

Regular Expressions

$r ::=$	Constant string	"hi"
	$r_1 \cdot r_2$	"hi" · "hello" "hi" · ("hello" "bye")
	$r_1 r_2$	"hi" "hello"
	r^* (Kleene-*)	("hi") [*]
	empty	

$['0' - '9']^5$: American zip codes

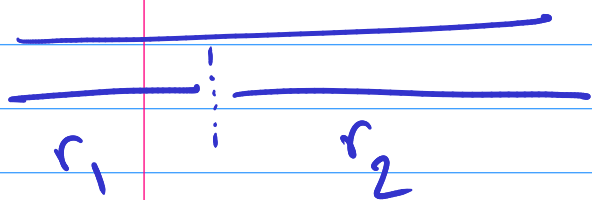
$['a' - 'z' A-Z]^+ \cdot @ \cdot [a-z A-Z]^+ \cdot (com | edu | org)$
: Email address

$['0' - '9']^+ = [0-9] \cdot [0-9]^*$
 $("-" | "") \cdot ['0' - '9']^+$

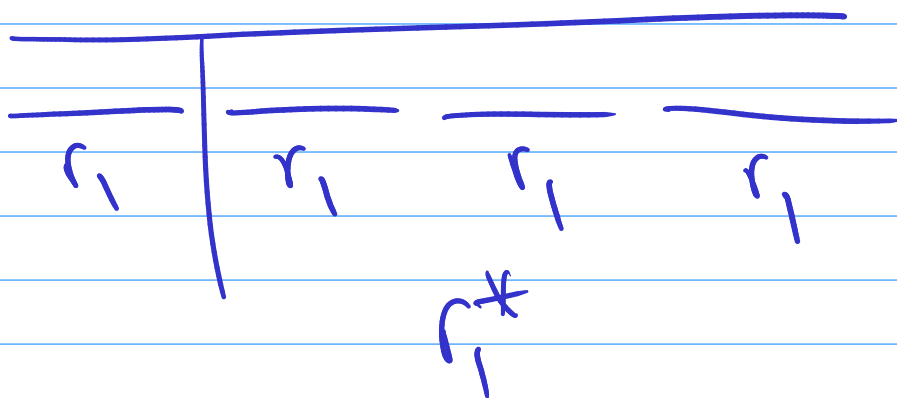
$(\text{"-"} | \text{" "}) \cdot [0-9]^+ \cdot \text{"."} [0-9]^*$

grep /sed

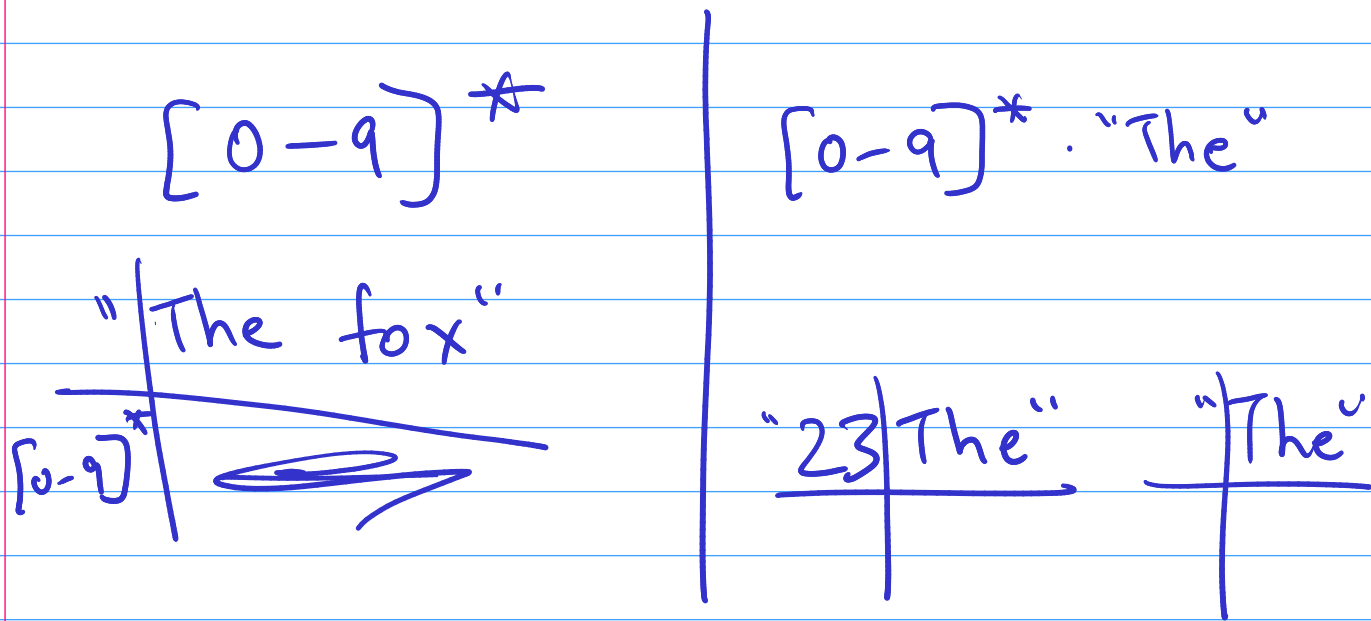
s matches $r_1 \cdot r_2$ if



"hi hello / ~~bye bye~~" \sim $(\text{hi/hello})^* \cdot \text{bye}^*$



$w \sim r_1^* = \begin{cases} w = \text{" "}, \text{ or} \\ w = w_1 w_2, w_1 \neq \text{" "}, \\ w_1 \sim r_1, w_2 \sim r_1^* \end{cases}$



```

[0-9]* — ans = true
for c in str:
    if c ∉ [0-9]:
        ans = false
  
```

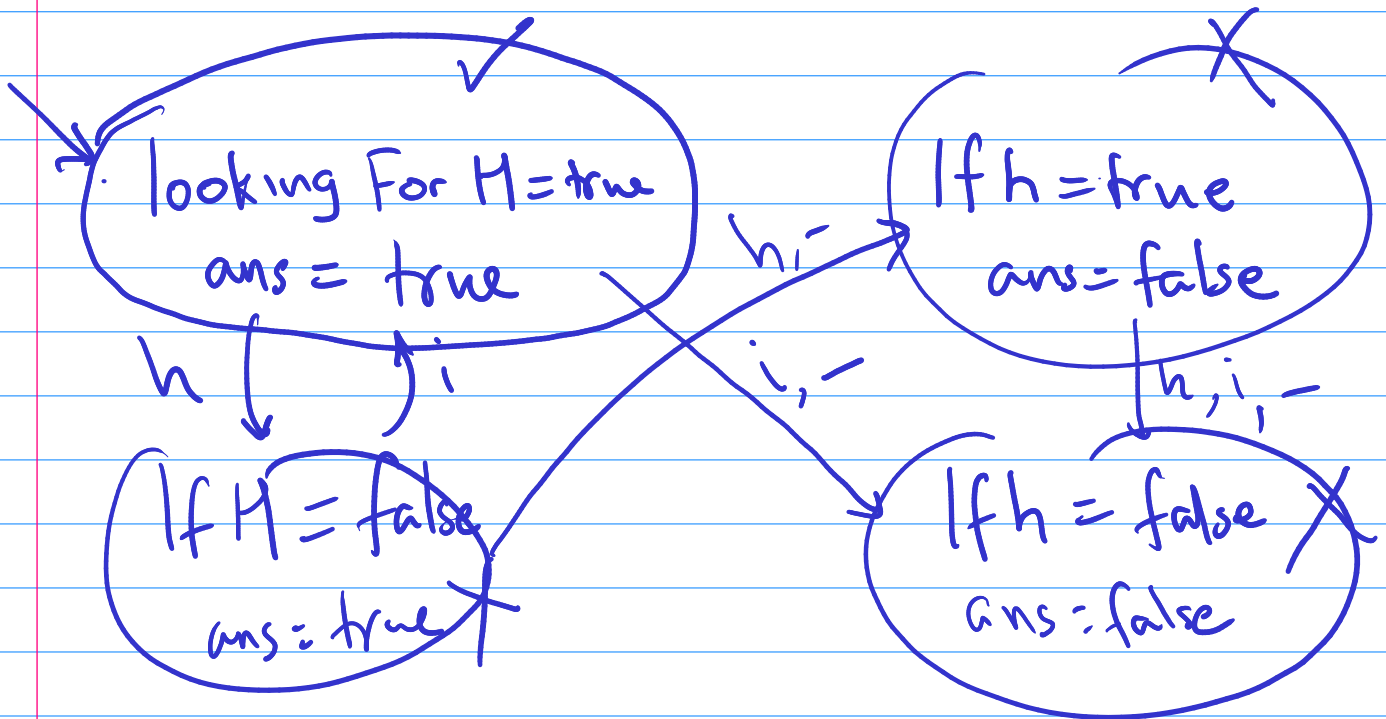
```

"hi"* — Step over in increments of 2
Look for hi
ans = true, looking for H = true
for c in str:
    if looking for H & c ≠ 'h':
        ans = false
    or not looking for H & c ≠ 'i':
        ans = false
  
```

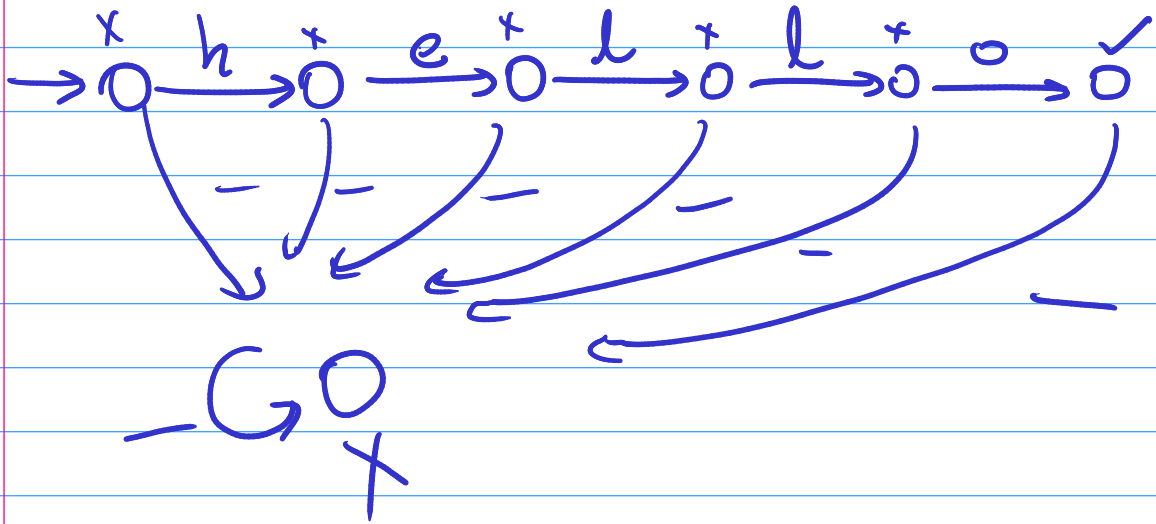
looking For H = not looking For H.

Check if string str matches pattern r

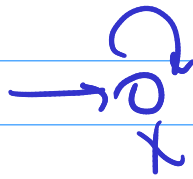
- Single left-to-right pass
- Each character processed in $O(1)$ time



"hello"

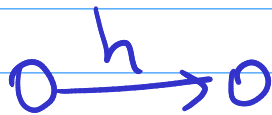
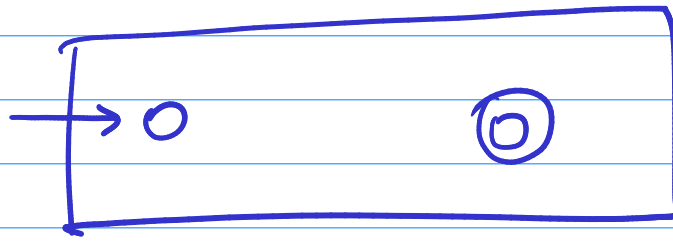
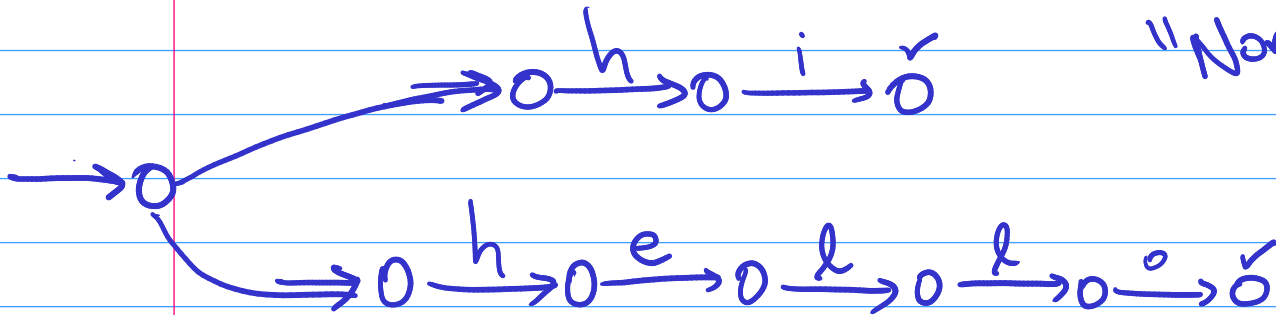


Bottom

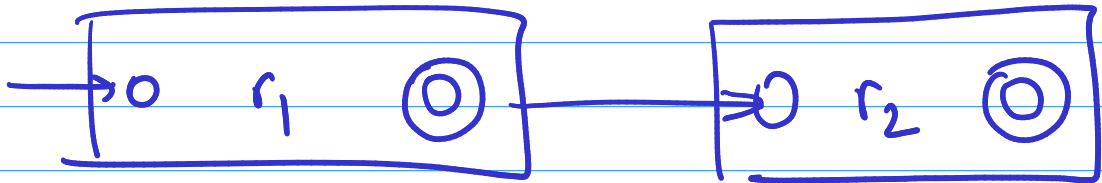


"hi" | "hello"

"Nondeterminism"



$r_1 \cdot r_2$



$h_1^* \cdot \text{bye}$

hi hi | hi | hi bye

