

1. Inicias con 5 y le agregas 6 (definición recursiva)
2. $\Sigma = \{a, b, c, d\}$ Tiene que empezar con b y hay a's pares (DR)
3. DFA De ahora abc pagado, antes pueden haber c's y d's y después a's y b's
4. Elevar cada elemento al cubo
5. Multiplos de 3
6. Expresión regular: 3 letras mayúsculas \rightarrow % \rightarrow Año (4 dígitos) \rightarrow 5 letras minúsculas

① I. Basis: $a_0 = 5$

II. Recursive step: $a_n = a_{(n-1)} + 6$

III. Closure: a_n

② I. Basis: $b \in \Sigma$

II. Recursive step: if $v \in \Sigma$ then

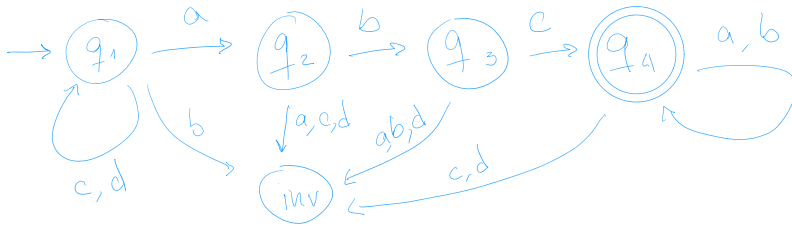
$vaca \in \Sigma$

$vb \in \Sigma$

$vc \in \Sigma$

$vd \in \Sigma$

③



④ #lang racket

(provide (all-defined-out))

(define (** lst)

(let loop

([lst+ lst]

[res '()]))

(if (empty? lst)

(reverse res)

(loop

loop

(cdr lst)

(cons (* (car lst) (car lst)) res)))

⑤ defmodule Listas do
 def multp3(lst), do: do_mult(lst, [])
 defp do_mult([], res), do: Enum.reverse(res)
 defp do_mult([head | tail], res) when rem(head, 3) == 0, do: do_mult(tail, res)
 defp do_mult([_ | tail], res), do: do_mult(tail, res)

⑥ $(A|B|C\dots|X|Y|Z)^3 \% (1|2|3\dots|8|9|0)^4 (a|b|c\dots|x|y|z)^5$

