

A Generative View of Cellular Automata: Language Generators via Atomic Gliders

Noa Izsak¹  and Dana Fisman² 

¹ CISP A Helmholtz Center for Information Security, Saarbrücken, Germany

² Ben Gurion University, Beer-Sheva, Israel

✉ noa.izsak@cispa.de

Introduction. Cellular automata (CA) are classical models of distributed computation and emergent behavior, in which simple local rules give rise to complex global dynamics. Beyond their role as dynamical systems, CA have also been studied in connection with formal language theory, most notably as language *recognizers* or as systems preserving language complexity under evolution [2–4]. In this work, we present a complementary perspective, recently introduced in [1], in which CA are viewed as genuine *language generators*. Rather than asking which languages can be recognized or preserved by a CA, we study the words induced by its evolution over time. Concretely, we focus on one-dimensional CA defined over bi-infinite grids, with arbitrary finite alphabets and neighborhood radii, evolving from sets of finite initial configurations. The generative view shifts the focus from recognition-based or invariance-based settings to unbounded space and time, enabling a symbolic analysis of CA dynamics. It further enables a structured account of the generated words via *atomic gliders*—a formally defined class of single-cell propagating structures whose interactions shape the resulting language. This perspective also connects to formal verification via language-theoretic analysis of the induced language.

Model and Generative View. A *one-dimensional cellular automaton* (CA) is a tuple $\mathcal{A} = (\Sigma, r, f)$, where: Σ is a finite alphabet, $r \in \mathbb{N} \cup \{0\}$ is the *radius*, and $f: \Sigma^{2r+1} \rightarrow \Sigma$ is the *local transition function*. The global transition maps each configuration $c \in \Sigma^{\mathbb{Z}}$ to its successor by applying f synchronously at every cell. We assume a designated *quiescent* state $\perp \in \Sigma$, satisfying $f(\perp \cdots \perp) = \perp$. A configuration is *finite* if it contains only finitely many non-quiescent cells; quiescence is then preserved along the entire orbit. Given a CA $\mathcal{A}$ and a finite initial configuration c_0 , the forward orbit is $\mathcal{O}(c_0) = \{c_t\}_{t \geq 0}$, where each c_{t+1} is obtained by applying the global transition to c_t . For each t , let $\text{supp}(c_t) = [l_t, r_t] \subseteq \mathbb{Z}$ denote the minimal interval containing all non-quiescent cells of c_t ; reading c_t over this interval yields a finite word $w_t \in \Sigma^*$. Given a set I of finite initial configurations, we define the *language induced by $\mathcal{A}$ from I* as $\mathcal{L}(\mathcal{A}, I) = \{w_t \mid c_0 \in I, c_t \in \mathcal{O}(c_0)\}$. This definition captures precisely the words that arise as the non-quiescent support of reachable configurations, and forms the basis of the generative view of CA. Following the regular model checking paradigm [1], we take I to be specified by a regular language over finite configurations, which allows succinctly describing families of initial configurations of unbounded size.

Atomic Gliders. The structure of $\mathcal{L}(\mathcal{A}, I)$ is analyzed in [1] through the emergence of localized structures called *atomic gliders*. Unlike classical gliders – which are multi-cell patterns that recur periodically with spatial displacement—an atomic glider is a *single-cell* entity carrying a symbol at a fixed velocity (over the integers); its persistence and interaction behavior are derived directly from the local rule f . When gliders with differing symbols and velocities co-occur, a *dominance relation* on the glider set (derived from f), determines which prevails, giving rise to the class of *(atomic) glider-expressible* CA languages ($\mathbb{CA}^{\text{Glider}}$) as a proper subclass of all CA-expressible languages ($\mathbb{CA}$). Since each glider travels at a fixed velocity, the boundaries of distinct segments of the support grow linearly in time, which is the structural reason that the induced languages exhibit the linear-exponent form described in the following.

Expressiveness. The results of [1] show that even when I is a regular language, $\mathcal{L}(\mathcal{A}, I)$ can exhibit significantly richer structure. Consider languages of the form $L = \{w_1^{e_1(n)} \cdots w_m^{e_m(n)} \mid n \in \mathbb{N}\}$, where each $w_i \in \Sigma^+$ and $e_i(n) = a_i \cdot n + b_i$ is a positive linear function. The main result of [1] establishes that any language of this form that is $\mathbb{CA}$ -expressible is also $\mathbb{CA}^{\text{Glider}}$ -expressible: within this family, glider semantics is complete for CA expressibility. In particular, this framework generates languages such as $\{a^n b^n c^n d^n \mid n \in \mathbb{N}\}$, which lie strictly beyond context-free languages. All such languages are bounded (contained in $w_1^* \cdots w_m^*$) since fixed glider velocities bound the rate at which the support can expand. Moreover, each such language has a linear Parikh image; by Parikh’s theorem, every bounded context-free language has a semi-linear Parikh image and can thus be expressed as a finite union of languages of this form.

Conclusions. The generative view positions CA as language generators whose expressive power (even under regular initial configurations) can surpass the context-free languages, while the generated languages remain bounded and carry a linear Parikh image. Atomic gliders provide the structural account for this expressiveness, and within the family of linearly growing repeated blocks, glider semantics characterizes exactly the CA-expressible languages. This perspective connects naturally to formal verification: reachability questions over $\mathcal{L}(\mathcal{A}, I)$, against regular specifications remain decidable when the generated language is regular or context-free, but require new techniques for the non-context-free cases that the framework exposes.

References

1. Fisman, D., Izsak, N.: Atomic gliders and cellular automata as language generators. In: International Conference on Verification, Model Checking, and Abstract Interpretation. pp. 58–81. Springer (2026), arXiv preprint arXiv:2511.12656
2. Nordahl, M.G.: Formal languages and finite cellular automata. *Comp. Syst.* (1989)
3. Smith, A.: Cellular automata and formal languages. In: FOCS. pp. 216–224 (1970)
4. Wolfram, S.: Statistical mechanics of cellular automata. *RMP* **55**(3), 601 (1983)