



# Proyecto Final - Primera Parte

## COMPLEJIDAD Y OPTIMIZACIÓN

Robinson Duque, Ph.D  
robinson.duque@correounivalle.edu.co

Septiembre de 2020

**Importante:** Esta primera parte del proyecto de curso está pensada para un total de 18 horas de trabajo individual (el equivalente a una dedicación de 6 horas de trabajo durante 3 semanas). El proyecto puede ser desarrollado en grupos de máximo tres (3) estudiantes.

### Introducción

La competencia de SAT (<http://www.satcompetition.org/>) es un evento competitivo para los solucionadores (i.e., solvers) del problema de la satisfacción booleana (SAT). Se organiza como un evento satélite dentro de la “International Conference on Theory and Applications of Satisfiability Testing”.

Las instancias SAT se proveen a través de un formato DIMACS. Para esta primer parte del proyecto, usted debe leer y entender sobre el formato en los siguientes links:

- <http://www.satcompetition.org/2009/format-benchmarks2009.html>
- <https://fairmut3x.wordpress.com/2011/07/29/cnf-conjunctive-normal-form-dimacs-format-explained/>

El siguiente ejemplo muestra una representación válida de una instancia SAT:

$$C = (A \vee \neg B \vee C) \wedge (B \vee D \vee E) \wedge (D \vee F)$$

### Representación DIMACS:

c Esto es un ejemplo

c

p cnf 6 3

1 -2 3 0

2 4 5 0

4 6 0

## 1. (80pts) Reducción SAT a X-SAT

En esta primera parte usted construirá un programa que reciba como entrada una instancia SAT en formato DIMACS y genere una instancia equivalente en X-SAT para un valor X específico. Aspectos a tener en cuenta:

- Debe crear un directorio llamado **SAT**

- Dentro de SAT debe poner un directorio llamado **Reductor** con los fuentes de su implementación. Puede utilizar el lenguaje de programación de su predilección para hacer la reducción.

- Dentro de SAT debe existir un subdirectorio llamado **InstanciasSAT** donde debe poner 100 instancias descargadas de uno de los tres benchmarks (Application, Hard combinatorial, Random) del 2014 (<http://www.satcompetition.org/>) o de la página SATLIB-Benchmark (<https://www.cs.ubc.ca/~hoos/SATLIB/benchm.html>).

- Dentro de SAT debe existir un subdirectorio llamado **X-SAT** donde se almacenarán las 100 instancias convertidas a X-SAT.

- Dentro de SAT debe crear un script en bash llamado **reducir.sh** que:

- Instale los paquetes que se requieran para poder ejecutar su programa
- Compile su programa de ser necesario
- Lea todas las instancias SAT del directorio InstanciasSAT y las convierta a X-SAT para un X específico utilizando su programa reductor

- La efectividad de su capacidad para automatizar su reductor se evaluará el día de la sustentación ejecutando la siguiente instrucción para un nuevo conjunto de instancias:

```
~$bash reducir.sh -x 5
```

- Elabore un video de máximo 4 minutos donde:

- Explique el programa reductor que implementó
- Realice una prueba y muestre que funciona convirtiendo una instancia SAT a 5-SAT
- Ejecute dos instancias Triviales de SAT y 5-SAT donde pueda mostrar que instancias positivas en SAT son instancias positivas en 5-SAT (de la misma manera para instancias negativas)

## 2. (20pts) Informe de Rendimiento

Para esta parte del proyecto, el grupo deberá desarrollar un corto informe (1 página máximo) donde compare tiempos de ejecución para 20 instancias SAT utilizando uno de los solvers ganadores del 2017 (<https://baldur.iti.kit.edu/sat-competition-2017/solvers/>) vs. tiempos de ejecución para las mismas 10 instancias X-SAT. El informe deberá contener:

- Una descripción del solver SAT que se utilizó.
- Un gráfico comparativo de los tiempos de ejecución entre las instancias SAT y las instancias X-SAT (El gráfico debe discriminar entre instancias Satisfactibles e Insatisfactibles).
- **Importante:** asigne un tiempo máximo de ejecución de 5 minutos para cada instancia.
- Cree un subdirectorío dentro de SAT llamado **Informe** donde guarde un pdf del informe realizado. Siga la convención *CódigoEstudiante.pdf* para almacenar los archivos.

## 3. Entrega

- La solución debe ser subida al campus virtual a más tardar el día **23 de Octubre a las 11:59 pm**. Se debe subir al campus virtual en el enlace correspondiente a este proyecto un archivo comprimido **.zip** que siga la convención *CódigoEstudiante1-CódigoEstudiante2-CódigoEstudiante3-Entrega1CyOpt20191.zip*. Este archivo debe contener el directorio **SAT** con los subdirectorios y archivos especificados.

### 3.1. Entregas Tardías o por Otros Medios

1. Este proyecto **sólo se recibirá a través del campus virtual**. Adicionalmente, sólo se evaluarán los directorios/archivos solicitados en los puntos 1 y 2 de la sección anterior. Cualquier otro tipo de correo o nota aclaratoria será descartado. Sólo se aceptan envíos por fuera del horario establecido bajo excusas de fuerza mayor validadas a través de la dirección de la Escuela.
2. Las entregas tarde serán penalizadas así: (-1pt) por cada hora de retraso o fracción. Por ejemplo, si usted realiza su entrega y el campus registra las 12:00 am (i.e., 1seg después de la hora de entrega), usted está incurriendo en la primer hora de retraso. Asegúrese con mínimo dos horas de anticipación que el link de carga funciona correctamente toda vez que es posible incurrir en una entrega tardía debido a los tiempos de respuesta.