

## Tightening the Loop Between Specification and Computation: BCDS Year Two

As the *Bulletin of Computer and Data Sciences* (BCDS) enters its second year, we have the privilege of reflecting on how a young venue can already shape conversations across theory, algorithms, and practice. The papers in this volume continue the journal’s founding promise: methodologically careful work with an eye toward real computational impact. They also illustrate a pattern we hope to keep encouraging—results that travel well. The best ideas in one subfield often become the missing instruments in another, and this year’s contributions remind us that bridges between areas are not luxuries but necessities for progress.

In *Cost-Optimal Reachability for Timed Automata with Diagonals via Zone Simulations*, the classical tension between expressiveness and tractability is treated with uncommon clarity. Timed automata with diagonal constraints capture realistic coordination among clocks but have long tested the limits of analysis. By advancing zone-based abstractions for cost-optimal reachability in the presence of diagonals, the paper shows how to preserve the semantic fidelity practitioners need while regaining algorithmic handleability. The contribution is not merely an asymptotic line in a proof; it is a design principle: if we get the symbolic abstraction right—refined enough to honor dependencies, economical enough to admit effective search—then optimality questions become accessible rather than aspirational. The implications for real-time systems are immediate. Schedulers, controllers, and embedded protocols stand to benefit when optimality is not a postscript but a property we can compute with confidence. Beyond the specifics, the paper underlines an editorial value we prize: marrying formal soundness to implementable procedures that others can extend and reuse.

*From Parikh Vectors to Parikh Matrices: Structure, Ambiguity, and Complexity* revisits a venerable notion—Parikh’s commutative image of a word—and carries it into a richer algebraic signature that remembers just enough order to be interesting. Parikh vectors flatten words to letter counts; Parikh matrices retain patterns of scattered subwords, opening a subtle terrain where structure and ambiguity collide. The authors map that terrain with three complementary lenses. Structurally, they identify invariants that explain what these matrices can and cannot distinguish. In terms of ambiguity, they chart when different words share the same matrix and why, clarifying the boundary between descriptive power and collapse. On the complexity side, they investigate core decision problems associated with these representations, drawing lines between what is feasible and what likely resists efficient algorithms. For BCDS, the lesson is twofold. First, small algebraic shifts can unlock large explanatory dividends. Second, clarity about ambiguity is not a failure of a representation but a form of understanding: knowing where expressiveness ends is as valuable as pushing it further. This is the kind of synthesis—conceptual, technical, and boundary-aware—that helps multiple subfields adopt a common language.

*Hybridizing Expansion and Search for QBF* offers a pragmatic manifesto for solver design in the age of heterogeneous instance families. Expansion-based reasoning and clause-learning search each

---

have proud traditions in quantified Boolean formulas, yet each has instance regimes where it shines and others where it strains. The paper’s central contribution is to treat hybridization not as a loose ensemble but as a carefully engineered dialogue between proof systems and heuristics. Expansion can simplify the quantifier prefix and expose structure; conflict-driven search can exploit learned information to prune vast spaces. The key is orchestration: deciding when to switch, how to share information without polluting invariants, and how to preserve certification so that solver performance does not outpace trust. By articulating mechanisms for information flow and by evaluating robustness across benchmarks with different structural signatures, the work exemplifies the *systems of reasoning* ethos we believe will shape the next wave of automated deduction and verification.

Taken together, these papers sketch a coherent picture of where computer and data sciences are heading. Symbolic abstractions that are faithful yet frugal; algebraic representations that balance expressiveness with analyzability; solver architectures that integrate complementary proof strategies rather than treating them as rivals. If there is a theme to our second year, it is that the most durable advances are those that tighten the loop between specification and computation: more precise models that still compile to efficient procedures; richer invariants that we can compute and compare; proof-producing systems that earn their results. That loop, once shortened, changes practice. It moves optimal scheduling from hope to method, word-combinatorics from intuition to tested boundary cases, and QBF solving from a collection of tactics to a principled conversation among them.

We are equally encouraged by how these projects embody the cultural commitments BCDS was founded on. The authors make their claims testable. They articulate assumptions and limitations. Where code, data, or benchmarks can be shared, they are; where they cannot, procedures and environments are specified so others can reconstruct results. Reviewers responded in kind, engaging not only with novelty but with clarity, soundness checks, and opportunities for sharpening exposition. This is how a healthy record of knowledge grows: by pairing ambition with verification and by valuing intelligibility as part of impact.

A journal’s second year is where aspirations become habits. Our habits will be simple ones. We will keep favoring work that explains as well as it invents. We will keep encouraging artifacts that let the community verify, compare, and build. We will keep inviting conversations across boundaries—between formal methods and learning, between combinatorics and language technology, between proof theory and systems engineering—because that is where many of tomorrow’s tools are quietly being forged today.

To the authors who entrusted BCDS with their best ideas, to the reviewers whose careful readings improved every paper they touched, and to the readers who will carry these results into classrooms, tools, and future theorems: thank you. Our measure of success will remain the same as in our first year—not citation counts alone, but whether the work we publish becomes something others can *use*. The papers highlighted here make a strong case that we are on the right path.

— Sayan Mukherjee  
Editor in Chief