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

Breandan Considine

March 7, 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:

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

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

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

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 $\sigma \sim \mathcal{L}(R)$:*

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

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

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 of the synthetic productions 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 $\mathcal{O}(n^k)$ 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 $\exists s.\big((p \overset{s}{\to} r) \in \delta \wedge (w \to s) \in P\big)$, or,
2. there is some midpoint $q \in Q$, $p \rightsquigarrow q \rightsquigarrow r$ such that $\exists x, z.\big((w \to xz) \in P \wedge \overbrace{\underbrace{p \rightsquigarrow q}_{x}, \underbrace{q \rightsquigarrow r}_{z}}^{w}\big)$.

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:

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

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

$$[\ell \oplus r]_v = [\ell_v \vee r_v] \tag{2}$$
$$[\ell \otimes r]_w = \bigvee_{x,z \,\in\, V} \{\ell_x \cdot r_z \mid (w \to xz) \in P\} \tag{3}$$

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 $\lceil \log_2 |Q| \rceil$ squaring steps to determine 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). $\square$