Deriving (finite) intersection non-emptiness, courtesy of Brzozowski

Breandan Considine

March 8, 2025

1 Syntax and Semantics

Definition 1 (Generalized Regex)**.** Let E be an expression defined by the grammar:

$$E ::= \varnothing \mid \varepsilon \mid \Sigma \mid E \cdot E \mid E \vee E \mid E \wedge E$$

Semantically, we interpret these expressions as denoting regular languages:

$$\begin{aligned}
\mathcal{L}(\ \varnothing\ ) &= \varnothing & \mathcal{L}(\ R \cdot S\ ) &= \mathcal{L}(R) \times \mathcal{L}(S)\\
\mathcal{L}(\ \varepsilon\ ) &= \{\varepsilon\} & \mathcal{L}(\ R \vee S\ ) &= \mathcal{L}(R) \cup \mathcal{L}(S)\\
\mathcal{L}(\ a\ ) &= \{a\} & \mathcal{L}(\ R \wedge S\ ) &= \mathcal{L}(R) \cap \mathcal{L}(S)
\end{aligned}$$

Definition 2 (Brzozowski, 1964)**.** To compute the quotient $\partial_a(L) = \{b \mid ab \in L\}$, we:

$$\begin{aligned}
\partial_a(\ \varnothing\ ) &= \varnothing & \delta(\ \varnothing\ ) &= \varnothing\\
\partial_a(\ \varepsilon\ ) &= \varnothing & \delta(\ \varepsilon\ ) &= \varepsilon\\
\partial_a(\ b\ ) &= \begin{cases} \varepsilon & \text{if } a = b \\ \varnothing & \text{if } a \neq b \end{cases} & \delta(\ a\ ) &= \varnothing\\
\partial_a(\ R \cdot S\ ) &= (\partial_a R) \cdot S \vee \delta(R) \cdot \partial_a S & \delta(\ R \cdot S\ ) &= \delta(R) \wedge \delta(S)\\
\partial_a(\ R \vee S\ ) &= \partial_a R \vee \partial_a S & \delta(\ R \vee S\ ) &= \delta(R) \vee \delta(S)\\
\partial_a(\ R \wedge S\ ) &= \partial_a R \wedge \partial_a S & \delta(\ R \wedge S\ ) &= \delta(R) \wedge \delta(S)
\end{aligned}$$

Theorem 1 (Recognition)**.** *For any regex R and $\sigma : \Sigma^*$, $\sigma \in \mathcal{L}(R) \iff \varepsilon \in \mathcal{L}(\partial_\sigma R)$, where:*

$$\partial_\sigma(R) : RE \to RE = \begin{cases} R & \textit{if } \sigma = \varepsilon \\ \partial_b(\partial_a R) & \textit{if } w = ab, a \in \Sigma \end{cases}$$

Theorem 2 (Generation)**.** *For any $(\varepsilon, \wedge)$-free regex, R, to generate a witness $\sigma \sim \mathcal{L}(R)$:*

$$\texttt{follow}(R) : RE \to 2^\Sigma = \begin{cases} \{R\} & \textit{if } R \in \Sigma \\ \texttt{follow}(S) & \textit{if } R = S \cdot T \\ \texttt{follow}(S) \cup \texttt{follow}(T) & \textit{if } R = S \vee T \end{cases}$$

$$\texttt{choose}(R) : RE \to \Sigma^* = \begin{cases} R & \textit{if } R \in \Sigma \\ \big(s \sim \texttt{follow}(R)\big) \cdot \texttt{choose}(\partial_s R) & \textit{if } R = S \cdot T \\ \texttt{choose}(R' \sim \{S, T\}) & \textit{if } R = S \vee T \end{cases}$$

2 Language intersection

Theorem 3 (Bar-Hillel, 1961)**.** *For any context-free grammar (CFG), $G = \langle V, \Sigma, P, S\rangle$, and nondeterministic finite automata, $A = \langle Q, \Sigma, \delta, I, F\rangle$, there exists a CFG $G_\cap = \langle V_\cap, \sigma_\cap, P_\cap, S_\cap\rangle$ such that $\mathcal{L}(G_\cap) = \mathcal{L}(G) \cap \mathcal{L}(A)$.*

Definition 3 (Salomaa, 1973)**.** One could construct $G_\cap$ like so,

$$\frac{q \in I \quad r \in F}{\big(S \to qSr\big) \in P_\cap}\sqrt{} \qquad \frac{(A \to a) \in P \qquad (q \overset{a}{\to} r) \in \delta}{\big(qAr \to a\big) \in P_\cap}\uparrow \qquad \frac{(w \to xz) \in P \qquad p, q, r \in Q}{\big(pwr \to (pxq)(qzr)\big) \in P_\cap}\bowtie$$

however most synthetic productions in $P_\cap$ will be non-generating or unreachable.

Theorem 4 (Considine, 2025)**.** *For every CFG, G, and every acyclic NFA (ANFA), A, there exists a decision procedure φ : CFG $\to$ ANFA $\to \mathbb{B}$ such that $\varphi(G, A) \models [\mathcal{L}(G) \cap \mathcal{L}(A) \neq \varnothing]$ which requires $\mathcal{O}\big((\log|Q|)^c\big)$ time using $\mathcal{O}\big((|V||Q|)^k\big)$ parallel processors for some $c, k < \infty$.*

Proof sketch. At least one of the following must hold for $w \in V$ to parse some path $p \rightsquigarrow r$ in A:

1. p steps directly to r in which case it suffices to check $\exists s.\big((p \overset{s}{\to} r) \in \delta \land (w \to s) \in P\big)$, or,
2. there is some midpoint $q \in Q$, $p \rightsquigarrow q \rightsquigarrow r$ such that $\exists x, z.\big((w \to xz) \in P \land \overbrace{\underbrace{p \rightsquigarrow q}_{x}, \underbrace{q \rightsquigarrow r}_{z}}^{w}\big)$.

This suggests a dynamic programming solution. Let M be a matrix of type $RE^{|Q|\times|Q|\times|V|}$ indexed by Q. Since we assumed δ is acyclic, there exists a topological sort of δ imposing a total order on Q such that M is strictly upper triangular (SUT). We will initialize it as follows:

$$M_0[r, c, v] = \bigvee_{a \in \Sigma} \{a \mid (v \to a) \in P \land (q_r \overset{a}{\to} q_c) \in \delta\} \tag{1}$$

The algebraic operations $\oplus, \otimes : RE^{2|V|} \to RE^{|V|}$ will be defined elementwise:

$$[\ell \oplus r]_v = [\ell_v \lor r_v] \tag{2}$$

$$[\ell \otimes r]_w = \bigvee_{x,z \in V}\{\ell_x \cdot r_z \mid (w \to xz) \in P\} \tag{3}$$

By slight abuse of notation, we will redefine the matrix exponential over this domain as:

$$\exp(M) = \sum_{i=0}^{\infty} M_0^i = \sum_{i=0}^{|Q|} M_0^i \text{ (since } M \text{ is SUT.)} \tag{4}$$

To solve for the fixpoint, we can instead use exponentiation by squaring:

$$S(2n) = \begin{cases} M_0, & \text{if } n = 1, \\ S(n) + S(n)^2 & \text{otherwise.} \end{cases} \tag{5}$$

Therefor, we only need a maximum of $\lceil \log_2 |Q| \rceil$ sequential steps to reach the fixpoint. Finally,

$$S_\cap = \bigvee_{q \in I,\, q' \in F} \exp(M)[q, q', S] \text{ and } \varphi = [S_\cap \neq \varnothing] \tag{6}$$

To decode a witness in case of non-emptiness, we simply `choose` (φ). □

3 Future work

Broadly interested in questions related to formal languages and finite model theory, following the Carnap program of logical syntax. Encoding semantics into syntax would allow us to do type checking and static analysis in the parser. A few lines of attack here:

Theorem 5 (Büchi–Elgot–Trakhtenbrot). *A language is regular iff it can be defined in MSO. For every MSO formula, there is a corresponding FSA. Complexity may be nonelementary.*

Theorem 6 (Pentus, 1997). *Lambek categorial grammars, a weak kind of substructural logic, recognize exactly the context-free languages.*

Theorem 7 (Lautemann, Schwentick & Thérien, 1995). *CFLs coincide with the class of strings definable by $\exists b \phi$ where ϕ is first order, b is a binary predicate symbol ...*

Theorem 8 (DeYoung & Pfenning, 2016). *Describes a certain equivalence between subsingleton logic, a weak kind of linear logic and automata.*

Theorem 9 (Glenn & Garsarch, 1996). *Explores the relation between WS1S and finite automata.*

Theorem 10 (Knuth & Wegner, 1968). *Attribute grammars allow some notation of semanticity.*