

Práctica 1: Preliminares matemáticos

Grado en Inteligencia Artificial

Autómatas y Lenguajes Formales\

2024/2025

Miembros del grupo

- Guillermo Blanco Núñez
- Fiz Garrido Escudero

Instrucciones de Ejecución

Para ejecutar los ejercicios, es necesario acceder al **Command Prompt** y programar la ruta con el comando `cd` a la ruta local donde se encuentre la carpeta del repositorio.

Una vez hecho esto, para ejecutar cada ejercicio se debe escribir `python` o `python3` dependiendo de la versión instalada en el equipo, seguido del nombre del archivo a ejecutar.

Ejecución del Ejercicio 1

1. Seleccione el tipo de lista que desea mostrar y la cantidad de teoremas a visualizar.
2. Ejemplos de ejecución:

```
python q.py -a 12
```

- o Este comando muestra por pantalla los 12 primeros elementos de **Q**, el conjunto de los racionales.

```
python q.py -u 12
```

- o Este comando muestra por pantalla los 12 primeros elementos de **Q** que realmente sean irreducibles.

Ejecución del Ejercicio 2

3. Introduzca un número `(n)` para ver los `n` primeros teoremas del sistema formal **mg~**, o introduzca una cadena de texto para comprobar si forma parte del sistema formal.
4. La cadena de texto introducida solo puede contener los caracteres `m`, `g` y `~`.
5. Ejemplos de ejecución:

```
python mg.py 12
```

- o Este comando muestra por pantalla los 12 primeros teoremas del sistema formal **mg~**, ordenados por anchura.

```
python mg.py m~g~
```

- o Este comando verifica si la cadena `m~g~` forma parte del sistema formal **mg~**. En este caso, mostrará por pantalla **yes**.

Ejecución del Ejercicio 3

6. Introduzca un número **(n)** para mostrar los **n** primeros teoremas del sistema formal **MIU**.
7. Se puede añadir **-NoRepeat** antes del número para evitar teoremas repetidos.
8. Ejemplos de ejecución:

```
python miu.py 12
```

- Muestra los 12 primeros teoremas del sistema formal **MIU**, ordenados por anchura.

```
python miu.py -NoRepeat 12
```

- Muestra los 12 primeros teoremas del sistema formal **MIU**, asegurándose de que ninguno esté repetido.

Explicaciones Adicionales

Ejercicio 1

Al ejecutar cualquiera de los dos modelos, se generan automáticamente tanto **-a** como **-u** para un valor de **n = 10,000,000** y se comparan los tiempos de ejecución en **nanosegundos**.

- Se concluye que el algoritmo **-a** es notablemente más eficiente que **-u**, aunque la diferencia varía ligeramente entre ejecuciones.
- Esto se debe a que **-u** realiza una comparación adicional para verificar si el numerador y el denominador son primos relativos antes de agregarlos.

Ejercicio 2

El programa muestra los teoremas del sistema formal **mg~** ordenados por anchura.

- El algoritmo de decisión comprueba primero si la cadena introducida es un axioma.
- Si no lo es, invierte el paso de producción del sistema formal y hace una llamada recursiva con la nueva cadena.
- Continúa este proceso hasta que la longitud de la cadena sea **4** y, si es par, verifica si es un axioma.
- Si no se llega a un axioma, se determina que la cadena no es parte del sistema formal **mg~**.

Ejercicio 3

Acertijo MU

- El ciclo de las **potencias de 2 módulo 3** alterna entre los valores **2** y **1**, similar al comportamiento del sistema **MIU**.
- En **MIU**, la **Regla 2** duplica la cadena, aumentando el número de **I** de la forma **2^n**.
- Esto genera un ciclo en el que el número de **I** sigue creciendo, pero nunca se alcanza un valor en el que **Regla 3** (que reemplaza **III** por **U**) elimine todas las **I**.

- Ningún valor de 2^n es divisible por 3, por lo que **MU no es posible**.

Procedimiento de Decisión

- **El procedimiento de decisión del sistema formal MIU no existe.**
 - Es imposible de implementar, ya que las reglas de producción de nuevos teoremas no tienen un orden definido.
 - Esto genera un **problema de decidibilidad**, lo que significa que **no existe un algoritmo** que, en un número finito de pasos, siempre responda correctamente con "sí" o "no".
-