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

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

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:

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

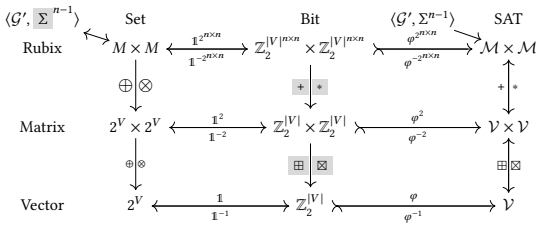
If we define $\uparrow \sigma_r := \{w \mid (w \to \sigma_r) \in P\}$, then initialize $M^0_{r+1=c}(\mathcal{G}', e) := \uparrow \sigma_r$ and solve for the fixpoint $M^* = M + M^2$,

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

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 [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 $\bigoplus_{c=1}^{n} M_{r,c} \otimes M_{c,r}$ 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]



where $\mathcal{V}$ is a function $\mathbb{Z}_2^{|V|} \to \mathbb{Z}_2$. Note that while always possible to encode $\mathbb{Z}_2^{|V|} \to \mathcal{V}$ using the identity function, φ^{-1} may not exist, as an arbitrary $\mathcal{V}$ might take on zero, one, or in general, multiple values in $\mathbb{Z}_2^{|V|}$. 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

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

x	w
	z

Right Quotient

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

x	w
	z

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., $\frac{\partial S}{\partial \overleftarrow{x}} : (\overleftarrow{V} \to S) \to \vec{\mathcal{V}}$ returns the set of admissible nonterminals to the RHS. One may also define a gradient operator, $\overleftarrow{\nabla} S : (\overleftarrow{\mathcal{V}} \to S) \to \vec{\mathcal{V}}$, 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.

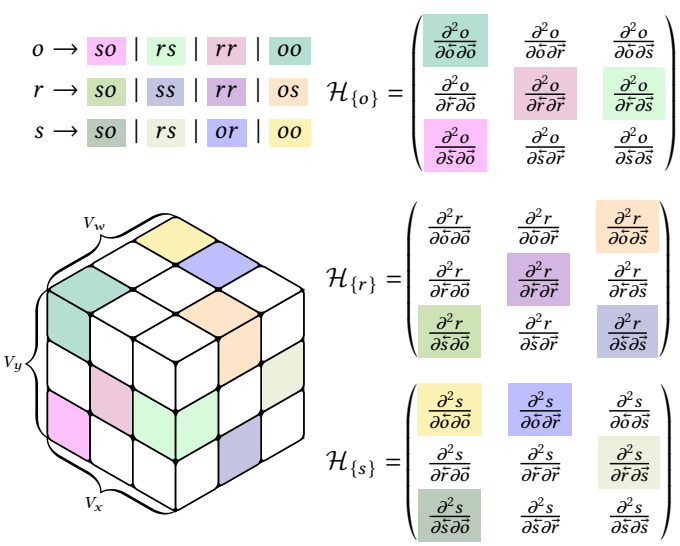


Figure 1. CFGs are witnessed by a rank-3 binary tensor, whose nonzero entries indicate CNF productions. Derivatives in this setting effectively condition the parse tensor. By backpropagating $\mathcal{H}$ across upper-triangular entries of $\mathcal{M}^*$, we constrain the superposition of admissible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

2.1 Deriving the hole locations

Valiant's procedure defines a bottom-up synthesizer for a string template with known hole locations. Given a string which contains an error, i.e. $\underset{\sim}{\sigma} : \Sigma^+$ s.t. $S \notin \Lambda^*_\sigma$, where should we put the holes in order for the string to parse? We can derive the hole locations by backpropagating Brzozowski's derivative across upper-triangular entries of $\mathcal{M}^*$. This constitutes to a top-down inference procedure, in which, following the tradition of Bird and Meertens, we define the exponential of a forest as a nested datatype called a *taiga*:

$$\textbf{data } \mathit{Tree}\ a = \varnothing \mid \langle a, \langle \mathit{Tree}\ a, \mathit{Tree}\ a \rangle \rangle$$
$$\textbf{data } \mathit{Forest}\ a = \varnothing \mid \langle \{a\}, \{\langle \mathit{Forest}\ a, \mathit{Forest}\ a \rangle\} \rangle$$
$$\textbf{data } \mathit{Taiga}\ a = \varnothing \mid \langle a, [\langle \mathit{Taiga}\ (\mathit{Taiga}\ a) \rangle^2] \rangle$$

This is needed because of the doubly-ambiguous nature of tree search: a single string may have a parse forest, and an incomplete string simultanously occupies a superposition of possible parse forests. When we encounter two adjacent parse forests which cannot be joined, we know that either (1) the left derivation must change (2) the right derivation must change or (3) there must be a hole in the middle which joins the two together. When recursing over the state space, we must simultaneously explore all three possibilities.

2.2 Context-Sensitive Reachability

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

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

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 $E(t_\cap, \mathcal{G}_i) = \{w_i \mid (w_i \to t_\cap) \in P_i\}$ and glue them together:

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

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

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

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 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