Continuing the proof of the Cook-Levin theorem:

2 more families of clauses to add to φ_w, dealing with scanned cells: (transition clauses)

For every $q, r \in Q$ and every $a, b \in \Gamma$ such that

$$\delta(q, a) = (r, b, R),$$

and for all $0 \le t < p$ and $0 \le i < p$, add the clauses

$$(s_{q,t} \wedge h_{i,t} \wedge c_{a,i,t} \to s_{r,t+1})$$
$$(\text{"} \quad \text{"} \quad \text{"} \to h_{i+1,t+1})$$
$$(\text{"} \quad \text{"} \quad \text{"} \to c_{b,i,t+1})$$

For all q, r, a, b s.t.

$$\delta(q, a) = (r, b, L),$$

& for all $0 \le t < p$ and $1 \le i \le p$, add clauses

$$(s_{q,t} \wedge h_{i,t} \wedge c_{a,i,t} \to s_{r,t+1})$$
$$(\text{"} \quad \text{"} \quad \text{"} \to h_{i-1,t+1})$$
$$(\text{"} \quad \text{"} \quad \text{"} \to c_{b,i,t+1})$$

— End of construction of φ_w —

Observe that any satisfying truth assignment to φ_w must correspond to an accepting computation (complete tableau).

<u>Conversely</u> if $V(\$w\#y)$ accepts for some y, then we can set the variables of φ_w to make it true.

Further, φ_w is computable from w (via f) in ptime.

$\therefore$ $A \le_p$ CNF-SAT via f

$\therefore$ CNF-SAT is NP-hard

$\therefore$ CNF-SAT is NP-complete (since CNF-SAT $\in$ NP).

<u>Prop</u>: Let L be any NP-hard language. If $L \in P$, then $P = NP$.

<u>Proof</u>: Let $A \in NP$ be arbitrary. Then $A \le_p L$ by assumption. But by a previous theorem, if $L \in P$ then $A \in P$.

Thus $NP \subseteq P$ ($A \in NP$ is arbitrary)

$\therefore$ $P = NP$ (since $P \subseteq NP$ is clear). ▨

~~Theory of NP-completeness:~~

Showing problems to be NP-complete (or NP-hard).

<u>Def</u>: 3-SAT is the decision problem

<u>Instance</u>: A Boolean fmla φ in cnf with exactly 3 literals in each clause (3-cnf).

<u>Question</u>: Is φ satisfiable?

3-SAT is a restriction of CNF-SAT, 3-SAT $\in$ NP.

<u>Prop</u>: 3-SAT is NP-complete.

<u>Proof</u>. Given cnf fmla φ, we construct a 3-cnf fmla φ' (in p-time) s.t. φ is sat iff φ' is sat, thus CNF-SAT $\le_p$ 3-SAT

Construct φ' via "local replacement"

For each clause of φ, replace with 1 or more clauses to make φ'.

Let C be a clause of φ.

If C has 3 literals then no replacement necessary.

If $C = (\ell_1 \vee \ell_2)$ (2 literals)

then replace C by the 2 clauses

$(\ell_1 \vee x) \wedge (\ell_2 \vee \bar{x})$ where

x is a fresh variable (occurring nowhere else).

If $C = \ell_1$, then replace with

$(\ell_1 \vee x \vee y) \wedge (\ell_1 \vee x \vee \bar{y}) \wedge (\ell_1 \vee \bar{x} \vee y) \wedge (\ell_1 \vee \bar{x} \vee \bar{y})$

x, y are fresh vars.

replacement clauses satisfied iff C is satisfied.

$C = (\ell_1 \vee \cdots \vee \ell_k)$ $(k \geq 4)$

replace C (first) with

$(\ell_1 \vee \ell_2 \vee x) \wedge (\bar{x} \vee \ell_3 \vee \cdots \vee \ell_k)$

x fresh.

Repeat replacement with 2nd clause, continue until all clauses have 3 literals. //