Reductions from 3-SAT

Def: $ALL_{DFA} = \{\langle D \rangle : L(D) = \Sigma^*\}$ [Σ is D's input alphabet]

$ALL_{DFA} \in P$ — BFS from start state, looking for a rejecting state, $L(D) = \Sigma^*$ iff none found.

Def: $ALL_{NFA} = \{\langle N \rangle : N \text{ is an NFA and } L(N) = \Sigma^*\}$

Prop: $\overline{ALL_{NFA}}$ is NP-hard.

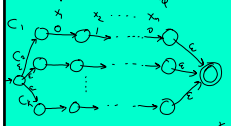
Proof: Reduce 3-SAT to ALL_{NFA} ($3\text{-SAT} \leq_p \overline{ALL_{NFA}}$) as follows:

Given a 3-cnf fmla φ

$\varphi = C_1 \wedge C_2 \wedge \cdots \wedge C_k$ (some k)

with variables $x_1, \ldots, x_n$ (some n)

Build in ptime an NFA N_φ:



N_φ accepts all binary strings except length-n binary strings encoding sat assignments to φ

Ex: $\varphi = (x_1 \vee \bar{x}_2 \vee x_n) \wedge \cdots$ (C_1)

For each $C_i = l_1 \vee l_2 \vee l_3$, label edge on path C_i corresp to l_1 "oppositely" from l_1, same for l_2 & l_3

where "oppositely" means $\begin{cases} 0 & \text{if } l_1 \text{ is pos var} \\ 1 & \text{if } l_1 \text{ is neg var} \end{cases}$

All other edges labeled 0,1

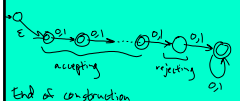
Consider any binary string a of length n

– If a encodes a sat assgnt to φ, then no accepting path (argued before)

– If a encodes non-sat assgnt, then there exists a C_i where all literals are made false by a. The corresp path can be taken to the accept state.

$\therefore$ a satisfies φ iff N_φ rejects a.

To make N_φ accept all strings of length $\neq n$, add one more path from the start state:



End of construction.

Clearly ptime, &

φ is sat $\iff$ N_φ does not accept all strings

$\iff \langle N_\varphi \rangle \in \overline{ALL_{NFA}}$

$\therefore 3\text{-SAT} \leq_p \overline{ALL_{NFA}}$

Cor: No ptime decision alg for ALL_{NFA} unless P=NP.

Cor: NFA minimization is NP-hard.

An NFA accepts all strings iff it minimizes to

If you can minimize NFAs in P-time, then you can decide ALL_{NFA} in p-time, thus P=NP.

Def: Let $k \geq 1$ be fixed. A graph G is k-colorable if there is a map $c: G.V \to \{1, \ldots, k\}$ such that no two adjacent vertices u, v have $c(u) = c(v)$.

The k-colorability problem:

k-COL:

Instance: A graph G

Q: Is G k-colorable?

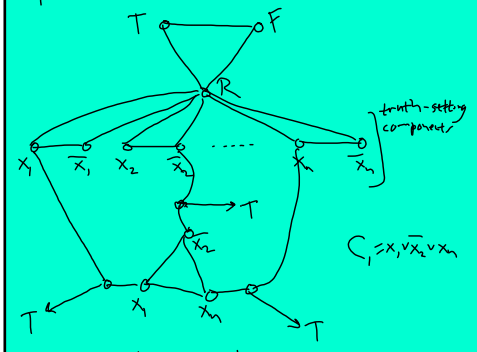
k-COL $\in P$ for $k=1$ & $k=2$.

Prop: 3-COL is NP-complete.

PF: 3-COL $\in$ NP (clear).

For NP-hardness, reduce from 3-SAT. Given a 3-cnf $\varphi = C_1 \wedge \cdots \wedge C_k$ (k clauses, $x_1, \ldots, x_n$ vars) we build in ptime a graph G_φ as follows

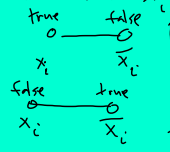
G_φ:



Have a similar "gadget" for each clause. End of construction. WTS:

φ sat $\iff$ G_φ is 3-colorable.

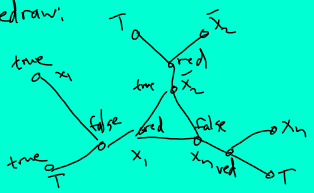
($\Rightarrow$) Let a be a sat assmt to φ. If a makes x_i true, then make



else

each $i = 1, \ldots, n$

Redraw:

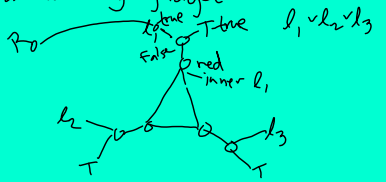


If, say, a makes x_1 true then can extend coloring to the C_1 gadget (similarly for the other 2 literals)

Similarly for all clauses. ($\Rightarrow$) ✓

($\Leftarrow$) Suppose G_φ has a 3-coloring.

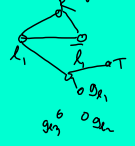
look at any gadget for clause



If inner l_1 is red, then l_1 in truth setting components must be colored true, corresp. to l_1 set to true.

Similarly for l_2 & l_3.

Holds for every clause gadget, so every clause has a true literal under the corresponding truth assmt.



$\therefore$ 3-SAT $\leq_p$ 3-COL

$\varphi \longmapsto G_\varphi$ ///

Next: k-COL is NP-complete for $k \geq 3$.