UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CIENCIAS

Práctica 01 Algoritmo No-Determista

PRESENTA

Villegas Salvador Kevin Ricardo

314173739

PROFESORA

María de Luz Gasca Soto

AYUDANTES

-Brenda Margarita Becerra Ruíz
-Malinali González Lara

ASIGNATURA

Complejidad
10 de Marzo de 2024

1. Ruta más corta: Dada una gráfica no dirigida G = (V,E), u, v vértices en V, y *k* entero positivo, ¿Existe una uv-trayectoria en G de peso menor a *k*?
    a. Forma canónica
    Sea *G = (V,E)* conexa, dos vértices *u, v* $\in$ *V* y un entero positivo *k*

    ¿Existe una *uv*-trayectoria en G de longitud menor o igual a *k*?

    b. Diseñar un algoritmo no-determinístico polinomial
        i. Fase adivinadora

        Elegimos un subconjunto de vértices *X* de *V* aleatoriamente. Hacemos ésto recorriendo la lista de vértices de *G* y con un random.Boolean indicando si el vértice estará en el subconjunto propuesto.

        ii. Fase verificadora

        Validamos primero que el subconjunto sea de longitud menor o igual *k*. Seguido con un algoritmo BFS o DFS verificamos si en el subconjunto propuesto *X* se forma una *uv*-trayectoria y si la trayectoria es menor o igual a *k*

        Si cumple con las condiciones anteriores diremos que es un ruta satisfactoria, False en otro caso.

2. 3-SAT.
    a. Forma canónica
    Sea un conjunto de cláusulas C={c1, c2, …, cn}, donde c*i* es una disyunción de exactamente 3 literales booleanos

    ¿Existe una asignación de valores de verdad para el conjunto de cláusulas booleanas, que satisfaga la cláusulas unidos por la conjunción?

    b. Diseñar un algoritmo no-determinístico polinomial
        i. Fase adivinadora

        Para cada literal de las cláusulas les asignamos un valor de verdad (True, False), esto es aleatoriamente.

        Tenemos una lista de literales con la finalidad de no reasignar valores dos veces a la misma literal y evitarnos el tema de T = True y T = False.

        List<Clausulas> clausulas

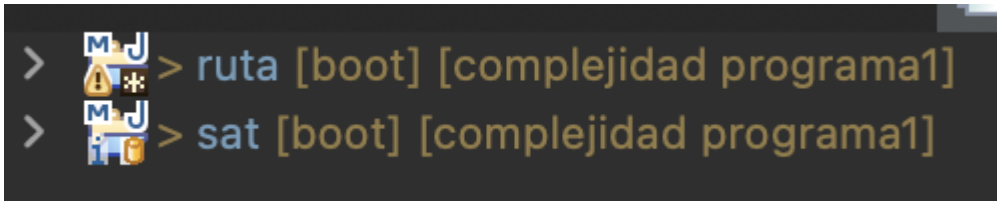
```
List<Literales> literales;
for (Literal l : clausulas){
        if (!literales.contains(l)){
                literales.add(new Literal(l, random.Boolean));
        } else {
                verificaLiteralNegada(l);
        }
}
```

ii. Fase verificadora

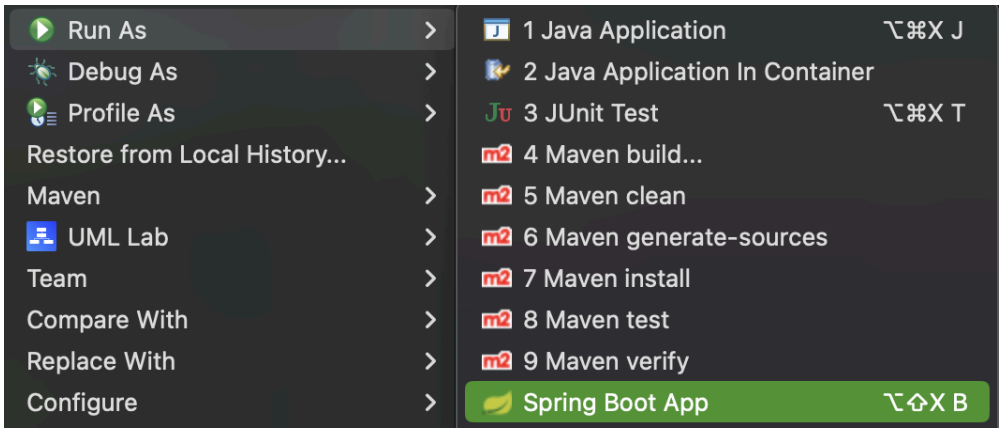
Validamos que cada cláusula |ci| = 3 y ci es una disyunción de literales. Validamos que cada cláusula ci tenga un valor de verdad igual a True, siendo todas las cláusulas True diremos que la fórmula es 3-SAT satisfable, False en otro caso.

Instrucciones para correr el programa.

- Opción 1
    1. Podemos abrir el programa importandolo en un IDE (recomendado Spring Tools o Eclipse)



    2. Una vez teniendo el programa importado, corremos el programa dando click derecho en la raíz del programa con la instrucción Run As -> Spring Boot App



En la consola del IDE veremos la siguiente salida

```
Fase adivinadora

Valores asignados a las literales:
        Literal [name = x, valor = true]
        Literal [name = y, valor = false]
        Literal [name = z, valor = false]
        Literal [name = k, valor = false]

Fase verificadora

        La fórmula: x + -y + z  *  x + y + z  *  -x + -y + -z  *  -x + k + -z
        Es 3-SAT satisfacible
```

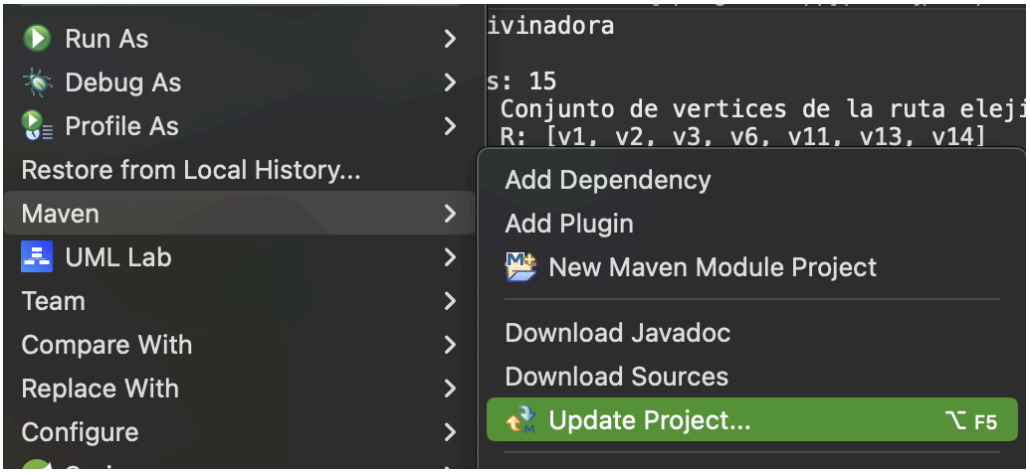
```
Fase adivinadora

vertices: 15
        Conjunto de vertices de la ruta elejida:
        R: [v1, v2, v3, v6, v11, v13, v14]

Fase verificadora

        La ruta anterior no es una ruta menor igual a k
```

En el caso donde arroje error al compilar por no encontrar la clase, debemos actualizar el mvn, dando click derecho en la raíz del proyecto Maven -> Update Project.

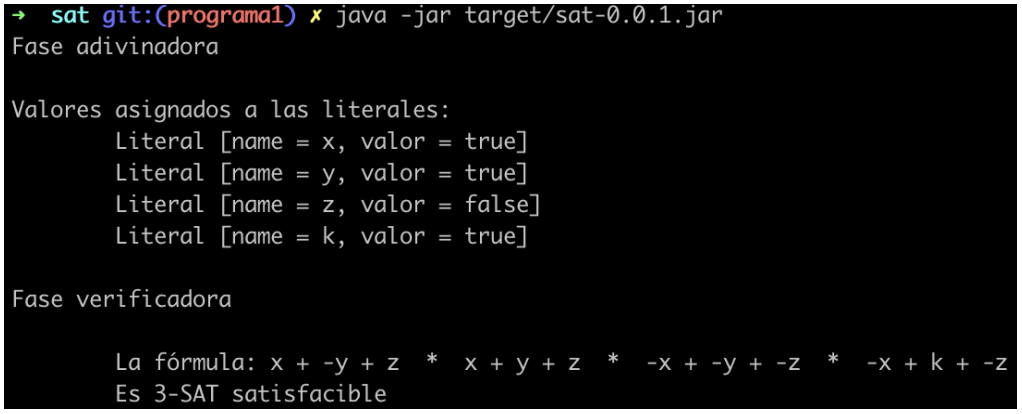


Una vez hecho este paso volvemos a correr el programa.

- Opción 2
    1. Nos posicionamos dentro de la ruta raíz del proyecto
        a. …/sat/
        b. …/ruta/
    2. Dentro de la ruta aplicamos la siguiente sentencia para compilar el programa:

…/sat $ mvn install -DskipTests
…/ruta $ mvn install -DskipTests

3. Para correr el programa ejecutamos el siguiente comando:
   …/sat $ java -jar target/sat-0.0.1.jar



   …/ruta $ java -jar target/ruta-0.0.1.jar