Formalization of Structural Resilience in Discrete-State Dynamical Systems

Boris Kriger
Institute of Integrative and Interdisciplinary Research
boriskriger@interdisciplinary-institute.org

Abstract

We present a formal framework for analyzing and ensuring resilience in discrete-state dynamical systems through structural properties of state transition graphs. While we focus on discrete-state systems, we discuss natural extensions to continuous and hybrid settings. The central contribution is the *Structural Resilience Principle*: a system can be made resilient to reaching unsafe states if and only if there exists a finite set of transitions whose removal disconnects all paths from operational to unsafe states. We prove that soft interventions (weight modifications) are fundamentally unstable under perturbation, while topological interventions (transition removal) provide guarantees independent of local dynamics. We situate this work within the broader literature on structural systems theory, graph-theoretic resilience, and formal verification, and demonstrate the framework through a concrete cybersecurity example. Extensions to residual threat analysis, liveness-preserving reconfiguration, and observability-aware design are developed, with discussion of connections to continuous and hybrid systems.

Keywords: structural resilience, dynamical systems, transition graphs, reachability analysis, safety verification, topological intervention, minimum cut

1 Introduction and Motivation

We consider dynamical systems whose evolution is determined by the structure of admissible state transitions. The fundamental objective is to design the transition structure such that certain classes of states remain unreachable from a specified set of initial states.

This structural approach complements traditional control-theoretic methods that rely on continuous parameter tuning, feedback design, or probabilistic bounds. Rather than asking whether a system *will* avoid dangerous states under nominal conditions, we ask whether it *can* reach them at all. When the answer is negative—when unsafe states are topologically disconnected from operational states—safety guarantees hold regardless of local dynamics, perturbations, or adversarial inputs.

1.1 Novelty and Contribution

This framework extends classical structural systems theory in a distinctive direction. Where Lin's structural controllability [1] characterizes when a system's state can be steered arbitrarily through parameter-independent properties of the (A, B) matrix pattern, we characterize when unsafe states can be made *unreachable* through parameter-independent properties of the transition graph. The key shift is summarized in Table 1.

Similarly, while supervisory control theory [14] asks how to restrict behavior to a legal language given fixed controllable/uncontrollable event partitions, we ask how to *reconfigure the transition structure itself* to achieve topological safety guarantees. The result is a framework where safety holds not "generically" or "for almost all parameters," but *absolutely*—for any local dynamics whatsoever.

Table 1: Comparison of classical structural theory and the present framework.

Classical Structural Theory	**This Framework**
Controllability of trajectories	Disconnection of reachability
Generic rank conditions	Graph-theoretic cuts
"Can we reach any state?"	"Can we prevent reaching unsafe states?"
Parameter-independence via matrix sparsity	Parameter-independence via topology

1.2 Scope and Limitations

We focus on discrete-state systems with finite or countably infinite state spaces, where transitions are explicitly enumerable. This framework directly applies to finite automata, Petri nets, discrete-event systems, and discretized approximations of continuous systems. Extensions to continuous-time, continuous-state, and hybrid systems require additional machinery discussed in Section 12.

The transition graph is assumed to be reconfigurable through deliberate intervention. While many physical systems have fixed dynamics, numerous engineered systems—network architectures, access control policies, software state machines, and configurable control systems—permit structural modification as a design or operational choice.

1.3 Safety-Liveness Trade-offs

A central tension in system design is the trade-off between *safety* (avoiding bad states) and *liveness* (achieving good states). Removing transitions to ensure safety may inadvertently block paths to desirable outcomes or create deadlocks. We address this explicitly in Section 4.3, introducing liveness-preserving reconfiguration constraints that ensure safety interventions do not compromise the system's ability to reach mission-critical states.

2 Related Work

2.1 Structural Systems Theory

The notion that system properties can be determined from structure rather than parameter values has deep roots. Lin's foundational work on structural controllability established that controllability of linear systems (A, B) can be generically determined from the zero/nonzero pattern of matrices alone, independent of specific numerical values [1]. Subsequent work extended these ideas to structural observability and structural stabilizability [2, 3].

Our framework shares this emphasis on structure over parameters but operates in a discrete-state setting where the relevant structure is a directed graph rather than a sparse matrix pattern. The key parallel is that both approaches seek properties that hold generically or universally across parameter variations.

2.2 Safety Verification and Reachability Analysis

Formal verification methods, particularly model checking and reachability analysis for hybrid systems, address related questions: can a system reach an unsafe state from a given initial set? Tools such as SPIN, UPPAAL, and SpaceEx compute reachable state sets and verify safety properties [4, 5, 6].

Our contribution differs in emphasis: rather than *verifying* whether an existing system is safe, we characterize *when and how* a system can be *made* safe through structural reconfiguration.

The Structural Resilience Principle (Theorem 6.1) provides necessary and sufficient conditions for the existence of such a reconfiguration.

2.3 Barrier Functions and Invariant Set Design

In continuous control theory, control barrier functions (CBFs) and invariant set methods ensure that system trajectories remain within safe regions [7, 8]. These approaches typically modify control inputs to render unsafe sets invariant or unreachable under the controlled dynamics.

Our topological intervention concept is analogous: removing transitions is equivalent to restricting the set of admissible control actions. The correspondence is clarified in Table 2.

Table 2: Correspondence between discrete and continuous safety frameworks.

Discrete (This Framework)	**Continuous (CBF/Invariant Sets)**
Transition graph (S, T)	Vector field $\dot{x} = f(x, u)$
Unsafe states B	Unsafe set $\mathcal{B} \subset \mathbb{R}^n$
Reachability $\text{Reach}_T(G)$	Forward reachable set $\mathcal{R}(G, t)$
Minimum (G, B)-cut	Control barrier function $h(x) \geq 0$
Transition removal	Input constraint $u \in \mathcal{U}_{\text{safe}}(x)$
Structural resilience	Forward invariance of safe set
Rule-independence (Theorem 10.3)	Robustness to control law variation

The structural guarantee we provide—unreachability under *any* local selection rule—is stronger than typical CBF guarantees, which depend on the specific control law and Lipschitz conditions.

2.4 Graph-Theoretic Resilience in Networks

The resilience of networked systems to node and edge failures has been extensively studied [9, 10]. Metrics such as connectivity, minimum cuts, and network robustness quantify how much damage a network can sustain before fragmentation. Algorithmic results on minimum cuts (Ford-Fulkerson, Dinic's algorithm, Karger's algorithm) provide efficient methods for identifying critical edges [11, 12, 13].

Our framework leverages these graph-theoretic tools but applies them to a different problem: rather than analyzing vulnerability to failures, we use cut-based reasoning to *design* systems where dangerous states are structurally isolated.

2.5 Discrete-Event and Supervisory Control

Ramadge-Wonham supervisory control theory addresses how to restrict a discrete-event system's behavior to satisfy specifications [14]. A supervisor disables controllable events to ensure the closed-loop system remains within a legal language.

Our reconfiguration operator $\mathcal{U}$ plays a role analogous to supervisory control: both restrict transitions to achieve safety. The key difference is that supervisory control typically assumes a fixed plant with controllable and uncontrollable events, while we consider reconfiguration of the transition structure itself.

2.6 Category-Theoretic Perspective

For readers in theoretical computer science, the reconfiguration operator $\mathcal{U}$ can be viewed category-theoretically. Transition systems form a category where objects are pairs (S, T) and morphisms are simulation relations. The operator $\mathcal{U} : T \mapsto T'$ with $T' \subseteq T$ defines a sub-object in this category. Structural resilience is then a property preserved by morphisms that respect the (G, B)-separation [21].

3 Formal Definitions

3.1 Basic Objects

Definition 3.1 (System Components)**.** Let:

- S be a non-empty, finite or countably infinite set of system states;
- $T \subseteq S \times S$ be the set of admissible transitions;
- $G \subseteq S$ be the set of operational (initial) states, with $G \neq \varnothing$;
- $B \subseteq S$ be the set of unsafe states, with $B \neq \varnothing$ and $G \cap B = \varnothing$;
- $M \subseteq S$ be the set of mission success (goal) states, with $M \cap B = \varnothing$.

The pair (S, T) forms a directed graph called the *transition graph*.

3.2 Reachability

Definition 3.2 (Reachability)**.** The reachability operator $\mathrm{Reach}_T : 2^S \to 2^S$ is defined as:

$$\mathrm{Reach}_T(A) = \{s \in S \mid \exists \text{ finite path } (a = s_0, s_1, \ldots, s_k = s) \text{ with } a \in A \text{ and } (s_i, s_{i+1}) \in T \ \forall i\}$$

By convention, $A \subseteq \mathrm{Reach}_T(A)$ (every state reaches itself via the empty path).

Remark 3.3 (Computation)**.** For finite graphs, $\mathrm{Reach}_T(A)$ is computable via breadth-first search in $O(|S|+|T|)$ time, or via algebraic methods using the transitive closure of the adjacency matrix.

3.3 Local Evolution Rule

Definition 3.4 (Local Evolution Rule)**.** A *local evolution rule* is a function $\pi : S \to S$ such that for all $s \in S$ with outgoing transitions, $\pi(s) \in \{s' \mid (s, s') \in T\}$.

We consider rules of the form $\pi(s) = \arg\min_{s':(s,s')\in T} \updownarrow(s, s')$ for some local cost functional $\updownarrow : S \times S \to \mathbb{R}$. The specific form of $\updownarrow$ determines local behavior, but structural resilience properties are independent of this choice.

4 Structural Reconfiguration

4.1 Reconfiguration Operator

Definition 4.1 (Reconfiguration)**.** A *reconfiguration operator* is a function $\mathcal{U} : 2^{S\times S} \to 2^{S\times S}$ mapping transition sets to transition sets. We focus on *subtractive* reconfigurations where $\mathcal{U}(T) \subseteq T$.

The set $\Delta T = T \setminus \mathcal{U}(T)$ represents transitions removed by the reconfiguration.

4.2 Definition of Structural Resilience

Definition 4.2 (Structural Resilience)**.** The system (S, T, G, B) is *structurally resilient* if:

$$\mathrm{Reach}_T(G) \cap B = \varnothing$$

Equivalently, for all $g \in G$ and $b \in B$, there is no directed path from g to b in the transition graph.

4.3 Liveness-Preserving Reconfiguration

Structural resilience ensures safety but does not guarantee liveness—the system's ability to reach desirable goal states. Removing transitions to isolate B might inadvertently disconnect paths to mission success states M, or create deadlocks.

Definition 4.3 (Liveness)**.** A transition structure T satisfies *liveness with respect to* M if for all $g \in G$:

$$\mathrm{Reach}_T(g) \cap M \neq \varnothing$$

Definition 4.4 (Deadlock-Freedom)**.** A transition structure T is *deadlock-free on* G if for all $s \in \mathrm{Reach}_T(G) \setminus M$, there exists s' such that $(s, s') \in T$.

Definition 4.5 (Valid Reconfiguration)**.** A reconfiguration $T' = \mathcal{U}(T)$ is *valid* if:

1. **Safety:** $\mathrm{Reach}_{T'}(G) \cap B = \varnothing$
2. **Liveness:** For all $g \in G$, $\mathrm{Reach}_{T'}(g) \cap M \neq \varnothing$
3. **Deadlock-freedom:** T' is deadlock-free on G

Proposition 4.6 (Existence of Valid Reconfiguration)**.** *A valid reconfiguration exists if and only if there exists a set* $\Delta T \subseteq T$ *such that:*

- ΔT *disconnects all paths from* G *to* B*, and*
- *For all* $g \in G$*, at least one path from* g *to some* $m \in M$ *avoids* ΔT*.*

Proof. Follows directly from the definitions and the path characterization of reachability. □

5 Soft Versus Topological Interventions

We introduce a weight function $w : T \to \mathbb{R}_{\geq 0}$ representing transition costs or penalties.

Definition 5.1 (Soft Intervention)**.** A *soft intervention* modifies the weight function: $(T, w) \mapsto (T, w')$ where $w' \neq w$ but the transition set T is unchanged.

Definition 5.2 (Topological Intervention)**.** A *topological intervention* (also called *edge-removal reconfiguration*) removes transitions: $(T, w) \mapsto (T', w|_{T'})$ where $T' \subsetneq T$.

5.1 Comparative Analysis

The distinction between soft and topological interventions involves fundamental trade-offs, summarized in Table 3.

Table 3: Comparison of intervention types.

Feature	**Soft Intervention**	**Topological Intervention**
Mechanism	Cost/penalty modification	Edge removal (disconnection)
Guarantee	Probabilistic/nominal	Absolute (topological)
Robustness	Sensitive to ϵ-perturbation	Rule-independent
Complexity	Continuous optimization	Discrete/graph algorithms
Reversibility	Easily adjustable	May require redesign
Functional impact	Preserves all paths	May reduce functionality

5.2 The Cost of Topological Interventions

While topological interventions provide stronger guarantees, they often carry higher functional and implementation costs:

- **Cybersecurity:** Removing a transition means closing a port, revoking a privilege, or air-gapping a network segment.
- **Power grids:** Removing a transition corresponds to opening a circuit breaker or decommissioning a transmission line.
- **Software systems:** Removing a transition may require code changes, API deprecation, or architectural refactoring.
- **Physical systems:** Many transitions reflect inherent physics and cannot be "removed" without physical modification.

5.3 Instability of Soft Constraints

To formalize the instability of soft interventions, we introduce a perturbation model.

Definition 5.3 (ϵ-Perturbed Selection)**.** Given a local cost functional $\updownarrow$ and $\epsilon > 0$, an *ϵ-perturbed selection rule* at state s selects any $s' \in \{s'' \mid (s, s'') \in T\}$ satisfying:

$$\updownarrow(s, s') \leq \min_{s'':(s,s'')\in T} \updownarrow(s, s'') + \epsilon$$

Lemma 5.4 (Instability of Soft Constraints)**.** *Let $(s_0, s_1, \ldots, s_k)$ be a path from G to B in T. For any weight function w with $w(s_i, s_{i+1}) < \infty$ for all i, and any $\epsilon > 0$, there exists an ϵ-perturbed selection rule under which this path is traversed with positive probability.*

Proof. Define $W = \max_i w(s_i, s_{i+1})$ and $\updownarrow(s, s') = w(s, s')$. At each state s_i, the transition (s_i, s_{i+1}) satisfies $\updownarrow(s_i, s_{i+1}) = w(s_i, s_{i+1}) \leq W$.

Choose $\epsilon > W - \min_{s':(s_i,s')\in T} w(s_i, s')$ (which is finite since $W < \infty$). Under ϵ-perturbed selection, transition (s_i, s_{i+1}) is admissible at each step. If ties are broken uniformly at random among admissible transitions, the path is selected with probability at least:

$$\prod_{i=0}^{k-1} \frac{1}{|\{s' : (s_i, s') \in T,\ w(s_i, s') \leq w(s_i, s_{i+1}) + \epsilon\}|} > 0$$

since each factor is positive. □

Corollary 5.5. *Soft interventions cannot guarantee* $\text{Reach}_T(G) \cap B = \varnothing$ *under ϵ-perturbed selection for any $\epsilon > 0$. Only removal of transitions provides structural guarantees.*

6 The Structural Resilience Principle

Theorem 6.1 (Structural Resilience Principle)**.** *There exists a subtractive reconfiguration $T' = \mathcal{U}(T)$ such that* $\text{Reach}_{T'}(G) \cap B = \varnothing$ *if and only if there exists a finite set $\Delta T \subseteq T$ such that removing ΔT disconnects all directed paths from G to B.*

Proof. ($\Rightarrow$) Suppose $T' = \mathcal{U}(T)$ achieves $\text{Reach}_{T'}(G) \cap B = \varnothing$. Define $\Delta T = T \setminus T'$.

Assume for contradiction that ΔT does not disconnect all paths from G to B. Then there exists a path $(g = s_0, s_1, \ldots, s_k = b)$ with $g \in G$, $b \in B$, and $(s_i, s_{i+1}) \in T'$ for all i. But then $b \in \text{Reach}_{T'}(G)$, contradicting $\text{Reach}_{T'}(G) \cap B = \varnothing$. Hence ΔT disconnects all paths.

($\Leftarrow$) Suppose $\Delta T \subseteq T$ is finite and disconnects all paths from G to B. Define $T' = T \setminus \Delta T$.

Assume for contradiction that $\text{Reach}_{T'}(G) \cap B \neq \varnothing$. Then there exists $b \in B$ and a path from some $g \in G$ to b using only transitions in T'. This path uses no transitions from ΔT, contradicting that ΔT disconnects all G-to-B paths. Hence $\text{Reach}_{T'}(G) \cap B = \varnothing$. □

Remark 6.2 (Graph-Theoretic Context)**.** Theorem 6.1 is an instantiation of the classical *cut-connectivity duality* in graph theory: a set of vertices (or edges) separates two vertex sets if and only if every path between them passes through the separating set.

Corollary 6.3. *Structural resilience is a topological property of the transition graph, independent of edge weights or local evolution rules.*

Corollary 6.4. *The minimum number of transitions whose removal achieves structural resilience equals the minimum (G, B)-cut in the transition graph.*

6.1 Computational Aspects

For finite transition graphs, the minimum (G, B)-cut can be computed in polynomial time. Introducing a super-source g^* connected to all $g \in G$ and a super-sink b^* connected from all $b \in B$, the minimum (G, B)-cut equals the minimum (g^*, b^*)-cut.

Algorithm complexity:

- Dinic's algorithm [12] computes maximum flow in $O(|S|^2|T|)$ time, or $O(|T|\sqrt{|S|})$ for unit-capacity graphs.
- The Karger-Stein randomized algorithm [13, 15] finds minimum cuts in $O(|S|^2 \log^3 |S|)$ expected time.
- Ford-Fulkerson with BFS (Edmonds-Karp) runs in $O(|S||T|^2)$ time.

Scalability for large systems: For very large graphs, modern approaches include symbolic representations via BDDs [16], parallel push-relabel implementations [22], and flow-based partitioning [23].

7 Illustrative Example: Access Control Reconfiguration

Consider a simplified network security model with states representing access levels:

$$S = \{\text{External}, \text{DMZ}, \text{Internal}, \text{Admin}, \text{Database}\}$$

The initial transition set T includes:

- $(\text{External}, \text{DMZ})$, $(\text{DMZ}, \text{Internal})$, $(\text{DMZ}, \text{External})$
- $(\text{Internal}, \text{Admin})$, $(\text{Internal}, \text{DMZ})$
- $(\text{Admin}, \text{Database})$, $(\text{Admin}, \text{Internal})$

With $G = \{\text{External}\}$ (attacker starting position), $B = \{\text{Database}\}$ (asset to protect), and $M = \{\text{Admin}\}$ (legitimate administrative access).

Analysis: The path External $\to$ DMZ $\to$ Internal $\to$ Admin $\to$ Database shows $\text{Reach}_T(G) \cap B \neq \varnothing$. The system is not structurally resilient.

Soft Intervention Attempt: Assign $w(\text{Admin}, \text{Database}) = 1000$. By Lemma 5.4, any ϵ-perturbed selection can still traverse this path.

Topological Intervention: The minimum (G, B)-cut is $\{(\text{Admin}, \text{Database})\}$ (size 1). Removing this transition yields T' with $\text{Reach}_{T'}(\text{External}) = \{\text{External}, \text{DMZ}, \text{Internal}, \text{Admin}\}$. Since Database $\notin \text{Reach}_{T'}(G)$, structural resilience is achieved.

Liveness Verification: Under T', the path External $\to$ DMZ $\to$ Internal $\to$ Admin shows $M \cap \text{Reach}_{T'}(G) \neq \varnothing$. The reconfiguration is valid.

8 Residual Threat Potential

Definition 8.1 (Predecessor Set)**.** The *predecessor set* of B is:

$$\text{Pre}_T(B) = \{s \in S \mid \text{Reach}_T(s) \cap B \neq \varnothing\}$$

Definition 8.2 (Residual Threat Function)**.** The *residual threat function* $R_T : S \to \mathbb{Z}_{\geq 0}$ is:

$$R_T(s) = |\operatorname{Reach}_T(s) \cap \operatorname{Pre}_T(B)|$$

Remark 8.3 (Computation)**.** The residual threat function is the cardinality of the intersection of two reachability sets, both computable via graph traversal in $O(|S| + |T|)$ time each, or via transitive closure of the adjacency matrix.

Proposition 8.4. *Structural resilience* ($\operatorname{Reach}_T(G) \cap B = \varnothing$) *implies* $R_T(g) = 0$ *for all* $g \in G$.

Proof. If $\operatorname{Reach}_T(G) \cap B = \varnothing$, then for all $g \in G$, no state in $\operatorname{Reach}_T(g)$ can reach B. Hence $\operatorname{Reach}_T(g) \cap \operatorname{Pre}_T(B) = \varnothing$, giving $R_T(g) = 0$. □

9 Observability and Signal Structure

Definition 9.1 (Observation Mapping)**.** An *observation mapping* is a function $O : S \to \Sigma$ projecting internal states to an observable signal space Σ.

Definition 9.2 (Extended Reconfiguration)**.** An *extended reconfiguration* modifies both transitions and observations:

$$\mathcal{U} : (T, O) \mapsto (T', O')$$

Proposition 9.3. *If* $\operatorname{Reach}_T(G) \cap B = \varnothing$*, then safety is preserved under any observation mapping* O *and any observation-based local rule.*

Proof. By Theorem 10.3, structural resilience implies rule-independence. Since observation-based rules are a subset of all local rules, the result follows. □

Remark 9.4. Observability remains relevant in two contexts: (1) when the system is *not* structurally resilient, observability influences which paths are likely to be taken; (2) in adaptive reconfiguration settings, observability determines which threats can be detected.

10 Robustness to Local Evolution Rules

Definition 10.1 (Rule Class)**.** A *rule class* Π is a set of local evolution rules $\pi : S \to S$.

Definition 10.2 (Robustness)**.** A transition structure T is *robust with respect to* Π if, for every $\pi \in \Pi$, all trajectories starting from G avoid B.

Theorem 10.3 (Rule-Independence)**.** *If* $\operatorname{Reach}_T(G) \cap B = \varnothing$*, then* T *is robust with respect to any rule class* Π.

Proof. A trajectory under any $\pi \in \Pi$ is a path in the transition graph. If no path from G to B exists, no trajectory can reach B. The rule π selects among existing transitions but cannot create new ones. □

Corollary 10.4. *Structural resilience provides guarantees that hold uniformly across all possible local dynamics, including adversarial, stochastic, and unknown dynamics.*

11 Summary of Main Results

Table 4 summarizes the main theoretical contributions.

Table 4: Summary of main results.

Result	Statement	Significance
Lemma 5.4	Soft constraints unstable under perturbation	Motivates topological interventions
Theorem 6.1	Resilience $\Leftrightarrow$ separating cut exists	Characterizes achievability
Theorem 10.3	Resilience implies rule-independence	Universal guarantees
Proposition 4.6	Valid reconfig. requires safety + liveness	Addresses deadlock concerns
Proposition 8.4	Resilience implies zero residual threat	Eliminates latent risk

12 Extensions and Connections

12.1 Continuous and Hybrid Systems

The discrete framework does not directly apply to continuous-state systems, but several connections exist:

Abstraction-Based Approaches: Continuous systems can be abstracted to finite transition systems that over-approximate reachable sets [17]. If the abstraction is structurally resilient, so is the concrete system.

Hybrid Automata: Guard conditions in hybrid automata—Boolean predicates that enable mode transitions—correspond directly to edges in our transition graph. Disabling a guard is equivalent to removing a transition.

12.2 Stochastic Transitions

When transitions have associated probabilities, structural resilience ($\text{Reach}_T(G) \cap B = \varnothing$) still implies zero probability of reaching B. Partial resilience requires different metrics and falls outside our structural guarantee framework.

12.3 Adaptive Reconfiguration

In dynamic environments, the reconfiguration operator might be applied repeatedly. Online algorithms for maintaining minimum cuts under edge insertions and deletions enable adaptive structural resilience [18].

12.4 Multi-Objective Reconfiguration

Different cuts may have different costs. Computational approaches include weighted minimum cut, integer linear programming [19], and parametric minimum cut [20].

13 Conclusion

We have established a formal framework for structural resilience in discrete-state dynamical systems. The central results are:

1. **Instability of Soft Constraints:** Weight-based interventions cannot guarantee unreachability under perturbation; only topological interventions provide structural guarantees.
2. **Structural Resilience Principle:** A system can be made structurally resilient if and only if a finite separating cut exists between operational and unsafe states.
3. **Rule-Independence:** Structural resilience holds uniformly across all local evolution rules.
4. **Liveness Preservation:** Valid reconfigurations must jointly satisfy safety and liveness constraints.

The framework transforms resilience engineering from continuous optimization over parameters to discrete optimization over graph structure, enabling application of efficient graph algorithms while providing stronger guarantees than probabilistic approaches.

References

[1] C.-T. Lin, "Structural controllability," *IEEE Transactions on Automatic Control*, vol. 19, no. 3, pp. 201–208, 1974.

[2] K. Murota, *Matrices and Matroids for Systems Analysis*. Springer, 2000.

[3] J.-M. Dion, C. Commault, and J. van der Woude, "Generic properties and control of linear structured systems: A survey," *Automatica*, vol. 39, no. 7, pp. 1125–1144, 2003.

[4] G. J. Holzmann, *The SPIN Model Checker: Primer and Reference Manual*. Addison-Wesley, 2004.

[5] G. Frehse *et al.*, "SpaceEx: Scalable verification of hybrid systems," in *Proc. CAV*, 2011, pp. 379–395.

[6] K. G. Larsen, P. Pettersson, and W. Yi, "UPPAAL in a nutshell," *International Journal on Software Tools for Technology Transfer*, vol. 1, no. 1–2, pp. 134–152, 1997.

[7] A. D. Ames *et al.*, "Control barrier functions: Theory and applications," in *Proc. ECC*, 2019, pp. 3420–3431.

[8] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[9] R. Albert, H. Jeong, and A.-L. Barabási, "Error and attack tolerance of complex networks," *Nature*, vol. 406, pp. 378–382, 2000.

[10] D. S. Callaway *et al.*, "Network robustness and fragility: Percolation on random graphs," *Physical Review Letters*, vol. 85, no. 25, pp. 5468–5471, 2000.

[11] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. MIT Press, 2009.

[12] E. A. Dinic, "Algorithm for solution of a problem of maximum flow in networks with power estimation," *Soviet Mathematics Doklady*, vol. 11, pp. 1277–1280, 1970.

[13] D. R. Karger, "Minimum cuts in near-linear time," *Journal of the ACM*, vol. 47, no. 1, pp. 46–76, 2000.

[14] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM Journal on Control and Optimization*, vol. 25, no. 1, pp. 206–230, 1987.

[15] D. R. Karger and C. Stein, "A new approach to the minimum cut problem," *Journal of the ACM*, vol. 43, no. 4, pp. 601–640, 1996.

[16] R. E. Bryant, "Symbolic Boolean manipulation with ordered binary-decision diagrams," *ACM Computing Surveys*, vol. 24, no. 3, pp. 293–318, 1992.

[17] P. Tabuada, *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer, 2009.

[18] M. Thorup, "Fully-dynamic min-cut," *Combinatorica*, vol. 27, no. 1, pp. 91–127, 2007.

[19] H. W. Hamacher and G. Ruhe, "On spanning tree problems with multiple objectives," *Annals of Operations Research*, vol. 52, no. 4, pp. 209–230, 1994.

[20] D. Gusfield and É. Tardos, "A faster parametric minimum-cut algorithm," *Algorithmica*, vol. 11, no. 3, pp. 278–290, 1994.

[21] R. Milner, *Communication and Concurrency*. Prentice Hall, 1989.

[22] A. V. Goldberg and R. E. Tarjan, "A new approach to the maximum-flow problem," *Journal of the ACM*, vol. 35, no. 4, pp. 921–940, 1988.

[23] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 359–392, 1998.