Facultad de Ciencias Exactas y Naturales

Escuela de Informática y Computación

Ingeniería en Sistemas

Paradigmas de Programación

Proyecto #1

Algoritmo de Markov

Fecha de entrega: domingo 21 de octubre, 2018

Profesor: Mag. Georges Alfaro S.

Estudiantes:

Estefanía Murillo Romero 117000387

Andrés Romero Hernández 402300958

Table of Contents

Planteamiento del problema

El problema consiste en construir un programa que pueda aplicar un conjunto de reglas de producción o sustitución a una hilera de entrada para producir un resultado determinado.

El conjunto de reglas a aplicar forma un sistema de reescritura o sustitución llamado algoritmo de Markov, concebido por el matemático ruso Andrey Andreevich Markov (1903-1979), quien hiciera contribuciones importantes a la teoría de algoritmos y funciones recursivas. Un algoritmo de Markov es un sistema completo según el modelo de Turing, lo cual quiere decir que permite, en teoría, evaluar cualquier función calculable. De esta manera, un algoritmo de Markov puede servir como modelo computacional. Un lenguaje de programación basado en este modelo debería definir la comparación y empate de patrones (pattern matching) como una operación fundamental.

Biblioteca PyQt5

Instalación

Pasos para ejecutar en el shell de Ubuntu:

1. Para Python 3: sudo apt-get install python3-pyqt5. Para Python 2: sudo apt install python-pyqt5
2. sudo apt-get install pyqt5-dev-tools
3. sudo apt-get install qttools5-dev-tools

Para convertir los archivos .ui a .py: pyuic5 USERINTERFACENAME.ui -o OUTPUTFILENAME.py -x

Pasos para ejecutar en el cmd Windows:

1. pip install PyQt5
2. pip install PyQt5-tools

Para convertir los archivos .ui a .py: pyuic5 -x USERINTERFACENAME.ui -o OUTPUTFILENAME.py

Análisis de Resultados

Archivos y clases

- Clase rule.py: Es la clase donde se almacenan los componentes de cada regla en particular. Contiene los siguientes atributos: id, patrón, sustitución, etiqueta y un booleano llamado esFinal.
- Archivo __init__.py: Es un archivo para importar la clase regla (rule.py).
- Archivo MainWindow.py: Es el archivo que implementa la interfaz gráfica principal.
- Archivo InputsGui.py: Es el archivo que implementa la interfaz gráfica de la ventana para importar archivos y ejecutar múltiples entradas de prueba.
- Markov.py: Es el archivo que implementa los métodos necesarios para la ejecución del programa.

Métodos Markov.py

VerifyParse

Este método devuelve true si el algoritmo está parseado.

ParseAlgorithm

Este método limpia los arrays de variables, símbolos, marcadores y reglas en caso de que estén previamente cargados con la información de otro algoritmo y carga estos mismos arrays con la información del nuevo algoritmo dado.

Run

Método que ejecuta el algoritmo.

RunMultipleInputs

Método que extrae cada entrada de la ejecución múltiple y corre ejecuta el algoritmo para cada una de ellas.

GetUserInput

Extrae lo que el usuario digita en el espacio de entrada.

GetMultipleInputs

Devuelve un array con todas las múltiples entradas.

RemoveSymbols

Elimina los símbolos que son iguales a las variables.

LoadAlgorithm

Método que extrae la información del algoritmo y la analiza.

CreateRule

Método que toma crea cada objeto regla y los almacena en el array de reglas.

Execute_algorithm

Método que ejecuta el algoritmo dependiendo de la entrada que dio el usuario.

ProcessRuleCase1

Procesa las reglas que en su patrón contiene variables.

ProcessRuleCase2

Procesa las reglas que en su patrón no contiene variables.

ApplySubstitution

Sustituye el patrón de la regla por su sustitución.

GetRuleRegex

Devuelve la expresión regular del patrón de la regla dada.

ContainsVariable

Método que devuelve un array de las variables que se encuentran en el patrón de la regla dada.

Remove_null

Método que elimina el símbolo de lambda mayúscula en un string.

NullRule

Método que recibe un patrón y un string de entrada. Si el patrón es un símbolo de lambda mayúscula (null) lo coloca al inicio del string de entrada.

VariablesFromString

Método que devuelve un array de las variables que hay en el string que se le pasa por parámetro.

VariablesFromPattern

Método que devuelve un array de variables en el mismo orden de como las va encontrando en el patrón.

Cómo utilizar el programa

En Windows puede ejecutarlo mediante el Markov.exe o por medio del cmd escribiendo python Markov.py
En Ubuntu se puede ejecutar mediante el Shell escribiendo python3 Markov.py

Con una hilera de entrada:

1. Cargue el algoritmo de preferencia o digítelo en el espacio correspondiente.
2. Click al botón de Parse.
3. Ingresar una hilera de prueba.
4. Click en el botón Run.
5. Observe los resultados en el espacio correspondiente.

Con múltiples hileras de entrada:

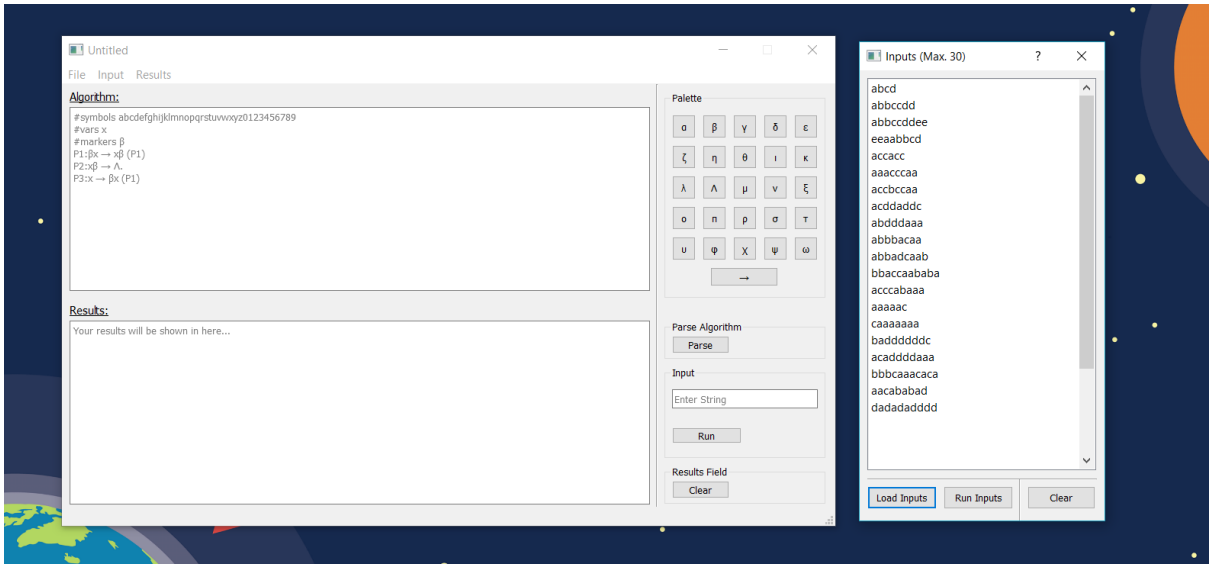
1. Cargue el algoritmo de preferencia o digítelo en el espacio correspondiente.
2. Click al botón de Parse.
3. En la barra de herramientas, click al espacio de Input.
4. Click en Multiple Inputs.
5. Cargue el archivo con múltiples hileras o digítelas en cada línea.
6. Click en el botón Run Inputs.
7. Observe los resultados en el espacio correspondiente.

Guardar los resultados:

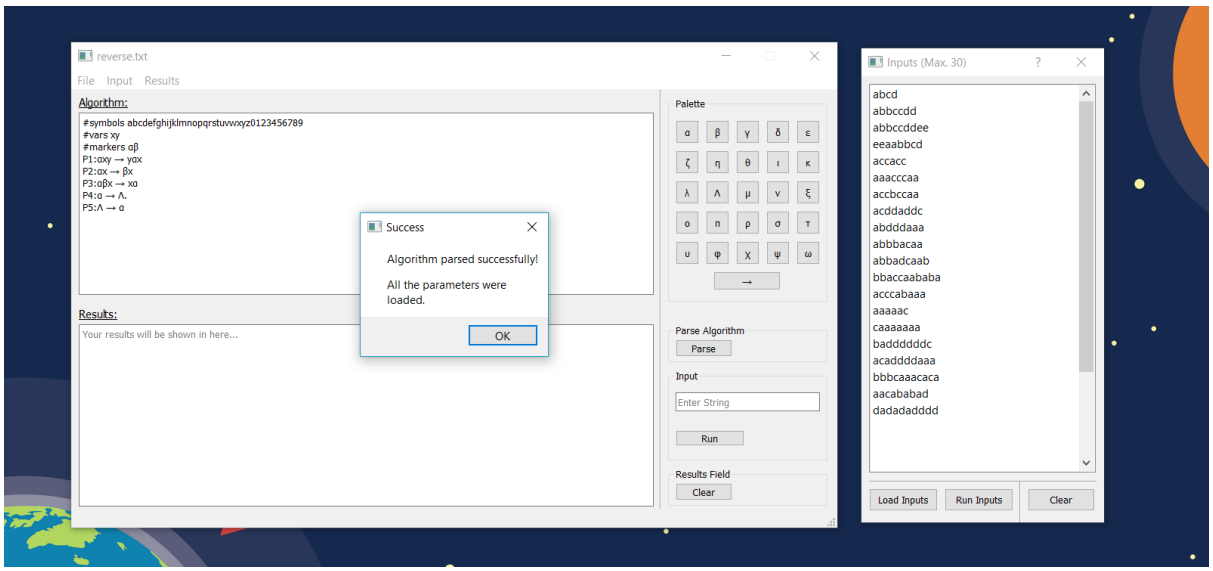
1. Ejecute el algoritmo de alguna manera previamente explicada.
2. En la barra de herramientas, click al espacio Results.
3. Click Save Results Log
4. Guarde el archivo con el nombre y en el lugar de preferencia.

Anexos

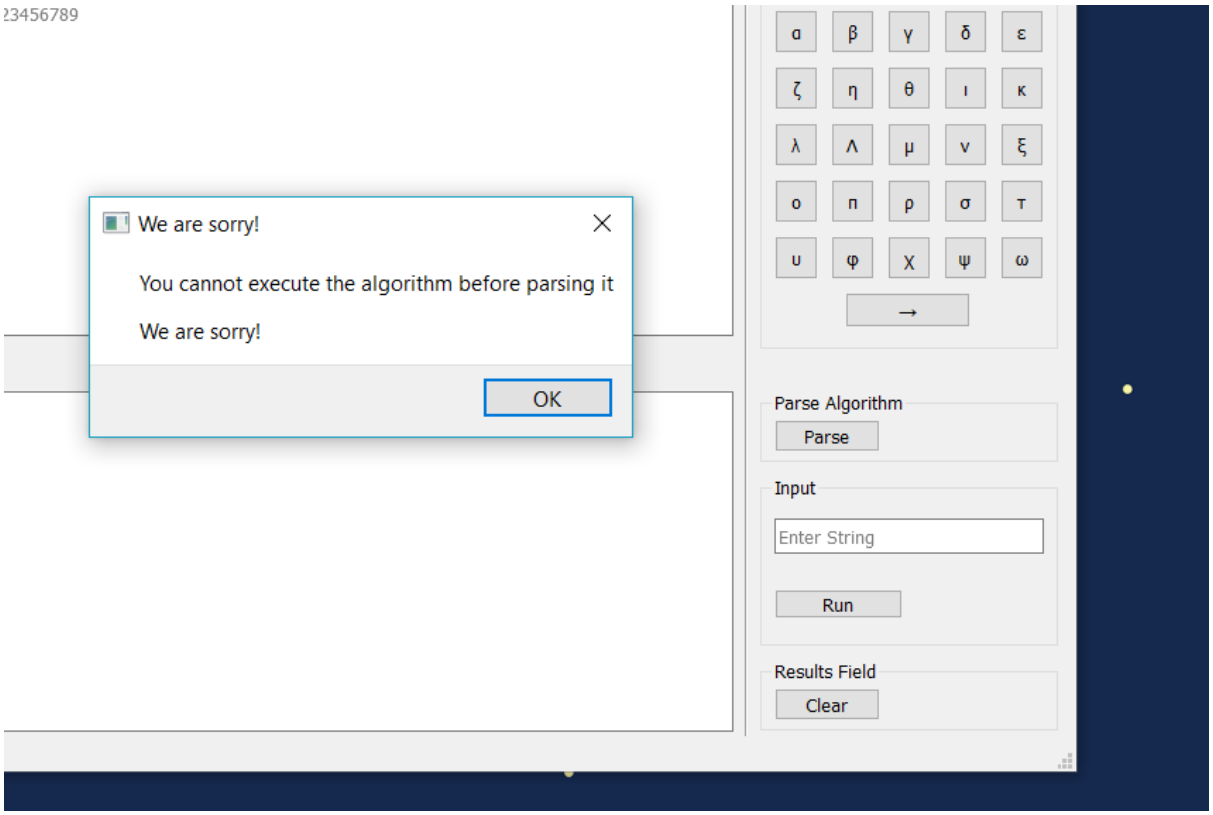
Interfaz Principal con ventana de Múltiples entradas abierta



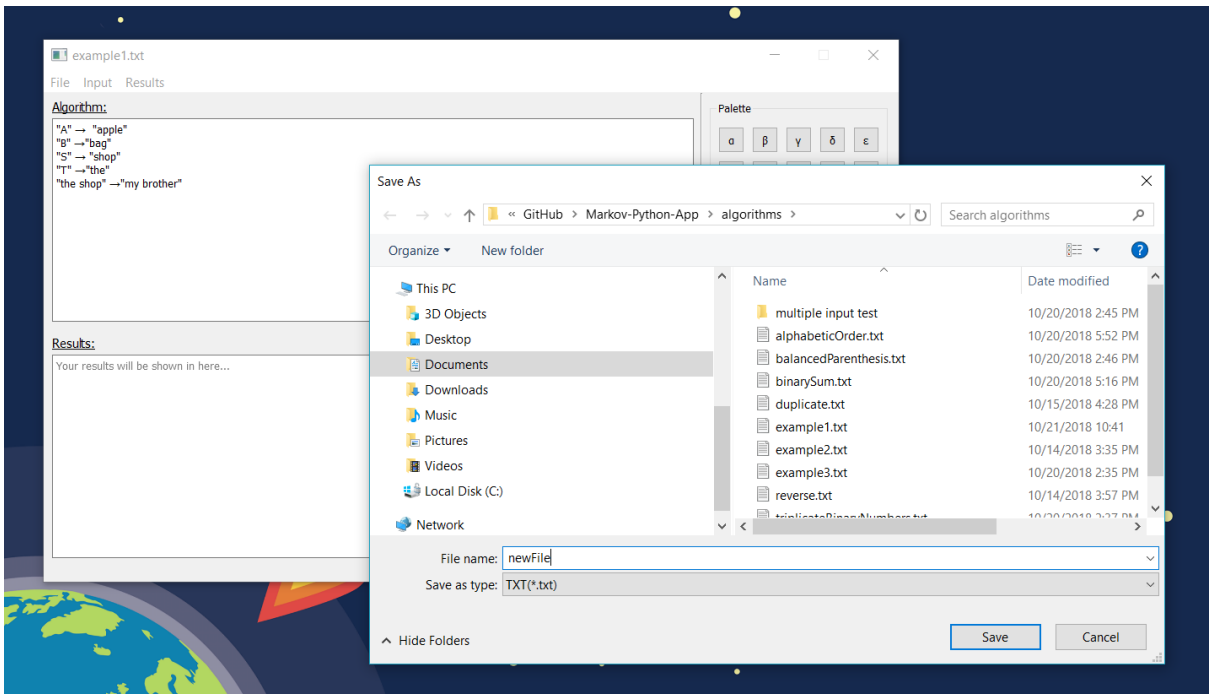
Mensaje de que el algoritmo fue parseado correctamente



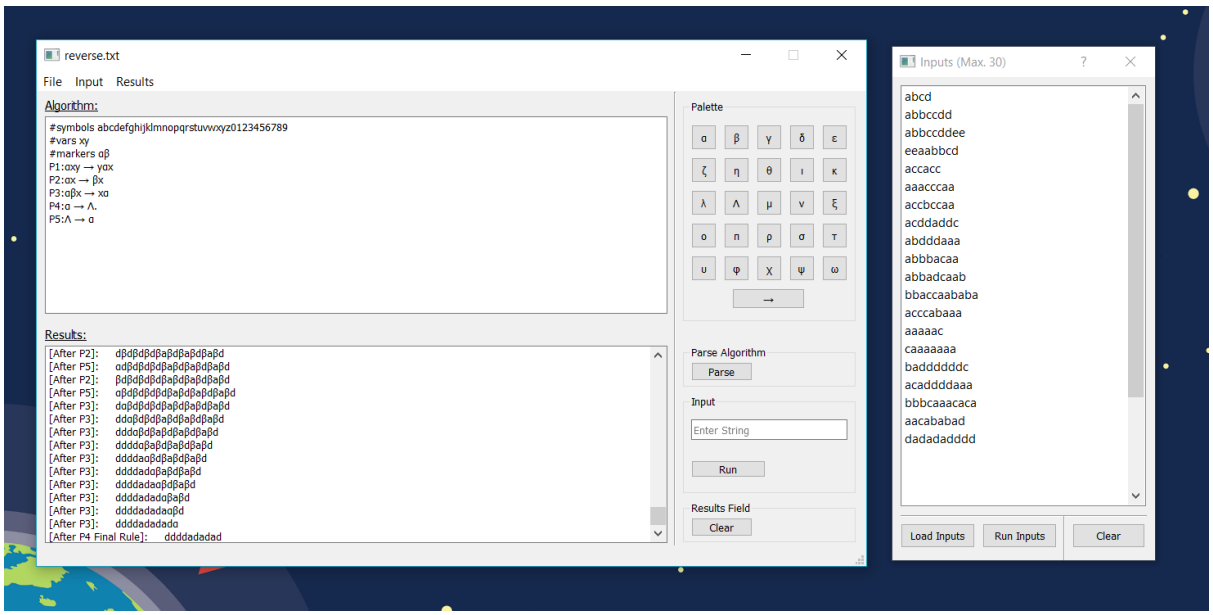
Mensaje de error, por no haber parseado el algoritmo antes de ejecutar.



Opción de guardar como



Resultado al ejecutar múltiples entradas



Resultado al ejecutar el algoritmo de duplicar (concatenar un string con él mismo)