# Completeness: 3COLOR

Weston Dransfield

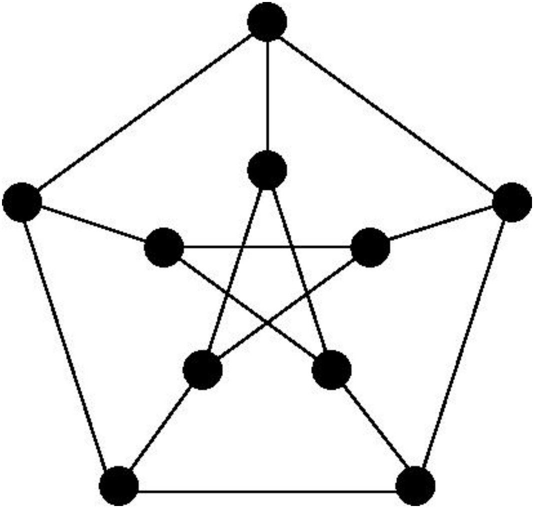
March 16, 2016

# Outline

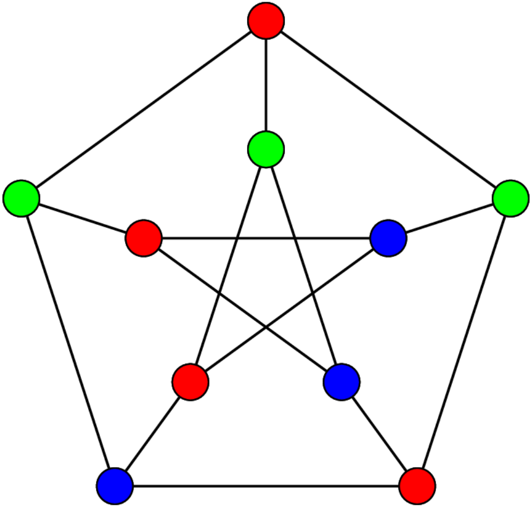
# Description

$\mathit{3COLOR} = \{\langle G \rangle \mid$ the nodes of G can be colored with three colors such that no two adjacent nodes are the same color $\}$

# Example



# Example



# The Problem

Is a given graph $G$ a member of ***3COLOR***?

# The Problem

Is a given graph $G$ a member of ***3COLOR***?

- This is tough to decide, but easy to verify!

# The Verifier

$V =$ "On input $\langle G, c \rangle$,

1. Check that c includes 3 colors.

# The Verifier

$V =$ "On input $\langle G, c \rangle$,

1. Check that c includes 3 colors.
2. Color each node of G as specified by c.

# The Verifier

$V =$ "On input $\langle G, c \rangle$,

1. Check that c includes 3 colors.
2. Color each node of G as specified by c.
3. For each node, check that each adjacent node is not the same color.

# The Verifier

$V =$ "On input $\langle G, c \rangle$,

1. Check that c includes 3 colors.
2. Color each node of G as specified by c.
3. For each node, check that each adjacent node is not the same color.
4. If all checks pass accept, otherwise reject."

# The Verifier

$V =$ "On input $\langle G, c \rangle$,

1. Check that c includes 3 colors.
2. Color each node of G as specified by c.
3. For each node, check that each adjacent node is not the same color.
4. If all checks pass accept, otherwise reject."

- Step 3 has largest time complexity of $O(n^2)$. 3COLOR is in NP because it can be verified in polynomial time.

# Constructing the Reduction

Construct a transformation $T$ from $3SAT$ to $3COLOR$.

# Constructing the Reduction

Construct a transformation $T$ from $3SAT$ to $3COLOR$.

1. Establish Truthiness

# Constructing the Reduction

Construct a transformation $T$ from $3SAT$ to $3COLOR$.

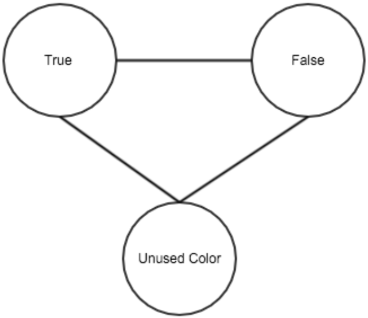
1. Establish Truthiness
2. Force variables to be true or false

# Constructing the Reduction

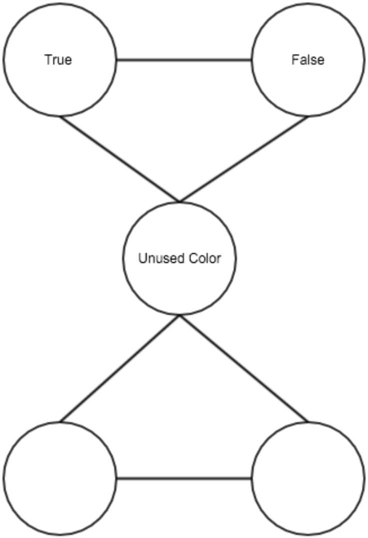
Construct a transformation $T$ from $3SAT$ to $3COLOR$.

1. Establish Truthiness
2. Force variables to be true or false
3. Use these subgraphs to create a graph that is 3 colorable iff the statement is satisfiable

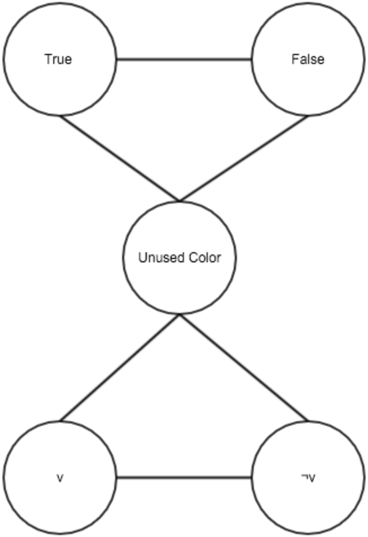
# Constructing the Reduction - Truthiness



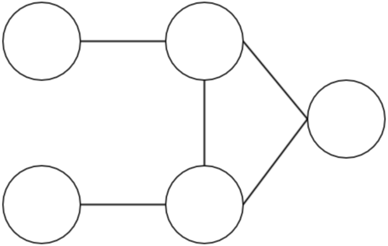
# Constructing the Reduction - Variables



# Constructing the Reduction - Variables



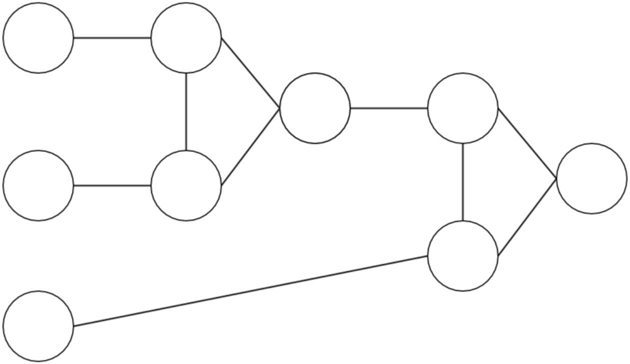
# Constructing the Reduction - OR



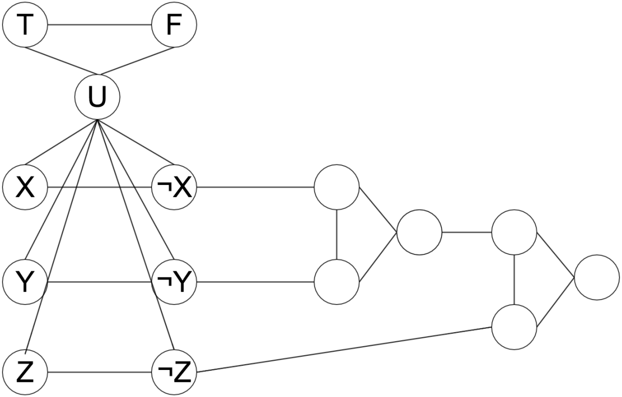
Output node is colored false if both input nodes are colored false

- Need to attach to truthiness gadget

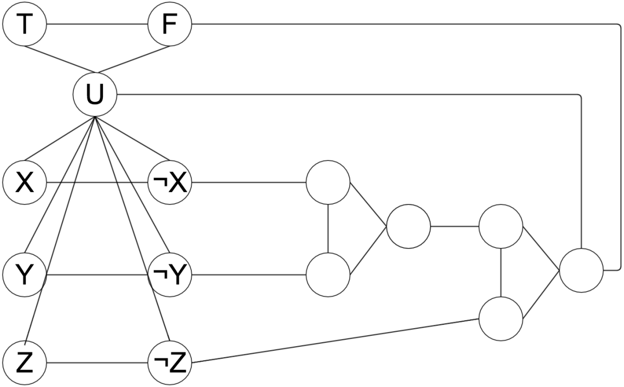
# Constructing the Reduction - OR

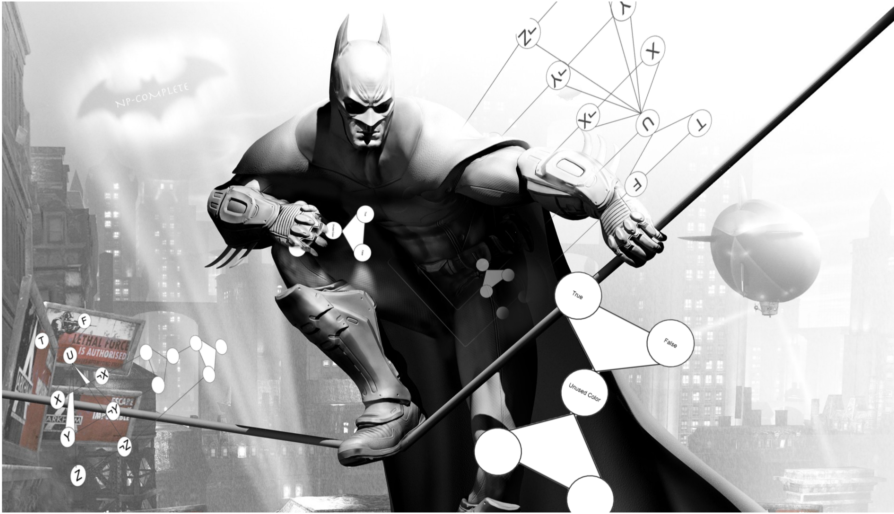


# Constructing the Reduction - Clause



# Constructing the Reduction - Clause




NP-COMPLETE
LETHAL FORCE
IS AUTHORISED
True
False
Unused Color

# Transformation

Transform expression $S$ to graph $G_s$ $T =$ "On input $\langle S \rangle$,

1. Construct the truthiness subgraph $T$

# Transformation

Transform expression $S$ to graph $G_s$ $T =$ "On input $\langle S \rangle$,

1. Construct the truthiness subgraph $T$
2. For each clause in $S$ add a 3 way OR gate subgraph $O_i$

# Transformation

Transform expression $S$ to graph $G_s$ $T =$ "On input $\langle S \rangle$,

1. Construct the truthiness subgraph $T$
2. For each clause in $S$ add a 3 way OR gate subgraph $O_i$
3. Connect the "output" node of $O_i$ to both the "false" and "unused" nodes of $T$

# Transformation

Transform expression $S$ to graph $G_s$ $T =$ "On input $\langle S \rangle$,

1. Construct the truthiness subgraph $T$
2. For each clause in $S$ add a 3 way OR gate subgraph $O_i$
3. Connect the "output" node of $O_i$ to both the "false" and "unused" nodes of $T$
4. For each variable in the $S$:

# Transformation

Transform expression $S$ to graph $G_s$ $T$ = "On input $\langle S \rangle$,

1. Construct the truthiness subgraph $T$
2. For each clause in $S$ add a 3 way OR gate subgraph $O_i$
3. Connect the "output" node of $O_i$ to both the "false" and "unused" nodes of $T$
4. For each variable in the $S$:

- Add nodes $v$ and $v_0$ connected by an edge

# Transformation

Transform expression $S$ to graph $G_s$ $T =$ "On input $\langle S \rangle$,

1. Construct the truthiness subgraph $T$
2. For each clause in $S$ add a 3 way OR gate subgraph $O_i$
3. Connect the "output" node of $O_i$ to both the "false" and "unused" nodes of $T$
4. For each variable in the $S$:

- Add nodes $v$ and $v_0$ connected by an edge
- Connect nodes $v$ and $v_0$ to the "unused" end of $t$

# Transformation

Transform expression $S$ to graph $G_s$ $T =$ "On input $\langle S \rangle$,

1. Construct the truthiness subgraph $T$
2. For each clause in $S$ add a 3 way OR gate subgraph $O_i$
3. Connect the "output" node of $O_i$ to both the "false" and "unused" nodes of $T$
4. For each variable in the $S$:

- Add nodes $v$ and $v_0$ connected by an edge
- Connect nodes $v$ and $v_0$ to the "unused" end of $t$
- Connect the corresponding node ($v_0$ or $v$) to one input of the clause's 3 way OR gate $O_i$"

# Example

$$(x \vee y \vee \neg z) \vee (\neg x \vee \neg y \vee z)$$

# Transformation - Forward

If boolean expression $S$ is satisfiable, $G_s$ is 3 colorable

- If $S$ is satisfialbe at least one literal in each clause is colored true.

# Transformation - Forward

If boolean expression $S$ is satisfiable, $G_s$ is 3 colorable

- If $S$ is satisfialbe at least one literal in each clause is colored true.
- Output of 3-way OR gate can be colored with true as the output. This leads to a valid 3 coloring

# Transformation - Backward

If graph $G_s$ is 3 colorable, $S$ is satisfiable

- A coloring of the graph forces the output of the 3-way OR gades to be colored true

# Transformation - Backward

If graph $G_s$ is 3 colorable, $S$ is satisfiable

- A coloring of the graph forces the output of the 3-way OR gades to be colored true
- For each clause in $S$ there must be at least one varaible colored true

# Transformation - Polynomial Time

- Truthiness nodes - $O(1)$

# Transformation - Polynomial Time

- Truthiness nodes - $O(1)$
- Variable T/F nodes - $O(n)$

# Transformation - Polynomial Time

- Truthiness nodes - $O(1)$
- Variable T/F nodes - $O(n)$
- $O(n)$ for $n$ clauses

# Transformation - Polynomial Time

- Truthiness nodes - $O(1)$
- Variable T/F nodes - $O(n)$
- $O(n)$ for $n$ clauses
- Overall - $O(n)$

# Sources

http://web.stanford.edu/class/archive/cs/cs103/cs103.1132/lectures/27/Small27.pdf
http://www.cs.princeton.edu/courses/archive/spring07/cos226/lectures/23Reductions.pdf