CS 228M: Logic in CS
Alternating Finite Automata

Ashwin Abraham

8th November, 2023

An Alternating Finite Automaton over a finite alphabet Σ is a 5-tuple $(Q, \Sigma, \delta, \varphi_0, F)$ where Q is a finite set of states, $F \subseteq Q$ is the set of final states, $\delta : Q \times \Sigma \to \texttt{WFF}(Q)$ is the transition function and $\varphi_0 \in \texttt{WFF}(Q)$ is the initial formula. Here, $\texttt{WFF}(Q)$ denotes the set of well formed formulae of propositional logic with the states in Q as the propositional variables, ie it is formed by the grammar $\varphi ::= \top | \bot | q \in Q | \varphi \land \varphi | \varphi \lor \varphi | \neg\varphi$.
We inductively define the function $\delta^* : \texttt{WFF}(Q) \times \Sigma \to \texttt{WFF}(Q)$ as follows:
For all $q \in Q, a \in \Sigma, \varphi, \psi \in \texttt{WFF}(Q)$:

1. $\delta^*(\top, a) = \top$
2. $\delta^*(\bot, a) = \bot$
3. $\delta^*(q, a) = \delta(q, a)$
4. $\delta^*(\varphi \lor \psi, a) = \delta^*(\varphi, a) \lor \delta^*(\psi, a)$
5. $\delta^*(\varphi \land \psi, a) = \delta^*(\varphi, a) \land \delta^*(\psi, a)$
6. $\delta^*(\neg\varphi, a) = \neg\delta^*(\varphi, a)$

Now, we inductively define the extended transition function $\hat{\delta} : \texttt{WFF}(Q) \times \Sigma^* \to \texttt{WFF}(Q)$:

1. $\hat{\delta}(\varphi, \varepsilon) = \varphi$
2. $\hat{\delta}(\varphi, xa) = \delta^*(\hat{\delta}(\varphi, x), a)$

Now, a word w is accepted by this automaton iff $\alpha \models \hat{\delta}(\varphi_0, w)$ where $\alpha : Q \to \{0, 1\}$ is the assignment such that $\alpha(q) = 1$ if $q \in F$ and $\alpha(q) = 0$ otherwise.
A regular NFA $(Q, \Sigma, \Delta, I, F)$ is a special case of an AFA $(Q, \Sigma, \delta, \varphi_0, F)$ where $\varphi_0 = \bigvee_{q \in I} q$ and $\delta(q, a) = \bigvee_{(q,a,p) \in \Delta} p$. It can easily be shown that $\hat{\delta}(\varphi_0, w) = \bigvee_{q \in \hat{\Delta}(I,w)} q$, and therefore, w will be accepted by the AFA iff $\hat{\Delta}(I, w) \cap F \neq \varnothing$, ie iff the NFA accepts w. An NFA with a demonic acceptance condition (ie a word is accepted iff every run ends in a final state) is also a special case of an AFA, with the "or"s ($\bigvee$) in the above construction replaced by "and"s ($\bigwedge$).
Usually AFAs are defined in a different way - as a 6-tuple $(Q_\forall, Q_\exists, \Sigma, \Delta, I, F)$ where the set of states $Q = Q_\forall \cup Q_\exists$ is partitioned into universal states and existential states, $\Delta \subseteq Q \times \Sigma \times Q$, $I \subseteq Q$ and $F \subseteq Q$. This is done to have a combination of angelic and demonic acceptance in the automaton. Words here have run trees instead of run strings. A run tree of a word w is a tree with vertices labelled with states in Q and edges labelled with alphabets in w such that:

1. The root is some state in I.
2. The edges are labelled with alphabets such that every traversal from the root to a node constructs a prefix of the word. This ensures that all nodes at the same depth must have the same label for their outgoing edges (if any).
3. If a node is labelled with a state in $q \in Q_\exists$ then it has exactly one child, labelled with $p \in Q$ such that $(p, a, q) \in \Delta$ where a is the label that any outgoing edge from the node must have. If there is no such p, then the node has no children.
4. If a node is labelled with a state $q \in Q_\forall$, then for every $(p, a, q) \in \Delta$ (where a is the label that any outgoing edge must have) it has a child labelled with $p \in Q$.

5. A run tree is accepted iff every leaf node is at depth $|w|$ and are labelled with states in F.

The word w is accepted by the AFA iff there exists some accepted run tree of w.
This AFA is also a special case of the AFA that we have defined - here the transition function on the universal states will correspond to the "and" of all possible next states, and the transition function on the existential states will correspond to the "or" of all possible next states. Formally, this AFA is equivalent to $(Q, \Sigma, \delta, \varphi_0, F)$, where $Q = Q_\forall \cup Q_\exists$, $\varphi_0 = \bigvee\limits_{q \in I} q$, for $q \in Q_\exists$, $\delta(q,a) = \bigvee\limits_{(q,a,p)\in\Delta} p$, and for $q \in Q_\forall$, $\delta(q,a) = \bigwedge\limits_{(q,a,p)\in\Delta} p$ if such p exist, otherwise $\delta(q,a) = \bot$.

Theorem 1. *The languages accepted by AFAs are the regular languages.*

Proof. Every NFA is also an AFA and so every regular language is accepted by some AFA. Thus, all we have to prove is that the language of every AFA is regular. We do this by constructing an equivalent DFA for any AFA. We use the fact that every propositional formula has an equivalent DNF. We write the DNFs in set notation, as members of $2^{2^{Q\times\{0,1\}}}$ where any $f \in 2^{2^{Q\times\{0,1\}}}$ corresponds to the DNF $\bigvee\limits_{s\in f} \bigwedge\limits_{(q,b)\in s} q^b$ where $q^0 = \neg q$ and $q^1 = q$.
Given a formula $\varphi \in \mathtt{WFF}(Q)$, let $\mathtt{DNF}(\varphi) \in 2^{2^{Q\times\{0,1\}}}$ denote an equivalent DNF. Given the AFA $(Q, \Sigma, \delta, \varphi_0, F)$, consider the DFA $(2^{2^{Q\times\{0,1\}}}, \Sigma, \delta', s_0, F')$ where $s_0 = \mathtt{DNF}(\varphi_0)$, $F' = \{f \in 2^{2^{Q\times\{0,1\}}} : \alpha \models \bigvee\limits_{s\in f} \bigwedge\limits_{(q,b)\in s} q^b\}$ where $\alpha : Q \to \{0,1\}$ is such that $\alpha(q) = 1$ if $q \in F$, otherwise $\alpha(q) = 0$ and $\delta' : 2^{2^{Q\times\{0,1\}}} \times \Sigma \to 2^{2^{Q\times\{0,1\}}}$ is such that $\delta'(t,a) = \mathtt{DNF}\left(\bigvee\limits_{s\in t} \bigwedge\limits_{(q,b)\in s} \delta(q,a)^b\right)$.
It is easy to show via induction that for any word $w \in \Sigma^*$, $\hat{\delta}'(s_0, w) = \mathtt{DNF}(\hat{\delta}(\varphi_0, w))$. Our DFA accepts w iff $\hat{\delta}'(s_0, w) \in F$, ie $\mathtt{DNF}\left(\hat{\delta}(\varphi_0, w)\right) \in F$, ie $\alpha \models \hat{\delta}(\varphi_0, w)$. Therefore, w is accepted by the DFA iff it is accepted by the AFA, and hence the DFA and AFA are equivalent. Therefore, AFAs accept the regular languages. $\square$