Algebra in Automata Theory

Ashwin Abraham

IIT Bombay

11th November, 2023

Table of Contents

Monoids

Definition

A monoid is a set A equipped with an associative binary operation $\cdot : A^2 \to A$ with an identity $e \in A$.

For brevity, we refer to A as the monoid, and for $a, b \in A$, we denote $\cdot(a, b)$ as ab. By associativity, we have $(ab)c = a(bc)$ for all $a, b, c \in A$, and so we neglect to include the brackets, and write abc instead. An important corollary to the definition of monoids is the uniqueness of the identity.

Corollary (Uniqueness of the Identity)

The identity element in a monoid is unique.

Proof.

This follows immediately from the definition of the identity. An element in e in a monoid A is an identity iff $\forall x \in A, ex = xe = x$. If e and e' are two identities, then we have $ee' = e$ and $ee' = e'$, ie $e' = e$. $\square$

Free Monoids

Example

The set of functions A^A from a set A to itself is a monoid under function composition. The identity function is the identity element of this monoid.

Example (Free Monoid)

For any set Σ, the set of strings of elements of Σ, denoted by Σ^*, is a monoid with the associative binary operation being concatenation and identity ε (the empty string). This monoid is known as the free monoid over Σ.

Submonoids

Definition (Submonoids)

A subset S of a monoid A is said to be a submonoid of A iff

1. $e \in S$
2. $\forall x, y \in S, xy \in S$ (S is closed under the monoid operation)

Theorem (Submonoids are closed under intersection)

If K and G are two submonoids of a monoid A, then so is $K \cap G$.

Definition (Submonoid generated by a set)

For a monoid A, the submonoid generated by a subset $S \subseteq A$, denoted by $\langle S \rangle$, is the smallest submonoid containing S, or equivalently, the intersection of all submonoids of A containing S.

The existence of such a submonoid is guaranteed by the closure of submonoids under intersection.

Generated Monoids

Definition (Generated monoid)

A monoid A is said to be generated by a subset $S \subseteq A$ iff $\langle S \rangle = A$.

Definition (Finitely generated monoid)

A monoid is said to be finitely generated iff it is generated by a finite subset of itself.

Example

For any alphabet Σ, $\Sigma^* = \langle \Sigma \rangle$, ie the free monoid over Σ is generated by Σ itself.

Definition (Homomorphism between Monoids)

If A and B are two monoids, then a function $f : A \to B$ is said to be a homomorphism from A to B iff $\forall x, y \in A, f(xy) = f(x)f(y)$ and $f(e_A) = e_B$.

Definition (Isomorphism between Monoids)

An isomorphism between A and B is a homomorphism that is bijective.

Definition (Isomorphic Monoids)

Two monoids A and B are said to be isomorphic if there exists an isomorphism between them.

Theorem

If $f : A \to B$ is an isomorphism from A to B then $f^{-1} : B \to A$ is an isomorphism from B to A.

By the above theorem, it is easy to see that isomorphism is an equivalence relation on monoids.

Monoid Congruences

Definition (Congruence)

An equivalence relation $\sim$ over a monoid A is a right congruence iff $\forall x, y, z \in A$, $x \sim y \implies xz \sim yz$. Similarly, $\sim$ is a left congruence iff $\forall x, y, z \in A, x \sim y \implies zx \sim zy$. We say $\sim$ is a congruence iff it is both a left congruence and a right congruence.

Theorem

An equivalence relation $\sim$ over a monoid A is a congruence iff $\forall a, b, x, y \in A, a \sim b \wedge x \sim y \implies ax \sim by$

Proof.

If $\sim$ is a congruence, then for any $a, b, x, y \in A$ if $a \sim b$ and $x \sim y$, then $ax \sim bx$ and $bx \sim by$, ie $ax \sim by$. On the contrary, if $\sim$ satisfies $\forall a, b, x, y \in A, a \sim b \wedge x \sim y \implies ax \sim by$, then for any $x, y, z \in A$, if $x \sim y$, then since $z \sim z$, we have $xz \sim yz$ and $zx \sim zy$, ie $\sim$ is a congruence. $\square$

Quotient Monoids over Congruences

Theorem

Given a congruence $\sim$ over a monoid A, the congruence class containing the identity, $[e]$ is a submonoid of A.

Proof.

We have $e \in [e]$ and for any $x, y \in [e]$ we have $x \sim e$ and $y \sim e$, and hence $xy \sim e$, ie $xy \in [e]$. □

Theorem (Quotient monoid over a congruence)

The set of congruence classes of a monoid A under a congruence $\sim$ themselves form a monoid, under the operation $\cdot$ where $[x] \cdot [y] = [xy]$ with identity $[e]$. This monoid, $A/\sim$, is called the quotient of A over $\sim$.

Proof.

Note that $[x][y]$ is well defined, because if $[x] = [u]$ and $[y] = [v]$ then $x \sim u$ and $y \sim v$, and since $\sim$ is a congruence, this means $xy \sim uv$, ie $[xy] = [uv]$. Now, since $[e]$ is clearly an identity, it can be seen that the set of congruence classes forms a monoid. □

Myhill-Nerode Theorem

As we have already seen, Σ^* is a monoid under concatenation, with identity ε. It is known as the *free* monoid over Σ, as given any monoid N and a function $f : \Sigma \to N$, we can define a homomorphism $\hat{f} : \Sigma^* \to N$ such that $\hat{f}(a) = f(a)$ for each $a \in \Sigma$. There are many ways to characterize regular languages via monoids. One of them is by the Myhill-Nerode Theorem.

Definition (Saturation)

An equivalence relation $\sim$ over Σ^* is said to saturate a language $L \subseteq \Sigma^*$ iff $\forall x, y \in \Sigma^*, x \sim y \implies (x \in L \iff y \in L)$.

Corollary

An equivalence relation $\sim$ over Σ^* saturates a language $L \subseteq \Sigma^*$ iff for any $x \in \Sigma^*$ either $[x] \subseteq L$ or $[x] \cap L = \emptyset$, which occurs iff $L = \bigcup_{x \in L} [x]$.

Myhill-Nerode Theorem

Theorem (Myhill-Nerode)

A language is regular iff there exists a right congruence of finite index saturating it.

Proof.

If $\sim$ is a right congruence of finite index over Σ^* saturating $L \subseteq \Sigma^*$, then consider the DFA $A = (\{[x] : x \in \Sigma^*\}, \Sigma, \delta, [\varepsilon], \{[x] : x \in L\})$[a] where $\delta : \{[x] : x \in \Sigma^*\} \times \Sigma \to \{[x] : x \in \Sigma^*\}$ is such that for any $x \in \Sigma^*$ and $a \in \Sigma$, $\delta([x], a) = [xa]$. Note that since $\sim$ is a right congruence, if $[x] = [y]$, ie $x \sim y$, then $xa \sim ya$, ie $[xa] = [ya]$, ie δ is well defined. Clearly, $\hat{\delta}([\varepsilon], w) = [w]$, for any $w \in \Sigma^*$. Now, $[w] \in \{[x] : x \in L\}$ iff there exists $x \in L$ such that $[w] = [x]$, ie $w \sim x$. Since $\sim$ saturates L, this occurs iff $w \in L$. □

[a] Since $\sim$ has finite index, ie finite number of equivalence classes, the number of states of this DFA is indeed finite.

Myhill-Nerode Theorem

Proof.

On the other hand, if $L \subseteq \Sigma^*$ is regular, ie it is recognized by the DFA $(Q, \Sigma, \delta, q_0, F)$ where Q is a finite set of states, with $q_0 \in Q$, $F \subseteq Q$ and $\delta : Q \times \Sigma \to Q$. Consider the equivalence relation $\sim$ on Σ^* where $x \sim y$ iff $\hat{\delta}(q_0, x) = \hat{\delta}(q_0, y)$. This is a right congruence, as for any $x, y, z \in \Sigma^*$, if $x \sim y$, ie $\hat{\delta}(q_0, x) = \hat{\delta}(q_0, y) = q$, then $\hat{\delta}(q_0, xz) = \hat{\delta}(q_0, yz) = \hat{\delta}(q, z)$. Furthermore, $\sim$ saturates L, since if $x \sim y$, then $\hat{\delta}(q_0, x) = \hat{\delta}(q_0, y)$, and $x \in L \iff \hat{\delta}(q_0, x) \in F \iff \hat{\delta}(q_0, y) \in F \iff y \in L$. Furthermore, the index of $\sim$ is at most $|Q|$, ie it is finite. This is as there exists an injection $f : \{[x] : x \in \Sigma^*\} \to Q$ where $f([x]) = \hat{\delta}(q_0, x)$. This is well defined, as if $[x] = [y]$, then $x \sim y$ and hence $\hat{\delta}(q_0, x) = \hat{\delta}(q_0, y)$, and is an injection, since if $[x] \neq [y]$, ie $x \nsim y$, then $\hat{\delta}(q_0, x) \neq \hat{\delta}(q_0, y)$, ie $f([x]) \neq f([y])$. Therefore, there exists a right congruence of finite index saturating L. □

The Nerode Equivalence

Definition (Nerode equivalence)

For any language $L \subseteq \Sigma^*$, we define the Nerode equivalence $\sim_L$ on Σ^* such that for any $x, y \in \Sigma^*$, $x \sim_L y$ iff $\forall z \in \Sigma^*, xz \in L \iff yz \in L$.

Theorem

For any language $L \subseteq \Sigma^*$, the Nerode equivalence $\sim_L$ is the coarsest[a] right congruence saturating it.

[a] An equivalence relation $\sim$ is said to be coarser than $\sim'$ iff $\sim' \subseteq \sim$

Corollary

A language $L \subseteq \Sigma^*$, is regular iff the Nerode equivalence $\sim_L$ has finite index.

This corollary holds since if L is regular, then there exists a right congruence of finite index saturating it, and since $\sim_L$ is at least as coarse as it, it too must have finite index. On the other hand, if $\sim_L$ has finite index, then by the Myhill-Nerode Theorem, L must be regular.

The Nerode Equivalence

Proof.

We first have to show that $\sim_L$ is a right congruence saturating L. This holds since for any $x, y, z \in \Sigma^*$, if $xz \nsim_L yz$, then $\exists u \in \Sigma^*$ such that exactly one of xzu and yzu are in L. This means there exists $v = zu \in \Sigma^*$ such that exactly one of xv and yv are in L, ie $x \nsim_L y$. Therefore, $x \sim_L y \implies xz \sim_L yz$, ie $\sim_L$ is a right congruence. Also, if exactly one of x and y are in L, then there exists $u = \varepsilon \in \Sigma^*$ such that exactly one of xu and yu are in L, ie $x \nsim_L y$. Therefore, $x \sim_L y \implies (x \in L \iff y \in L)$, ie $\sim_L$ saturates L.

Now, it remains to show that for any right congruence $\sim_L$ saturating L, and any $x, y \in \Sigma^*$, $x \sim y \implies x \sim_L y$. This holds since if $x \sim y$, then for any $z \in \Sigma^*$, $xz \sim yz$ (since $\sim$ is a right congruence), and since $\sim$ saturates L, this means that $xz \in L \iff yz \in L$ for any $z \in \Sigma^*$, ie $x \sim_L y$. $\square$

Minimal DFA

Theorem (Minimal DFA)

If $L \subseteq \Sigma^*$ is a regular language, $(\{[x]_L : x \in \Sigma^*\}, \Sigma, \delta, [\varepsilon]_L, \{[x]_L : x \in L\})$ is the unique (upto isomorphism) minimal DFA recognizing L, where $[x]_L$ denotes the equivalence class of the Nerode equivalence $\sim_L$ containing x, and δ is defined such that $\delta([x]_L, a) = [xa]_L$ for any $x \in \Sigma^*, a \in \Sigma$.

Proof.

By the corollary presented earlier, since L is regular, the Nerode equivalence is of finite index, and hence this DFA indeed has a finite number of states. Also, since the Nerode equivalence is a right congruence, $[x]_L = [y]_L \implies x \sim_L y \implies xa \sim_L ya \implies [xa]_L = [ya]_L$, ie δ is well defined. For this DFA, $\hat{\delta}([\varepsilon]_L, w) = [w]_L$ and so a word w is accepted iff $\hat{\delta}([\varepsilon], w) \in \{[x]_L : x \in L\}$, which occurs iff $\exists z \in L, z \sim_L w$, and since $\sim_L$ saturates L, this occurs iff $w \in L$. Therefore, the language recognized by this DFA is L. □

Minimal DFA

Proof (Contd.)

Now, for any DFA $(Q, \Sigma, \delta, q_0, F)$ recognizing L, consider the equivalence $\sim$ where for any $x, y \in \Sigma^*$, $x \sim y$ iff $\hat{\delta}(q_0, x) = \hat{\delta}(q_0, y)$. As shown earlier, $\sim$ is a right congruence over Σ^* saturating L, and for any $x, y \in \Sigma^*$, $x \sim y \implies x \sim_L y$. Consider the function $f : \{[x]_L : x \in \Sigma^*\} \to 2^Q$ where $f([x]) = \{\hat{\delta}(q_0, y) : y \sim_L x\}$ for every $x \in \Sigma^*$. Note that $|f([x])| \geq 1$ for each $x \in \Sigma^*$, since $\hat{\delta}(q_0, x) \in f([x]_L)$ for every word x. Also for any $x, y \in \Sigma^*$, if $f([x]_L) \cap f([y]_L) \neq \emptyset$, then $\exists u \sim_L x, v \sim_L y$ such that $\hat{\delta}(q_0, u) = \hat{\delta}(q_0, v)$, ie $u \sim v$, which implies that $u \sim_L v$, and hence $x \sim_L y$, ie $[x]_L = [y]_L$. From this, we deduce that $|Q| \geq \sum\limits_{C \in dom(f)} |f(C)|$ and since $f(C) \geq 1$ for each equivalence class C, $|Q|$ must be at least the index of $\sim_L$, which is the number of states in the previously constructed DFA. Therefore, the DFA constructed from the Nerode equivalence is minimal. Finally, we show that if equality holds, then the automaton is isomorphic to the one constructed from the Nerode equivalence. $\square$

Minimal DFA

Proof (Contd.)

If equality holds, then we must have $|f([x]_L)| = 1$ for every $x \in \Sigma^*$ and the range of f must cover Q. Therefore, for any words $x, y \in \Sigma^*$, $x \sim_L y \implies \hat{\delta}(q_0, x) = \hat{\delta}(q_0, y)$, ie $\sim$ and $\sim_L$ coincide. This means that the function $f^* : \{[x]_L : x \in \Sigma^*\} \to Q$ such that $f^*([x]_L) = \hat{\delta}(q_0, x)$ is a bijection. This function is well defined, since if $[x]_L = [y]_L$, then $x \sim_L y$, ie $x \sim y$, ie $\hat{\delta}(q_0, x) = \hat{\delta}(q_0, y)$, and it is an injection, since if $f^*([x]_L) = f^*([y]_L)$, then $\hat{\delta}(q_0, x) = \hat{\delta}(q_0, y)$, ie $x \sim y$, ie $[x]_L = [y]_L$. Since the range of f covers Q, for every $q \in Q$ there exists $x \in \Sigma^*$ such that $q \in f([x])$. Since $f^*([x]_L) \in f([x]_L)$ and $f([x]_L)$ is a singleton, we have $f^*([x]_L) = q$, which means that f^* is also a surjection, ie it is a bijection. Now, note that $f^*([\varepsilon]_L) = \hat{\delta}(q_0, \varepsilon) = q_0$, and $f^*(\{[x]_L : x \in L\}) = \{\hat{\delta}(q_0, x) : x \in L\} = F$. Equality holds in the previous equation, since f^* is a bijection, which means for every $q \in F$, there is some $x \in \Sigma^*$ such that $f([x]_L) = \hat{\delta}(q_0, x) = q$. Since $\hat{\delta}(q_0, x) \in F$, we get $x \in L$. Finally, note that $f^*([xa]_L) = \hat{\delta}(q_0, xa) = \delta(f^*([x]_L), a)$, all of which show that f^* is an isomorphism between the two automata. $\square$

Monoids as Recognizers of Languages

Definition (Language recognized by a Monoid)

Given a monoid M and a subset $X \subseteq M$, and a homomorphism $h : \Sigma^* \to M$, we call the language $h^{-1}(X) \subseteq \Sigma^*$ as the language recognized by X with respect to h. We say that a language $L \subseteq \Sigma^*$ is recognized by a monoid M if there exists $X \subseteq M$ and a homomorphism $h : \Sigma^* \to M$ such that L is recognized by X with respect to h.

Theorem

A language $L \subseteq \Sigma^*$ is regular iff it is recognized by a finite monoid.

To prove this, we will introduce a few important congruences and monoids.

The Syntactic Congruence

Definition (Syntactic Congruence)

For any language $L \subseteq \Sigma^*$, we define the syntactic congruence $\sim_L$ such that $x \sim_L y \iff \forall u, v \in \Sigma^*, uxv \in L \iff uyv \in L$.

Theorem

The syntactic congruence $\sim_L$ is the coarsest congruence saturating L.

Proof.

Firstly, note that if for any $x, y, z \in \Sigma^*$, if $x \sim_L y$, then $xz \sim_L yz$ and $zx \sim_L zy$, ie $\sim_L$ is a congruence. This is because if $xz \nsim_L yz$, then $\exists u, v \in \Sigma^*$ such that exactly one of $uxzv$ and $uyzv$ are in L, which means $\exists u, v' = zv \in \Sigma^*$ such that exactly one of uxv' and uyv' are in L, and similarly if $zx \nsim_L zy$ then $x \nsim_L y$. Note that if $x \sim_L y$, then $\varepsilon x \varepsilon \in L \iff \varepsilon y \varepsilon \in L$, ie $x \in L \iff y \in L$, ie $\sim_L$ saturates L. $\square$

Syntactic Monoids

Proof (Contd.)

Now, if $\sim$ is any congruence saturating L, then for any $x, y \in \Sigma^*$, if $x \sim y$, then for every $u, v \in \Sigma^*$, $uxv \sim uyv$ (since $\sim$ is a congruence). Now, since $\sim$ saturates L, $uxv \sim uyv \implies (uxv \in L \iff uyv \in L)$. Therefore, $x \sim y \implies \forall u, v \in \Sigma^*, uxv \in L \iff uyv \in L$, ie for every $x, y \in \Sigma^*$, $x \sim y \implies x \sim_L y$. □

Definition (Syntactic Monoid)

The quotient monoid $\Sigma^* / \sim_L$ where $\sim_L$ is the syntactic congruence of L over Σ^* is known as the syntactic monoid of L.

Theorem

Every language $L \subseteq \Sigma^*$ is recognized by its syntactic monoid.

Proof.

Take $[L]_L = \{[x]_L : x \in L\} \subseteq \Sigma^* / \sim_L$ and $h : \Sigma^* \to \Sigma^* / \sim_L$ as the homomorphism such that $h(w) = [w]_L$. Then $h^{-1}([L]_L) = L$. □

Syntactic Monoids

Theorem

If a language $L = h^{-1}(X)$ where $X \subseteq M$ is a subset of a monoid M and $h : \Sigma^* \to M$ is a homomorphism, then there is a homomorphism $h_L : h(\Sigma^*) \to \Sigma^* / \sim_L$ such that $h_L \circ h$ is the canonical homomorphism mapping an element of Σ^* to the equivalence class of $\sim_L$ containing it.

Proof.

Let $\eta_L : \Sigma^* \to \Sigma^* / \sim_L$ denote the canonical homomorphism, satisfying $\eta_L(x) = [x]_L$ for any $x \in \Sigma^*$. Consider the equivalence relation $\sim_h$ over Σ^* where $x \sim_h y$ iff $h(x) = h(y)$. It is easy to see that this is a congruence saturating L, which means that $x \sim_h y \implies x \sim_L y$. Define h_L to be such that $h_L(m) = \left[h^{-1}(m)\right]_L$, for any $m \in h(\Sigma^*)$. Note that since $m \in h(\Sigma^*)$, $h^{-1}(m)$ is non-empty, and if $x, y \in h^{-1}(m)$, then $h(x) = h(y) = m$, ie $x \sim_h y$ which means that $x \sim_L y$, ie $[x]_L = [y]_L$. Therefore, h_L is well defined. h_L is also a homomorphism from $h(\Sigma^*)$ to $\Sigma^* / \sim_L$. □

Syntactic Monoids

Proof (Contd.)

To see this, note that $h_L(e) = h_L(h(\varepsilon)) = [\varepsilon]_L$ and if $p, q \in h(\Sigma^*)$, say $p = h(x)$ and $q = h(y)$ for some $x, y \in \Sigma^*$, then $h_L(pq) = h_L(h(x)h(y))$ $= h_L(h(xy)) = [xy]_L = h_L(p)h_L(q)$.
Now, for any $x \in \Sigma^*$, $h_L(h(x)) = \left[h^{-1}(h(x))\right]_L = [x]_L$, ie $h_L \circ h = \eta_L$. □

Corollary

The syntactic monoid of a language is the smallest monoid recognizing it.

Proof.

We have already shown that the syntactic monoid of a language L recognizes it and for any other monoid M recognizing it with subset X and homomorphism h, we have shown that there exists a homomorphism $h_L : h(\Sigma^*) \to \Sigma^*/\sim_L$ such that $h_L \circ h = \eta_L$, where η_L is the canonical homomorphism from Σ^* to $\Sigma^*/\sim_L$. η_L is a surjection, and therefore, h_L must be a surjection from $h(\Sigma^*)$ to $\Sigma^*/\sim_L$. Hence $|h(\Sigma^*)| \geq |\Sigma^*/\sim_L|$, ie $|M| \geq |\Sigma^*/\sim_L|$. If L is regular, ie the syntactic monoid is finite, then it is the unique minimal monoid recognizing L upto isomorphism. □

Transition Monoids

Definition (Transition Monoid)

The transition monoid of a DFA $(Q, \Sigma, \delta, q_0, F)$ is the submonoid of Q^Q generated by $\{\hat{\delta}_a : a \in \Sigma\}$, where $\hat{\delta}_a(q) = \delta(q, a)$ for any $q \in Q, a \in \Sigma$.

We take the binary operation of Q^Q to be flipped function composition, ie for $f, g \in Q^Q$, $fg = g \circ f$. The transition monoid has underlying set $\{\hat{\delta}_x : x \in \Sigma^*\}$ where $\hat{\delta}_x(q) = \hat{\delta}(q, x)$ for any $q \in Q, x \in \Sigma^*$. $\hat{\delta}_\varepsilon$ is the identity function, and $\hat{\delta}_x\hat{\delta}_y = \hat{\delta}_{xy}$. Note that the transition monoid is finite.

Theorem

The language of any automaton is recognized by its transition monoid.

Proof.

For a DFA $(Q, \Sigma, \delta, q_0, F)$ with transition monoid T, take the subset $X = \{f \in T : f(q_0) \in F\}$ and homomorphism $h : \Sigma^* \to T$ such that $h(x) = \hat{\delta}_x$. $h^{-1}(X)$ is the language of this DFA. $\square$

Transition Monoids

Theorem (Isomorphism between Syntactic and Transition Monoids)

If a language $L \subseteq \Sigma^*$ is regular, then its syntactic monoid is isomorphic to the transition monoid of the minimal DFA recognizing it.

Proof.

For a regular language L, let $\sim_L$ denote the syntactic congruence and $\sim$ denote the Nerode equivalence. Let T denote the transition monoid of the minimal DFA recognizing L. Consider the function $f : \Sigma^*/\sim_L \to T$ such that for any $x \in \Sigma^*$, $f([x]_L) = \hat{\delta}_x$. This is well defined as for any $x, y, p, q \in \Sigma^*$, if $[x]_L = [p]_L$ and $[y] = [q]$, ie $x \sim_L p$ and $y \sim q$, then $yx \sim qx$, and $qx \sim_L qp$, which implies that $qx \sim qp$ (since $\sim_L \subseteq \sim$) Therefore, $yx \sim qp$, ie $[yx] = [qp]$, ie $\hat{\delta}_x(y) = \hat{\delta}_p(q)$. Also, for any $x, y \in \Sigma^*$, if $f([x]_L) = f([y]_L)$, then for any $u \in \Sigma^*$, $\hat{\delta}_x(u) = \hat{\delta}_y(u)$, ie $[ux] = [uy]$, ie for any $u, v \in \Sigma^*, uxv \in L \iff uyv \in L$, ie $x \sim_L y$ and hence $[x]_L = [y]_L$, ie f is an injection. Since f is clearly a surjection, which means f is a bijection. Now, $f([x]_L [y]_L) = f([xy]_L) = \hat{\delta}_{xy} = \hat{\delta}_x \hat{\delta}_y$ $= f([x]_L) f([y]_L)$, hence f is an isomorphism between $\Sigma^*/\sim_L$ and T. □

Syntactic Monoids and Transition Monoids

Theorem

A language L is regular iff its syntactic monoid is finite.

Proof.

If L is regular, then its syntactic monoid is isomorphic to the transition monoid of the minimal DFA recognizing L, and hence it is finite. On the other hand, if the syntactic monoid of L is finite, then the DFA $(\Sigma^*/\sim_L, \Sigma, \delta, [\varepsilon]_L, \{[x]_L : x \in L\})$, where $\delta([x]_L, a) = [xa]_L$ for any $x \in \Sigma^*, a \in \Sigma$ recognizes the language L, ie L is regular. □

This construction works for any language recognized by a finite monoid. If $L = h^{-1}(X)$ where X is a subset of a finite monoid M and $h : \Sigma^* \to M$ is a homomorphism, then L is accepted by the DFA $(M, \Sigma, \delta, h(\varepsilon), X)$, where $\delta(m, a) = mh(a)$ for any $m \in M, a \in \Sigma$. It is easy to see that $\hat{\delta}(h(\varepsilon), w) = h(w)$ which is in X iff $w \in L$. With these results, it is quite easy to show that a language is regular iff it is accepted by a finite monoid.

Star-Free Languages

Definition (Star-Free Language)

The class of star free languages with alphabet Σ is the smallest collection of languages containing $\emptyset$ and $\{a\}$ (for each $a \in \Sigma$) that is closed under union, intersection, complement, and concatenation.

The star-free languages are therefore those that can be written as regular expressions without using the Kleene star.

Theorem

The class of star free languages is precisely the class of languages definable in FO[$<$]

Proof.

We first prove that every star free language can be defined in FO[$<$]. This is done by structural induction. Clearly, $\emptyset$ and the set $\{a\}$ for $a \in \Sigma$ can be defined in FO ($\exists x(x \neq x)$ and $\exists x(a(x) \wedge \forall y(x = y))$). For any FO sentences φ and ψ, $L(\varphi)^c = L(\neg\varphi)$, $L(\varphi) \cup L(\psi) = L(\varphi \vee \psi)$ and $L(\varphi) \cap L(\psi) = L(\varphi \wedge \psi)$. $\square$

Proof (Contd.)

For any sentence φ and formula $\eta(x)$ with a free variable x, let φ_η denote the sentence obtained by recursively replacing every $\exists x(.)$ with $\exists x\,[\eta(x) \wedge (.)]$ and by replacing every $\forall x(.)$ with $\forall x\,[\eta(x) \implies (.)]$. Now, $L(\varphi)L(\psi) = L\left(\exists p\,[(\varphi_{\eta_1} \wedge \psi_{\neg\eta_1}) \vee (\varphi_{\eta_2} \wedge \psi_{\neg\eta_2})]\right)$ where $\eta_1(x)$ is $x < p$ and $\eta_2(x)$ is $x < p \vee x = p$. Hence, by structural induction, every star free language is definable in FO[$<$].

Next, we prove that every FO[$<$] definable language is star-free. WLOG, assume the formula φ is written in prenex normal form with only existential quantifiers. For any such FO formula φ with free variables $x_1, \ldots x_n$, we associate a regular expression $L(\varphi)$ over the alphabet $\Sigma \times \{0,1\}^n$ such that $(w, s_1, \ldots s_n) \in L(\varphi)$ iff each s_i has a single occurence of 1 and $w \models_s \varphi$, where s is interpreted as an assignment to the free variables (x_i is assigned to the position of the 1 in s_i). For variables $x_1, \ldots x_n$, let $\mathtt{Val}(x_1, \ldots x_n)$ denote the following regular expression over $\Sigma \times \{0,1\}^n$: $\square$

Proof (Contd.)

$$\begin{pmatrix}\Sigma\\0\\\mathcal{B}\\\vdots\\\mathcal{B}\end{pmatrix}^* \begin{pmatrix}\Sigma\\1\\\mathcal{B}\\\vdots\\\mathcal{B}\end{pmatrix} \begin{pmatrix}\Sigma\\0\\\mathcal{B}\\\vdots\\\mathcal{B}\end{pmatrix}^* \cap \begin{pmatrix}\Sigma\\\mathcal{B}\\0\\\vdots\\\mathcal{B}\end{pmatrix}^* \begin{pmatrix}\Sigma\\\mathcal{B}\\1\\\vdots\\\mathcal{B}\end{pmatrix} \begin{pmatrix}\Sigma\\\mathcal{B}\\0\\\vdots\\\mathcal{B}\end{pmatrix}^* \cap \dots \begin{pmatrix}\Sigma\\\mathcal{B}\\\mathcal{B}\\\vdots\\0\end{pmatrix}^* \begin{pmatrix}\Sigma\\\mathcal{B}\\\mathcal{B}\\\vdots\\1\end{pmatrix} \begin{pmatrix}\Sigma\\\mathcal{B}\\\mathcal{B}\\\vdots\\0\end{pmatrix}^*$$

where $\mathcal{B} = \{0, 1\}$. This denotes the allowed words for formulae with free variables $x_1 \dots x_n$. For atomic formulae, the following hold:

1. $L(x_1 = x_2) = \begin{pmatrix}\Sigma\\0\\0\\\mathcal{B}\\\vdots\\\mathcal{B}\end{pmatrix}^* \begin{pmatrix}\Sigma\\1\\1\\\mathcal{B}\\\vdots\\\mathcal{B}\end{pmatrix} \begin{pmatrix}\Sigma\\0\\0\\\mathcal{B}\\\vdots\\\mathcal{B}\end{pmatrix}^* \cap \texttt{Val}(x_1, \dots x_n)$

Star-Free Languages

Proof (Contd.)

2. $$L(x_1 < x_2) = \begin{pmatrix} \Sigma \\ 0 \\ 0 \\ \mathcal{B} \\ \vdots \\ \mathcal{B} \end{pmatrix}^* \begin{pmatrix} \Sigma \\ 1 \\ 0 \\ \mathcal{B} \\ \vdots \\ \mathcal{B} \end{pmatrix} \begin{pmatrix} \Sigma \\ 0 \\ 0 \\ \mathcal{B} \\ \vdots \\ \mathcal{B} \end{pmatrix}^* \begin{pmatrix} \Sigma \\ 0 \\ 1 \\ \mathcal{B} \\ \vdots \\ \mathcal{B} \end{pmatrix} \begin{pmatrix} \Sigma \\ 0 \\ 0 \\ \mathcal{B} \\ \vdots \\ \mathcal{B} \end{pmatrix}^* \cap \texttt{Val}(x_1, \ldots x_n)$$

3. $$L(a(x_1)) = \begin{pmatrix} \Sigma \\ 0 \\ \mathcal{B} \\ \vdots \\ \mathcal{B} \end{pmatrix}^* \begin{pmatrix} a \\ 1 \\ \mathcal{B} \\ \vdots \\ \mathcal{B} \end{pmatrix} \begin{pmatrix} \Sigma \\ 0 \\ \mathcal{B} \\ \vdots \\ \mathcal{B} \end{pmatrix}^* \cap \texttt{Val}(x_1, \ldots x_n)$$

And for formulae φ and ψ with free variables $x_1, \ldots x_n$ we have: $\square$

Proof (Contd.)

1. $L(\varphi \wedge \psi) = L(\varphi) \cap L(\psi)$
2. $L(\varphi \vee \psi) = L(\varphi) \cup L(\psi)$
3. $L(\neg\varphi) = \mathtt{Val}(x_1, \dots x_n) - L(\psi)$
4. $L(\exists x \varphi) = \mathtt{Proj}_x(L(\varphi))$

Where $\mathtt{Proj}_x$ corresponds to projecting out (ie deleting) the row corresponding to x in the regular expression. Hence, for sentences φ we will get an equivalent regular expression over Σ. This regular expression will be star-free, since none of the operations listed above (union, intersection, set difference[a] and projection) introduce the Kleene star, and all the regular expressions for the atomic formulae and $\mathtt{Val}$ are star-free. To see this, note that: □

[a]Note that $A - B = A \cap B^c$

Star-Free Languages

Proof (Contd.)

1. $$\begin{pmatrix} \Sigma \\ 0 \\ \mathcal{B} \\ \vdots \\ \mathcal{B} \end{pmatrix}^* = \emptyset^c - \emptyset^c \begin{pmatrix} \Sigma \\ 1 \\ \mathcal{B} \\ \vdots \\ \mathcal{B} \end{pmatrix} \emptyset^c$$
2. $$\begin{pmatrix} \Sigma \\ 0 \\ 0 \\ \mathcal{B} \\ \vdots \\ \mathcal{B} \end{pmatrix}^* = \emptyset^c - \emptyset^c \begin{pmatrix} \Sigma \\ 1 \\ \mathcal{B} \\ \mathcal{B} \\ \vdots \\ \mathcal{B} \end{pmatrix} \emptyset^c - \emptyset^c \begin{pmatrix} \Sigma \\ \mathcal{B} \\ 1 \\ \mathcal{B} \\ \vdots \\ \mathcal{B} \end{pmatrix} \emptyset^c$$

Therefore, the language of any first order sentence φ is star-free, and hence the class of star-free languages is precisely the set of languages definable in FO[<]. □

Schutzenberger's Theorem

Definition (Aperiodic Monoid)

A finite monoid M is said to be aperiodic iff there exists a natural n such that for every $m \in M$, $m^{n+1} = m^n$.

Theorem (Schutzenberger)

A language is star-free iff it is recognized by an aperiodic monoid.

Before proving this, let us use the following lemma to restate the theorem.

Lemma

A language is recognized by an aperiodic monoid iff its syntactic monoid is aperiodic.

We can therefore restate the theorem like so:

Theorem (Schutzenberger)

A language is star-free iff its syntactic monoid is aperiodic.

Schutzenberger's Theorem

Proof (of Lemma).

The "if" part of the lemma is trivial, since every language is recognized by its syntactic monoid. For the converse, if the aperiodic monoid M recognizes the language $L \subseteq \Sigma^*$, then there exists a subset $X \subseteq M$ and homomorphism $h : \Sigma^* \to M$ such that $h^{-1}(X) = L$. We have also shown that there will exist a homomorphism $h_L : h(\Sigma^*) \to \Sigma^*/\sim_L$ such that $h_L \circ h = \eta_L$, where η_L is the canonical homomorphism from Σ^* to the syntactic monoid. Since η_L is surjective, h_L must be surjective as well. Since M is aperiodic, there exists a natural n such that for every $m \in M$, $m^{n+1} = m^n$. For any $p \in \Sigma^*/\sim_L$ let $t \in h(\Sigma^*)$ be such that $h_L(t) = p$. $p^{n+1} = h_L(t)^{n+1} = h_L(t^{n+1}) = h_L(t^n) = p^n$. Hence the syntactic monoid is aperiodic as well. □