

Propagation of Syntax Errors in Context-Sensitive Languages using the Lifted Brzozowski Derivative

Anonymous Author(s)

Abstract

Brzozowski (1964) defines the derivative of a regular language as the suffixes that 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 show how context-sensitive language recognition can be reduced into a tensor algebra over finite fields, drawing a loose analogy to partial differentiation in Euclidean spaces. In addition to its theoretical contributions, our method has yielded applications to incremental parsing, code completion and program repair. For example, we use it to repair syntax errors and perform sketch-based program synthesis, among other language decision problems.

1 Introduction

Recall that a CFG is a quadruple consisting of terminals (Σ), nonterminals (V), productions ($P: V \rightarrow (V \mid \Sigma)^*$), and a start symbol, (S). It is a well-known fact that every CFG is reducible to *Chomsky Normal Form*, $P': V \rightarrow (V^2 \mid \Sigma)$, in which every production takes one of two forms, either $w \rightarrow xz$, or $w \rightarrow t$, where $w, x, z: V$ and $t: \Sigma$. For example, the CFG, $P := \{S \rightarrow SS \mid (S) \mid ()\}$, corresponds to the CNF:

$$P' = \{ S \rightarrow QR \mid SS \mid LR, \quad L \rightarrow (, \quad R \rightarrow), \quad Q \rightarrow LS \}$$

Given a CFG, $\mathcal{G}' : \langle \Sigma, V, P, S \rangle$ in CNF, we can construct a recognizer $R: \mathcal{G}' \rightarrow \Sigma^n \rightarrow \mathbb{B}$ for strings $\sigma: \Sigma^n$ as follows. Let 2^V be our domain, 0 be \emptyset , \oplus be \cup , and \otimes be defined as:

$$X \otimes Y := \{ w \mid \langle x, z \rangle \in X \times Y, (w \rightarrow xz) \in P \} \quad (1)$$

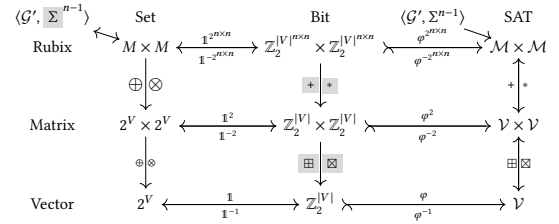
If we define $\uparrow \sigma_r := \{V \mid (V \rightarrow \sigma_r) \in P\}$, then initialize $M_{r+1=c}^0(\mathcal{G}', e) := \uparrow \sigma_r$ and solve for the fixpoint $M^* = M + M^2$,

$$M^0 := \begin{pmatrix} \emptyset & \sigma_1^\uparrow & \emptyset & \dots & \emptyset \\ & \ddots & \ddots & \ddots & \ddots \\ \emptyset & & \emptyset & \dots & \sigma_n^\uparrow \\ & & \emptyset & \dots & \emptyset \end{pmatrix} \Rightarrow M^* = \begin{pmatrix} \emptyset & \sigma_1^\uparrow & \mathcal{V} & \dots & \mathcal{V}^* \\ & \ddots & \ddots & \ddots & \ddots \\ \emptyset & & \emptyset & \dots & \mathcal{V} \\ & & \emptyset & \dots & \sigma_n^\uparrow \\ \emptyset & \dots & \dots & \dots & \emptyset \end{pmatrix}$$

this yields the recognizer, $R(\mathcal{G}', \sigma) := S \in \mathcal{V}^*? \Leftrightarrow \sigma \in \mathcal{L}(\mathcal{G})$? Full details of the bisimilarity between parsing and matrix multiplication can be found in Valiant [4] and Lee [3], who shows its time complexity to be $\mathcal{O}(n^\omega)$ where ω is the least matrix multiplication upper bound (currently, $\omega < 2.77$).

2 Method

Note that $\bigoplus_{k=1}^n M_{ik} \otimes M_{kj}$ 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



where \mathcal{V} is a function $\mathbb{Z}_2^{|V|} \rightarrow \mathbb{Z}_2$. Note that while always possible to encode an element of $\mathbb{Z}_2^{|V|}$ into \mathcal{V} , φ^{-1} may not exist, as an arbitrary \mathcal{V} might take on zero, one, or in general, multiple values in $\mathbb{Z}_2^{|V|}$. Let us consider two cases, where Σ^n is either left- or right-constrained, i.e., $xz, zx: \Sigma^{|x|+|z|}$.

Valiant’s \otimes operator, which solves for the set of productions unifying known factors in a binary CFG, implies the existence of a left- and right-quotient, which yield the set of nonterminals that may appear to either 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

Right Quotient

$$\frac{\partial f}{\partial x} = \{ y \mid (w \rightarrow xy) \in P \}$$

$$\frac{\partial f}{\partial y} = \{ x \mid (w \rightarrow xy) \in P \}$$



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., $\frac{\partial S}{\partial x}: (\tilde{V} \rightarrow S) \rightarrow \tilde{\mathcal{V}}$ returns the set of admissible nonterminals to the RHS. One may also define a gradient operator, $\nabla S: (\tilde{V} \rightarrow S) \rightarrow \tilde{\mathcal{V}}$, which simultaneously tracks the partials with respect to a set of multiple LHS nonterminals produced by a fixed root.

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

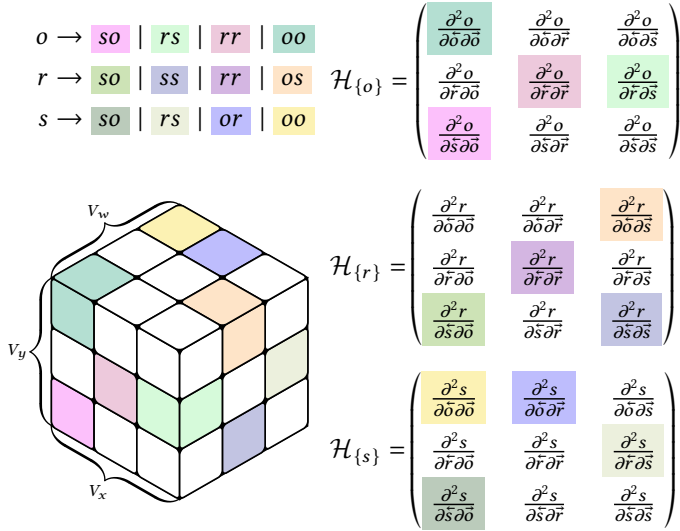


Figure 1. CFGs are witnessed by a rank-3 binary tensor, whose nonzero entries indicate CNF productions. Derivatives in this setting effectively condition the parse tensor. By backpropagating \mathcal{H} across upper-triangular entries of \mathcal{M}^* , we constrain the superposition of admissible parse forests.

If the root itself is unknown, we can define an operator, $\mathcal{H}_{\mathcal{W} \subseteq \mathcal{V}} : (\vec{\mathcal{V}} \times \vec{\mathcal{V}} \times \mathcal{W}) \rightarrow (\vec{\mathcal{V}} \times \vec{\mathcal{V}} \rightarrow \mathcal{W})$, which tracks second-order partial derivatives for all roots in \mathcal{W} . Unlike differential calculus on smooth manifolds, partials in this calculus do not necessarily commute depending on the CFG. By allowing the matrix \mathcal{M}^* to contain bitvector variables representing holes in σ , we obtain a set of multilinear equations whose solutions exactly correspond to the set of admissible repairs and their corresponding parse forests. Specifically, the repairs coincide with holes in the superdiagonal $\mathcal{M}_{r+1=c}^*$, and the parse forests occur along upper-triangular entries $\mathcal{M}_{r+1 < c}^*$. In the follow case, \mathcal{M}^* is left-constrained, although the holes may (in general) appear anywhere in σ :

$$\mathcal{M}^* = \begin{pmatrix} \emptyset & \uparrow \sigma_1 & \mathcal{L}_{1,3} & \mathcal{L}_{1,3} & \mathcal{V}_{1,4} & \dots & \mathcal{V}_{1,n} \\ \vdots & \vdots & \uparrow \sigma_2 & \mathcal{L}_{2,3} & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \uparrow \sigma_3 & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \mathcal{V}_{4,4} & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \mathcal{V}_{n,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \emptyset & \dots & \dots & \dots & \dots & \dots & \emptyset \end{pmatrix}$$

We also need the constraint that no two conflicting non-terminals may be active at any given time... TODO: describe uniqueness constraint

2.1 Context-Sensitive Reachability

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

$$P_1 := \{ S \rightarrow LR, \quad L \rightarrow ab \mid aLb, \quad R \rightarrow c \mid cR \}$$

$$P_2 := \{ S \rightarrow LR, \quad R \rightarrow bc \mid bRc, \quad L \rightarrow a \mid aL \}$$

Note that \mathcal{L}_\cap generates the language $\{ a^d b^d c^d \mid d > 0 \}$, 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 $E(t_\cap, \mathcal{G}_i) = \{ w_i \mid (w_i \rightarrow t) \in P_i \}$ and glue them together:



Figure 2. Orientations of a $\bigcap_{i=1}^4 \mathcal{L}(\mathcal{G}_i) \cap \Sigma^6$ configuration.

As $c \rightarrow \infty$, this shape approximates a circular cone whose symmetric axis intersects orthonormal CNF unit productions $w_i \rightarrow t_\cap$, with $S_i \in \mathcal{V}_i^*$ encoded by bitvectors on the base perimeter. Equations of this form are equiexpressive with the family of CSLs realizable by finite CFL intersection.

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

Although emptiness for CSLs is, in general, undecidable, unsatisfiability corresponds to emptiness of bounded CSLs. Anyhow, since we are working with bounded-width CSLs, everything collapses down to finite languages, which are always closed. Thus, Bar-Hillel and other elegant constructions become trivial when so restricted. So we can use it to decide impossible substrings and other decision problems which are typically intractable in more general settings.

3 Conclusion

This technique can be used to repair syntax errors in programming languages. It also has the nice advantage that it can provide partial parse trees for invalid strings, i.e., we can decode the tree from the structure of \mathcal{M} directly without any special error recovery.

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. http://maveric.uwaterloo.ca/reports/1964_JACM_Brzozowski.pdf

- [3] 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>
- [4] 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>