



# PUCMM

Pontificia Universidad Católica  
Madre y Maestra

Facultad de Ciencias de las Ingenierías

Departamento de Ingeniería de Sistemas y Computación

Tarea 3 (T3):

T3-2SAT

Por:

Ansël E. Corona Coste 2014-0031

Profesor:

J.R Felipe Nuñez

Asignatura:

Diseño y Análisis de Algoritmos

ISC-405-T-001

Santiago de los Caballeros

República Dominicana

## **Marco teórico:**

El formato del archivo es el siguiente. En cada caso, el número de variables y el número de cláusulas es el mismo, y este número se especifica en la primera línea del archivo. Cada línea subsiguiente especifica una cláusula a través de sus dos literales, con un número que indica la variable y un signo "-" denota lógica "no". Por ejemplo, la segunda línea del primer archivo de datos es "-16808 75250", que indica la cláusula  $\neg x_{16808} \vee x_{75250}$ .

Su tarea consiste en determinar cuál de los 6 casos son satisfiable, y que son insatisfiable. En la respuesta, introduzca una cadena de 6 bits, donde el bit  $i$ -ésimo debe ser 1 si la instancia  $i$ -ésima es satisfiable, y 0 en caso contrario. Por ejemplo, si usted piensa que los 3 primeros casos son satisfiable y el último 3 no son, entonces usted debe responder con 111000 en la respuesta.

DISCUSIÓN: Esta asignación es deliberadamente abierta, y se puede poner en práctica lo que el algoritmo 2SAT desea. Por ejemplo, 2SAT reduce al cálculo de los componentes fuertemente conexos de un gráfico adecuado (con dos vértices por variable y dos bordes dirigidos por cláusula, usted debe pensar en los detalles). Esto podría ser una opción especialmente atractiva para alguno que ha codificado un algoritmo SCC. Como alternativa, puede utilizar el algoritmo de búsqueda local aleatorizado de Papadimitriou. (El algoritmo de clases es probablemente demasiado lento como se ha dicho, por lo que es posible que desee hacer una o más simples modificaciones en él --- incluso si esto significa romper el análisis realizado en clases --- para asegurar que se ejecute en una cantidad razonable de tiempo.) Un tercer enfoque es a través de retroceso.

## Código:

```
import random
import math
def main():
    f=open("2SAT5.txt","r")
    num_vars=int(f.readline())
    inst={}
    choices=[True, False]
    tcclauses={}
    for i in range(1,num_vars+1):
        tcclauses[i]=[]
    for i in range(1,num_vars+1):
        s=f.readline().rstrip()
        variables=s.split(' ')
        if int(variables[0])<0: tcclauses[0-int(variables[0])].append(variables)
        else: tcclauses[int(variables[0])].append(variables)
        if variables[0]!=variables[1]:
            if int(variables[1])<0: tcclauses[0-
int(variables[1])].append(list(reversed(variables)))
            else: tcclauses[int(variables[1])].append(list(reversed(variables)))
        inst[str(i)]=random.choice(choices)
    f.close()
    for i in range(1,num_vars+1):
        if len(tcclauses[i])==0: del tcclauses[i]
    oldnum=num_vars+1
    newnum=num_vars
    while oldnum>newnum:
        oldnum=newnum
        for tc in tcclauses.keys():
            if len(tcclauses[tc])==0:
                del tcclauses[tc]
            else:
                first = tcclauses[tc][0][0]
                sameSign=True
                for i in tcclauses[tc]:
                    if i[0]!=first:
                        sameSign=False
                        break
                if sameSign:
                    for tcc in tcclauses[tc]:
                        var1=tcc[0]
                        var2=tcc[1]
                        if int(var1)<0: var1=str(0-int(var1))
                        if int(var2)<0: var2=str(0-int(var2))
                        var11=True
                        var22=True
                        if int(tcc[0])<0:
                            if inst[str(0-int(tcc[0]))]: var11=False
                            else: var11=True
                        else: var11=inst[tcc[0]]
                        if int(tcc[1])<0:
                            if inst[str(0-int(tcc[1]))]: var22=False
                            else: var22=True
                        else: var22=inst[tcc[1]]
                        if not(var11 or var22):
                            inst[var1]=not inst[var1]
                        if tcc[0]!=tcc[1]:
                            tcclauses[int(var2)].remove(list(reversed(tcc)))
```

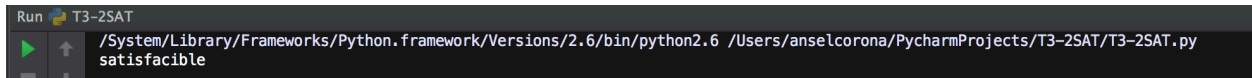
```

        tclauses[tc].remove(tcc)
        if len(tclauses[tc])==0:
            del tclauses[tc]
        newnum-=1
flag=False
for varc in tclauses.keys():
    if len(tclauses[varc])==0: del tclauses[varc]
    else:
        for c in tclauses[varc]:
            var1=c[0]
            var2=c[1]
            if int(var1)<0: var1=str(0-int(var1))
            if int(var2)<0: var2=str(0-int(var2))
            if c[0]!=c[1]:
                tclauses[int(var2)].remove(list(reversed(c)))
for i in range(0,int(math.log(num_vars,2))+1):
    j=1
    while (not flag) and j<(2*num_vars*num_vars):
        flag=True
        for varc in tclauses.keys():
            for c in tclauses[varc]:
                var1=c[0]
                var2=c[1]
                if int(var1)<0: var1=str(0-int(var1))
                if int(var2)<0: var2=str(0-int(var2))
                var11=True
                var22=True
                if int(c[0])<0:
                    if inst[str(0-int(c[0]))]: var11=False
                    else: var11=True
                else: var11=inst[c[0]]
                if int(c[1])<0:
                    if inst[str(0-int(c[1]))]: var22=False
                    else: var22=True
                else: var22=inst[c[1]]
                if not(var11 or var22):
                    if random.choice([1,2])==1: inst[var1]=not inst[var1]
                    else: inst[var2]=not inst[var2]
                    flag=False
                    break
            if flag:
                print "satisfacible"
                break
        j=j+1
    if not flag: print "no satisfacible"

if __name__ == '__main__':
    main()

```

## Conclusión:

A screenshot of a PyCharm Run console window. The title bar shows 'Run' and 'T3-2SAT'. The console output displays the command path '/System/Library/Frameworks/Python.framework/Versions/2.6/bin/python2.6 /Users/anselcorona/PycharmProjects/T3-2SAT/T3-2SAT.py' followed by the output 'satisfacible'.

La respuesta es mostrada.

Recibí satisfacible para los files “2SAT3” y “2SAT5”, para los otros despues de muuuuuuuuuuuucho esperar (muchas u para enfatizar), recibí “no satisfacible.”

A veces aliarse de la investigación puede ayudar mucho...

T3✓✓