MIARFID Máster en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

José Javier Calvo Moratilla
2021/2022

Computación Natural
Trabajo Stickers

1. Introducción

La presente memoria describe el trabajo realizado en el diseño de un módulo de análisis de moléculas completas para sistemas de stickers regulares, dónde dado un sistema y una molécula completa comprobar si el sistema lo genera o no.

La implementación del código se ha realizado con el lenguaje de programación Python en el entorno de Google Colab.

Se ha utilizado una técnica basada en backtracking recursivo para poder comprobar todas las combinaciones posibles eficientemente.

En primer lugar se introduce brevemente los aspectos más importantes de la operación de stickers para la implementación del trabajo.

En segundo lugar se realiza una explicación de la estructura de datos elegida para la creación de los stickers.

En tercer lugar se describen las funciones implementadas para la resolución del problema.

En cuarto lugar se realizan las pruebas pertinentes para comprobar el correcto funcionamiento del programa.

En último lugar se concluye la memoria, describiendo los puntos más importantes del trabajo realizado..

2. Stickers

La operación de Stickers consiste en aplicar operaciones de dominó a una cadena o molécula ‘x’ y ‘y’, partiendo de un axioma definido, siendo la “x” la hebra superior y la ‘y’ la inferior, para así determinar que una molécula forma parte o no del lenguaje de stickers.

Las operaciones efectuadas por los dominós se aplican a las hebras superior o inferior por la izquierda o por la derecha.

Si la solución va avanzando hasta un estado final se van aplicando operaciones recursivas. Si se encuentra una solución no viable dicho camino retorna un resultado nulo, para así no seguir explorando dicho camino.

En cambio si se computa la molécula y se logra llegar al axioma objetivo se demuestra que la molécula forma parte del lenguaje de Stickers.

3. Estructura de datos

Para la definición de las hebras superior e inferior de la molécula se ha utilizado para cada una de ellas una variable de tipo String, para poder realizar operaciones de manera sencilla tanto por la parte izquierda y derecha de la cadena.

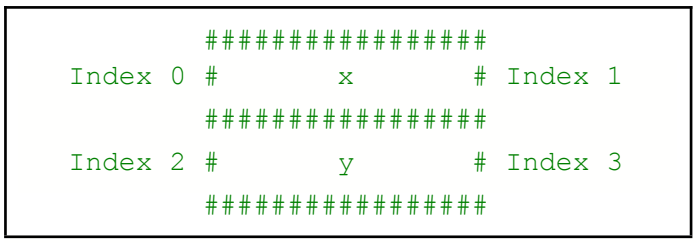
```
Molécula:  aabb
```

El axioma se define también en una variable de tipo String y se utiliza para comprobar que se ha llegado al axioma objetivo, aceptando así la molécula como parte del lenguaje de stickers.

```
Axioma:  a
```

Para definir los dominos se utiliza una lista de listas de 4 elementos, dónde el órden de los índices corresponden a las operaciones realizadas de hebra superior a inferior, de izquierda a derecha.

```
[['a','b','', ''],['b','a','', '']
```



Si en el dominó aparece un símbolo del vocabulario ‘a’ o ‘b’ se efectúa la operación en su lugar respectivo, si aparece la cadena vacía ‘ ‘ no se efectúa operación.

Para la solución dentro del backtracking se ha utilizado una cadena tipo String en la cuál nodo a nodo se ha ido guardando el camino realizado, incluyendo la cadena computada en dicho punto.

En cada nodo la solución incluida como parámetro tiene localizado en el último carácter de la cadena el número del dominó a aplicar, siendo el número uno para el primer dominó.

4. Funciones

4.1. op_domino:

La función ‘op_domino’ se encarga de aplicar a una molécula la operación de dominó, obteniendo como entrada la molécula ‘x’, ‘y’, el dominó y el axioma.

Si el dominó correspondiente se puede aplicar la función da como resultado la molécula resultante.

En el caso de que no se pueda aplicar la operación la función de un estado booleano ‘False’ como respuesta.

Se utilizan los índices de la cadena String para poder realizar las operaciones, obteniendo las subcadenas correspondientes.

En la función se deben de controlar dos situaciones más, la primera que una de las dos hebras puede llegar al axioma antes que la otra, por ello si una de las hebras corresponde con el axioma se cambia por la cadena vacía para poder diferenciarlo del símbolo ‘a’ o ‘b’ que aparece en el vocabulario, para no discriminarlo de una operación de dominó.

La segunda situación corresponde a identificar cuando una hebra ya ha sido computada del todo, después de la comprobación del axioma, en la cuál se

da como resultado la cadena espacio ‘ ‘ para diferenciar un estado final utilizado como comprobación por la función computación.

4.2. paso_valido:

La función “paso_valido” ejecuta la operación de dominó, si no se puede ejecutar se retorna un valor “False”, en el caso de ser completamente factible se retorna la molécula computada. La función es útil en el proceso de backtracking ya que permite podar las soluciones no factibles, reduciendo el coste computacional.

4.3. es_solucion:

La función “es_solución” se encarga de guardar una solución cuando ya es final, convirtiendo la solución en una hoja del árbol para así no seguir con una exploración más profunda.

4.4. solve_dominos:

La función “solve_dominos” se encarga de ejecutarse recursivamente utilizando las funciones “paso_válido” y “es_solucion”. Cada vez que se ejecuta se explora una nueva solución descendiente de un nodo concreto.

4.2. computación:

La función ‘computación’ realiza una búsqueda basada en la técnica backtracking dónde en cada rama se abren n caminos posibles (dominos totales),explorando las soluciones factibles y podando implícitamente las soluciones no factibles.

Si la solución recibida como parámetro de entrada está formada por la molécula ‘x’ y ‘y’ y tienen como contenido el string “espacio”, la solución es final y se almacena.

Se ejecuta la función computación, si no se puede aplicar la operación del dominó indicada por el estado correspondiente no se sigue explorando dicho camino, si de lo contrario se puede aplicar una operación se exploran los siguientes hijos para un nodo concreto y se instancian ejecutando recursivamente la función solve_dominos.

Al final del cómputo, si la lista resultante tiene soluciones, la molécula se indica que forma parte del lenguaje de stickers, por lo contrario si no hay ninguna solución se considera que la molécula no es generada por dicho sistema de stickers.

5. Pruebas

Se han definido cuatro pruebas para probar el funcionamiento del programa implementado, obteniendo los siguientes resultados:

Ejemplo 1 Xavier, aceptada

Molécula	`'aabb'`
Axioma	`'a'`
Dominós	`[['a','b','', ''],['b','a','', ''],['','','a', 'b'],['','','b', 'a']]`
Resultado	`La molécula: aabb Sí es generada por el lenguaje de stickers`
Soluciones	`{'3131', '1313', '3311', '1133', '1331', '3113'}`

Ejemplo 2 Xavier, rechazada

Molécula	`'aaabbb'`
Axioma	`'a'`
Dominós	`[['a','a','', ''],['b','b','', ''],['','','a', 'a'],['','','b', 'b']]`
Resultado	`La molécula: aaabbb NO es generada por el lenguaje de stickers`

Ejemplo 4, página 15 Stickers, aceptada

Molécula	`'cccbbbabbb'`
Axioma	`'a'`
Dominós	`[['b','b','', ''],['c','','', ''],['','','b', ''],['','','c', 'b']]`
Resultado	`La molécula: cccbbbabbb Sí es generada por el lenguaje de stickers`
Soluciones	`Tiene un gran número, referenciado en el cuaderno de google colab.`

Ejemplo 5, página 15 Stickers, rechazada

Molécula	`'cccbbabbb'`
Axioma	`'a'`
Dominós	`[['b','b','', ''],['c','','', ''],['','','b', ''],['','','c', 'b']]`
Resultado	`La molécula: cccbbabbb NO es generada por el lenguaje de stickers`

6. Conclusiones

La utilización del entorno Google colab ha facilitado la implementación de la solución, pudiendo ser programado desde diferentes ordenadores de manera sencilla y realizando las ejecuciones desde cada una de las celdas del cuaderno en formato ‘ipynb’.

La ejecución de código es rápida, algo que indica que la estructura de datos elegida ha sido acertada para la realización del cómputo y para las comparaciones de cadenas correspondientes.

La técnica backtracking implícita en la recursividad utilizada para la tarea no ha indicado una mejora, pero se ha implementado para poder explorar todas las soluciones posibles, podando las soluciones que no son factibles, reduciendo el coste del algoritmo.

Se ha dotado a la solución la posibilidad de definir el axioma como parámetro inicial, para así poder realizar más pruebas, ya que inicialmente el código no contemplaba el uso de axiomas, simplemente el cómputo hasta obtener la cadena vacía.

La realización del trabajo de stickers me ha permitido comprender en mayor profundidad la teoría dada en la asignatura, siendo una manera muy adecuada para poder visualizar el funcionamiento del cómputo de una molécula con operaciones de stickers.