Prácticas de Autómatas y Lenguajes. Curso 2024/25
Práctica 0: Expresiones regulares

Duración: 1 semana.
Entrega: Semana del 30 de septiembre, cada grupo antes de su clase.
Peso: 10% de la nota de prácticas.

Descripción del enunciado:

En esta práctica diseñaremos expresiones regulares y usaremos el módulo re (*regular expression operations*) de Python para comprobarlas. Una expresión regular es una cadena de caracteres que se utiliza para describir o encontrar patrones dentro de otras cadenas, en base al uso de delimitadores y ciertas reglas de sintaxis. Conviene que estudies el contenido de la presentación Expresiones Regulares disponible en moodle antes de afrontar esta práctica.

Es importante que tengas en cuenta que las expresiones regulares usadas habitualmente incluyen una sintaxis más rica que la utilizada en clase de teoría. Es recomendable por tanto que estudies la sintaxis de las expresiones regulares utilizadas en el módulo re de Python y atiendas a las explicaciones de tu profesor de prácticas. En el siguiente enlace tienes un tutorial alternativo: https://docs.python.org/3/howto/regex.html

Como es probable que no hayas utilizado el lenguaje de programación Python con anterioridad, a continuación te facilitamos algunos enlaces a tutoriales que te pueden resultar útiles. No obstante para la realización de esta primera práctica no es necesario programar, sino simplemente diseñar las expresiones regulares que te pedimos.

Tutoriales de python:

Documentación oficial:

- https://docs.python.org/3/tutorial/ (en inglés)
- https://docs.python.org/es/3/tutorial/ (en español)

Lo más básico de Python para programadores de C:

- https://engineering.purdue.edu/~milind/datascience/2018spring/notes/lecture-2.pdf

Tutoriales de Python más detallados:

- https://anandology.com/python-practice-book/getting-started.html
- https://inst.eecs.berkeley.edu/~cs188/fa20/project0/

Guías de estilo:

- https://gist.github.com/kamikaze-lab/df2b4df198605abad846 (en español)
- https://www.python.org/dev/peps/pep-0008/ (en inglés)

Uso de conda:

- https://docs.conda.io/projects/conda/en/4.6.0/_downloads/52a95608c49671267e40c689e0bc00ca/conda-cheatsheet.pdf
- https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html

Instrucciones:

Edita el fichero *regular_expressions.py* y completa las expresiones regulares para las variables `RE1` a `RE6` de acuerdo con las descripciones de los ejercicios siguientes. La expresión `RE0` se facilita a modo de ejemplo. Puedes probar tus expresiones regulares ejecutando el código *test_p0.py*, que contiene varios ejemplos.

Ejercicio 0:

Diseña una expresión regular sobre el alfabeto {a,b} que represente las cadenas no vacías que terminan en *a*. Asigna la expresión regular a la variable `RE0`.

Ejercicio 1 (2 puntos):

Diseña una expresión regular sobre el alfabeto {a,b,c} que represente las cadenas no vacías que contienen al menos una *a* y al menos una *b*. Asigna la expresión regular a la variable `RE1`.

Ejercicio 2 (2 puntos):

Diseña una expresión regular que represente un número arbitrario, con o sin signo y con o sin parte decimal. Asigna la expresión regular a la variable `RE2`. Ten en cuenta que:

- Los números negativos van precedidos del signo menos.
- Los números positivos (y el cero) no llevan signo.
- Los números pueden tener o no parte decimal. En caso de tenerla, el separador es un punto.
- Los números con separador decimal y sin parte decimal son posibles, por ejemplo `23.` ó `-142.`.
- Los números que solo tienen parte decimal también son posibles, por ejemplo `.41` ó `-.45`.
- La parte entera de un número no puede tener ceros a la izquierda.

Ejercicio 3 (2 puntos):

Diseña una expresión regular que represente direcciones web de una de las dos formas siguientes: *www.uam.es/extensión* o *moodle.uam.es/extensión*. En ambos casos la *extensión* es una cadena formada por letras minúsculas y el símbolo /, que puede estar vacía. En ningún caso podrán aparecer dos símbolos / seguidos. Asigna la expresión regular a la variable `RE3`.

Ejercicio 4 (2 puntos):

Diseña una expresión regular que represente expresiones aritméticas con números enteros positivos (se excluye el 0) y los símbolos +, -, * y /. Asigna la expresión regular a la variable `RE4`.

Ejercicio 5 (2 puntos):

Diseña una expresión regular que represente expresiones aritméticas con números enteros positivos (se excluye el 0) y los símbolos +, -, * y /, en las que se permite el uso de paréntesis sin anidamiento. Asigna la expresión regular a la variable `RE5`.

Normas de entrega:

La entrega la realizará sólo uno de los miembros de la pareja a través de la tarea disponible en Moodle.

Sólo se debe entregar el fichero *regular_expressions.py* con las expresiones regulares pedidas en los ejercicios 1-5.