

Some Remarks on the Cook-Levin Proof

Depends on the size n of x alone.

① Most of the formula $\bigwedge (u \vee w) \wedge (v \vee w) \wedge (\bar{u} \vee \bar{v} \vee \bar{w})$ is independent of the bits in x .

Substitute bit values of x , e.g. $x = 01101$

$\bigwedge (w_0) \wedge (\bar{x}_1) \wedge (x_2) \wedge (x_3) \wedge (\bar{x}_4) \wedge (x_5)$
" $\forall z \vee \bar{z}$ " the only part of $f(x) = \phi_x$ that depends on x "

② The reduction goes directly to 3SAT where the only size-3 clauses have all-negative literals.

③ This is the "up to 3" defn of 3SAT, but we can meet the strict definition by finding a fixed ψ that must be set false to satisfy ψ . Then add z as a disjunct to all other clauses.

③' By further similar tricks, you can arrange that every satisfying assgmt to the modified 3SAT makes 2 literals true and 1 false, or 1 literal true and 2 false. "Not All Equal 3SAT"

Substitute bit values of x , e.g. $x = 01101$

$\bigwedge (w_0) \wedge (\bar{x}_1) \wedge (x_2) \wedge (x_3) \wedge (\bar{x}_4) \wedge (x_5)$
" $\forall z \vee \bar{z}$ " the only part of $f(x) = \phi_x$ that depends on x "

where the only size-3 clauses have all-negative literals. "widget" ψ

but we can meet the strict definition by finding a fixed finite strict-3CNF formula ψ with a variable z .

Then add z as a disjunct to all other clauses.

arrange that every satisfying assgmt to the modified 3CNF formula ϕ'_x either makes 2 literals true and 1 false, or 1 literal true and 2 false. "Not All Equal 3SAT"

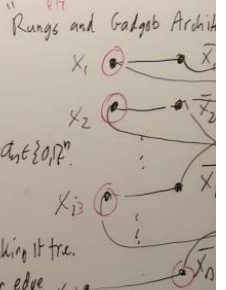
Some Remarks on the Cook-Levin Proof

Depends on the size n of x alone

Most of the formula $\bigwedge_{v \in V} (u \vee w) \wedge (v \vee w) \wedge (\bar{u} \vee \bar{v} \vee \bar{w})$ is independent of the bit in x .

Example: $\phi = (\bar{x}_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2 \vee x_4)$

$\bigwedge (W_0) \wedge (\bar{x}_1) \wedge (x_2)$
 "the only part of ϕ "
 "Rungs and Gadget Architecture"



IND. SET: INST: An undirected graph G , an integer k .

Q: $\exists S \subseteq V: |S| \geq k$ and $\bigwedge_{(u,v) \in E} (u,v) \notin E$?

Correctness (\Rightarrow): If ϕ is satisfiable, we can take any sat. assgt $a_1, \dots, a_n \in \{0,1\}^n$.

Define S_a by choosing the corresponding rung nodes

$$\begin{cases} x_i & \text{if } a_i = 1 \\ \bar{x}_i & \text{if } a_i = 0 \end{cases}$$

and choosing a node in each clause gadget for the literal making it true.

That literal and its rung partner are not connected by an edge, because all "crossing edges" go to the opposite literal. S_a is independent.

Converse (\Leftarrow): If S is indep. of size k , then S must have one node from each clause gadget, and the corresponding assgt to the rung nodes is enough to satisfy ϕ .

Example: $\phi = (\bar{x}_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2 \vee x_4) \wedge (x_2 \vee \bar{x}_3 \vee \bar{x}_4)$

$\bigwedge (W_0) \wedge (\bar{x}_1) \wedge (x_2) \wedge (x_3)$
 "the only part of ϕ "
 "Rungs and Gadget Architecture"

it. assgt $a_1, \dots, a_n \in \{0,1\}^n$
 rung nodes

The literal making it true.

connected by an edge, x_{n+1}

site literal. S_a is independent.
 not have one node from each clause gadget,
 rung nodes is enough to satisfy ϕ .

Overall Idea: Subsets S of V (obeying some constraints)

will be in 1-1 correspondence with assignments a to ϕ .

The chosen nodes in S from the rung nodes will define the assignment.

① Allocate rung (x_i, \bar{x}_i) for each i .
 ② For every clause $(x_i \vee \bar{x}_j \vee x_k)$ (eg.) allocate three nodes for a clause gadget and connect them in a triangle.

This completes $\forall(G): x_i, x_3$.

After step 2, already the max possible size of an ind. set in G is $n+m$. Take $k = n+m$.

③ For every literal $+x_i$ in any clause C_j , add an edge from x_i in C_j to \bar{x}_i in the rung. And for \bar{x}_i make edge go to x_i in the rung. This completes the construction, whose

complexity is $\text{poly}(n)$ time since this made just one "pass" over ϕ .

