

The Formulas of Cook's Theorem

- **Representation of the Computation**

$\#b_0\#b_1\#\dots\#b_{p(n)}$ where each b_i is an ID representing a whole computation state. The b_i consist of $p(n)$ symbols each. Each symbol represent one tape symbol. The symbol at the input position is a compound symbol representing the symbol on the tape, the state and the move.

- **Decision Variables for SAT**

$C_{i,X} \in \{true, false\}$ indicating that symbol X is in position i of the computation representation.

- **Formulas for SAT**

1. The formula fixes exactly one symbol for each position.

$$\bigwedge_i \left[\bigvee_X C_{i,X} \wedge \neg \left(\bigvee_{Y \neq X} (C_{i,X} \wedge C_{i,Y}) \right) \right]$$

2. The first ID is delimited by $\#$.

$$C_{0,\#} \wedge C_{p(n)+1,\#}$$

3. The first symbol is a valid start state and truthfully reflects the first tape symbol (Remember: it's an NTM)

$$C_{1,Y_1} \vee C_{1,Y_2} \vee \dots \vee C_{1,Y_k}$$

where Y_i are composites of the original first symbol and all possible valid start states.

4. The 2nd to n -th symbol truthfully reflect the remaining symbols on the tape in the start state.

$$\bigwedge_{2 \leq i \leq n} C_{i, a_i}$$

5. The unneeded symbols of the first ID are blank

$$\bigwedge_{n \leq i \leq p(n)} C_{i, B}$$

6. The final ID contains an accepting state

$$\bigvee_{p(n)(p(n)+1) < i < (p(n)+1)^2} \left(\bigvee_{X \in F} C_{i, X} \right)$$

7. ID n follows correctly from ID $n - 1$

$$\bigwedge_{p(n) < j < (p(n)+1)^2} \left(\bigvee_{W, X, Y, Z \text{ s.t. } f(W, X, Y, Z)} \left((C_{j-p(n)-2, W} \wedge C_{j-p(n)-1, X} \wedge C_{j-p(n), Y} \wedge C_{j, Z}) \right) \right)$$

here $f(W, X, Y, Z)$ is a predicate that is true for all symbol combinations such that if in the $n - 1$ st ID the symbol W is in position $j - 1$, X in position j and Y in position $j + 1$ then Z is in position j in the n -th ID. Recall that the composite symbols represent a tape symbol plus the move.

Note that the formulas generated in the transformation have length $O(p^2(n))$. To compute them, the TM only needs to be able to count up to $p^2(n)$. This can obviously be done in $O(\log(n))$ storage. So every language in \mathcal{NP} is log space reducible to SAT. Since a log space reduction cannot take more than polynomial, SAT is \mathcal{NP} -complete.