Fundamental Computer Science
Lecture 4
SAT and its variants

Denis Trystram
MoSIG1 and M1Info – University Grenoble-Alpes

March, 2021

Content

Variants of SAT

- 2SAT
- NAE-SAT
- 3SAT
- max2SAT

Recall about the satisfiability problem

- $X = \{x_1, x_2, \ldots, x_n\}$: set of variables
- $C = \{C_1, C_2, \ldots, C_m\}$: set of clauses
- $\mathcal{F} = C_1 \wedge C_2 \wedge \ldots \wedge C_m$

$$\mathrm{SAT} = \{\langle \mathcal{F} \rangle \mid \mathcal{F} \text{ is a satisfiable Boolean formula } \}$$

kSAT

- $k\mathrm{SAT}$: each clause has exactly k literals
  - example of $2\mathrm{SAT}$: $(x_1 \vee \bar{x}_2) \wedge (x_2 \vee x_3) \wedge (x_2 \vee \bar{x}_3)$
  - example of $3\mathrm{SAT}$: $(x_1 \vee \bar{x}_2 \vee x_4) \wedge (x_2 \vee x_3 \vee \bar{x}_4)$

kSAT

- $k\mathrm{SAT}$: each clause has exactly k literals
  - example of $2\mathrm{SAT}$: $(x_1 \vee \bar{x}_2) \wedge (x_2 \vee x_3) \wedge (x_2 \vee \bar{x}_3)$
  - example of $3\mathrm{SAT}$: $(x_1 \vee \bar{x}_2 \vee x_4) \wedge (x_2 \vee x_3 \vee \bar{x}_4)$
- $2\mathrm{SAT} \in \mathcal{P}$
- $3\mathrm{SAT} \in \mathcal{NP}$

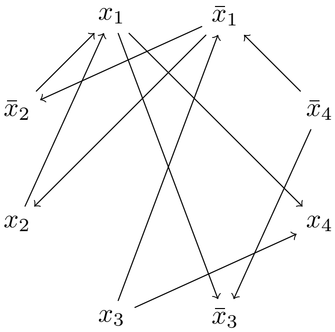
$2\mathrm{SAT} \in \mathcal{P}$

The idea is to transform the problem in a path algorithm in graph.

- Construct the graph G as follows
  - add a vertex for each literal $x \in X \cup \bar{X}$
  - for each clause $x \vee y$, add the arcs $(\bar{x}, y)$ and $(\bar{y}, x)$
    correspond to implications $\bar{x} \Rightarrow y$ and $\bar{y} \Rightarrow x$

$2\text{SAT} \in \mathcal{P}$

$$\mathcal{F} = (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee \bar{x}_3) \wedge (x_1 \vee x_2) \wedge (\bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee x_4)$$



We want $(\bar{x}_1 \vee x_4) = \text{TRUE}$

- arc (x_1, x_4) means:
  - if $x_1 = \text{T}$ then x_4 should be T
  - if $x_4 = \text{F}$ then x_1 should be F
- arc $(\bar{x}_4, \bar{x}_1)$ means:
  - if $\bar{x}_4 = \text{T}$ then $\bar{x}_1$ should be T
  - if $\bar{x}_1 = \text{F}$ then $\bar{x}_4$ should be F

$2\mathrm{SAT} \in \mathcal{P}$

Lemma

If there is a path from x to y in G, then there is also a path from $\bar{y}$ to $\bar{x}$.

Proof:

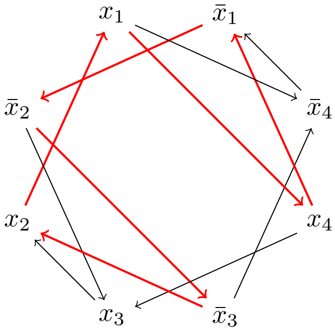
- By construction:
  - we add an arc (a, b) if $(\bar{a} \vee b)$ exists in $\mathcal{F}$
  - but if $(\bar{a} \vee b)$ exists in $\mathcal{F}$, then we add also the arc $(\bar{b}, \bar{a})$
- Apply the argument for all arcs in the path from x to y

$2\mathrm{SAT} \in \mathcal{P}$

Lemma

If there is a variable x such that G has both a path from x to $\bar{x}$ and a path from $\bar{x}$ to x, then $\mathcal{F}$ is not satisfiable.

$$\mathcal{F} = (x_1 \vee \bar{x}_2) \wedge (x_2 \vee \bar{x}_3) \wedge (x_3 \vee \bar{x}_4) \wedge (x_4 \vee \bar{x}_1) \wedge (\bar{x}_4 \vee \bar{x}_1) \wedge (x_2 \vee x_3)$$



If $x_1 = \mathrm{TRUE}$, then
x_4 should be TRUE, and then
$(\bar{x}_4 \vee \bar{x}_1)$ is not satisfiable

If $x_1 = \mathrm{FALSE}$, then
x_2 should be FALSE, and then
$\bar{x}_3$ should be FALSE, and then
$(x_2 \vee x_3)$ is not satisfiable

$2\text{SAT} \in \mathcal{P}$

Lemma

If there is a variable x such that G has both a path from x to $\bar{x}$ and a path from $\bar{x}$ to x, then $\mathcal{F}$ is not satisfiable.

Proof:

- assume that $\mathcal{F}$ is satisfiable (for contradiction)

$2\text{SAT} \in \mathcal{P}$

Lemma

If there is a variable x such that G has both a path from x to $\bar{x}$ and a path from $\bar{x}$ to x, then $\mathcal{F}$ is not satisfiable.

Proof:

- assume that $\mathcal{F}$ is satisfiable (for contradiction)
- case 1: $x = \text{TRUE}$

$$\begin{array}{ccccccccc} x & \longrightarrow & \cdots & \longrightarrow & a & \longrightarrow & b & \longrightarrow & \cdots & \longrightarrow & \bar{x} \\ \text{T} & & & & \text{T} & & \text{F} & & & & \text{F} \end{array}$$

There should be an arc (a, b) with $a = \text{T}$ and $b = \text{F}$.

$2\text{SAT} \in \mathcal{P}$

Lemma

If there is a variable x such that G has both a path from x to $\bar{x}$ and a path from $\bar{x}$ to x, then $\mathcal{F}$ is not satisfiable.

Proof:

- assume that $\mathcal{F}$ is satisfiable (for contradiction)
- case 1: $x = \text{TRUE}$

$$\begin{array}{ccccccccccc} x & \longrightarrow & \cdots & \longrightarrow & a & \longrightarrow & b & \longrightarrow & \cdots & \longrightarrow & \bar{x} \\ \text{T} & & & & \text{T} & & \text{F} & & & & \text{F} \end{array}$$

There should be an arc (a, b) with $a = \text{T}$ and $b = \text{F}$.
That is, $(\bar{a} \vee b)$ is not satisfiable.

$2\mathrm{SAT} \in \mathcal{P}$

Lemma

If there is a variable x such that G has both a path from x to $\bar{x}$ and a path from $\bar{x}$ to x, then $\mathcal{F}$ is not satisfiable.

Proof:

- assume that $\mathcal{F}$ is satisfiable (for contradiction)
- case 1: $x = \mathrm{TRUE}$

$$\begin{array}{ccccccccccc} x & \longrightarrow & \cdots & \longrightarrow & a & \longrightarrow & b & \longrightarrow & \cdots & \longrightarrow & \bar{x} \\ \mathrm{T} & & & & \mathrm{T} & & \mathrm{F} & & & & \mathrm{F} \end{array}$$

There should be an arc (a, b) with $a = \mathrm{T}$ and $b = \mathrm{F}$.
That is, $(\bar{a} \vee b)$ is not satisfiable.
Hence, x cannot be TRUE.

$2\text{SAT} \in \mathcal{P}$

Lemma

If there is a variable x such that G has both a path from x to $\bar{x}$ and a path from $\bar{x}$ to x, then $\mathcal{F}$ is not satisfiable.

Proof:

- assume that $\mathcal{F}$ is satisfiable (for contradiction)
- case 1: $x = \text{TRUE}$

$$\begin{array}{ccccccccccc} x & \longrightarrow & \cdots & \longrightarrow & a & \longrightarrow & b & \longrightarrow & \cdots & \longrightarrow & \bar{x} \\ \text{T} & & & & \text{T} & & \text{F} & & & & \text{F} \end{array}$$

  There should be an arc (a, b) with $a = \text{T}$ and $b = \text{F}$.
  That is, $(\bar{a} \vee b)$ is not satisfiable.
  Hence, x cannot be TRUE.
- case 2: $x = \text{FALSE}$
  Same arguments give that x cannot be FALSE.

$2\mathrm{SAT} \in \mathcal{P}$

Lemma

If there is a variable x such that G has both a path from x to $\bar{x}$ and a path from $\bar{x}$ to x, then $\mathcal{F}$ is not satisfiable.

Proof:

- assume that $\mathcal{F}$ is satisfiable (for contradiction)
- case 1: $x = \mathrm{TRUE}$

$$\begin{array}{ccccccccccc} x & \longrightarrow & \cdots & \longrightarrow & a & \longrightarrow & b & \longrightarrow & \cdots & \longrightarrow & \bar{x} \\ \mathrm{T} & & & & \mathrm{T} & & \mathrm{F} & & & & \mathrm{F} \end{array}$$

  There should be an arc (a, b) with $a = \mathrm{T}$ and $b = \mathrm{F}$.
  That is, $(\bar{a} \vee b)$ is not satisfiable.
  Hence, x cannot be TRUE.
- case 2: $x = \mathrm{FALSE}$
  Same arguments give that x cannot be FALSE.
- Then, $\mathcal{F}$ is not satisfiable, a contradiction.

2SAT $\in \mathcal{P}$

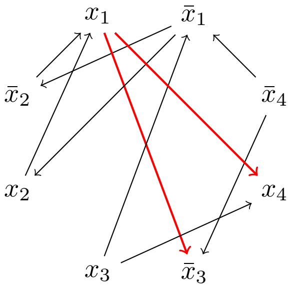
Algorithm

1. **while** there are non-assigned variables **do**
2. Select a literal a for which there is not a path from a to $\bar{a}$.
3. Set $a = \mathrm{TRUE}$.
4. Assign TRUE to all reachable literals from a.
5. Eliminate all assigned variables from G.

$2\mathrm{SAT} \in \mathcal{P}$

$$\mathcal{F} = (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee \bar{x}_3) \wedge (x_1 \vee x_2) \wedge (\bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee x_4)$$

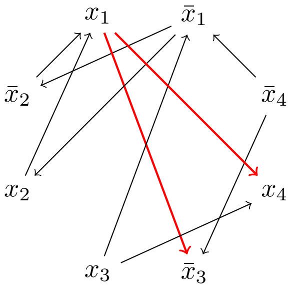
Select $\bar{x}_2$ and set $\bar{x}_2 = \mathrm{TRUE}$

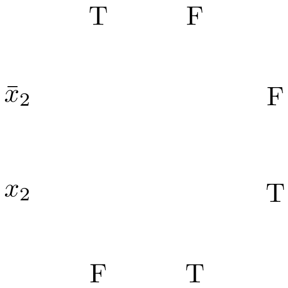


$2\mathrm{SAT} \in \mathcal{P}$

$$\mathcal{F} = (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee \bar{x}_3) \wedge (x_1 \vee x_2) \wedge (\bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee x_4)$$

Select $\bar{x}_2$ and set $\bar{x}_2 = \mathrm{TRUE}$





$2\text{SAT} \in \mathcal{P}$

Lemma (Correctness of the algorithm)

Consider a literal a selected in Line 2 of the algorithm. There is no path from a to both b and $\bar{b}$.

Proof:

- Assume there are paths from a to b and from a to $\bar{b}$.

$2\mathrm{SAT} \in \mathcal{P}$

Lemma (Correctness of the algorithm)

Consider a literal a selected in Line 2 of the algorithm. There is no path from a to both b and $\bar{b}$.

Proof:

- Assume there are paths from a to b and from a to $\bar{b}$.
- Then, there are paths from $\bar{b}$ to $\bar{a}$ and from b to $\bar{a}$ (by the first lemma)

$2\text{SAT} \in \mathcal{P}$

Lemma (Correctness of the algorithm)

Consider a literal a selected in Line 2 of the algorithm. There is no path from a to both b and $\bar{b}$.

Proof:

- Assume there are paths from a to b and from a to $\bar{b}$.
- Then, there are paths from $\bar{b}$ to $\bar{a}$ and from b to $\bar{a}$ (by the first lemma)
- Thus, there are paths from a to $\bar{a}$ (passing through b) and from $\bar{a}$ to a (passing through $\bar{b}$)

$2\text{SAT} \in \mathcal{P}$

Lemma (Correctness of the algorithm)

Consider a literal a selected in Line 2 of the algorithm. There is no path from a to both b and $\bar{b}$.

Proof:

- Assume there are paths from a to b and from a to $\bar{b}$.
- Then, there are paths from $\bar{b}$ to $\bar{a}$ and from b to $\bar{a}$ (by the first lemma)
- Thus, there are paths from a to $\bar{a}$ (passing through b) and from $\bar{a}$ to a (passing through $\bar{b}$)
- a cannot be selected by the algorithm, a contradiction.

3SAT $\in$ NP-complete

3SAT $\in \mathcal{NP}$

- given an assignment of variables, scan all clauses to check if they evaluate to TRUE

$3\mathrm{SAT} \in \mathrm{NP}$-complete

$3\mathrm{SAT} \in \mathcal{NP}$

- given an assignment of variables, scan all clauses to check if they evaluate to TRUE

$\mathrm{SAT} \leq_{\mathrm{P}} 3\mathrm{SAT}$

- given any formula $\mathcal{F}$ of SAT, we construct a formula $\tau(\mathcal{F})$ of $3\mathrm{SAT}$
  - replace each clause $(a_1 \vee a_2 \vee \ldots \vee a_\ell)$ in $\mathcal{F}$

3SAT $\in$ NP-complete

3SAT $\in \mathcal{NP}$

- given an assignment of variables, scan all clauses to check if they evaluate to TRUE

SAT $\leq_{\mathrm{P}}$ 3SAT

- given any formula $\mathcal{F}$ of SAT, we construct a formula $\tau(\mathcal{F})$ of 3SAT
  - replace each clause $(a_1 \vee a_2 \vee \ldots \vee a_\ell)$ in $\mathcal{F}$
  - if $\ell = 2$, add an extra variable z:
    $(a_1 \vee a_2) = (a_1 \vee a_2 \vee z) \wedge (a_1 \vee a_2 \vee \bar{z})$

3SAT $\in$ NP-complete

3SAT $\in \mathcal{NP}$

- given an assignment of variables, scan all clauses to check if they evaluate to TRUE

SAT $\leq_{\text{P}}$ 3SAT

- given any formula $\mathcal{F}$ of SAT, we construct a formula $\tau(\mathcal{F})$ of 3SAT
  - replace each clause $(a_1 \vee a_2 \vee \ldots \vee a_\ell)$ in $\mathcal{F}$
  - if $\ell = 2$, add an extra variable z:
    $(a_1 \vee a_2) = (a_1 \vee a_2 \vee z) \wedge (a_1 \vee a_2 \vee \bar{z})$
    Similarly for $\ell = 1$ by adding two variables

3SAT $\in$ NP-complete

3SAT $\in \mathcal{NP}$

- given an assignment of variables, scan all clauses to check if they evaluate to TRUE

SAT $\leq_{\mathrm{P}}$ 3SAT

- given any formula $\mathcal{F}$ of SAT, we construct a formula $\tau(\mathcal{F})$ of 3SAT
  - replace each clause $(a_1 \vee a_2 \vee \ldots \vee a_\ell)$ in $\mathcal{F}$
  - if $\ell = 2$, add an extra variable z:
    $(a_1 \vee a_2) = (a_1 \vee a_2 \vee z) \wedge (a_1 \vee a_2 \vee \bar{z})$
    Similarly for $\ell = 1$ by adding two variables
  - if $\ell > 3$, add $\ell - 3$ variables z_i and replace the clause by the $\ell - 2$ following clauses

    $$(a_1 \vee a_2 \vee z_1) \wedge (\bar{z}_1 \vee a_3 \vee z_2) \wedge (\bar{z}_2 \vee a_4 \vee z_3) \wedge \ldots \wedge (\bar{z}_{\ell-3} \vee a_{\ell-1} \vee a_\ell)$$

Proof (1)

$\mathcal{F}$ is satisfiable iff $\tau(\mathcal{F})$ is satisfiable

$(\Rightarrow)$

- assume that $\mathcal{F}$ is satisfiable
- then some a_i is TRUE for all clauses
- use the same assignment for the common variables of $\mathcal{F}$ and $\tau(\mathcal{F})$
- set $z_j = \text{TRUE}$ for $1 \leq j \leq i - 2$
- set $z_j = \text{FALSE}$ for $i - 1 \leq j \leq \ell - 3$
- all the clauses of $\tau(\mathcal{F})$ are satisfied

Proof (2)

$\mathcal{F}$ is satisfiable iff $\mathcal{F}' = \tau(\mathcal{F})$ is satisfiable

$(\Leftarrow)$

- assume that $\mathcal{F}'$ is satisfiable
- at least one of the literals a_i should be TRUE for each clause
- if not, then z_1 should be TRUE which implies that z_2 should be TRUE, etc.
- hence, the clause $(\bar{z}_{\ell-3} \vee a_{\ell-1} \vee a_\ell)$ is not satisfiable, contradiction
- then there is an assignment that satisfies $\mathcal{F}$

Exercise 3SAT-NAE

SAT not all equal.
Prove that $\text{3SAT-NAE} \in \text{NP-}\textsc{complete}$, where

$\text{SAT-NAE} = \{\langle \mathcal{F} \rangle \mid \mathcal{F}$ is a satisfiable with at least one true literal and at least one false literal in each clause$\}$

Tip for the reduction:

- Show first that: $3SAT \leq_P \text{4SAT-NAE}$ (add an extra boolean variable in each clause)
- $\text{4SAT-NAE} \leq_P \text{3SAT-NAE}$ (break each 4-clause into 2 3-clauses)

MAX-2SAT $\in$ NP-COMPLETE

$\text{MAX-2SAT} = \{\langle \mathcal{F}, k \rangle \mid \mathcal{F}$ is a formula with k TRUE clauses$\}$

MAX-2SAT $\in$ NP-complete

$\text{MAX-2SAT} = \{\langle \mathcal{F}, k \rangle \mid \mathcal{F}$ is a formula with k TRUE clauses$\}$

$\text{MAX-2SAT} \in \mathcal{NP}$

- given an assignment of variables, scan all clauses to check if there are at least k of them evaluated to TRUE

MAX-2SAT $\in$ NP-complete

$\text{MAX-2SAT} = \{\langle \mathcal{F}, k \rangle \mid \mathcal{F}$ is a formula with k TRUE clauses$\}$

MAX-2SAT $\in \mathcal{NP}$

- given an assignment of variables, scan all clauses to check if there are at least k of them evaluated to TRUE

3SAT $\leq_{\text{P}}$ MAX-2SAT

1. given any formula $\mathcal{F}$ of 3SAT, we construct a formula $\mathcal{F}'$ of MAX-2SAT
   - replace each clause $(x \vee y \vee z)$ by the 10 following clauses

     $$(x) \wedge (y) \wedge (z) \wedge (\bar{x} \vee \bar{y}) \wedge (\bar{y} \vee \bar{z}) \wedge (\bar{z} \vee \bar{x}) \wedge (w) \wedge (\bar{w} \vee x) \wedge (\bar{w} \vee y) \wedge (\bar{w} \vee z)$$

   - $k = 7m$ (m is the number of clauses)

MAX-2SAT $\in$ NP-complete

$\text{MAX-2SAT} = \{\langle \mathcal{F}, k \rangle \mid \mathcal{F}$ is a formula with k TRUE clauses$\}$

MAX-2SAT $\in \mathcal{NP}$

- given an assignment of variables, scan all clauses to check if there are at least k of them evaluated to TRUE

3SAT $\leq_P$ MAX-2SAT

1. given any formula $\mathcal{F}$ of 3SAT, we construct a formula $\mathcal{F}'$ of MAX-2SAT
    - replace each clause $(x \vee y \vee z)$ by the 10 following clauses

      $$(x) \wedge (y) \wedge (z) \wedge (\bar{x} \vee \bar{y}) \wedge (\bar{y} \vee \bar{z}) \wedge (\bar{z} \vee \bar{x}) \wedge (w) \wedge (\bar{w} \vee x) \wedge (\bar{w} \vee y) \wedge (\bar{w} \vee z)$$

    - $k = 7m$ (m is the number of clauses)
2. $\mathcal{F}'$ has $O(n + m)$ variables and $O(m)$ clauses

MAX-2SAT $\in$ NP-COMPLETE

3SAT $\leq_{\mathrm{P}}$ MAX-2SAT

Recall replace each clause $(x \vee y \vee z)$ by

$$(x) \wedge (y) \wedge (z) \wedge (\bar{x} \vee \bar{y}) \wedge (\bar{y} \vee \bar{z}) \wedge (\bar{z} \vee \bar{x}) \wedge (w) \wedge (\bar{w} \vee x) \wedge (\bar{w} \vee y) \wedge (\bar{w} \vee z)$$

3. $\mathcal{F}$ is satisfiable iff $\mathcal{F}'$ has at least k satisfied clauses
    - assume that $\mathcal{F}$ is satisfiable
    - if $x = \mathrm{T}$, $y = \mathrm{F}$ and $z = \mathrm{F}$, then set $w = \mathrm{F}$: 7 satisfied clauses
    - if $x = \mathrm{T}$, $y = \mathrm{T}$ and $z = \mathrm{F}$, then set $w = \mathrm{F}$: 7 satisfied clauses
    - if $x = \mathrm{T}$, $y = \mathrm{T}$ and $z = \mathrm{T}$, then set $w = \mathrm{T}$: 7 satisfied clauses
    - in all cases, there are 7 satisfied clauses in $\mathcal{F}'$ for each clause of $\mathcal{F}$

MAX-2SAT $\in$ NP-complete

3SAT $\leq_P$ MAX-2SAT

Recall replace each clause $(x \vee y \vee z)$ by

$$(x) \wedge (y) \wedge (z) \wedge (\bar{x} \vee \bar{y}) \wedge (\bar{y} \vee \bar{z}) \wedge (\bar{z} \vee \bar{x}) \wedge (w) \wedge (\bar{w} \vee x) \wedge (\bar{w} \vee y) \wedge (\bar{w} \vee z)$$

3. $\mathcal{F}$ is satisfiable iff $\mathcal{F}'$ has at least k satisfied clauses
   - assume that $\mathcal{F}$ is satisfiable
   - if $x = \mathrm{T}$, $y = \mathrm{F}$ and $z = \mathrm{F}$, then set $w = \mathrm{F}$: 7 satisfied clauses
   - if $x = \mathrm{T}$, $y = \mathrm{T}$ and $z = \mathrm{F}$, then set $w = \mathrm{F}$: 7 satisfied clauses
   - if $x = \mathrm{T}$, $y = \mathrm{T}$ and $z = \mathrm{T}$, then set $w = \mathrm{T}$: 7 satisfied clauses
   - in all cases, there are 7 satisfied clauses in $\mathcal{F}'$ for each clause of $\mathcal{F}$

   - contrapositive: assume that $\mathcal{F}$ is not satisfiable
   - there is one clause for which $x = y = z = \mathrm{F}$
   - then, in $\mathcal{F}'$ we correspondingly have:
     – 4 satisfied clauses if $w = \mathrm{T}$
     – 6 satisfied clauses if $w = \mathrm{F}$
   - hence, in $\mathcal{F}'$ there are less than k clauses that are satisfied