Linear Conjunctive Reachability as Tensor Completion

Anonymous Author(s)

Abstract

Brzozowski (1964) defines a regular expression derivative as the suffixes which complete a known prefix. In this work, we establish a Galois connection with Valiant's (1975) fixpoint construction in the context-free setting, and further extend their work into the hierarchy of bounded context-sensitive languages realizable by finite CFL intersection. We then show how to decide various language recognition, intersection and membership queries using multilinear systems of equations over finite fields. In addition to its theoretical value, this connection demonstrates a number of practical applications in incremental parsing, code completion and program repair.

1 Introduction

Recall that a CFG is a quadruple consisting of terminals (Σ), nonterminals (V), productions ($P\colon V \to (V \mid \Sigma)^*$), and a start symbol, ($S$). All CFGs are reducible to *Chomsky Normal Form*, $P'\colon V \to (V^2 \mid \Sigma)$, where every production has either the form $w \to xz$, or $w \to t$, where $w, x, z : V$ and $t : \Sigma$. Given a CFG, $\mathcal{G}' : \langle \Sigma, V, P, S\rangle$ in CNF, we can construct a recognizer $R : \mathcal{G}' \to \Sigma^n \to \mathbb{B}$ for strings $\sigma : \Sigma^n$ as follows. Let 2^V be our domain, 0 be $\varnothing$, $\oplus$ be $\cup$, and $\otimes$ be defined as:

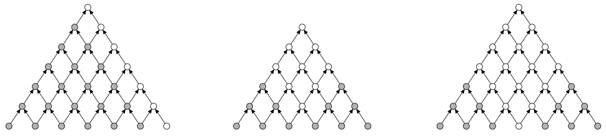
$$X \otimes Z := \left\{\, w \mid \langle x, z\rangle \in X \times Z, (w \to xz) \in P \,\right\} \qquad (1)$$

If we define $\sigma_r^{⤉} := \{w \mid (w \to \sigma_r) \in P\}$, then initialize $M^0_{r+1=c}(\mathcal{G}', e) := \sigma_r^{⤉}$ and solve for the fixpoint $M^* = M + M^2$,

$$M^0 := \begin{pmatrix} \varnothing & \sigma_1^{\rightarrow} & \varnothing & \cdots & \varnothing \\ \vdots & & \ddots & \ddots & \vdots \\ & & & \ddots & \varnothing \\ & & & & \sigma_n^{\uparrow} \\ \varnothing & \cdots & \cdots & \cdots & \varnothing \end{pmatrix} \Rightarrow M^* = \begin{pmatrix} \varnothing & \sigma_1^{\rightarrow} & \Lambda & \cdots & \Lambda_\sigma^* \\ \vdots & & \ddots & \ddots & \vdots \\ & & & \ddots & \Lambda \\ & & & & \sigma_n^{\uparrow} \\ \varnothing & \cdots & \cdots & \cdots & \varnothing \end{pmatrix}$$

we obtain the recognizer, $R(\mathcal{G}', \sigma) := S \in \Lambda_\sigma^*? \Leftrightarrow \sigma \in \mathcal{L}(\mathcal{G})?$ Full details of the bisimilarity between parsing and matrix multiplication can be found in Valiant [8] and Lee [4], who shows its complexity to be $\mathcal{O}(n^\omega)$ where $\omega < 2.77$.

Further optimizations, e.g., incremental Levenshtein edits with quadratic cost in terms of $|\Sigma^*|$ are possible under mild sparsity assumptions. Depicted below as trellis automata [5], incremental parsing is also closely related to *dynamic matrix inversion* [6] in the linear algebraic setting, and *incremental transitive closure* in the literature on dynamic graphs [3].



Append: $\mathcal{O}(n+1)$ Delete: $\mathcal{O}\left(\frac{1}{4}(n-1)^2\right)$ Insert: $\mathcal{O}\left(\frac{1}{4}(n+1)^2\right)$

2 Galois representation

Note that $\bigoplus_{c=1}^{n} M_{r,c} \otimes M_{c,r}$ has cardinality bounded by $|V|$ and is thus representable as a fixed-length vector using the characteristic function, $\mathbb{1}$. In particular, $\oplus, \otimes$ are redefined as $\boxplus, \boxtimes$ over bitvectors so the following diagram commutes,[1]

$$\begin{tikzcd}
\langle \mathcal{G}', \Sigma^{n-1} \rangle \arrow[dr, leftrightarrow] & \text{Set} & \text{Bit} & \langle \mathcal{G}', \Sigma^{n-1} \rangle \arrow[dr] & \text{SAT} \\
\text{Rubix} & M \times M \arrow[r, leftrightarrow, "\mathbb{1}^{2^{n\times n}}", "\mathbb{1}^{-2^{n\times n}}"'] \arrow[d, "\oplus"', "\otimes"] & \mathbb{F}_2^{|V|^{n\times n}} \times \mathbb{F}_2^{|V|^{n\times n}} \arrow[rr, tail, "\varphi^{2^{n\times n}}", "\varphi^{-2^{n\times n}}"'] \arrow[d, "+"', "*"] & & \mathcal{M} \times \mathcal{M} \arrow[d, leftrightarrow, "+"', "*"] \\
\text{Matrix} & 2^V \times 2^V \arrow[r, leftrightarrow, "\mathbb{1}^2", "\mathbb{1}^{-2}"'] \arrow[d, "\oplus"', "\otimes"] & \mathbb{F}_2^{|V|} \times \mathbb{F}_2^{|V|} \arrow[rr, tail, "\varphi^2", "\varphi^{-2}"'] \arrow[d, "\boxplus"', "\boxtimes"] & & \mathcal{V} \times \mathcal{V} \arrow[d, leftrightarrow, "\boxplus"', "\boxtimes"] \\
\text{Vector} & 2^V \arrow[r, leftrightarrow, "\mathbb{1}", "\mathbb{1}^{-1}"'] & \mathbb{F}_2^{|V|} \arrow[rr, tail, "\varphi", "\varphi^{-1}"'] & & \mathcal{V}
\end{tikzcd}$$

where $\mathcal{V}$ is a function $\mathbb{F}_2^{|V|} \to \mathbb{F}_2$. Note that while always possible to encode $\mathbb{F}_2^{|V|} \to \mathcal{V}$ using the identity function, φ^{-1} may not exist, as an arbitrary $\mathcal{V}$ might have zero, one, or in general, multiple solutions in $\mathbb{F}_2^{|V|}$. Although holes may occur anywhere, let us consider two cases in which Σ^+ is strictly left- or right-constrained, i.e., $x\,z, x\,z : \Sigma^{|x|+|z|}$.

Valiant's $\otimes$ operator, which yields the set of productions unifying known factors in a binary CFG, naturally implies the existence of a left- and right-quotient, which yield the set of nonterminals that may appear the right or left side of a known factor and its corresponding root. In other words, a known factor not only implicates subsequent expressions that can be derived from it, but also adjacent factors that may be composed with it to form a given derivation.

Left Quotient

$$\frac{\partial}{\partial \overleftarrow{x}} = \left\{\, z \mid (w \to xz) \in P \,\right\}$$

$$\begin{array}{|c|c|} \hline x & w \\ \hline \multicolumn{1}{c|}{} & z \\ \cline{2-2} \end{array}$$

Right Quotient

$$\frac{\partial}{\partial \overrightarrow{z}} = \left\{\, x \mid (w \to xz) \in P \,\right\}$$

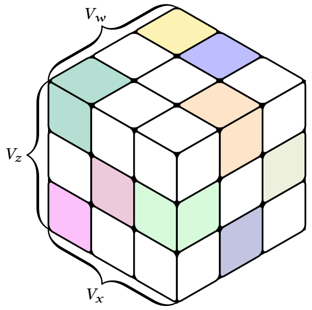
$$\begin{array}{|c|c|} \hline x & w \\ \hline \multicolumn{1}{c|}{} & z \\ \cline{2-2} \end{array}$$

The left quotient coincides with the derivative operator first proposed by Brzozowski [2] and Antimirov [1] over regular languages, lifted into the context-free setting (our work). When the root and LHS are fixed, e.g., $\frac{\partial S}{\partial \overleftarrow{x}} : (\overleftarrow{V} \to S) \to \overrightarrow{\mathcal{V}}$ returns the set of admissible nonterminals to the RHS. One may also consider a gradient operator, $\overleftarrow{\nabla} S : (\overleftarrow{\mathcal{V}} \to S) \to \overrightarrow{\mathcal{V}}$, which simultaneously tracks the partials with respect to a set of multiple LHS nonterminals produced by a fixed root.

[1] Hereinafter, we use gray highlighting to distinguish between expressions containing only constants from those which may contain free variables.

$$\begin{aligned} o &\to so \mid rs \mid rr \mid oo \\ r &\to so \mid ss \mid rr \mid os \\ s &\to so \mid rs \mid or \mid oo \end{aligned}$$

$$\mathcal{H}_{\{o\}} = \begin{pmatrix} \frac{\partial^2 o}{\partial\overleftarrow{o}\partial\overrightarrow{o}} & \frac{\partial^2 o}{\partial\overleftarrow{o}\partial\overrightarrow{r}} & \frac{\partial^2 o}{\partial\overleftarrow{o}\partial\overrightarrow{s}} \\ \frac{\partial^2 o}{\partial\overleftarrow{r}\partial\overrightarrow{o}} & \frac{\partial^2 o}{\partial\overleftarrow{r}\partial\overrightarrow{r}} & \frac{\partial^2 o}{\partial\overleftarrow{r}\partial\overrightarrow{s}} \\ \frac{\partial^2 o}{\partial\overleftarrow{s}\partial\overrightarrow{o}} & \frac{\partial^2 o}{\partial\overleftarrow{s}\partial\overrightarrow{r}} & \frac{\partial^2 o}{\partial\overleftarrow{s}\partial\overrightarrow{s}} \end{pmatrix}$$



$$\mathcal{H}_{\{r\}} = \begin{pmatrix} \frac{\partial^2 r}{\partial\overleftarrow{o}\partial\overrightarrow{o}} & \frac{\partial^2 r}{\partial\overleftarrow{o}\partial\overrightarrow{r}} & \frac{\partial^2 r}{\partial\overleftarrow{o}\partial\overrightarrow{s}} \\ \frac{\partial^2 r}{\partial\overleftarrow{r}\partial\overrightarrow{o}} & \frac{\partial^2 r}{\partial\overleftarrow{r}\partial\overrightarrow{r}} & \frac{\partial^2 r}{\partial\overleftarrow{r}\partial\overrightarrow{s}} \\ \frac{\partial^2 r}{\partial\overleftarrow{s}\partial\overrightarrow{o}} & \frac{\partial^2 r}{\partial\overleftarrow{s}\partial\overrightarrow{r}} & \frac{\partial^2 r}{\partial\overleftarrow{s}\partial\overrightarrow{s}} \end{pmatrix}$$

$$\mathcal{H}_{\{s\}} = \begin{pmatrix} \frac{\partial^2 s}{\partial\overleftarrow{o}\partial\overrightarrow{o}} & \frac{\partial^2 s}{\partial\overleftarrow{o}\partial\overrightarrow{r}} & \frac{\partial^2 s}{\partial\overleftarrow{o}\partial\overrightarrow{s}} \\ \frac{\partial^2 s}{\partial\overleftarrow{r}\partial\overrightarrow{o}} & \frac{\partial^2 s}{\partial\overleftarrow{r}\partial\overrightarrow{r}} & \frac{\partial^2 s}{\partial\overleftarrow{r}\partial\overrightarrow{s}} \\ \frac{\partial^2 s}{\partial\overleftarrow{s}\partial\overrightarrow{o}} & \frac{\partial^2 s}{\partial\overleftarrow{s}\partial\overrightarrow{r}} & \frac{\partial^2 s}{\partial\overleftarrow{s}\partial\overrightarrow{s}} \end{pmatrix}$$

Figure 1. CFGs are witnessed by a rank-3 tensor, whose nonempty inhabitants indicate CNF productions. Gradients in this setting effectively condition the parse tensor M by constraining the superposition of admissible parse forests.

3 Context-sensitive reachability

It is well-known that the family of CFLs is not closed under intersection. For example, consider $\mathcal{L}_\cap := \mathcal{L}(\mathcal{G}_1) \cap \mathcal{L}(\mathcal{G}_2)$:

$$\begin{aligned} P_1 &:= \left\{ S \to LR, \quad L \to ab \mid aLb, \quad R \to c \mid cR \right\} \\ P_2 &:= \left\{ S \to LR, \quad R \to bc \mid bRc, \quad L \to a \mid aL \right\} \end{aligned}$$

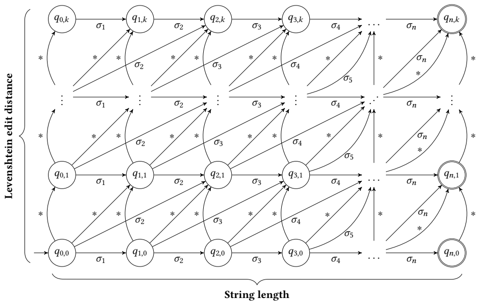
Note that $\mathcal{L}_\cap$ generates the language $\left\{ a^d b^d c^d \mid d > 0 \right\}$, which according to the pumping lemma is not context-free. We can encode $\bigcap_{i=1}^{c} \mathcal{L}(\mathcal{G}_i)$ as a polygonal prism with upper-triangular matrices adjoined to each rectangular face. More precisely, we intersect all terminals $\Sigma_\cap := \bigcap_{i=1}^{c} \Sigma_i$, then for each $t_\cap \in \Sigma_\cap$ and CFG, construct an equivalence class $E(t_\cap, \mathcal{G}_i) = \{w_i \mid (w_i \to t_\cap) \in P_i\}$ and bind them together.

$$\bigwedge_{t \in \Sigma_\cap} \bigwedge_{j=1}^{c-1} \bigwedge_{i=1}^{|\sigma|} E(t_\cap, \mathcal{G}_j) \equiv_{\sigma_i} E(t_\cap, \mathcal{G}_{j+1}) \tag{2}$$

Figure 2. Orientations of a $\bigcap_{i=1}^{4} \mathcal{L}(\mathcal{G}_i) \cap \Sigma^6$ configuration. As $c \to \infty$, this shape approximates a circular cone whose symmetric axis joins σ_i with orthonormal unit productions $w_i \to t_\cap$, and $S_i \in \Lambda_\sigma^*$? represented by the outermost bitvector inhabitants. Equations of this form are equiexpressive with the family of CSLs realizable by finite CFL intersection.

4 Levenshtein reachability

Levenshtein reachability is recognized by the nondeterministic infinite automaton (NIA) whose topology $\mathcal{L} =$ can be factored into a product of (a) the monotone Chebyshev topology , equipped with horizontal transitions accepting σ_i and vertical transitions accepting Kleene stars, and (b) the monotone knight's topology , equipped with transitions accepting σ_{i+2}. The structure of this space can be finitely approximated by an acyclic NFA [7], populated by accept states within radius k of $q_{n,0}$, or equivalently, a left-linear CFG whose productions bisimulate the transition dynamics:



Let $G(\utilde{\sigma} : \Sigma^*, d : \mathbb{N}^+) \mapsto \mathbb{G}$ be the construction described above accepting a string, $\utilde{\sigma}$, an edit distance, d, and returning a grammar that accepts the language of all strings within Levenshtein radius d of $\utilde{\sigma}$. To find the language edit distance and corresponding least-distance edit(s), we must find the least d such that $\mathcal{L}_d^\cap := \mathcal{L}\big(G(\utilde{\sigma}, d)\big) \cap \mathcal{L}(\mathcal{G}')$ is nonempty, i.e.: (1) $\tilde{\sigma} \in \mathcal{L}(\mathcal{G}')$, and (2) $\Delta(\utilde{\sigma}, \tilde{\sigma}) \leq d^* \iff \tilde{\sigma} \in \mathcal{L}\big(G(\utilde{\sigma}, d^*)\big)$, and (3) $\nexists \sigma' \in \mathcal{L}(\mathcal{G}').[\Delta(\utilde{\sigma}, \sigma') < d^*]$. To satisfy these criteria, it suffices to check $d \in (0, d_{\max}]$ by encoding the Levenshtein automata and the original grammar as a single polynomial, call it $\varphi_d[\cdot]$, then gradually introduce new accepting states at increasing radii until either (1) a satisfying assignment is found or (2) $d_{\max}$ is attained. Defined recursively,

$$\varphi_{d+1} := \begin{cases} \varphi\big[q_{\{i,j \mid n-i+j \leq 1\}} \subset \Lambda_{\tilde{\sigma}}^*\big(G(\utilde{\sigma}, d_{\max})\big)\big] & \text{if } d = 1\text{, or} \\ \varphi_d \oplus \varphi\big[q_{\{i,j \mid n-i+j = d+1\}} \subset \Lambda_{\tilde{\sigma}}^*\big(G(\cdot, \cdot)\big)\big] & \text{otherwise.} \end{cases}$$

φ will surely terminate in at most either the number of steps required to overwrite every symbol in $\utilde{\sigma}$, or the length of the shortest string in $\mathcal{L}(\mathcal{G}')$, whichever is greater.

5 Conclusion

Not only is multilinear algebra over finite fields an expressive language for inference, but also a natural one for studying decision problems on formal languages themselves. Galois fields enjoy rich connections to algebraic complexity theory, and are particularly amenable to both SAT encoding and GPU acceleration. In this work, we illustrate a few applications for code completion and syntax repair in context-free and linear conjunctive languages. In the future, we plan to extend our method to stochastic grammars like PCFGs and HMMs.

References

[1] Valentin Antimirov. 1996. Partial derivatives of regular expressions and finite automaton constructions. Theoretical Computer Science 155, 2 (1996), 291–319.

[2] Janusz A Brzozowski. 1964. Derivatives of regular expressions. Journal of the ACM (JACM) 11, 4 (1964), 481–494.

[3] Kathrin Hanauer, Monika Henzinger, and Christian Schulz. 2021. Recent advances in fully dynamic graph algorithms. arXiv preprint arXiv:2102.11169 (2021).

[4] Lillian Lee. 2002. Fast context-free grammar parsing requires fast boolean matrix multiplication. Journal of the ACM (JACM) 49, 1 (2002), 1–15. https://arxiv.org/pdf/cs/0112018.pdf

[5] Alexander Okhotin. 2004. On the equivalence of linear conjunctive grammars and trellis automata. RAIRO-Theoretical Informatics and Applications 38, 1 (2004), 69–88.

[6] Piotr Sankowski. 2004. Dynamic transitive closure via dynamic matrix inverse. In 45th Annual IEEE Symposium on Foundations of Computer Science. IEEE, 509–517.

[7] Klaus U Schulz and Stoyan Mihov. 2002. Fast string correction with Levenshtein automata. International Journal on Document Analysis and Recognition 5 (2002), 67–85.

[8] Leslie G Valiant. 1975. General context-free recognition in less than cubic time. Journal of computer and system sciences 10, 2 (1975), 308–315. http://people.csail.mit.edu/virgi/6.s078/papers/valiant.pdf