)$ defined by[15]

$$\pi(n)[g]_{\mathbb{U}} = T \iff T \in \mathrm{sd}(\mathcal{N}(\mathbb{U})),\ |T| = n + 1,\ g \in \mathbf{C}_{\mathbb{U}}(T), \tag{29}$$

satisfying, for every $m, n \in \mathrm{Ob}\boldsymbol{\Delta}$ and $\alpha \in \boldsymbol{\Delta}(m, n)$,

$$\pi(m) \circ \widetilde{\mathbf{R}}_{\mathbb{U}}(\alpha) = \mathbf{R}_{\mathbb{U}}(\alpha) \circ \pi(n). \tag{30}$$

[15] Here we again omit the parentheses around the argument of the map $\pi(n)$ to simplify notation.

In other words, $\pi : \widetilde{\mathbf{R}}_{\mathbb{U}} \to \mathbf{R}_{\mathbb{U}}$ is a natural transformation (which is a rather special case of a trisp map, Kozlov (2008, Definition 2.48)), and it induces a geometric realization $|\pi| : \left|\widetilde{\mathbf{R}}_{\mathbb{U}}\right| \to |\mathbf{R}_{\mathbb{U}}|$ defined, up to quotients, as

$$|\pi|(\sigma, \xi) = (\pi(n)\sigma, \xi) \tag{31}$$

for all $n \in \mathrm{Ob}\boldsymbol{\Delta}$, $\sigma \in \widetilde{\mathbf{R}}_{\mathbb{U}}(n)$ and $\xi \in \Delta^{[n+1]}$.

Remark 6.2.4 A vertex of the reduced nerve is a 0-simplex $u \in \mathbf{R}_{\mathbb{U}}(0)$ of the form $u = \{\varsigma(x)\}$ for some $x \in X$, witnessed by the $\mathbb{U}$-small $\{\varsigma(x)\}$-simplex $e_x : \Delta^{\{\varsigma(x)\}} \to X$ defined by $e_x(\mathbf{e}_{\{\varsigma(x)\}}) \triangleq x$, noting that $\mathbf{e}_{\{\varsigma(x)\}}$ is the only point of $\Delta^{\{\varsigma(x)\}}$. If $y \in \varsigma^{-1}(\varsigma(x))$, then $[e_x]_{\mathbb{U}} = [e_y]_{\mathbb{U}}$ holds if and only if there is a path from x to y in $\varsigma^{-1}(\varsigma(x))$, since such a path constitutes a $\mathbb{U}$-homotopy between the $\mathbb{U}$-small u-simplices e_x and e_y. Thus, the map from X to $\widetilde{\mathbf{R}}_{\mathbb{U}}(0)$ given by $x \mapsto [e_x]_{\mathbb{U}}$ is a lift of the map from X to $\mathcal{N}_{red}(\mathbb{U})$ given by $x \mapsto \{\varsigma(x)\}$, through the map $\pi(0)$.

Definition 6.2.5 (Canonical $\mathbb{U}$-covering). The natural transformation π from (29), as well as its geometric realization $|\pi|$ from (31) will both be referred to as the canonical $\mathbb{U}$-covering map. □

Despite it *not* being a covering map—Example 6.0.2 serves as a good illustration of this fact—the map $|\pi|$ has, nevertheless, some properties reminiscent of a branched cover, such as the fact that it sends open simplices homeomorphically onto open simplices of the same dimension (however, the local degree is not constant). Section 6.3 uses the canonical $\mathbb{U}$-cover to elucidate the path correspondence problem, further establishing π in a role analogous to that of a covering map.

6.3 Tame paths vs. paths in the reduced nerve

The path correspondence problem is concerned with constructing a discrete refinement of $\mathcal{N}(\mathbb{U})$ for which combinatorial paths in the 1-skeleton of the model correspond to tame paths in the labeled space $(X, \mathrm{AP}, \mathbb{U}, L)$. The trisp $\widetilde{\mathbf{R}}_{\mathbb{U}}$ is precisely such a model, as shown by the following lemma.

Lemma 6.3.1 *Let $\mathbb{U}$ be an admissible cover of X over AP. Every $\mathbb{U}$-tame path in X induces a combinatorial path in the 1-skeleton of the connectivity trisp $\widetilde{\boldsymbol{R}}_{\mathbb{U}}$. Every combinatorial path in the 1-skeleton of $\widetilde{\boldsymbol{R}}_{\mathbb{U}}$ is induced by a $\mathbb{U}$-tame path.*

Proof It suffices to prove the lemma for tame paths whose image under ς is finite (the infinite case follows by induction), in correspondence with finite combinatorial paths in the 1-skeleton of $\widetilde{\mathbf{R}}_{\mathbb{U}}$. Suppose $(c, \mathcal{J} = \{J_k\}_{k=1}^m)$ is a $\mathbb{U}$-tame path in X, and let $\sigma_k = \varsigma(c(t_k))$, $t_k \in J_k$, $k \in [m]$. Then, every pair $T_k := \{\sigma_k, \sigma_{k+1}\}$, $k \in [m-1]$ admits a $\mathbb{U}$-small singular T_k-simplex $g_k : \Delta^{T_k} \to X$ constructed as

$$g_k\left((1-s)\mathbf{e}_{\sigma_k} + s\mathbf{e}_{\sigma_{k+1}}\right) \triangleq c\left((1-s)t_k + st_{k+1}\right).$$

Thus, by Lemma 5.1.2, each T_k is a 1-simplex (an edge) of $\mathbf{R}_{\mathbb{U}}$, and the $[g_k]_{\mathbb{U}} \in \widetilde{\mathbf{R}}_{\mathbb{U}}(1)$ form an edge-path in $\widetilde{\mathbf{R}}_{\mathbb{U}}$, as required.

Conversely, suppose $f_k = \{[g_k]_{\mathbb{U}}\}_{k=1}^m$ is an edge-path in $\widetilde{\mathbf{R}}_{\mathbb{U}}$. By Definition 6.1.6 and Remark 6.2.4, there are vertices $v_k = [e_{x_k}]_{\mathbb{U}} \in \widetilde{\mathbf{R}}_{\mathbb{U}}(0)$, $k \in [m+1]$ such that $\{v_k, v_{k+1}\} = \{\partial_0 f_k, \partial_1 f_k\}$. For each $k \in [m]$, construct the following:

- Let $p_k : [0,1] \to X$ be linear reparametrizations of g_k so that $p_k(0)$ coincides with the endpoint of g_k lying in v_k and $p_k(1)$ coincides with the endpoint of g_k lying in v_{k+1};
- Let $a_k : [0,1] \to X$ be a path in v_k from x_k to $p_k(0)$;
- Let $b_k : [0,1] \to X$ be a path in v_{k+1} from $p_k(1)$ to x_{k+1}.

Let q_k be the concatenation of a_k, p_k, b_k, for $k \in [m]$, and let c be the concatenation of $q_1, \ldots, q_m$, in that order. Then c is the desired tame path (with an appropriately constructed partition of its domain into intervals). □

Corollary 6.3.2 *Let $\mathbb{U}$ be an admissible cover of X over AP. Then, every $\mathbb{U}$-tame path in X induces a combinatorial path in the 1-skeleton of $\mathbf{R}_{\mathbb{U}}$—or, equivalently, in $\mathcal{N}_{red}(\mathbb{U})$.*

Proof If c is a tame path in X, let $\tilde{\gamma}$ be the edge-path in $\widetilde{\mathbf{R}}_{\mathbb{U}}$ induced by it. Then $\pi(1)\tilde{\gamma}$ is the required edge-path in $\mathbf{R}_{\mathbb{U}}$. □

The canonical $\mathbb{U}$-covering π is useful for making explicit the path correspondence between X and the reduced nerve.

Definition 6.3.3 (Path-lifting property). Let $\mathbb{U}$ be an admissible cover of X. We say that $\mathbb{U}$ has the path-lifting property, if for every edge-path γ in $\mathbf{R}_{\mathbb{U}}$ and any $\tilde{v}_1 \in \pi(0)^{-1}\partial_0\gamma$ there exists an edge-path $\tilde{\gamma}$ in $\widetilde{\mathbf{R}}_{\mathbb{U}}$ with $\partial_0\tilde{\gamma} = \tilde{v}_1$ that satisfies $\pi(1)\tilde{\gamma} = \gamma$.

The conditions for there being a valid correspondence between tame paths in X and edge paths in $\mathcal{N}_{red}(\mathbb{U})$ are now clarified by the following result.

Theorem 6.3.4 (Path correspondence criterion). *Let $\mathbb{U}$ be an admissible cover of X over AP. Then the following are equivalent:*

1. *For every $x \in X$ and every edge-path γ in $\mathcal{N}_{red}(\mathbb{U})$ emanating from $\varsigma(x)$ there exists a tame path c in X emanating from x and inducing γ.*
2. *$\mathbb{U}$ has the path-lifting property.*

Proof To prove (1)⇒(2), given γ, apply the construction in the proof of Lemma 6.3.1 to the path c to obtain the desired $\tilde{\gamma}$. For the converse, given an edge-path γ in $\mathbf{R}_{\mathbb{U}}$ and $\tilde{v}_1 \in \pi(0)^{-1}\partial_0\gamma$, find $x \in X$ such that $v_1 = [e_x]_{\mathbb{U}}$ and a lift $\tilde{\gamma}$ with $\partial_0\tilde{\gamma} = v_1$. Then, Lemma 6.3.1 (converse direction) provides the desired tame path. □

The following corollary of Remark 6.2.4 and Theorem 6.3.4 now explains the intuition behind the failure of the path correspondence in Example 1.2.3.

Corollary 6.3.5 *Let $\mathbb{U}$ be an admissible cover of X over AP. If $\varsigma^{-1}(\sigma)$ is path-connected for all realized $\sigma \in \mathcal{N}(\mathbb{U})$, then every edge-path in $\mathcal{N}_{red}(\mathbb{U})$ is induced from a tame path.* □

Proof It suffices to show that $\mathbb{U}$ has the path-lifting property. By the present assumption, it follows from Remark 6.2.4 that the map $\pi(0)$ is a bijection. By construction, the map $\pi(1)$ is a surjection respecting the endpoint maps ∂_0 and ∂_1. It follows that every edge-path in $\mathbf{R}_{\mathbb{U}}$ lifts to an edge path in $\widetilde{\mathbf{R}}_{\mathbb{U}}$. □

It is unclear at the moment whether or not the path-lifting property may hold for other reasons. For example, standard covering space theory (where the path-lifting property is guaranteed) provides an alternative sufficient condition:

Corollary 6.3.6 *Let $\mathbb{U}$ be an admissible cover of X over AP. If the canonical $\mathbb{U}$-covering induces a covering map of geometric realizations, then every edge-path in $\mathcal{N}_{red}(\mathbb{U})$ is induced from a tame path.* □

In conclusion, the result of Lemma 6.3.1 seems to point towards $\widetilde{\mathbf{R}}_{\mathbb{U}}$ as a preferred discrete model for the purpose of path planning, at the price of this model being, potentially, much larger than the reduced nerve $\mathcal{N}(\mathbb{U})$. Studying the homotopy properties of $\widetilde{\mathbf{R}}_{\mathbb{U}}$ in relation to those of X seems like a natural direction for continuing the present work, with an emphasis on obtaining some clarity regarding the conditions under which $\mathcal{N}_{red}(\mathbb{U})$ and $\widetilde{\mathbf{R}}_{\mathbb{U}}$ may be homotopy equivalent to X.

Author contributions All authors contributed equally to this work.

Funding This research was supported in part by AFOSR award numbers FA9550-19-1-0169, and FA9550-22-1-0429. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsoring agency.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

References

Baryshnikov, Y.: Topological perplexity of feedback stabilization. J. Appl. Comput. Topol. **7**, 75–87 (2023). https://doi.org/10.1007/s41468-022-00098-2

Bhattacharya, S., Lipsky, D., Ghrist, R., Kumar, V.: Invariants for homology classes with application to optimal search and planning problem in robotics. Ann. Math. Artif. Intell. **67**(3), 251–281 (2013). https://doi.org/10.1007/s10472-013-9357-7

Burridge, R.R., Rizzi, A.A., Koditschek, D.E.: Sequential composition of dynamically dexterous robot behaviors. Int. J. Robot. Res. **18**(6), 534–555 (1999). https://doi.org/10.1177/02783649922066385

Brockett, R.W.: Asymptotic Stability and Feedback Stabilization. In: Brockett, R.W., Millman, R.S., Sussman, H.J. (eds.) Differential Geometric Control Theory. Progress in Mathematics, vol. 27, pp. 181–191 (1983). https://courses.ece.ucsb.edu/ECE594/594D_W10Byl/hw/Brockett83.pdf

Baryshnikov, Y., Shapiro, B.: How to run a centipede: a topological perspective. In: Geometric Control Theory and Sub-Riemannian Geometry, pp. 37–51 (2014). https://doi.org/10.1007/978-3-319-02132-4_3

Belta, C., Sadraddini, S.: Formal methods for control synthesis: An optimization perspective. Annu. Rev. Control Robot. Auton. Syst. **2**(1), 115–140 (2019). https://doi.org/10.1146/annurev-control-053018-023717

Farber, M.: Topological complexity of motion planning. Discrete Comput. Geom. **29**(2), 211–221 (2003). https://doi.org/10.1007/s00454-002-0760-9

Fainekos, G.E., Girard, A., Kress-Gazit, H., Pappas, G.J.: Temporal logic motion planning for dynamic robots. Automatica **45**(2), 343–352 (2009). https://doi.org/10.1016/j.automatica.2008.08.008

Fainekos, G.E., Kress-Gazit, H., Pappas, G.J.: Hybrid controllers for path planning: A temporal logic approach. In: Proceedings of the 44th IEEE Conference on Decision and Control, pp. 4885–4890 (2005a). https://doi.org/10.1109/CDC.2005.1582935

Fainekos, G.E., Kress-Gazit, H., Pappas, G.J.: Temporal logic motion planning for mobile robots. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pp. 2020–2025. IEEE, Barcelona, Spain (2005b). https://doi.org/10.1109/ROBOT.2005.1570410

Holmes, P., Full, R.J., Koditschek, D.E., Guckenheimer, J.: The dynamics of legged locomotion: models, analyses, and challenges. SIAM Rev. **48**, 207–304 (2006). https://doi.org/10.1137/S0036144504445133

Karaman, S., Frazzoli, E.: Linear temporal logic vehicle routing with applications to multi-UAV mission planning. Int. J. Robust Nonlinear Control **21**(12), 1372–1395 (2011). https://doi.org/10.1002/rnc.1715

Kress Gazit, H., Fainekos, G.E., Pappas, G.J.: Temporal-logic-based reactive mission and motion planning. IEEE Trans. Rob. **25**(6), 1370–1381 (2009). https://doi.org/10.1109/TRO.2009.2030225

Kress Gazit, H., Lahijanian, M., Raman, V.: Synthesis for robots: Guarantees and feedback for robot behavior. Annu. Rev. Control Robot. Auton. Syst. **1**(1), 211–236 (2018). https://doi.org/10.1146/annurev-control-060117-104838

Koditschek, D.E.: Task encoding: Toward a Scientific Paradigm for Robot Planning and Control. Robot. Auton. Syst. **9**(1), 5–39 (1992). https://doi.org/10.1016/0921-8890(92)90031-S

Koditschek, D.E.: What is robotics? Why do we need it and how can we get it? Annu. Rev. Control Robot. Auton. Syst. **4**(1), 1–33 (2021). https://doi.org/10.1146/annurev-control-080320-011601

Kozlov, D.: Combinatorial Algebraic Topology. Algorithms and Computation in Mathematics, vol. 21. Springer, Berlin (2008). https://doi.org/10.1007/978-3-540-71962-5

Koditschek, D.E., Rimon, E.: Robot navigation functions on manifolds with boundary. Adv. Appl. Math. **11**(4), 412–442 (1990). https://doi.org/10.1016/0196-8858(90)90017-S

Karaman, S., Sanfelice, R.G., Frazzoli, E.: Optimal control of mixed logical dynamical systems with linear temporal logic specifications. In: 2008 47th IEEE Conference on Decision and Control, pp. 2117–2122 (2008). https://doi.org/10.1109/CDC.2008.4739370

Kvalheim, M.D.: Relationships Between Necessary Conditions for Feedback Stabilizability. (preprint) (2023)

Lamnabhi Lagarrigue, F., Annaswamy, A., Engell, S., Isaksson, A., Khargonekar, P., Murray, R.M., Nijmeijer, H., Samad, T., Tilbury, D., Hof, P.: Systems & Control for the Future of Humanity, Research Agenda: Current and Future Roles, Impact and Grand Challenges. Annu. Rev. Control **43**, 1–64 (2017)

Mavrogiannis, C., Balasubramanian, K., Poddar, S., Gandra, A., Srinivasa, S.S.: Winding through: Crowd navigation via topological invariance. IEEE Robot. Autom. Lett. **8**(1), 121–128 (2023). https://doi.org/10.1109/LRA.2022.3223024

Marinakis, D., Dudek, G.: Pure topological mapping in mobile robotics. IEEE Trans. Rob. **26**(6), 1051–1064 (2010). https://doi.org/10.1109/TRO.2010.2081410

Maierhofer, S., Moosbrugger, P., Althoff, M.: Formalization of intersection traffic rules in temporal logic. In: 2022 IEEE Intelligent Vehicles Symposium (IV), pp. 1135–1144 (2022). https://doi.org/10.1109/IV51971.2022.9827153

Maler, O., Nickovic, D.: Monitoring temporal properties of continuous signals. In: International Symposium on Formal Techniques in Real-time and Fault-tolerant Systems, pp. 152–166 (2004). https://doi.org/10.1007/978-3-540-30206-3_12 . Springer

Montgomery, R., Sanfelice, R.G.: 6. Hybrid Control, Morse Theory and Ivan Kupka **12**, 135–152 (2024)

Ouaknine, J., Worrell, J.: On the decidability of metric temporal logic. In: 20th Annual IEEE Symposium on Logic in Computer Science (LICS'05), pp. 188–197 (2005). https://doi.org/10.1109/LICS.2005.33 . IEEE

Pokorny, F., Hawasly, M., Ramamoorthy, S.: Multiscale topological trajectory classification with persistent homology. In: Proceedings of Robotics: Science and Systems X 2014, (2014). https://doi.org/10.15607/RSS.2014.X.054

Pappas, G.J., Lafferriere, G., Sastry, S.: Hierarchically consistent control systems. IEEE Trans. Autom. Control **45**(6), 1144–1160 (2000). https://doi.org/10.1109/9.863598

Ren, W., Calbert, J., Jungers, R.: Zonotope-Based Controller Synthesis for LTL Specifications. In: 2021 60th IEEE Conference on Decision and Control (CDC), pp. 580–585 (2021). https://doi.org/10.1109/CDC45484.2021.9683150

Rimon, E., Koditschek, D.E.: Exact robot navigation using artificial potential functions. IEEE Trans. Robot. Autom. **8**(5), 501–518 (1992). https://doi.org/10.1109/70.163777

Roohi, N., Viswanathan, M.: Revisiting mitl to fix decision procedures. In: International Conference on Verification, Model Checking, and Abstract Interpretation, pp. 474–494 (2017). https://doi.org/10.1007/978-3-319-73721-8_22 . Springer

Sickert, S., Esparza, J., Jaax, S., Křetínský, J.: Limit-deterministic büchi automata for linear temporal logic. In: Chaudhuri, S., Farzan, A. (eds.) Computer Aided Verification, pp. 312–332. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-41540-6_17

Shoukry, Y., Nuzzo, P., Balkan, A., Saha, I., Sangiovanni-Vincentelli, A.L., Seshia, S.A., Pappas, G.J., Tabuada, P.: Linear Temporal Logic Motion Planning for Teams of Underactuated Robots Using Satisfiability modulo Convex Programming. In: 2017 IEEE 56th Annual Conference on Decision and Control (CDC), pp. 1132–1137. IEEE, Melbourne, Australia (2017). https://doi.org/10.1109/CDC.2017.8263808

Sahin, Y.E., Nilsson, P., Ozay, N.: Multirobot coordination with counting temporal logics. IEEE Trans. Rob. **36**(4), 1189–1206 (2019)

Tokuda, S., Yamakita, M., Oyama, H., Takano, R.: Convex Approximation for LTL-based Planning. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 9863–9869 (2021). https://doi.org/10.1109/IROS51168.2021.9636801

Wells, M., Kingston, Z., Lahijanian, M., Kavraki, L.E., Vardi, M.Y.: Finite-Horizon Synthesis for Probabilistic Manipulation Domains. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 6336–6342. IEEE, Xi'an, China (2021). https://doi.org/10.1109/ICRA48506.2021.9561297

Wang, Y., Pajic, M.: Hyperproperties for robotics: Motion planning via HyperLTL. In: IEEE International Conference on Robotics and Automation, Paris, France, pp. 8462–8468 (2020). https://doi.org/10.1109/ICRA40945.2020.9196874