NP-Completeness of HAMPATH

Stepan Kuznetsov

Discrete Math Bridging Course, HSE University

The Hamiltonian Path Problem

- We consider the **Hamiltonian path** problem for directed graph: given a directed graph and two vertices, s and t, does there exist a path from s to t which traverses each vertex exactly once.

The Hamiltonian Path Problem

- We consider the **Hamiltonian path** problem for directed graph: given a directed graph and two vertices, s and t, does there exist a path from s to t which traverses each vertex exactly once.
- We denote this problem by HAMPATH.

The Hamiltonian Path Problem

- We consider the **Hamiltonian path** problem for directed graph: given a directed graph and two vertices, s and t, does there exist a path from s to t which traverses each vertex exactly once.
- We denote this problem by HAMPATH.

Theorem

HAMPATH is NP-complete.

The Hamiltonian Path Problem

- The fact that HAMPATH belongs to NP is established by definition: the path itself is a hint, polynomial in size and polynomially checkable.

The Hamiltonian Path Problem

- The fact that HAMPATH belongs to NP is established by definition: the path itself is a hint, polynomial in size and polynomially checkable.
- NP-hardness is proved by backwards reduction:

$$\text{3-SAT} \leq_m^P \text{HAMPATH}.$$

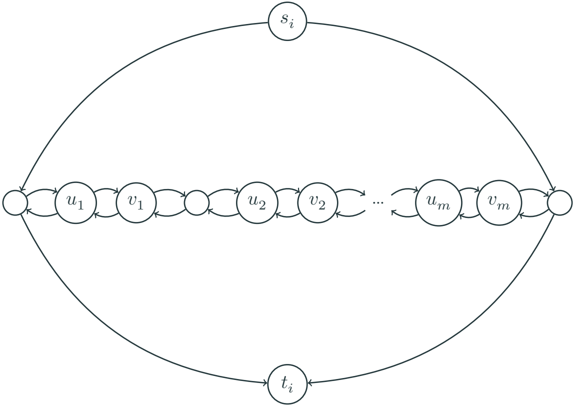
Backwards Reduction

Backwards reduction means that we need to construct a polynomial algorithm which takes a 3-CNF φ and constructs a directed graph G_φ with the following property:

G_φ has a Hamiltonian path from s to t
if and only if
φ is satisfiable.

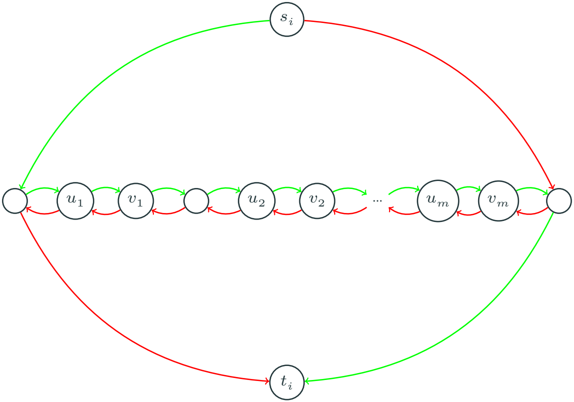
Gadget

Let φ include m clauses. For each *variable* x_i of φ we construct the following subgraph called *gadget:*



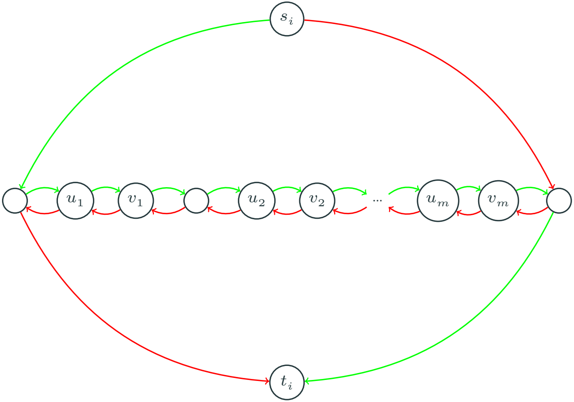
Gadget

In a Hamiltonian path, this gadget can be traversed, from s_i to t_i, only in the following two ways:



Gadget

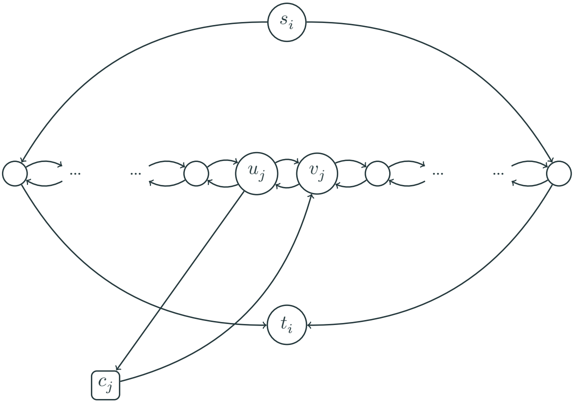
In a Hamiltonian path, this gadget can be traversed, from s_i to t_i, only in the following two ways:



Green means true, red means false.

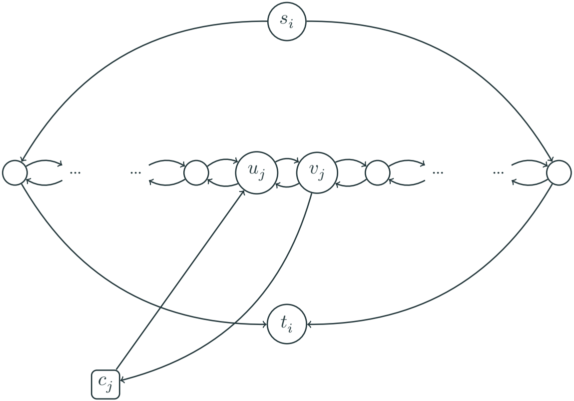
Modelling Clauses

Next, for each clause C_j we add a designated vertex c_j. If C_j includes x_i:



Modelling Clauses

If C_j includes $\neg x_i$:



Modelling Satisfiability

- If C_j includes x_i, then it can be visited, while traversing the i-th gadget, only on the green path ($x_i = 1$).

Modelling Satisfiability

- If C_j includes x_i, then it can be visited, while traversing the i-th gadget, only on the green path ($x_i = 1$).
- Dually, if C_j includes $\neg x_i$, then we can use only the red path ($x_i = 0$).

Modelling Satisfiability

- If C_j includes x_i, then it can be visited, while traversing the i-th gadget, only on the green path ($x_i = 1$).
- Dually, if C_j includes $\neg x_i$, then we can use only the red path ($x_i = 0$).
- Since C_j should be visited (exactly once), we have to satisfy at least one of its literals.

Modelling Satisfiability

- If C_j includes x_i, then it can be visited, while traversing the i-th gadget, only on the green path ($x_i = 1$).
- Dually, if C_j includes $\neg x_i$, then we can use only the red path ($x_i = 0$).
- Since C_j should be visited (exactly once), we have to satisfy at least one of its literals.
- Finally, we connect the gadgets: $s_1 = s$, $s_2 = t_1$, …, $s_n = t_{n-1}$, $t_n = t$.

Modelling Satisfiability

- If C_j includes x_i, then it can be visited, while traversing the i-th gadget, only on the green path ($x_i = 1$).
- Dually, if C_j includes $\neg x_i$, then we can use only the red path ($x_i = 0$).
- Since C_j should be visited (exactly once), we have to satisfy at least one of its literals.
- Finally, we connect the gadgets: $s_1 = s$, $s_2 = t_1$, ..., $s_n = t_{n-1}$, $t_n = t$.
- There is a Hamiltonian path from s to t in G_φ iff φ is satisfiable.