REfO: Regular Expressions for Objects

¿Y para que sirve?

Para usar expresiones regulares sobre secuencias de **lo que sea**. Pueden ser caracteres, números o cualquier objeto python! Además, secuencia no necesita ser una lista (o una string), puede ser solo un iterador.

¿Un ejemplo? Dada una lista de números enteros buscar un número par seguido de un número primo seguido de un número divisible por 3.
¿Otro? Dado un secuencia de paquetes TCP encontrar los paquetes que marcan el principio y el fin de un stream.

Posta, ¿Para que lo usan?

Atrapar patrones complejos en lenguage natural sin escribir código ad-hoc. Por ejemplo **matchear "What is a set comprehension?"** con una expresion regular que dice:

> Empieza con "What",
> Sigue el verbo "be",
> Opcionalmente sigue un determinante (a, an, the)
> Sigue uno o más sustantivos

Sin autómatas!

¿Porque? Porque introducir cosas como capture groups, flags para greegy/non-greedy y otros hace al código poco elegante y muy probablemente de complejidad exponencial. REfO esta implementado usando una **maquina virtual**. Aca hay un ejemplo del bytecode:

> 1- Consumir "a"
> 2- Consumir "a"
> 3- Forkear al paso 1 y al paso 4
> 4- Forkear al paso 5 y al paso 6
> 5- Consumir "b"
> 6- Producir un match

Python **re** puede tomar tiempo **exponencial**

Si si, probá:

```
>>> import re
>>> re.match("(:?a?){100}a{100}", "a" * 100)
```

Python usa un algoritmo basado en automatas y hace una forma de backtracking que tiene peor caso exponencial.
REfO usa un algoritmo **polinomial** ;)

Rafael Carrascosa @ machinalis

https://github.com/machinalis/refo