

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 Σ^* denote the set of all finite strings consisting of letters from Σ . A language $\mathcal{L} \subseteq \Sigma^*$ 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, $\Delta: (Q \times \Sigma) \rightarrow 2^Q$ is the transition function, $S \subseteq Q$ is the initial states, 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 i th 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 $P \subseteq \Sigma^*$ and negative set $N \subseteq \Sigma^*$ with P and N disjoint, does there exist 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 . Self-loops are permitted here. 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 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} .

We first construct a formula that will force \mathcal{A} to accept a word w if and only if the formula is satisfied. Let $y_{i,t}^w$ denote be an indicator variable to show that \mathcal{A} is at state q_i at time t , with $1 \leq i \leq n$ and $1 \leq t \leq |w| + 1$. Note that since each letter of w acts as a transition, the automata will occupy $|w| + 1$ possibly repeated states during its accepting run. Then:

$$\rho_w \equiv \bigwedge_{1 \leq t \leq |w|+1} \left[\bigwedge_{1 \leq i,j \leq n} \neg (y_{i,t}^w \wedge y_{j,t}^w) \right]$$

Forces the automata to be in only one state at any given time t while reading w . To accompany this, let $e_{i,j,\sigma}$ to denote that \mathcal{A} has a transition edge from q_i to q_j on letter σ . Similarly:

$$\pi_w \equiv \bigwedge_{1 \leq t \leq |w|} \left[\bigvee_{1 \leq i,j \leq n} (y_{i,t}^w \wedge y_{j,t+1}^w \wedge e_{i,j,w_t}) \right]$$

Additionally, we can force boundary conditions to ensure that \mathcal{A} begins reading w on a starting state and ends on an accepting state:

$$\gamma_w \equiv \left[\bigvee_{1 \leq i,j \leq n} (e_{i,j,w_1} \wedge s_i) \right] \vee \left[\bigvee_{1 \leq i,j \leq n} (e_{i,j,w_{|w|}} \wedge f_j) \right]$$

Where s_i and f_j are indicator variables to express that q_i is a starting state and q_j is a final state respectively. Then finally we set:

$$\varphi_w \equiv \rho_w \wedge \pi_w \wedge \gamma_w$$

Then \mathcal{A} will only accept w if and only if φ_w is satisfiable. Finally:

$$\Phi_{P,N} \equiv \left[\bigwedge_{u \in P} \varphi_u \right] \wedge \left[\bigwedge_{v \in N} \neg \varphi_v \right]$$

By construction, $\Phi_{P,N}$ is true if and only if \mathcal{A} accepts all P and rejects all N .

AAAAAAAAAAAAAAAAAAAA

For any $w \in \Sigma^*$, consider the following encoding:

$$\pi_w \equiv \bigwedge_{1 \leq t < |w|+1} \left(\bigvee_{i \neq j \neq k} e_{i,j,w_t} \wedge e_{j,k,w_{t+1}} \right)$$

This forces the existence of a path on \mathcal{A} for w . The number of states in the walk of an accept will be $|w| + 1$, since each letter in w acts as transition between states. Additionally, to force w to stop on a final state, we may either have every state be a final state, or just set the following formula:

$$\varphi_w \equiv \left[\bigwedge_{i \neq j} (e_{i,j,w_1} \implies s_i) \right] \wedge \left[\bigwedge_{i \neq j} (e_{i,j,w_{|w|}} \implies f_j) \right]$$

Essentially, this forces each w to begin processing on an initial state as indicated by s_i , and end on a final state as indicated by f_j . Thus, the conjunction $\pi_w \wedge \varphi_w$ will be true if and only if \mathcal{A} accepts w . Hence, the satisfaction of its negation will then force \mathcal{A} to reject w . We leverage this in order to define over P and N :

$$\Phi_{P,N} = \left[\bigwedge_{u \in P} (\pi_u \wedge \varphi_u) \right] \wedge \left[\bigwedge_{v \in N} \neg (\pi_v \wedge \varphi_v) \right]$$

Thus, $\Phi_{P,N}$ defines a formula that will make \mathcal{A} to accept P and reject N .

The variables of the $\Phi_{P,N}$ are the indicators for edges of form $e_{i,j,\sigma}$ and final states of form f_j . Suppose that Σ is known. If $\Phi_{P,N}$ is satisfiable, then $\mathcal{A} = (\Sigma, Q, \Delta, S, F)$ can be extracted as:

$$\begin{aligned} Q &= \{q_1, \dots, q_n\} \\ \Delta &= \{((q_i, \sigma), q_j) : e_{i,j,\sigma} = \top\} \\ S &= \{q_i : s_i = \top\} \\ F &= \{q_j : f_j = \top\} \end{aligned}$$

Note that Δ is slightly overloaded.