Computability and Complexity
COSC 4200

2SAT is in P

Theorem

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

Proof. Given a 2CNF formula ϕ, we construct a directed graph G as follows.

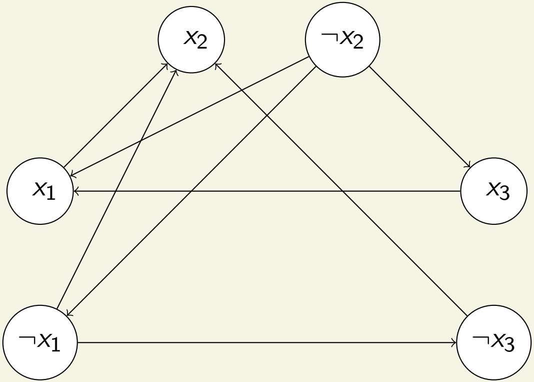
- The vertices of G are the variables of ϕ and their negations.
- There is an edge from vertex α to vertex β if there is a clause $(\neg\alpha \vee \beta)$ or $(\beta \vee \neg\alpha)$ in ϕ.

In other words, we put an edge from α to β if the implication $\alpha \to \beta$ is represented in ϕ.

For an example, for the formula

$$\phi = (x_1 \vee x_2) \wedge (x_1 \vee \neg x_3) \wedge (\neg x_1 \vee x_2) \wedge (x_2 \vee x_3)$$

we construct the following graph:



There is a nice symmetry in the graph, in that if there is a path from α to β, there is also a path from $\neg\beta$ to $\neg\alpha$.

Claim

ϕ is unsatisfiable if and only if there is a variable x such that there are paths from x to $\neg x$ and from $\neg x$ to x in G.

Proof of Claim. Suppose that for some variable x, there is a path from x to $\neg x$ and from $\neg x$ to x. Suppose also that ϕ is satisfiable by some assignment τ.

Proof of Claim. Suppose that for some variable x, there is a path from x to $\neg x$ and from $\neg x$ to x. Suppose also that ϕ is satisfiable by some assignment τ. There are two cases:

Case 1: $\tau(x) = T$. Since there is a path in G from x to $\neg x$, and $\neg x = F$ under τ, there must be some edge (α, β) along the path with $\tau(\alpha) = T$ but $\tau(\beta) = F$. But then $(\neg\alpha \vee \beta)$ is a clause in ϕ and it is not satisfied by τ, a contradiction.

Proof of Claim. Suppose that for some variable x, there is a path from x to $\neg x$ and from $\neg x$ to x. Suppose also that ϕ is satisfiable by some assignment τ. There are two cases:

Case 1: $\tau(x) = T$. Since there is a path in G from x to $\neg x$, and $\neg x = F$ under τ, there must be some edge (α, β) along the path with $\tau(\alpha) = T$ but $\tau(\beta) = F$. But then $(\neg\alpha \vee \beta)$ is a clause in ϕ and it is not satisfied by τ, a contradiction.

Case 2: $\tau(x) = F$. Since there is a path in G from $\neg x$ to x and $x = F$ under τ, there must be some edge (α, β) along the path with $\tau(\alpha) = T$ but $\tau(\beta) = F$. But then $(\neg\alpha \vee \beta)$ is a clause in ϕ and it is not satisfied by τ, a contradiction.

Proof of Claim. Suppose that for some variable x, there is a path from x to $\neg x$ and from $\neg x$ to x. Suppose also that ϕ is satisfiable by some assignment τ. There are two cases:

Case 1: $\tau(x) = T$. Since there is a path in G from x to $\neg x$, and $\neg x = F$ under τ, there must be some edge (α, β) along the path with $\tau(\alpha) = T$ but $\tau(\beta) = F$. But then $(\neg\alpha \vee \beta)$ is a clause in ϕ and it is not satisfied by τ, a contradiction.

Case 2: $\tau(x) = F$. Since there is a path in G from $\neg x$ to x and $x = F$ under τ, there must be some edge (α, β) along the path with $\tau(\alpha) = T$ but $\tau(\beta) = F$. But then $(\neg\alpha \vee \beta)$ is a clause in ϕ and it is not satisfied by τ, a contradiction.

In either case we have a contradiction, so ϕ must be unsatisfiable.

Now assume that for all x, there is no path from x to $\neg x$ and from $\neg x$ to x.

We want to show that ϕ is satisfiable.

Now assume that for all x, there is no path from x to $\neg x$ and from $\neg x$ to x.

We want to show that ϕ is satisfiable. To do so, we repeat the following until all vertices have a T/F assignment:

- Pick a vertex α in G whose truth value has not been defined, such that there is no path from α to $\neg\alpha$. We know such a vertex exists by our assumption. Initially we can take any variable and use either it or its negation.

Now assume that for all x, there is no path from x to $\neg x$ and from $\neg x$ to x.

We want to show that ϕ is satisfiable. To do so, we repeat the following until all vertices have a T/F assignment:

- Pick a vertex α in G whose truth value has not been defined, such that there is no path from α to $\neg\alpha$. We know such a vertex exists by our assumption. Initially we can take any variable and use either it or its negation.
- Assign α, and all vertices reachable from α, the value T.

Now assume that for all x, there is no path from x to $\neg x$ and from $\neg x$ to x.

We want to show that ϕ is satisfiable. To do so, we repeat the following until all vertices have a T/F assignment:

- Pick a vertex α in G whose truth value has not been defined, such that there is no path from α to $\neg\alpha$. We know such a vertex exists by our assumption. Initially we can take any variable and use either it or its negation.
- Assign α, and all vertices reachable from α, the value T.
- Assign the negation of all those vertices the value F. In other words, assign $\neg\alpha$, and all vertices from which $\neg\alpha$ can be reached, the value F.

We observe that we can always assign all vertices reachable from α the value of T. If a vertex β can be reached from α and it were already assigned F, then by our algorithm, we would have already assigned α F because β can be reached from it.

We observe that we can always assign all vertices reachable from α the value of T. If a vertex β can be reached from α and it were already assigned F, then by our algorithm, we would have already assigned α F because β can be reached from it.

Also, the negation of every node reachable from α can be assigned F. If both β and $\neg\beta$ are reachable from α, then by symmetry, there are paths to $\neg\alpha$ from both β and $\neg\beta$. This means there is a path from α to $\neg\alpha$, which contradictions our assumption.

Since for all x, there is no path from x to $\neg x$ or there is no path from $\neg x$ to x, all vertices will eventually be assigned a truth value. For every edge (α, β) in the graph, we have either

- $(\alpha = T) \to (\beta = T)$,
- $(\alpha = F) \to (\beta = F)$, or
- $(\alpha = F) \to (\beta = T)$.

This is because the construction ensures the false implication, $(\alpha = T) \to (\beta = F)$, never occurs. Therefore all clauses of ϕ are satisfied by this assignment. $\square$ Claim

Now we can describe an algorithm for $2\mathrm{SAT}$.

- We input ϕ and construct the graph G.
- For each variable x in ϕ, check to see if there is a path from x to $\neg x$ and a path from $\neg x$ to x.
- If there is such a set of paths for any variable, we reject. Otherwise, we accept.

Since reachability can be computed in polynomial time (e.g. using BFS, DFS, etc.), we can solve $2\mathrm{SAT}$ in polynomial time. $\square$

Theorem

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

In fact, a stronger result is true:

Theorem

$\mathrm{2SAT} \in \mathrm{NL}$.

Theorem

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

In fact, a stronger result is true:

Theorem

$2\mathrm{SAT} \in \mathrm{NL}$.

Proof. The following algorithm shows $2\mathrm{SAT}^c \in \mathrm{NL}$.

- We input ϕ, guess a variable x.
- Guess paths from x to $\neg x$ and from $\neg x$ to x.
- If both paths are found, accept.
- Otherwise, reject.

Theorem

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

In fact, a stronger result is true:

Theorem

$2\mathrm{SAT} \in \mathrm{NL}$.

Proof. The following algorithm shows $2\mathrm{SAT}^c \in \mathrm{NL}$.

- We input ϕ, guess a variable x.
- Guess paths from x to $\neg x$ and from $\neg x$ to x.
- If both paths are found, accept.
- Otherwise, reject.

This algorithm is correct by the analysis in the previous proof.
Thus $2\mathrm{SAT} \in \mathrm{coNL}$. Since $\mathrm{NL} = \mathrm{coNL}$, we have $2\mathrm{SAT} \in \mathrm{NL}$. $\square$

Theorem

$\mathrm{2SAT}$ *is* NL*-complete.*

Proof. We will show $\mathrm{PATH}^c \leq_{\mathrm{L}} \mathrm{2SAT}$. This will establish that $A \leq_{\mathrm{L}} \mathrm{2SAT}$ for every $A \in \mathrm{coNL}$. Since $\mathrm{NL} = \mathrm{coNL}$, it follows that $\mathrm{2SAT}$ is NL-complete.

Theorem

$\mathrm{2SAT}$ *is* NL*-complete.*

Proof. We will show $\mathrm{PATH}^c \leq_{\mathrm{L}} \mathrm{2SAT}$. This will establish that $A \leq_{\mathrm{L}} \mathrm{2SAT}$ for every $A \in \mathrm{coNL}$. Since $\mathrm{NL} = \mathrm{coNL}$, it follows that $\mathrm{2SAT}$ is NL-complete.

Let $G = (V, E)$, s, and t be an instance of PATH^c. Our formula ϕ will have V as variables and have $|E| + 2$ clauses: (s), $(\neg t)$, and $(\neg u \vee v)$ for each $(u, v) \in E$.

$$\phi = (s) \wedge (\neg t) \wedge \bigwedge_{(u,v)\in E} (\neg u \vee v)$$

Note that ϕ is computable by a log-space transducer.

Suppose there is a path from s to t:

$$s \to u_1 \to u_2 \cdots u_k \to t$$

Suppose there is a path from s to t:

$$s \to u_1 \to u_2 \cdots u_k \to t$$

Then

$$(\neg s \vee u_1), (\neg u_2 \vee u_3), \cdots, (\neg u_{k-1} \vee u_k), (\neg u_k \vee t)$$

are all clauses in ϕ.

Suppose there is a path from s to t:

$$s \to u_1 \to u_2 \cdots u_k \to t$$

Then

$$(\neg s \vee u_1), (\neg u_2 \vee u_3), \cdots, (\neg u_{k-1} \vee u_k), (\neg u_k \vee t)$$

are all clauses in ϕ.

If these clauses are all satisfied by some assignment τ, then $(\neg s \vee t)$ is satisfied by τ as well.

Suppose there is a path from s to t:

$$s \to u_1 \to u_2 \cdots u_k \to t$$

Then

$$(\neg s \vee u_1), (\neg u_2 \vee u_3), \cdots, (\neg u_{k-1} \vee u_k), (\neg u_k \vee t)$$

are all clauses in ϕ.

If these clauses are all satisfied by some assignment τ, then $(\neg s \vee t)$ is satisfied by τ as well.

Thus τ does not satisfy ϕ since it has clauses (s) and $(\neg t)$. Therefore ϕ is unsatisfiable.

Suppose there is a path from s to t:

$$s \to u_1 \to u_2 \cdots u_k \to t$$

Then

$$(\neg s \vee u_1), (\neg u_2 \vee u_3), \cdots, (\neg u_{k-1} \vee u_k), (\neg u_k \vee t)$$

are all clauses in ϕ.

If these clauses are all satisfied by some assignment τ, then $(\neg s \vee t)$ is satisfied by τ as well.

Thus τ does not satisfy ϕ since it has clauses (s) and $(\neg t)$. Therefore ϕ is unsatisfiable.

This shows $\langle G, s, t \rangle \in \mathrm{PATH} \Rightarrow \phi \notin \mathrm{2SAT}$.

Now suppose there is no path from s to t. Let

$$\begin{aligned} S &= \{v \in V \mid v \text{ is reachable from } s\} \\ T &= \{v \in V \mid t \text{ is reachable from } v\} \\ M &= V - (S \cup T) \end{aligned}$$

Then S, T, and M are disjoint. There are no edges from S to $M \cup T$ and there are no edges from $S \cup M$ to T.

Now suppose there is no path from s to t. Let

$$\begin{array}{rcl} S & = & \{v \in V \mid v \text{ is reachable from } s\} \\ T & = & \{v \in V \mid t \text{ is reachable from } v\} \\ M & = & V - (S \cup T) \end{array}$$

Then S, T, and M are disjoint. There are no edges from S to $M \cup T$ and there are no edges from $S \cup M$ to T.

Define an assignment τ by

$$\tau(v) = \begin{cases} T & \text{if } v \in S \\ T & \text{if } v \in M \\ F & \text{if } v \in T \end{cases}$$

This assignment satisfies ϕ because it sets $\tau(s) = T$, $\tau(t) = F$, and for any edge (u, v) the clause $(\neg u \vee v)$ is satisfied because $\tau(u) = \tau(v)$.

Now suppose there is no path from s to t. Let

$$\begin{aligned} S &= \{v \in V \mid v \text{ is reachable from } s\} \\ T &= \{v \in V \mid t \text{ is reachable from } v\} \\ M &= V - (S \cup T) \end{aligned}$$

Then S, T, and M are disjoint. There are no edges from S to $M \cup T$ and there are no edges from $S \cup M$ to T.

Define an assignment τ by

$$\tau(v) = \begin{cases} T & \text{if } v \in S \\ T & \text{if } v \in M \\ F & \text{if } v \in T \end{cases}$$

This assignment satisfies ϕ because it sets $\tau(s) = T$, $\tau(t) = F$, and for any edge (u, v) the clause $(\neg u \vee v)$ is satisfied because $\tau(u) = \tau(v)$.

This proves $\langle G, s, t \rangle \notin \mathrm{PATH} \Rightarrow \phi \in \mathrm{2SAT}$.

We have shown

$$\langle G, s, t\rangle \in \mathrm{PATH} \Leftrightarrow \phi \notin \mathrm{2SAT}.$$

Equivalently,

$$\langle G, s, t\rangle \in \mathrm{PATH}^c \Leftrightarrow \phi \in \mathrm{2SAT}.$$

Thus $\mathrm{PATH}^c \leq_{\mathrm{L}} \mathrm{2SAT}$. $\square$