# Is Everything Regular?

$\Sigma$ is finite

$\Sigma^*$ is countably infinite

$\Delta = \Sigma \cup \{ \text{"}\varepsilon\text{"}, \text{"}\Phi\text{"}, \text{"}\cup\text{"}, \text{"}\circ\text{"}, \text{"}*\text{"}, \text{"}(\text{"}, \text{"})\text{"} \}$

$\Delta^*$ is countable

every reg. lang. is $L(x)$ for some $x \in \Delta^*$

$\therefore$ set of regular languages is countable

Set of all languages $= 2^{\Sigma^*}$
is uncountable.

$\therefore$ non-regular languages exist.
(in fact, "most" are non-regular.)

$\Sigma = \{a, b\}$

$L_1 = \{ x \mid \#_a(x) = \#_b(x) \}$

$L_2 = \{ x \mid \#_{ab}(x) = \#_{ba}(x) \}$

abbbabaaba

$L_1$ is not regular, $L_2$ is.

to be shown

---

$L_3 = \{ ww \mid w \in \Sigma^* \}$ $\Sigma = \{a,b\}$

$\varepsilon$

a a

b b

ab ab

ba ba

aa aa

bbbb

$\vdots$

find middle;

does left = right?

aba

abba

} not in $L_3$

Intuitively, a DFA accepting $L_3$ must "remember" left half when it crosses middle, and "memory" = "state" but as $|w| \to \infty$, this will overwhelm any finite memory. Made rigorous below ...

$\Sigma = \{a, b\}$

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA

if $x$ and $y$ take $M$ from $q_0$ to $q$, & if $xz \in L$ for some $z$ then so is $yz$ (or neither)

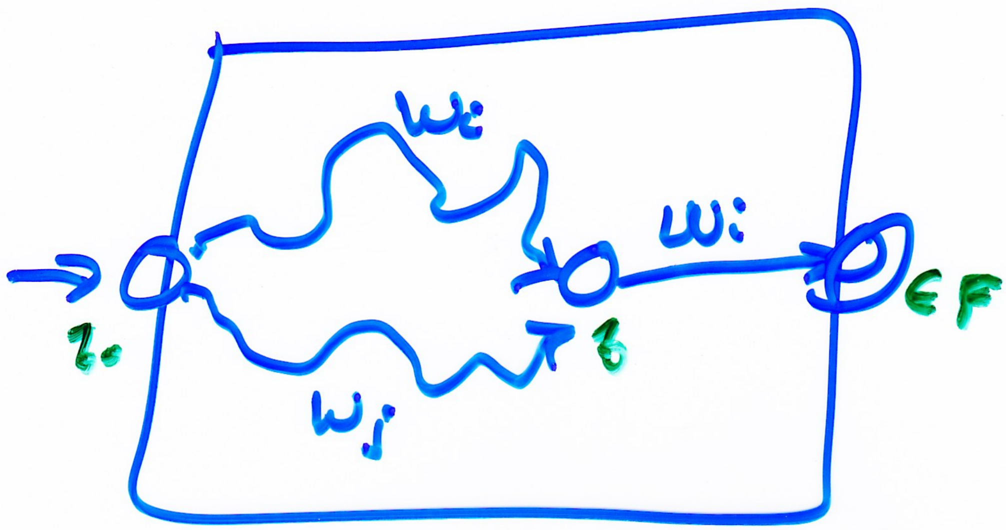
~~Supp~~ Let $p = |Q|$; pick $k$ so that $2^k > p$

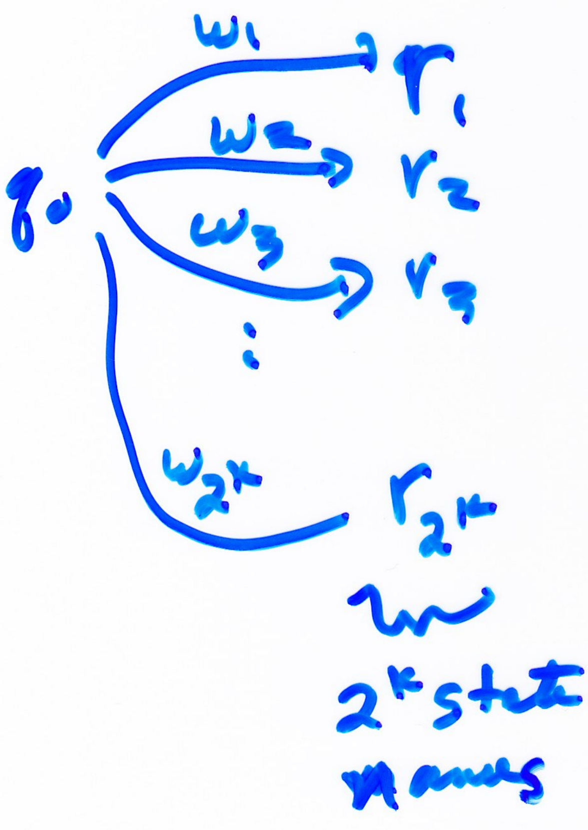
consider $\left.\begin{matrix} w_1 \\ w_2 \\ \vdots \\ w_{2^k} \end{matrix}\right\}$ all $2^k$ different strings of length $k$, where $2^k > p$

$\exists_{q \in Q}$ $\exists$ $i \neq j$ st $w_i$ & $w_j$ both take $M$ to same fixed state $q$.
(by pigeon hole principle)

if $M$ accepts $w_i w_i$ it also accepts $w_j w_i \notin L_3$

$\therefore$ $M$ does not accept $L_3$





Since $2^k > p$, list of state names $r_1 - r_{2^k}$ has duplicates, i.e., $\exists\, i \neq j$ st $\underbrace{r_i = r_j}_{= q \text{ on prev. slide}}$ (but $w_i \neq w_j$)

# An Alternate Proof

$L_3 = \{ ww \mid w \in \{a, b\}^* \}$

Assume for contradiction that $L_3$ is regular. Let $M = (Q \ldots)$ be a DFA ~~accepting~~ $L_3$

Let $p = |Q|$.

consider $x_i = a^i b \quad 0 \le i \le p$

$p+1$ different $x_i$'s.

$\exists q \in Q \ \exists \ i \ne j \quad \begin{array}{l} 0 \le i \le p \\ 0 \le j \le p \end{array}$

st M reaches $q$ on both $x_i$ & $x_j$

WLOG $i < j$

M accepts $x_i x_i$ and $x_j x_j$

it also accepts $x_j x_i = a^j b a^i b$

but $x_j x_i \notin L_3$ since left half $\ne$ right half. [NB: it's true but not sufficient to say "$x_j \ne x_i$" since $x_j$ is not left half. Instead, point is that, since $j > i$, both b's in right half, left half all a's, ~~etc~~. ]

## An Alternate Proof

$L_3 = \{ ww \mid w \in \{a,b\}^* \}$

Assume for contradiction that $L_3$ is regular. Let $M = (Q \ldots)$ be a DFA accepting $L_3$

Let $p = |Q|$.

consider $x_i = a^i b \quad 0 \le i \le p$

$p+1$ different $x_i$'s.

Note importance of $b$; without it, conclusion falls apart

$\exists q \in Q \ni i \ne j \quad \begin{array}{l} 0 \le i \le p \\ 0 \le j \le p \end{array}$

st $M$ reaches $q$ on both $x_i$ & $x_j$

WLOG $i < j$

$M$ accepts $x_i x_i$ and $x_j x_j$

it also accepts $x_j x_i = a^j b a^i b$

but $x_j x_i \notin L_3$ since left half $\ne$ right half. [NB: it's true that not sufficient to say "$x_j \ne x_i$" since $x_j$ is not left half. Instead, point is that, since $j > i$, both $b$'s in right half, left half all $a$'s, etc.]


$a^{i+j}\, b\, a^{i+j}\, b$ ← go around loop once

$a^{i}\, b\, a^{i+j}\, b$ ← zero times

$a^{i+2j}\, b\, a^{i+j}\, b$ ← twice

$a^{i+3j}\, b\, a^{i+j}\, b$ ← 3 times

$\vdots$

$\forall k \geq 0 \quad a^{i+kj}\, b\, a^{i+j}\, b \in L(M)$

# Notes on these proofs

All versions are proof by contradiction: assume some DFA M accepts L3. M of course has some fixed (but unknown number of states, p. All versions also relied on the intuition that to accept L3, you need to "remember" the left half of the string when you reach the middle, "memory" = "states", and since every DFA has only a finite number of states, you can force it to "forget" something, i.e., force it into the *same* state on two *different* strings. Then a "cut and paste" argument shows that you can replace one string with the other in a longer, accepted, string, proving that M accepts something it shouldn't.

Version 1 (slide 15-3): pick a length large enough so that there are more strings of that length than states in M.

Version 2 (slide 15-5): pick increasingly long strings of a simple form until the same thing happens. The argument is a little more subtle here, since the string length, hence the midpoint, changes when you do the cut-and-paste, and so you have to argue that *where ever* the middle falls, left half != right half. Some cleverness in picking "long strings of a simple form" makes this possible; in this case the "b" in "$a^i b$" is a handy marker.

Version 3 (slide 15-7): Generalizing version 2, an accepted string longer than p always forces M around a loop. The substring defining the loop can be removed or repeated indefinitely, generating many simple variants of the initial string. With careful choice of the initial string, you can often prove that not all of these variants should be accepted. Again, some subtlety in these proofs because you need to allow for any start point/ length for the loop.

Not all proofs of non-regularity are about "left half/right half", of course, so the above isn't the whole story, but variations on these themes are widely used. Version 3 is especially versatile, and is the heart of the "pumping lemma".