NP-complete Problems

HP, TSP, 3COL, 0/1IP

Dimitris Diamantis

$\mu\Pi\lambda\forall$

November 6, 2014

HAMILTON PATH is NP-Complete

Definition

Given an undirected graph $G(V, E)$, is there a *simple* path containing each vertex exactly once?

Claim

HAMILTON PATH is in NP

- Certificate: A list of nodes
- Certifier: Check that all nodes of G are indeed contained in the list exactly once. Check that for each two consecutive nodes in the list there is indeed an edge in G connecting them.

In order to prove that HAMILTON PATH is NP-Complete we will perform a reduction from 3SAT to HAMILTON PATH

3SAT to HAMILTON PATH

Definition

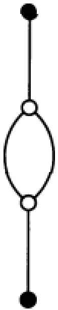
3SAT Given a finite set of clauses in CNF form, each containing exactly three literals, is there some truth assignment for the variables which satisfies all of the clauses?

In order to move from the domain of 3SAT to the domain of HAMILTON PATH, we have to respect the following:

- Each variable has a **choice** between values **true** and **false**
  - Choice gadget
- There is **consistency** of the variables' values, across all clauses. Meaning that all occurrences of x have the same value and all occurrences of $\neg x$ have the opposite value.
  - Consistency gadget
- Finally the clauses impose the **constraints** that must be satisfied in each 3SAT instance.
  - Constraint gadget

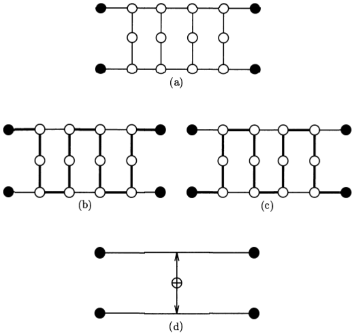
Choice gadget

- It is a subgraph of G. Imagine it being approached from above.
- A choice is made: which edge to follow. Let us assume that the left edge corresponds to **true** and the right edge corresponds to **false**
- It is connected to the rest of the graph via its endpoints, depicted as full dots.



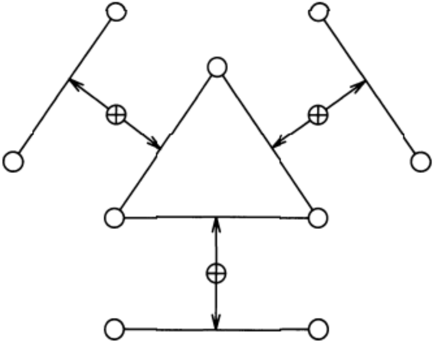
Consistency gadget

- It is a subgraph of G, connected to the rest of the graph via its endpoints (full dots)
- Assume that there is a HAMILTON PATH *not starting or ending* in this subgraph.
- Then it can only be traversed in two possible ways.
- Think of it as only two separate edges. When one is traversed, the other one is not. *Exclusive-OR* edges.



Constraint gadget

- This gadget represents the clauses of the 3SAT instance.
- Each side of the triangle represents a literal
- Assume that if the side of a triangle is chosen, then that literal is **false**.
- At least on literal has to be **true**. Otherwise every edge of the triangle is traversed. Hence, there is no HAMILTON PATH.



3SAT to HAMILTON

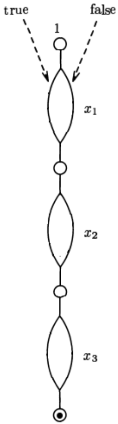
Example

Let ϕ be $(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)$

3SAT to HAMILTON

Example

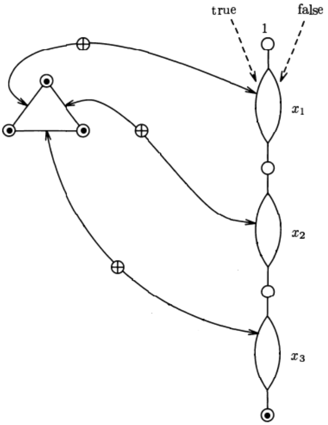
Let ϕ be $(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)$



3SAT to HAMILTON

Example

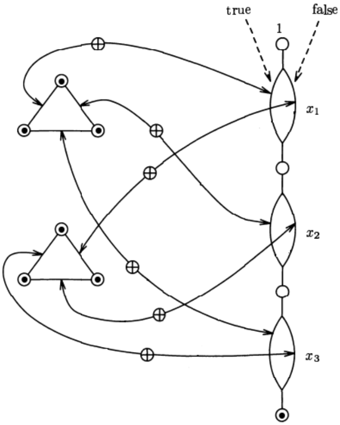
Let ϕ be $(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)$



3SAT to HAMILTON

Example

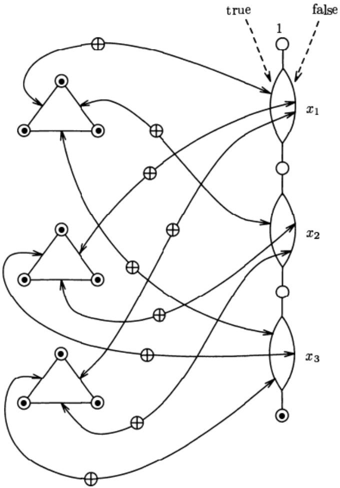
Let ϕ be $(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)$



3SAT to HAMILTON

Example

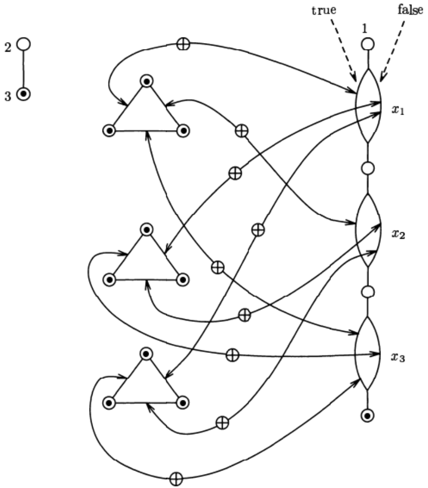
Let ϕ be $(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)$



3SAT to HAMILTON

Example

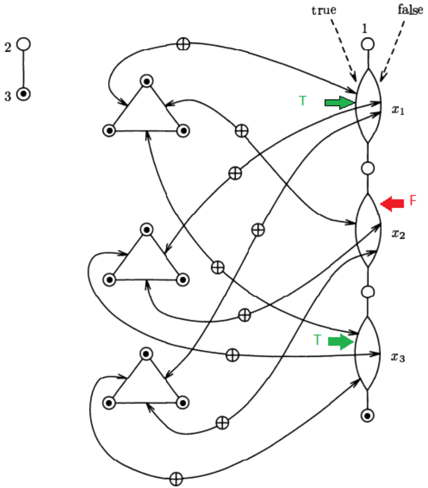
Let ϕ be $(x_1 \lor x_2 \lor x_3) \land (\neg x_1 \lor \neg x_2 \lor \neg x_3) \land (\neg x_1 \lor \neg x_2 \lor x_3)$



3SAT to HAMILTON

Example

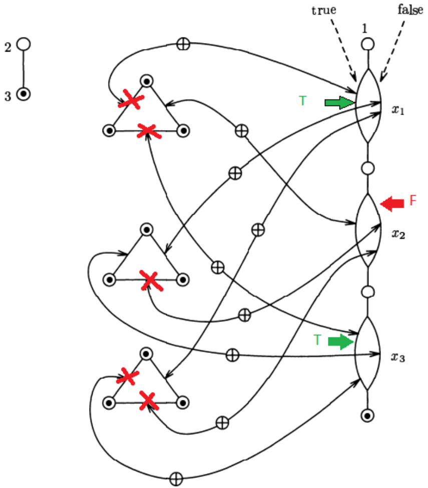
Let ϕ be $(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)$



3SAT to HAMILTON

Example

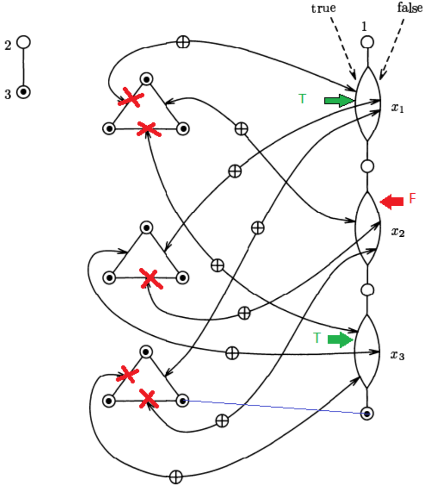
Let ϕ be $(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)$



3SAT to HAMILTON

Example

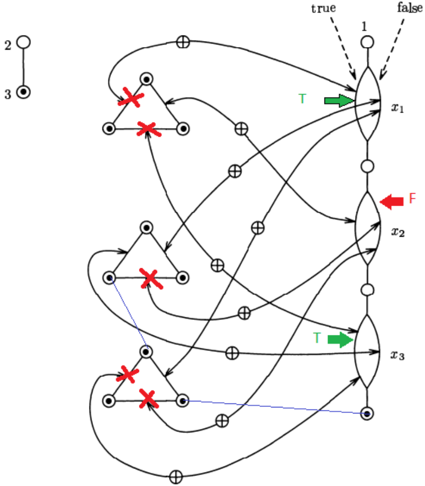
Let ϕ be $(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)$



3SAT to HAMILTON

Example

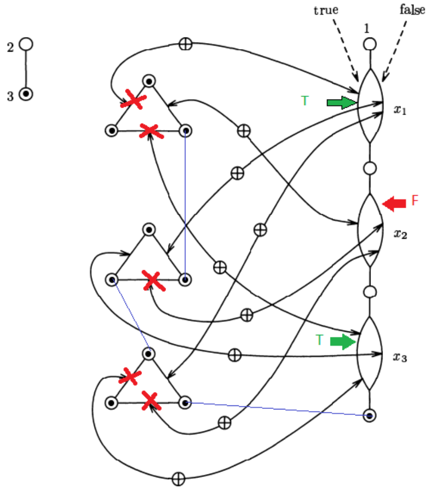
Let ϕ be $(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)$



3SAT to HAMILTON

Example

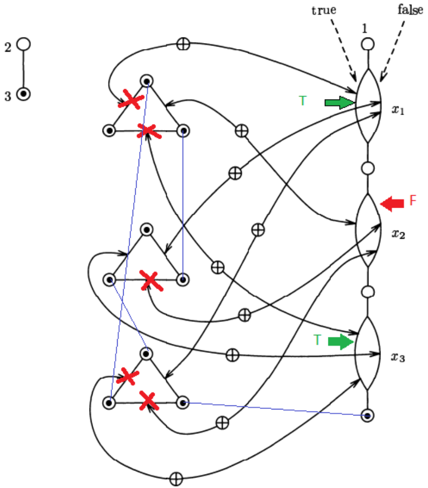
Let ϕ be $(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)$



3SAT to HAMILTON

Example

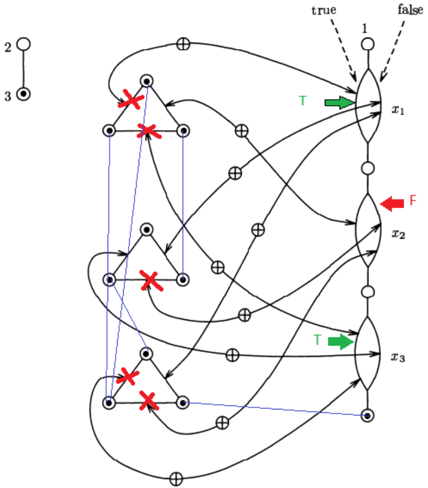
Let ϕ be $(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)$



3SAT to HAMILTON

Example

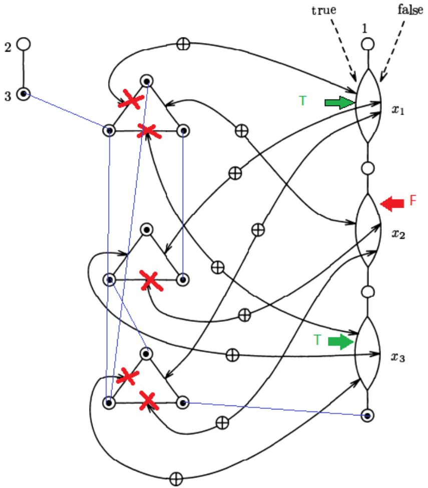
Let ϕ be $(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)$



3SAT to HAMILTON

Example

Let ϕ be $(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)$



3SAT to Hamilton

Claim

There is a truth assignment for instance ϕ of 3SAT if and only if there is a HAMILTON PATH in graph G

if direction

- Since there is a HAMILTON PATH, we traverse the choice gadget thus giving values to the literals of ϕ
- We then proceed to the constraint gadgets. For each triangle, we cannot traverse all 3 sides, otherwise there would be no HAMILTON PATH
- Since not all sides of each triangle are traversed, not all literals can be **false**. Hence all clauses are satisfied and the value assignment from the choice gadget is actually a *truth* assignment

Claim

There is a truth assignment for instance ϕ of 3SAT if and only if there is a HAMILTON PATH in graph G

else if direction

- Since there is a *truth* assignment for ϕ, we use that to traverse the choice gadget.
- For the constraint gadgets (triangles) we use the remaining (if any) sides of each triangle that are able to be traversed along with the clique that connects their nodes

Traversing the constraint gadgets as well gives us a HAMILTON PATH

TSP(D) is NP-COMPLETE

Definition

Given a list of cities, the distances between each pair of cities and a budget B, is there a route that visits each city exactly once and returns to the origin city(tour) and is of cost less or equal to B?

Claim

TSP(D) is in NP

- Certificate: A list of cities
- Certifier: Check that the list of cities is indeed a *tour*. Check that the tour's cost is less than or equal to B.

In order to prove that TSP(D) is NP-Complete we will perform a reduction from HAMILTON PATH to TSP(D)

HAMILTON PATH to TSP(D)

Given an instance of HAMILTON PATH, we construct an instance of TSP(D) as follows:

- We introduce n cities, one for each node on the given graph G
- If there is an edge connecting nodes i and j in G, then we set the distance of the corresponding cities to 1. Otherwise, we set it to 2.
- We set $B = n + 1$

Claim

Graph G has a HAMILTON PATH if and only if there is a tour of cost B or less in the TSP(D) instance

HAMILTON PATH to TSP(D)

Claim

Graph G has a HAMILTON PATH if and only if there is a tour of cost B or less in the TSP(D) instance

Suppose graph G has no HAMILTON PATHS.

- Then every tour must have at least two edges of distance 2. If this were not true, we could remove one edge of distance 2 and obtain a HAMILTON PATH, which is a contradiction. The total cost of this tour is at least $(n-2)*1+2*2=n+2>B$

Suppose G has a HAMILTON PATH

- This HAMILTON PATH along with the addition of an edge from the last node to the starting node, produces a tour
- This tour has a cost of at most $(n-1)*1+2=n+1=B$

k-COLORING

Definition

Given a graph G(V,E), can we color its vertices using k colors such that no two adjacent nodes have the same color?

For $k = 2$ the problem can be solved in linear time.

- Test if the graph is bipartite. How?
- Perform **depth-first** search (works with breadth-first search as well).
- Visit the nodes in a **pre-order** fashion.
- Color each child the opposite color from its parent.
- Return a 2-color scheme for the graph or an odd length cycle.

3-COLORING is NP-Complete

Claim

3-COLORING is in NP

- Certificate: For each node, a color from $\{0, 1, 2\}$
- Certifier: Check if for each edge (u, v), the color assigned to u is different than the one assigned to v.

In order to prove that 3-COLORING is NP-Complete, we will perform a reduction from NAESAT to 3-COLORING.

Definition

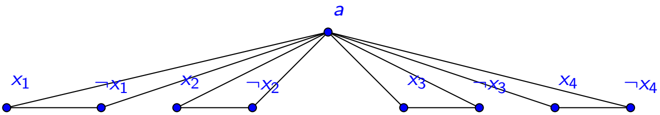
NAESAT Let ϕ be a *3-CNF* formula. Is there a truth assignment for ϕ where in no clause all three literals are **equal** in truth value?

NAESAT to 3-COLORING

- We are given a set of clauses $C_1, ..., C_m$, each with three literals involving variables $x_1, ..., x_n$. We shall construct a graph G that will be an instance of the 3-COLORING problem.
- For each literal we construct a **triangle** $[a, x_i, \neg x_i]$. All these triangles associated with our literals, have a common node: a

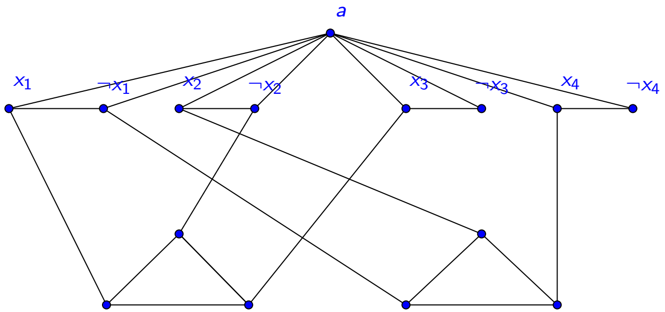
Example

Let ϕ be $(x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4)$



NAESAT to 3-COLORING

- Each clause C_i is also represented by a triangle $[C_i1, C_i2, C_i3]$ in our graph. Finally, there is an edge connecting C_{ij} with the node that represents the *jth* literal of C_i.



NAESAT to 3-Coloring

Claim

Graph G can be colored with 3 colors $\{0, 1, 2\}$ if and only if the given instance of NAESAT is satisfiable

else if direction.

- Suppose that a NAESAT truth assignment exists.
- We color node *a* with color 2.
- Literal Triangles: We color the 2 remaining nodes according to the truth assignment.
- Clause Triangles: We pick 2 literals in each clause triangle with different truth values and color them with different colors in $\{0, 1\}$. If literal is **true** then we assign color 1, if literal is **false** we assign color 0
- We color the third literal with color 2.

NAESAT to 3-Coloring

Claim

Graph G can be colored with 3 colors $\{0, 1, 2\}$ if and only if the given instance of NAESAT is satisfiable

if direction

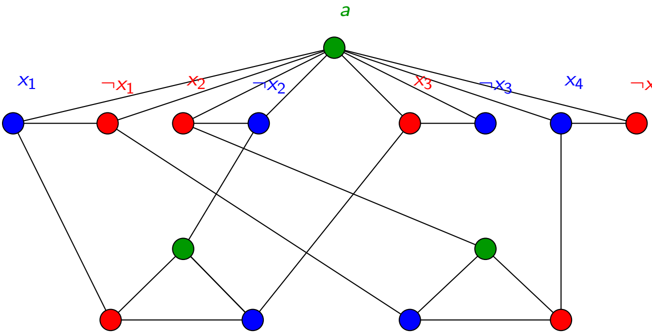
- Suppose that G is indeed 3 colorable.
- Literal Triangles: Node a takes color 2 and so in the *literal* triangles, x_i and $\neg x_i$ take colors 1 and 0 respectively.
- If x_i takes color 1, we think that the variable is **true**, otherwise it is **false**
- Clause Triangles: If all literals in a clause are **true**, then the corresponding clause triangle cannot be colored, since color 1 cannot be used. Same situation if all literals are **false**.

NASEAT to 3-Coloring

Example

Let ϕ be $(x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4)$

Truth assignment is $(x_1, x_2, x_3, x_4) = (T, F, F, T)$



0/1 IP is NP-Complete

Definition

Given a matrix A and a vector b, is there a vector x with values from $\{0, 1\}$ such that $Ax \geq b$?

Claim

0/1 IP is in NP.

- Certificate: Values for $x_1, ..., x_n$
- Certifier: Check that each value $x_i \in \{0, 1\}$ and that each constraint is met.

In order to show that 0/1 IP is NP-complete we will perform a reduction from 3SAT to 0/1 IP

3SAT to 0/1 IP

Definition

3SAT Given a finite set of clauses in CNF form, each containing exactly three literals, is there some truth assignment for the variables which satisfies all of the clauses?

From the given set of clauses we will construct a corresponding instance of 0/1 IP.

- Let $\phi = C_1, \ldots, C_k$ a set of clauses in 3SAT form, involving variables $x_1, \ldots, x_n$
- Corresponding matrix A will be a k x n matrix where

$$c_{ij} = \begin{cases} 1 & \text{if } x_j \in C_i \\ -1 & \text{if } \overline{x_j} \in C_i \\ 0 & \text{otherwise} \end{cases}$$

- We set $b_i = 1 -$ (the number of complemented variables in clause i)

3SAT to 0/1 IP

Hence for some clause C_j of the 3SAT instance, the possible corresponding inequalities are

$$\begin{cases} 1x_{j1} + 1x_{j2} + 1x_{j3} & \geq 1 \\ 1x_{j1} + 1x_{j2} - 1x_{j3} & \geq 0 \\ 1x_{j1} - 1x_{j2} - 1x_{j3} & \geq -1 \\ -1x_{j1} - 1x_{j2} - 1x_{j3} & \geq -2 \end{cases}$$

- If there is a truth assignment for ϕ then all possible inequalities for each clause are satisfied.
- If there is no truth assignment for ϕ, then assuming C_j is not satisfied, no possible inequality can be satisfied.
- If the inequality is satisfied then so is the corresponding clause in ϕ
- If the inequality is not satisfied then neither is the clause in ϕ