

## **Inteligencia Artificial I - CI5437**

**Profesor:** Carlos Infante

### **Integrantes:**

Daniela Caballero, 14-10140

Emmanuel Bandres, 14-10071

Fernando Gonzalez, 08-10464

## **Proyecto III: CNF con SAT Solver**

### **Especificaciones del entorno de prueba:**

- **Memoria:** 12 GB RAM

- **Procesamiento:** Google Colab

En este proyecto se resolvió el problema de verificar si es posible asignar partidos para un torneo con un determinado número de participantes, las fechas y horas válidas y el conjunto de restricciones especificado en el enunciado, utilizando el SAT Solver Glucose con la representación del problema en CNF.

Para ello las variables se identificaron con un día, hora, jugador local y jugador visitante en strings que se utilizaron como llaves de un diccionario. Cada vez que se crea una variable se añade una entrada al diccionario con ese string como llave y un número entero como valor, asignado de forma secuencial. Se generan las cláusulas requeridas por las restricciones y se escriben en un archivo llamado cnf.txt que es utilizado como la entrada de Glucose. Para finalizar este paso, se guarda el diccionario generado utilizando la librería pickle en un archivo llamado keys.pkl.

Luego, se ejecuta Glucose con el archivo cnf.txt generado y se revisa el output r.txt del programa para comprobar que el problema sea satisfacible. Se carga el archivo de keys.pkl y se reconstruyen las variables positivas en r.txt, obteniendo de nuevo el día, hora, jugador local y jugador visitante.

Por último, se crea el archivo de iCalendar utilizando la librería ics.

Para todas las restricciones dadas se generaron las combinaciones de variables necesarias que las cumplen en la forma clausal de CNF. La mayoría de las cláusulas calculadas contienen dos literales negativos para garantizar que solo uno de ellos debe ser verdadero. Una parte de la primera restricción es la única que genera cláusulas con una mayor cantidad de literales, todos positivos, para asegurar que cada participante juegue al menos una vez con cada uno de los otros participantes.

## Cómo ejecutarlo

Para ejecutar el proyecto primero hay que dirigirse a la carpeta `simp` dentro del directorio `glucose` y escribir el comando **`make rs`** para compilar Glucose. Después en la carpeta principal se debe ejecutar el archivo `main.py`, con el comando **`python3 main.py nombrePrueba.json`**, donde `nombrePrueba.json` puede ser el nombre de cualquier archivo de prueba que se encuentra en la carpeta `Test`. Se necesita Python 3.6 en adelante.

## Resultados

### Casos de prueba fáciles

Caso	Variables	Cláusulas	Tiempo CNF	Tiempo Glucose	Tiempo Total
1	288	18876	0.131 s	0.014 s	0.153 s
2	144	7302	0.024 s	0.006 s	0.036 s
3	1100	165320	0.521 s	0.187 s	0.718 s
4	3600	977790	3.058 s	1.726 s	4.8 s

Se probaron cuatro casos de prueba en formato JSON, donde la cantidad de participantes está en un rango de 3 a 6 y hay variedad en la cantidad de días y horas disponibles. Podemos observar que para todos los casos el tiempo en generar el archivo CNF es mayor al tiempo empleado por Glucose para verificar su satisfacibilidad.

### Casos de prueba medios

Caso	Variables	Cláusulas	Tiempo CNF	Tiempo Glucose	Tiempo Total
------	-----------	-----------	------------	----------------	--------------

1	9016	3862040	12.666 s	8.554 s	21.251 s
2	7920	2826072	9.101 s	5.263 s	14.398 s
3	17010	9523170	29.978 s	2.568 s	32.586 s
4	12096	6298056	19.854 s	1.811 s	21.705 s

Para los casos de prueba medios también se crearon y probaron cuatro casos de prueba, donde la cantidad de participantes varía de 8 a 10 y la duración de los torneos de 21 a 26 días. El rango de horas es amplio para incrementar la dificultad, más la restricción de que cada participante puede jugar máximo una vez al día hace que no todos los bloques de horarios sean utilizados. Se observa nuevamente que el tiempo en pasar el JSON a CNF es mayor al tiempo de ejecución de Glucose para todos los casos.

#### Casos de prueba difíciles

<b>Caso</b>	<b>Variables</b>	<b>Cláusulas</b>	<b>Tiempo CNF</b>	<b>Tiempo Glucose</b>	<b>Tiempo Total</b>
1	39936	34045596	106.939 s	9.930 s	116.946 s
2	45864	39183326	127.731 s	11.563 s	139.394 s
3	47880	39639810	127.123 s	14.087 s	141.314 s
4	63360	58556400	302.163 s	18.633 s	320.926 s

Los cuatro casos de prueba más difíciles que se probaron tienen desde 13 hasta 16 participantes y en todos los casos la duración del torneo es igual o superior a un mes. Se puede apreciar en mayor medida la diferencia del tiempo en generar el CNF y en encontrar una solución con Glucose. Para la gran cantidad de variables y cláusulas que se construyen es de resaltar que Glucose logra verificar la satisfacibilidad de los casos en poco tiempo con los recursos disponibles.

Con estos casos de prueba los archivos que se generaron y que contenían todas las cláusulas fueron bastante grandes, alrededor de 500 MB y llegando a alcanzar los 800 MB para el último caso.

## Conclusión

Podemos ver que en todos los casos la parte que más tiempo tomó durante la ejecución fue la creación del archivo DIMACS cnf.txt. Calcular la solución con Glucose fue relativamente rápido, mientras que la reconstrucción de la solución y creación del calendario siempre se terminó casi instantáneamente.

Utilizar nuestra solución para calcular las fechas de los eventos de un torneo valdría la pena para torneos no muy grandes. Incluso el caso más complicado que utilizamos (44 días, 13 horas y 16 participantes) se tardó un poco más de 5 minutos el cual probablemente se puede reducir considerablemente al usar un calendario más ajustado con menos variables posibles, como en el 3<sup>er</sup> caso de prueba difícil (38 días, 12 horas y 15 participantes) el cual duró 141 segundos en ejecutarse, menos de la mitad que el caso más grande.

Eventualmente, al hacer torneos más grandes, va a llegar un punto donde sería más eficiente asignar las fechas a mano, pero con la cantidad de equipos y los horarios de campeonatos en la vida real, no creemos que esto sea un problema.