Pumping Lemma

Non-Regular language

ex. $C = \{w \mid w$ has equal number of 1s and 0s$\}$

The Lemma

Let L be a regular language

then there is a positive integer P when

$\forall w \in L$ where $|w| \geq P$, there is a way to write

$w = xyz$ where

1. $|y| > 0$
2. $|xy| \leq P$
3. $\forall i \geq 0$, $xy^iz \in L$

Generalized Pumping Lemma

Same as above, but for strings $awb \in L$, $axy^izb \in L$

(only need to pump middle, not beginning/end)

Minimizing DFA's

Theorem: for every language L there is a unique minimal-state DFA M^* such that $L = L(M^*)$.

There is an efficient algorithm that will produce such an M^*

Extending transition function δ to strings

Given DFA $M = (Q, \Sigma, \delta, q_0, F)$, extend δ to a function

$\Delta : Q \times \Sigma^* : Q$ as follows:

$$\Delta(q, \varepsilon) = q$$
$$\Delta(q, \sigma) = \delta(q, \sigma)$$
$$\Delta(q, \sigma_1 \ldots \sigma_{k+1}) = \delta(\Delta(q, \sigma_1 \ldots \sigma_k), \sigma_{k+1})$$

(note $\Delta(q, w)$: the state of M reached after reading w from q)

$\hat{\delta}(q_0, w) \in F \iff M$ accepts w

Distinguishability

Def. $w \in \Sigma^*$ distinguishes states q_1, q_2 if exactly one of $\hat{\delta}(q_1, w)$, $\hat{\delta}(q_2, w) \in F$

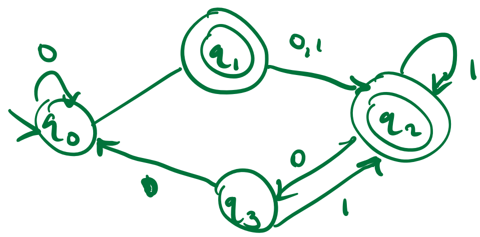
Def: states p, q are distinguishable if there exists such a w $(p \not\sim q)$

Def: states p, q are indistinguishable if there exists no such w $(p \sim q)$

Pairs of indistinguishable states are redundant.

$\sim$ is an equivalence operation; partitions Q into disjoint equivalence classes

ex.



All states are distinguishable.

Minimization Algorithm

Input: DFA M

Output: DFA M_{min} where $L(M) = L(M_{min})$

where M_{min} has no inaccessible states

and all states in M_{min} are distinguishable

M_{min} is the unique minimal DFA equivalent to M

States of M_{min} will be equivalence classes of M

Discovery via Table-Filling Algorithm

(DP to mark indistinguishable states)

Steps:

1. Remove all inaccessible states from M
2. Run Table-Filling Algorithm to get:

   $EQIV_M = \{[q] \mid q$ is an accessible state of $M\}$
3. Define: $M_{min} = (Q_{min}, \Sigma, \delta_{min}, q_{0\,min}, F_{min})$

   where

   $Q_{min} = EQIV_{min}$

   $q_{0\,min} = [q_0]$

   $F_{min} = \{[q] \mid q \in F\}$

   $\delta_{min}([q], \sigma) = [\delta(q, \sigma)]$

Myhill-Nerode Theorem

Definitions

Let $L \subseteq \Sigma^*$, $x, y \in \Sigma^*$

Then $x \equiv_L y$ if $\forall z \in \Sigma^*$,

$[xz \in L \Longleftrightarrow yz \in L]$

$\equiv_L$ is an equivalence relation

The theorem

A language L is regular iff the number of equivalence classes of $\equiv_L$ is regular

Usefulness: if we show that there is a distinguishing string w for L, then all strings in L are distinguishables so the language has infinite equivalence classes, so it is not regular.

Learning DFA's

PAC Learning (Probably Approximately Correct)

learn concepts of classes (e.g. cats and dogs)
and learn to distinguish

→ Instance space X
→ Concept class C (fn's over X)
→ Hypothesis class H (fn's over X)

Proper learning: $H = C$

Algorithm A PAC-learns C if:

$\forall c \in C$, $\forall D \in X$ (distributions),
A gets as input $(x_1, c(x_1)) \ldots (x_m, c(x_m))$
where x_i distributed according to D and outputs
$h \in H$ where

$$Pr_A\left[Pr_{x \in D}[h(x) \neq c(x)] > \delta \right] < \varepsilon$$

PAC-Learning DFA's

Given regular language L, examples $w_1 \ldots w_m$ $(+, -)$
learn a small DFA consistent with these inputs

Define $L^?$ to be "partially defined language" where
$w \in L^?$, $w \notin L^?$, or unknown

$L^?$ distinguishes x, y if $\exists z$ where either $xz \in L^?$, $yz \notin L^?$
or vice versa

If $x \equiv_L y$ then x, y cannot be distinguished by $L^?$

Related to Myhill-Nerode but set of examples has to be
carefully selected: PAC-learning w/ membership queries