→ **RUDRATA CYCE** is **NP-Complete**

→ Every problem in **NP** $\leq_p$ **CIRCUIT SAT**

→ **CIRCUIT SAT** $\leq_p$ **3SAT**

→ **Approximation Algos.**
RUORATA CYCLE

INPUT: A directed Graph $G$

Sol: A directed cycle passing through every vertex exactly once
**Theorem:** Rudrata Cycle is NP-Complete

**Proof:**

1) **Rudrata Cycle** $\in$ NP
   
   (exercise)

2) Some NP-complete problem $\leq_p$ Rudrata Cycle

**3SAT problem**

**Input:** A 3-SAT formula

$$\bigwedge (x_1 \lor y_1 \lor \bar{z}_1) \land \cdots \land (z_u \lor w_v \lor p)$$

**Sol:** An assignment to variables satisfying
$3\text{SAT} \leq_p \text{RUUDRATA CYCLE}$

**3SAT instance:**
\[
\phi = (x \lor y \lor z) \land (\overline{y} \lor z \lor \overline{w}) \land (x \lor y \lor \overline{w})
\]

**Reduction:**

**RUUDRATA CYCLE**

*Input:* Directed Graph $G=(V,E)$

*Algorithm*

*Solution:*

**Such that:**

\(\exists a \text{ satisfying assignment to formula } \phi\)

\[\iff \exists \text{ a Rudrata cycle in graph } G.\]
Intuition:

\[ x \in \{0, 15\} \]

1 → Left to Right
or
0 → Right to Left
Clause C can be visited only if the tour is going L to R on x and y to L to R. 

\[
C = (x, y, \{v \in V \mid z \})
\]
CIRCUIT SAT

INPUT: 1) Circuit with AND/OR/NOT
        2) $n$ inputs

SOLUTION: An assignment of boolean values so that output = 1.
Theorem: Every problem in $NP$ \leq_p \text{Circuit SAT}.

Proof: For concreteness, consider factorization.
FACTORIZATION
Input: An n-bit number N
Sol: \( p, q, p \cdot q > 1 \) and \( p - q = N \)

CIRCUIT SAT
Input: Circuit C
Sol: \( x \land C(x) = 1 \)

BASIC IDEA:

NP problems have verification algs
All computation can be done on circuits
Proof. Factorization has an verification algo

\[ \text{VERIFY} \left( \text{INPUT} \begin{array}{c} \text{Number } N \end{array}, \text{Solution} \begin{array}{c} \text{Numbers } p, q \end{array} \right) \]

\{ \text{Check if } p \cdot q = N \}

\text{AND } p > 1 \}
Circuit

MULTIPLY

REDUCTION

I'm not sure what the specific details of the diagram are, but it appears to be related to some kind of reduction process, possibly involving multiplication and factorization.
CIRCUIT SAT

Input: A circuit $C$

Solution: An assignment $x$, s.t.

$C(x) = 1$

3SAT

Input: A 3SAT formula

Solution: A satisfying assignment.
<table>
<thead>
<tr>
<th><strong>INDSET:</strong></th>
<th><strong>CLIQUE</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>INPUT:</strong> Graph $G = (V, E)$ integer $K$</td>
<td><strong>INPUT:</strong> Graph $G' = (V', E')$ integer $K$</td>
</tr>
<tr>
<td><strong>SOL:</strong> An independent set of size $K$</td>
<td><strong>SOL:</strong> A clique of size $K$</td>
</tr>
<tr>
<td>INDOSET:</td>
<td>VERTEX COVER</td>
</tr>
<tr>
<td>----------</td>
<td>--------------</td>
</tr>
<tr>
<td><strong>INPUT:</strong> Graph $G = (V,E)$ integer $K$</td>
<td><strong>INPUT:</strong> Graph $G' = (V',E')$ integer $l$</td>
</tr>
<tr>
<td><strong>SOL:</strong> An independent set of size $K$</td>
<td><strong>SOL:</strong> A vertex cover of size $l$</td>
</tr>
</tbody>
</table>
**Approximation Algorithm**

Def: For a minimisation problem \( P \), \((\alpha > 1)\)

an algorithm is an \( \alpha \)-approximation algorithm

if \( \forall \) input \( I \) from \( P \)

\[
\text{ALG-OUTPUT}(I) \leq \alpha \cdot \text{OPT}(I)
\]

[For maximisation problem]

\[
\text{ALG-OUTPUT}(I) \geq \frac{1}{\alpha} \cdot \text{OPT}(I)
\]
**Minimum Vertex Cover** is NP-hard.

**INPUT:** Graph $G = (V, E)$

**SOL:** A vertex cover $SCV$ of smallest size

**Definition:** $S$ is a vertex cover if every edge is covered by $S$

i.e. $(uv) \in E \Rightarrow u \in S$ or $v \in S$ or both
A 2-Approximation AlgO

→ Pick a maximal matching $M$

(Maximal matching: keep adding edges until one can't)

Output $S = \{\text{both endpoints of edges in } M\}$

Purple: Maximal Matching edges

FACT 1: $S$ is a vertex cover.
Proof: $M$ is a maximal matching
Every edge overlaps some edge in $M$. 
Every edge overlaps some endpoint of \( M \) of a vertex in \( S \).

\[ |S| = 2 \quad \text{Maximal matching } M \]

[Fact: \text{OPTIMAL VERTEX COVER}] \Rightarrow [\text{Maximal Matching } M]

Proof: Every vertex cover has at least one vertex per edge of \( M \).
\( \text{OPT} \geq |M| \)

\(|S| = 2 \cdot |\text{Maximal Matching}| \leq 2 \cdot |\text{Optimal Vertex Cover}| \)