Separating Strings with Automata and SAT

1 Introduction

In this sketch we study how to use SAT to construct a non-deterministic finite automata (NFA) that separates two finite sets of strings. This yields a decision procedure for calculating the metric distance between two regular sets with respect to the (inverse) automata size metric.

2 Preliminaries

An alphabet Σ is a finite set of unique symbols. Let $\Sigma^\star$ denote the set of all finite strings consisting of letters from Σ. A language $\mathcal{L} \subseteq \Sigma^\star$ is then a countable set of strings of Σ. Often Σ is implicitly assumed.

A non-deterministic finite automata (NFA) $\mathcal{A}$ is represented as a tuple $\mathcal{A} = (\Sigma, Q, \Delta, s, F)$ where Σ is a finite alphabet, Q is a finite set of states, $\Delta\colon (Q \times \Sigma) \to 2^Q$ is the transition function, $s \in Q$ is the inital state, and $F \subseteq Q$ is a set of final states.

For some string w we say that $\mathcal{A}$ accepts w if there exists a sequence of transitions on which $\mathcal{A}$ ends in a final state when reading w. We abuse notation and say that $\mathcal{A}$ accepts w if $\mathcal{A}(w) = 1$. Similarly, we say $\mathcal{A}(w) = 0$ if $\mathcal{A}$ rejects w.

Let w_i denote the ith letter of the string w, and ε be the empty string.

3 Separating Sets

The question that this sketch attempts to answer can be formalized as follows:

Question 1. *Given a positive set $P \subseteq \Sigma^\star$ and negative set $N \subseteq \Sigma^\star$ with P and N disjoint, does there eixst an NFA $\mathcal{A} = (\Sigma, Q, \Delta, s, F)$ with $|Q| = n$ such that for all $u \in P$ in the positive set $\mathcal{A}(u) = 1$, but for all $v \in N$ in the negative set $\mathcal{A}(v) = 0$.*

We achieve this by constructing a boolean satisfiability formula that encodes P and N, and is satisfiable if and only if such $\mathcal{A}$ exists. There are several high-level insights that we leverage:

(1) NFAs can be represented as directed multi-edge graphs where each edge is labeled by one letter from Σ. In other words, let $e_{i,j,\sigma}$ be an indicator variable encodes the indicator of a transition from state q_i to state q_j on the letter σ.

(2) For each $u \in P$, we can create a formula φ_u that forces a sequence of edge walks resulting in a final state in $\mathcal{A}$. Similarly for each $v \in N$ we can encode a sequence that will force a rejection of v in $\mathcal{A}$.

For a particular $u \in P$, consider the following encoding:

$$\pi_u \equiv \bigwedge_{1 \leq t < |u|} \left(\bigvee_{i \neq j \neq k} e_{i,j,u_t} \wedge e_{j,k,u_{t+1}} \right)$$

This forces the existence of a path on $\mathcal{A}$ for u. Additionally, to force u to stop on a final state, we may either have every state be a final state, or just set the following formula:

$$\phi_u \equiv \bigwedge_{i \neq j} \left(e_{i,j,u_{|u|}} \implies f_j \right)$$

Where f_j is an indicator denoting that q_j is a final state.