More on Hamiltonian Paths

Stepan Kuznetsov

Discrete Math Bridging Course, HSE University

Hamiltonian Path

- Let us comment a bit more on NP-completeness of the **Hamiltonian path** problem.

Hamiltonian Path

- Let us comment a bit more on NP-completeness of the **Hamiltonian path** problem.
- A Hamiltonian path is a path which traverses each **vertex** of a given graph exactly once.

Hamiltonian Path

- Let us comment a bit more on NP-completeness of the **Hamiltonian path** problem.
- A Hamiltonian path is a path which traverses each **vertex** of a given graph exactly once.
- Such a path can be either a **cycle** or have distinct start and end.

The Hamiltonian Path Problem

- On the previous lecture, we have sketched a proof of NP-completeness for the problem of existence of a Hamiltonian path in a **directed** graph, with fixed start s and end t.

The Hamiltonian Path Problem

- On the previous lecture, we have sketched a proof of NP-completeness for the problem of existence of a Hamiltonian path in a **directed** graph, with fixed start s and end t.
- This problem is called HAMPATH.

The Hamiltonian Path Problem

- On the previous lecture, we have sketched a proof of NP-completeness for the problem of existence of a Hamiltonian path in a **directed** graph, with fixed start s and end t.
- This problem is called HAMPATH.
- In our construction s had no incoming edges and t had no outgoing ones.

The Hamiltonian Path Problem

- On the previous lecture, we have sketched a proof of NP-completeness for the problem of existence of a Hamiltonian path in a **directed** graph, with fixed start s and end t.
- This problem is called HAMPATH.
- In our construction s had no incoming edges and t had no outgoing ones.
- Thus, (1) we do not have to fix s and t in advance; (2) by identifying s and t, we get NP-completeness for HAMCYCLE.

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:

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

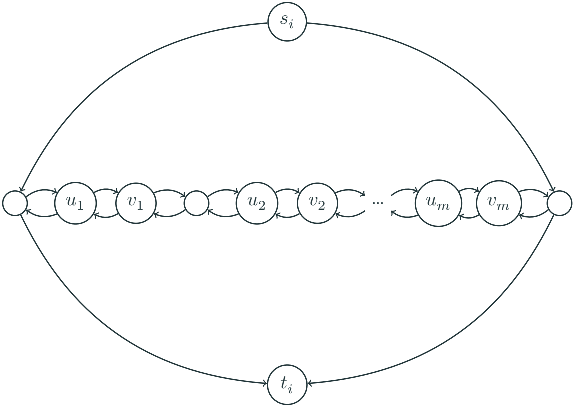
Backwards Reduction

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.

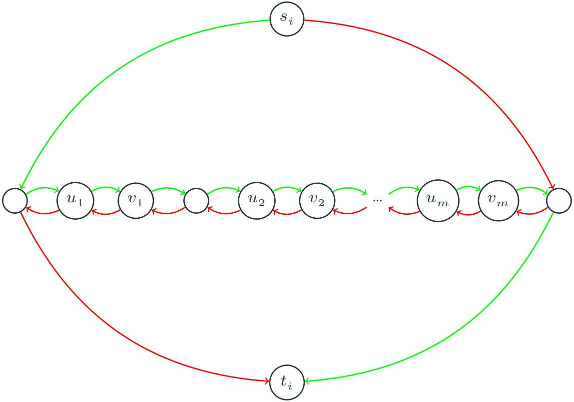
Gadget

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



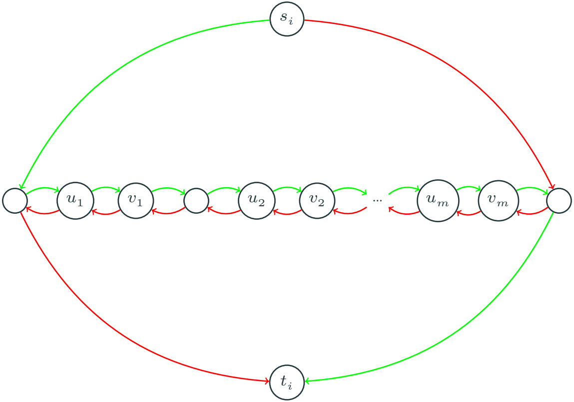
Gadget

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.

Encoding the 3-CNF

- Let $x_1, \ldots, x_n$ be **variables** used in our 3-CNF (Boolean formula) φ.

Encoding the 3-CNF

- Let $x_1, \dots, x_n$ be **variables** used in our 3-CNF (Boolean formula) φ.
- Let $C_1, \dots, C_m$ be the **clauses** of φ. (Each C_j is of the form like $x_2 \vee \neg x_6 \vee x_7$.)

Encoding the 3-CNF

- Let $x_1, \dots, x_n$ be **variables** used in our 3-CNF (Boolean formula) φ.
- Let $C_1, \dots, C_m$ be the **clauses** of φ. (Each C_j is of the form like $x_2 \vee \neg x_6 \vee x_7$.)
- The 3-CNF φ is modelled in our graph G_φ as follows:

$$\begin{array}{r}
\text{variables} \Rightarrow \text{gadgets} \\
\text{clauses } C_j \Rightarrow \text{extra vertices } c_j \\
\hline
\text{satisfying assignment} \Rightarrow \text{Hamiltonian path}
\end{array}$$

Connecting Gadgets

- Gagdets are “stacked” one upon another: $s_{i+1} = t_i$.

Connecting Gadgets

- Gagdets are “stacked” one upon another: $s_{i+1} = t_i$.
- In a Hamiltonian path, on the path from $s = s_1$ to $t = t_n$, we have to traverse all gadgets.

Connecting Gadgets

- Gagdets are “stacked” one upon another: $s_{i+1} = t_i$.
- In a Hamiltonian path, on the path from $s = s_1$ to $t = t_n$, we have to traverse all gadgets.
- The i-th gadget can be traversed either by the green, or by the red path.

Connecting Gadgets

- Gagdets are “stacked” one upon another: $s_{i+1} = t_i$.
- In a Hamiltonian path, on the path from $s = s_1$ to $t = t_n$, we have to traverse all gadgets.
- The i-th gadget can be traversed either by the green, or by the red path.
- This encodes the value of x_i in our assignment: $x_i = 1$ or $x_i = 0$.

Modelling Clauses

- For each clause C_j, we have an additional vertex c_j (the same for all gadgets!).

Modelling Clauses

- For each clause C_j, we have an additional vertex c_j (the same for all gadgets!).
- If C_j includes x_i, there are edges which allow visiting c_j on the green path of the i-th gadget: $u_j \to c_j \to v_j$.

Modelling Clauses

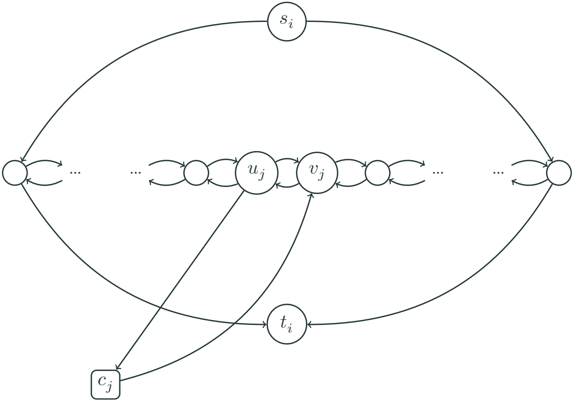
- For each clause C_j, we have an additional vertex c_j (the same for all gadgets!).
- If C_j includes x_i, there are edges which allow visiting c_j on the green path of the i-th gadget: $u_j \to c_j \to v_j$.
- For $\neg x_i$ in C_j, we can visit it on the red path: $v_j \to c_j \to u_j$.

Modelling Clauses

- For each clause C_j, we have an additional vertex c_j (the same for all gadgets!).
- If C_j includes x_i, there are edges which allow visiting c_j on the green path of the i-th gadget: $u_j \to c_j \to v_j$.
- For $\neg x_i$ in C_j, we can visit it on the red path: $v_j \to c_j \to u_j$.
- Thus, if (and only if) the assignment satisfies φ, the path traversing the gadgets can be augmented by visiting all c_j's, thus becoming a Hamiltonian path.

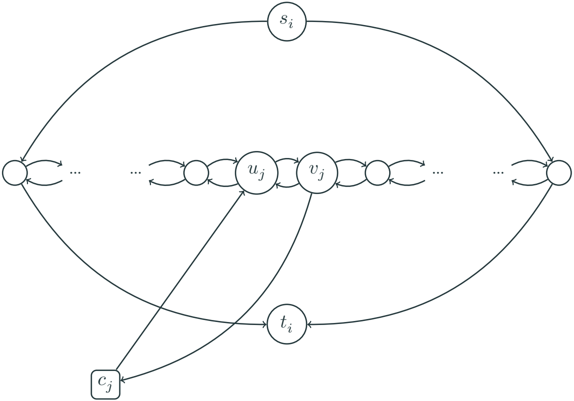
Modelling Clauses

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.

Undirected Graphs

- As already noticed, fixing s and t in HAMPATH is unnecessary, and identifying them gives NP-hardness of HAMCYCLE.

Undirected Graphs

- As already noticed, fixing s and t in HAMPATH is unnecessary, and identifying them gives NP-hardness of HAMCYCLE.
- However, it is crucial for our construction that the graph is **directed**.

Undirected Graphs

- As already noticed, fixing s and t in HAMPATH is unnecessary, and identifying them gives NP-hardness of HAMCYCLE.
- However, it is crucial for our construction that the graph is **directed**.
- If we make edges undirected, each c_j would be traversable on **both** red and green paths, thus, the graph will **always** have a Hamiltonian path.

Undirected Graphs

- Let us denoted the undirected versions of HAMPATH and HAMCYCLE problems by UHAMPATH and UHAMCYCLE, resp.

Undirected Graphs

- Let us denoted the undirected versions of HAMPATH and HAMCYCLE problems by UHAMPATH and UHAMCYCLE, resp.
- We show their NP-hardness by reducing from their undirected counterparts:

$$\text{3-SAT} \leq_m^P \text{HAMPATH} \leq_m^P \text{UHAMPATH}$$
$$\text{3-SAT} \leq_m^P \text{HAMCYCLE} \leq_m^P \text{UHAMCYCLE}$$

Undirected Graphs

- Let us denoted the undirected versions of HAMPATH and HAMCYCLE problems by UHAMPATH and UHAMCYCLE, resp.
- We show their NP-hardness by reducing from their undirected counterparts:

$$\text{3-SAT} \leq_m^P \text{HAMPATH} \leq_m^P \text{UHAMPATH}$$
$$\text{3-SAT} \leq_m^P \text{HAMCYCLE} \leq_m^P \text{UHAMCYCLE}$$

- The idea is to model direction by replacing each vertex with a more complicated subgraph.

Undirected Graphs

- Each vertex v is replaced by three vertices: v_{in}, v_{mid}, and v_{out}, connected as follows:

Undirected Graphs

- Each vertex v is replaced by three vertices: v_{in}, v_{mid}, and v_{out}, connected as follows:

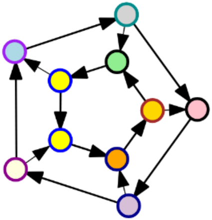
- Each edge $u \to v$ of the original graph is replaced by $u_{\mathrm{out}} - v_{\mathrm{in}}$.

Undirected Graphs

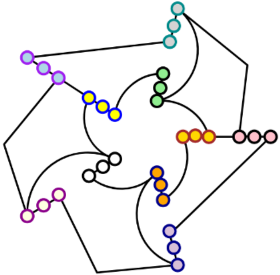
- Each vertex v is replaced by three vertices: v_{in}, v_{mid}, and v_{out}, connected as follows:

- Each edge $u \to v$ of the original graph is replaced by $u_{\text{out}} - v_{\text{in}}$.
- The Hamiltonian path can traverse v_{mid} only between v_{in} and v_{out}.

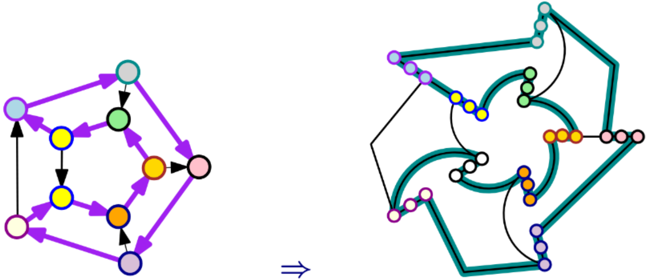
Undirected Graphs



$\Rightarrow$



Undirected Graphs



Undirected Graphs

- The construction transforms a directed graph G to an undirected graph G'.

Undirected Graphs

- The construction transforms a directed graph G to an undirected graph G'.
- Each Hamiltonian (directed) path in G maps onto a Hamiltonian (undirected) path in G'.

Undirected Graphs

- The construction transforms a directed graph G to an undirected graph G'.
- Each Hamiltonian (directed) path in G maps onto a Hamiltonian (undirected) path in G'.
- Conversely, from a Hamiltonian path in G' (from s_{mid} to t_{mid}) we can reconstruct a **directed** Hamiltonian path in the original graph G, taking only the 'mid'-vertices.

Undirected Graphs

- u_{mid} – u_{out} – v_{in} – v_{mid} maps to $u \to v$, while u_{mid} – u_{in} – v_{out} – v_{mid} does not (it is $u \leftarrow v$).

Undirected Graphs

- u_{mid} – u_{out} – v_{in} – v_{mid} maps to $u \to v$, while u_{mid} – u_{in} – v_{out} – v_{mid} does not (it is $u \leftarrow v$).
- Thus, the function $f\colon G \mapsto G'$ provides the necessary reductions:

$$\text{HAMPATH} \leq_m^P \text{UHAMPATH}$$
$$\text{HAMCYCLE} \leq_m^P \text{UHAMCYCLE}$$

Undirected Graphs

- u_{mid} – u_{out} – v_{in} – v_{mid} maps to $u \to v$, while u_{mid} – u_{in} – v_{out} – v_{mid} does not (it is $u \leftarrow v$).
- Thus, the function $f\colon G \mapsto G'$ provides the necessary reductions:

$$\text{HAMPATH} \leq_m^P \text{UHAMPATH}$$
$$\text{HAMCYCLE} \leq_m^P \text{UHAMCYCLE}$$

- On the next lecture, we shall discuss NP-completeness of yet another problem: 3-COLOR.