CS21
Decidability and Tractability

Lecture 20
February 22, 2021

Outline
• NP-complete problems: independent set, vertex cover, clique
• NP-complete problems: Hamilton path and cycle, Traveling Salesperson Problem
• NP-complete problems: Subset Sum

Vertex cover
• Definition: given a graph $G = (V, E)$, a vertex cover in $G$ is a subset $V' \subseteq V$ such that for all $(u, w) \in E$, $u \in V'$ or $w \in V'$
• A search problem:
  given $G$, find the smallest vertex cover
• corresponding language (decision problem):
  $VC = \{(G, k) : G$ has a VC of size $\leq k\}$

Vertex Cover is NP-complete
• Theorem: the following language is NP-complete:
  $VC = \{(G, k) : G$ has a VC of size $\leq k\}$
• Proof:
  – Part 1: $VC \in$ NP. Proof?
  – Part 2: $VC$ is NP-hard.
    • reduce from?

Vertex Cover is NP-complete
• We are reducing from the language:
  $IS = \{(G, k) : G$ has an IS of size $\geq k\}$
  to the language:
  $VC = \{(G, k) : G$ has a VC of size $\leq k\}$
• How are IS, VC related?
• Given a graph $G = (V, E)$ with $n$ nodes
  – if $V' \subseteq V$ is an independent set of size $k$
  – then $V - V'$ is a vertex cover of size $n - k$
• Proof:
  – suppose not. Then there is some edge with neither endpoint in $V - V'$. But then both endpoints are in $V$.
  contradiction.
Vertex Cover is NP-complete

- How are IS, VC related?

- Given a graph $G = (V, E)$ with $n$ nodes
  - if $V' \subseteq V$ is a vertex cover of size $k$
  - then $V-V'$ is an independent set of size $n - k$

- Proof:
  - suppose not. Then there is some edge with both endpoints in $V-V'$. But then neither endpoint is in $V'$. contradiction.

Clique

- Definition: given a graph $G = (V, E)$, a clique in $G$ is a subset $V' \subseteq V$ such that for all $u, v \in V'$, $(u, v) \in E$
- A search problem:
  - given $G$, find the largest clique
- corresponding language (decision problem):
  - $\text{CLIQUE} = \{(G, k) : G \text{ has a clique of size } \geq k\}$

Clique is NP-complete

Theorem: the following language is NP-complete:

$\text{CLIQUE} = \{(G, k) : G \text{ has a clique of size } \geq k\}$

- Proof:
  - Part 1: $\text{CLIQUE} \in \text{NP}. \text{ Proof?}$
  - Part 2: $\text{CLIQUE}$ is NP-hard.
    - reduce from?

Clique is NP-complete

- We are reducing from the language:

  $\text{IS} = \{(G, k) : G \text{ has an IS of size } \geq k\}$

  to the language:

  $\text{CLIQUE} = \{(G, k) : G \text{ has a CLIQUE of size } \geq k\}$. 

Clique is NP-complete

- How are IS, CLIQUE related?

- Given a graph $G = (V, E)$, define its complement $G' = (V, E' = \{(u,v) : (u,v) \notin E\})$
  - if $V' \subseteq V$ is an independent set in $G$ of size $k$
  - then $V'$ is a clique in $G'$ of size $k$

- Proof:
  - Every pair of vertices $u, v \in V'$ has no edge between them in $G$. Therefore they have an edge between them in $G'$.
Clique is NP-complete

- How are IS, CLIQUE related?
- Given a graph $G = (V, E)$, define its complement $G' = (V, E' = \{(u,v) : (u,v) \notin E\})$
  - if $V' \subseteq V$ is a clique in $G'$ of size $k$
  - then $V'$ is an independent set in $G$ of size $k$

- Proof:
  - Every pair of vertices $u,v \in V'$ has an edge between them in $G'$. Therefore they have no edge between them in $G$.

The reduction:
- given an instance of IS: $(G, k)$ $f$ produces the pair $(G', k)$
- $f$ poly-time computable?
- YES maps to YES?
  - IS of size $\geq k$ in $G$ $\Rightarrow$ CLIQUE of size $\geq k$ in $G'$
- NO maps to NO?
  - CLIQUE of size $\geq k$ in $G'$ $\Rightarrow$ IS of size $\geq k$ in $G$

Hamilton Path

- Definition: given a directed graph $G = (V, E)$, a Hamilton path in $G$ is a directed path that touches every node exactly once.

- A language (decision problem):
  $\text{HAMPATH} = \{(G, s, t) : G \text{ has a Hamilton path from } s \text{ to } t\}$

Theorem: the following language is NP-complete:
$\text{HAMPATH} = \{(G, s, t) : G \text{ has a Hamilton path from } s \text{ to } t\}$

- Proof:
  - Part 1: $\text{HAMPATH} \in \text{NP}$. Proof?
  - Part 2: $\text{HAMPATH}$ is NP-hard.
    - reduce from?

HAMPATH is NP-complete

- We are reducing from the language:
  $3\text{SAT} = \{ \varphi : \varphi \text{ is a 3-CNF formula that has a satisfying assignment } \}$

  to the language:
  $\text{HAMPATH} = \{(G, s, t) : G \text{ has a Hamilton path from } s \text{ to } t\}$

- We want to construct a graph from $\varphi$ with the following properties:
  - a satisfying assignment to $\varphi$ translates into a Hamilton Path from $s$ to $t$
  - a Hamilton Path from $s$ to $t$ can be translated into a satisfying assignment for $\varphi$

- We will build the graph up from pieces called gadgets that “simulate” the clauses and variables of $\varphi$. 
HAMPATH is NP-complete

- The variable gadget (one for each $x_i$):
  - $x_i$ true:
  - $x_i$ false:

February 22, 2021

HAMPATH is NP-complete

$\varphi = (x_1 \lor x_2 \lor \neg x_3) \land (\neg x_1 \lor x_4 \lor x_3) \land \ldots \land (\ldots)$

- How to ensure that all $k$ clauses are satisfied?
  - need to add nodes
    - can be visited in path if the clause is satisfied
    - if visited in path, implies clause is satisfied by the assignment given by path through variable gadgets

February 22, 2021

HAMPATH is NP-complete

$\varphi = (x_1 \lor x_2 \lor \neg x_3) \land (\neg x_1 \lor x_4 \lor x_3) \land \ldots \land (\ldots)$

- Clause gadget allows “detour” from “assignment path” for each true literal in clause
  - for clause 1
  - for clause 2

February 22, 2021

HAMPATH is NP-complete

$\varphi = (x_1 \lor x_2 \lor \neg x_3) \land (\neg x_1 \lor x_4 \lor x_3) \land \ldots \land (\ldots)$

- $f(\varphi)$ is this graph (edges to/from clause nodes not pictured)
  - $f$ poly-time computable?
  - # nodes = $O(km)$
HAMPATH is NP-complete

\[ \phi = (x_1 \lor x_2 \lor \neg x_3) \land (\neg x_4 \lor x_5) \land \ldots \]

- YES maps to YES?
- first form path from satisfying assignment.
- pick true literal in each clause and add detour

Case 1 (positive occurrence of \( v \) in clause):

- path must visit \( y \)
- must enter from \( x \) or \( z \)
- must exit to \( z \) (\( x \) is taken)
- \( x, c \) are taken, can’t happen

Case 2 (negative occurrence of \( v \) in clause):

- path must visit \( y \)
- must enter from \( x \) or \( z \)
- must exit to \( z \) (\( x \) is taken)
- \( x \) is taken, can’t happen

Undirected Hamilton Path

- HAMPATH refers to a directed graph.
- Is it easier on an undirected graph?

A language (decision problem):

\[ UHAMPATH = \{ (G, s, t) : \text{undirected } G \text{ has a Hamilton path from } s \text{ to } t \} \]