1 An equivalence relation on strings

1.1 Preliminaries

Equivalence Relations

Definition 1. A binary relation $R \subseteq A \times A$ is an *equivalence* relation iff

Reflexivity For every $a \in A$, $(a, a) \in R$,

Symmetry For every $a, b \in A$, if $(a, b) \in R$ then $(b, a) \in R$, and

Transitivity For every $a, b, c \in A$, if $(a, b) \in R$ and $(b, c) \in R$ then $(a, c) \in R$.

For an equivalence relation $\equiv$, we will often write $a \equiv b$ instead of $(a, b) \in \equiv$.

Definition 2. For an equivalence relation $\equiv \subseteq A \times A$, the *equivalence class* of $a \in A$ (denoted $[a]_\equiv$) is given by

$$[a]_\equiv = \{b \in A \mid b \equiv a\}$$

The *index* of $\equiv$, denoted as $\#(\equiv)$, is the number of equivalence classes of $\equiv$. We will say that $\equiv$ has *finite index* if $\#(\equiv)$ is a finite number.

Example 3. Consider the relation $=_3 \subseteq \mathbb{N} \times \mathbb{N}$ such that $(i, j) \in =_3$ iff $i \bmod 3 = j \bmod 3$. It is easy to see that $=_3$ is reflexive, symmetric, and transitive (and hence an equivalence relation).

The equivalence class of 5 is given by

$$[5]_{=_3} = \{i \in \mathbb{N} \mid i \bmod 3 = 2 = 5 \bmod 3\}$$

The relation $=_3$ has 3 equivalence classes given by

$$A_0 = \{i \in \mathbb{N} \mid i \bmod 3 = 0\}$$
$$A_1 = \{i \in \mathbb{N} \mid i \bmod 3 = 1\}$$
$$A_2 = \{i \in \mathbb{N} \mid i \bmod 3 = 2\}$$

Thus, $\#(=_3) = 3$.

Let us consider another equivalence relation $= \subseteq \mathbb{N} \times \mathbb{N}$ such that $(i, j) \in =$ iff $i = j$. Now the equivalence class for any number i is $[i]_= = \{i\}$. The collection of all equivalence classes of $=$ is $\{\{i\} \mid i \in \mathbb{N}\}$. Thus $\#(=)$ is infinite.

1.2 An Equivalence Relation on Strings

A Language theoretic equivalence

Definition 4. For any $L \subseteq \Sigma^*$, define $\equiv_L \subseteq \Sigma^* \times \Sigma^*$ such that

$$x \equiv_L y \text{ iff } \forall z \in \Sigma^*.\ xz \in L \leftrightarrow yz \in L$$

Proposition 5. *For any language L, $\equiv_L$ is an equivalence relation.*

Proof left as exercise. ______

Examples

Example 6. Let $L = \{w \mid w \text{ has an odd number of 0s and 1s }\}$. Observe that $110 \equiv_L 000$ because for any $z \in \{0,1\}^*$

$$110z \in L \text{ iff } z \text{ has an odd number of 1s and an even number of 0s iff } 000z \in L$$

In fact, $[110]_{\equiv_L} = \{w \mid w \text{ has an even number of 1s and an odd number of 0s}\}$. Consider

$$\begin{aligned} A_{ee} &= \{w \mid w \text{ has an even number of 0s and 1s}\} \\ A_{oe} &= \{w \mid w \text{ has an even number of 0s and an odd number of 1s}\} \\ A_{eo} &= \{w \mid w \text{ has an odd number of 0s and an even number of 1s}\} \\ A_{oo} &= \{w \mid w \text{ has an odd number of 0s and 1s}\} \end{aligned}$$

Now for any $x, y \in A_{ee}$, we can show that $x \equiv_L y$. Let z be any string. $xz \in L$ iff z has an odd number of 0s and 1s iff $yz \in L$. Similarly one can show that any pair of strings in A_{oe} (or A_{eo} or A_{oo}) are equivalent w.r.t. $\equiv_L$.

On the other hand, if x and y belong to different sets above then $x \not\equiv_L y$. For example, let $x \in A_{ee}$ and $y \in A_{oe}$. Then $x10 \in L$ (because $x10$ has an odd number of 0s and 1s). But $y10 \notin L$ because $y10$ has an even number of 0s and an odd number of 1s. The other cases are similar.

Thus, the collection of equivalence classes of $\equiv_L$ is $\{A_{ee}, A_{oe}, A_{eo}, A_{oo}\}$. Therefore $\#(\equiv_L) = 4$.

Example 7. Let $P = \{w \mid w \text{ contains 001 as a substring }\}$. Observe that $x = 10 \not\equiv_P y = 100$ because taking $z = 1$, we have $xz = 101 \notin P$ but $yz = 1001 \in P$. On the other hand, $1001 \equiv_P 001$ because for every z, we have $1001z \in L$ and $001z \in L$. The equivalence classes of $\equiv_P$ are

$$\begin{aligned} &A_{001} = \{w \mid w \text{ has 001 as a substring}\} \\ &A_0 = \{w \mid w \text{ does not have 001 as substring and ends in 0 and the second last symbol is not 0}\} \\ &A_{00} = \{w \mid w \text{ does not have 001 as substring and ends in 00}\} \\ &A_1 = \{w \mid w \text{ does not have 001 as substring and ends in 1 or is } \epsilon\} \end{aligned}$$

One can show that for any two strings x, y that belong to the same set (in the above listing), $x \equiv_P y$, and x, y belong to different sets then $x \not\equiv_P y$. We show these for one particular case; the rest can be similarly established. Consider $x, y \in A_0$. Now $xz \in P$ iff either z has 001 as a substring or z begins with 01 iff $yz \in L$. On the other hand, suppose $x \in A_0$ and $y \in A_{00}$. Take $z = 1$. Now $yz = y1 \in P$ because the last 3 symbols of yz is 001. On the other hand $xz = x1 \notin L$ because $x1$ does not have 001 as a substring.

Since the collection of all equivalence classes of $\equiv_P$ is $\{A_{001}, A_0, A_{00}, A_1\}$, $\#(\equiv_P) = 4$.

Example 8. Consider $L_{0n1n} = \{0^n1^n \mid n \geq 0\}$. Consider $x = 0^i$ and $y = 0^j$ with $i \neq j$. $x \not\equiv_{L_{0n1n}} y$ because $0^i1^i \in L_{0n1n}$ but $0^j1^i \notin L_{0n1n}$. In fact, for any i, $[0^i]_{\equiv_{L_{0n1n}}} = \{0^i\}$. If we consider strings of the form 0^i1^j where $1 \leq j \leq i$, we have $[0^i1^j]_{\equiv_{L_{0n1n}}} = \{0^k1^\ell \mid k - \ell = i - j\}$ because $0^i1^jz \in L_{0n1n}$ iff $z = 1^{j-i}$ iff $0^k1^\ell \in L_{0n1n}$ when $k - \ell = i - j$. Finally, when we consider any two strings x and y such that x and y are not of the form 0^i1^j, where $j \leq i$, we have xz and yz are never in the set L_{0n1n}, and so (vaccuously) $x \equiv_{L_{0n1n}} y$.

Based on the above analysis, $\#(\equiv_{L_{0n1n}})$ is infinite.

Properties of $\equiv_L$

Proposition 9. *For any language L, if $x \equiv_L y$ then for any w, $xw \equiv_L yw$.*

Proof. Assume for contradiction that $x \equiv_L y$ but for some w, $xw \not\equiv_L yw$. Since $xw \not\equiv_L yw$, there is a z such that either ($xwz \in L$ and $ywz \notin L$) or ($xwz \notin L$ and $ywz \in L$). In either case, we can conclude that $x \not\equiv_L y$ because taking $z' = wz$, we have $xz' \in L$ and $xz' \notin L$ (or $xz' \notin L$ and $yz' \in L$). This contradicts the assumption that $x \equiv_L y$. □

---

2 Myhill-Nerode Theorem

Regular languages have finite index

Proposition 10. *Let L be recognized by DFA M with initial state q_0. If $\hat{\delta}_M(q_0, x) = \hat{\delta}_M(q_0, y)$ then $x \equiv_L y$.*

Proof. This proof is essentially the basis of all our DFA lower bound proofs. We repeat the crux of the argument again here.

Suppose x, y are such that $\hat{\delta}_M(q_0, x) = \hat{\delta}_M(q_0, y)$. It follows that for any $z \in \Sigma^*$, $\hat{\delta}_M(q_0, xz) = \hat{\delta}_M(q_0, yz)$. Hence, xz is accepted by M iff yz if accepted by M. In other words, $xz \in \mathbf{L}(M) = L$ iff $yz \in \mathbf{L}(M) = L$. Thus, $x \equiv_L y$. □

Corollary 11. *Let L be a regular language and let $k = \#(\equiv_L)$. If M is a DFA that recognizes L and suppose M has n states, then $n \geq k$.*

Proof. This our lower bound proof technique. It is the contrapositive of the previous proposition because it says that if $x \not\equiv_L y$ then $\hat{\delta}_M(q_0, x) \neq \hat{\delta}_M(q_0, y)$. □

Corollary 12. *If L is regular then $\equiv_L$ has finite index.*

Proof. If L is regular then there is a DFA M recognizing L. Suppose M has n states. Then by proposition, we have $\#(\equiv_L) \leq n$, and thus, $\equiv_L$ has finitely many equivalence classes. □

Finite Index implies Regularity

Proposition 13. *Let $L \subseteq \Sigma^*$ be such that $\#(\equiv_L)$ is finite. Then L is regular.*

Proof. Our proof will construct a DFA that recognizes L. Since $\equiv_L$ has finite index, let $E_1, E_2, \ldots E_k$ be the set of all the equivalence classes of $\equiv_L$. The states of the DFA M^L recognizing L will be the equivalence classes of $\equiv_L$. The formal construction is as follows. The DFA $M^L = (Q^L, \Sigma, \delta^L, q_0^L, F^L)$ where

- $Q^L = \{E_1, \ldots E_k\}$,

- $q_0^L = [\epsilon]_{\equiv_L}$
- $F^L = \{[x]_{\equiv_L} \mid x \in L\}$; observe that F^L is well-defined because if $x \in L$ and $x \equiv_L y$ then $x\epsilon \in L \;\Rightarrow\; y\epsilon = y \in L$.
- And δ^L is given by
$$\delta^L([x]_{\equiv_L}, a) = [xa]_{\equiv_L}$$
Notice that δ^L is well defined because if $x \equiv_L y$ then $xa \equiv_L ya$.

Correctnes of the above construction requires us to prove that $\mathbf{L}(M^L) = L$, i.e., $\forall w.\ w \in \mathbf{L}(M^L)$ iff $w \in L$. As for all DFA correctness proofs, this one will also be proved by induction on $|w|$ by strengthening this statement. We will show

$$\forall w.\ \hat{\delta}_{M^L}(q_0^L, w) = \{[w]_{\equiv_L}\}$$

First observe that if the stronger statement is established then correctness follows because w is accepted by M^L iff $\hat{\delta}_{M^L}(q_0^L, w)(= \{[w]_{\equiv_L}\}) \cap F^L \neq \emptyset$ iff $[w]_{\equiv_L} \in F^L$ iff $w \in L$ (by definition of F^L).

To complete the proof we will show

$$\forall w.\ \hat{\delta}_{M^L}(q_0^L, w) = \{[w]_{\equiv_L}\}$$

by induction on $|w|$.

- **Base Case** When $|w| = 0$, $w = \epsilon$. We know that $\hat{\delta}_{M^L}(q_0, \epsilon) = \{q_0\} = \{[\epsilon]_{\equiv_L}\}$ since $q_0 = [\epsilon]_{\equiv_L}$
- **Ind. Hyp.** Assume that $\hat{\delta}_{M^L}(q_0, w) = \{[w]_{\equiv_L}\}$ for all w s.t. $|w| < n$.
- **Ind. Step** Consider $w = ua$ such that $a \in \Sigma$ and $u \in \Sigma^{n-1}$.

$$\begin{aligned}\hat{\delta}_{M^L}(q_0, w = ua) &= \{\delta^L(q, a)\} \text{ where } \hat{\delta}_{M^L}(q_0, u) = \{q\} \\ &= \{\delta^L([u]_{\equiv_L}, a)\} \text{ because by ind. hyp. } q = [u]_{\equiv_L} \\ &= \{[ua = w]_{\equiv_L}\}\text{because of the defn. of } \delta^L\end{aligned}$$

□

Corollary 14. *If L is such that $\#(\equiv_L) = k$ then the DFA with the fewest states that recognizes L has k states.*

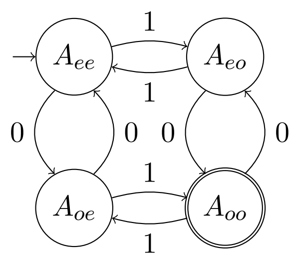
Proof. We previously showed that $\#(\equiv_L)$ is lower bound on the number of statates that any DFA recognizing L must have. The above construction of the DFA in fact shows that there is a DFA recognizing L that has exactly k states. Thus, it must be the DFA with fewest states. □

Example

Example 15. Consider $L = \{w \mid w \text{ has an odd number of 0s and 1s }\}$. We previously observed that the equivalence classes of $\equiv_L$ are

$$\begin{aligned}
A_{ee} &= \{w \mid w \text{ has an even number of 0s and 1s}\} \\
A_{oe} &= \{w \mid w \text{ has an even number of 0s and an odd number of 1s}\} \\
A_{eo} &= \{w \mid w \text{ has an odd number of 0s and an even number of 1s}\} \\
A_{oo} &= \{w \mid w \text{ has an odd number of 0s and 1s}\}
\end{aligned}$$

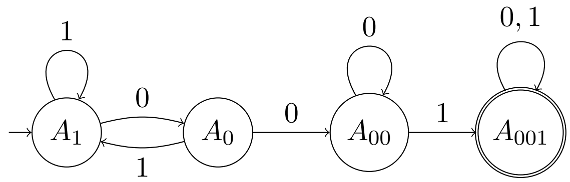
Now for $w \in A_{ee}$, $w0 \in A_{oe}$ and $w1 \in A_{eo}$. Thus in DFA M^L the transtion from A_{ee} on 0 will go to A_{oe} and on 1 will go to A_{eo}. Similarly we can figure out the other transtions. The resulting DFA looks like



Example 16. For the language $P = \{w \mid w \text{ contains 001 as a substring }\}$, we saw that the set of equivalence classes are

$A_{001} = \{w \mid w \text{ has 001 as a substring}\}$
$A_0 = \{w \mid w \text{ does not have 001 as substring and ends in 0 and the second last symbol is not } 0\}$
$A_{00} = \{w \mid w \text{ does not have 001 as substring and ends in 00}\}$
$A_1 = \{w \mid w \text{ does not have 001 as substring and ends in 1 or is } \epsilon\}$

Once again we can figure out transitions easily. For example, for $w \in A_{001}$, $w0$ and $w1$ are A_{001}. The resulting DFA is



Myhill-Nerode Theorem

Theorem 17. *L is regular iff $\equiv_L$ has finitely many equivalence classes.*

Proof. Follows from all the observation made so far. □
