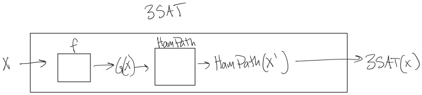
Theorem: Hamitonian Path is NP-Complete

Pf • Hamiltonian Path ∈ NP (see NP class)

- Hamiltonian Path ∈ NP-Hard

↳ $\boxed{\text{Show } 3\text{-}SAT \le_p \text{HamPath}}$ ↴

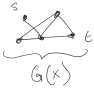

3SAT
x
f
G(x)
HamPath
HamPath(x')
3SAT(x)

↓ Need to prove this reduction exists

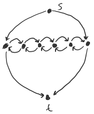
1. Describe f
2. Analyze runtime of f (polynomial)
3. Prove HamPath(G(x)) = Yes iff 3SAT(x) = Yes (correct output!)
   a) If HamPath(G(x)) = Yes, 3SAT(x) = Yes
   b) If 3SAT(x) = Yes, HamPath(G(x)) = Yes

1. Describe reduction from 3SAT to HamPath

$\underbrace{(u_1 \vee \neg u_3 \vee u_6) \wedge \ldots}_{x} \Rightarrow$



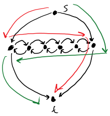
Idea: Use
Graph Gadget (a little graph that implements some functionality)

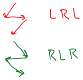


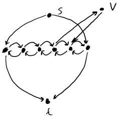
How many Hamiltonian Paths are there from s to t?

A) 0
B) 1
C) 2
D) >2

2 paths





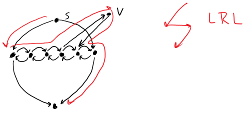


How many Hamiltonian Paths are there from s to t?

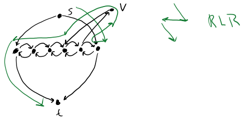
A) 0　　C) 2

B) 1　　D) >2

1 path



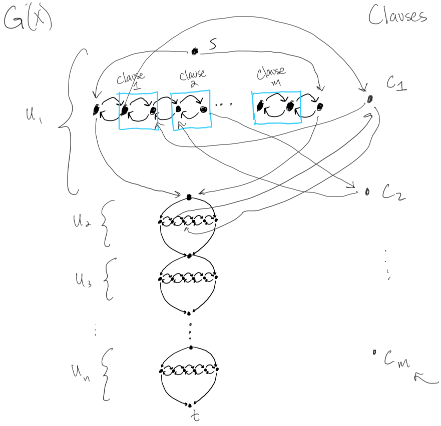
Reverse direction of edges to v:



One gadget for each variable u_i:

- LRL path ⟷ u_i = True
- RLR path ⟷ u_i = False

G(x)


Clauses
S
Clause 1
Clause 2
Clause m
C_1
C_2
C_m
u_1
u_2
u_3
u_n
t

$X = (u_1 \vee \neg u_2) \wedge (\neg u_2 \dots$

- u_1 is in clause 1
- $\neg u_1$ is in clause 2

- $\neg u_2$ is in clause1

⋮

← #of clauses

(Visual description of reduction)

Group Work:

Create graphs for the following formulas:

$x = (u_1 \vee u_2) \wedge (\neg u_1 \vee \neg u_2)$

$x = (u_1) \wedge (\neg u_1 \vee u_2) \wedge (\neg u_1 \vee \neg u_2)$

$x = (u_1 \vee u_2) \wedge (\neg u_1 \vee \neg u_2)$

$G(x)$

ex: $x = (u_1) \wedge (\neg u_1 \vee u_2) \wedge (\neg u_1 \vee \neg u_2)$ 3SAT(x) = No (no satisfying assignment)

$G(x)$

Because only 3 variables in a clause, only $8 \cdot \binom{n}{3} = O(n^3)$ possible clauses

2. Runtime: $|x| = (\text{size of 3SAT input}) = O(\overset{\#\text{ clauses}}{m}) = O(\overset{\#\text{variables}}{n^3})$

Runtime: need to create list of edges in $G(x)$

- # of vertices is $O(nm) = O(n^4)$
- # of edges is $O(n^8)$

Runtime: $O((n^3)^{8/3}) = O(|x|^{8/3})$

$\boxed{\text{Polynomial!}}$

3.

a) If HamPath(G(x)) = Yes, 3SAT(x) = Yes
b) If 3SAT(x) = Yes, HamPath(G(x)) = Yes

a) If HamPath(G(x)) = Yes, ∃ a path!

→ That path must go LRL or RLR through each gadget } assign $u_i = \{$ True / False based on LRL / RLR

→ It must hit each c_i vertex once.

- Must hit c_j from a gadget u_i and return to u_i (otherwise would skip vertices)
- Then that $u_i / \neg u_i$ causes c_j to be satisfied.

get a satisfying assigment → 3SAT(x) is Yes

b) If 3SAT(x) = Yes, there is a $\underbrace{\text{satisfying assignment}}$

- u_i = True or False
- Clause j has has a least one satisfying literal (Pick 1!)

If u_i = True ⇒ RLR
u_i = False ⇒ LRL

Pick $u_i / \neg u_i$ that satisfies clause j and jog out to c_j from gadget u_i. So there is a Hamiltonian Path in $G(x)$.

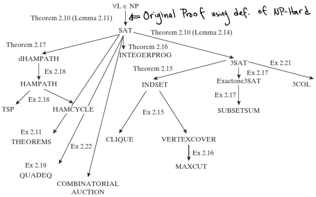
□ QED

Once you prove Π is NP-Hard, can use in place of 3-SAT:

Theorem: • If $\overset{\text{NP-Hard}}{\Pi} \leq_p Q$, then $Q \in$ NP-Hard:

$\begin{array}{c} A \\ \downarrow \\ B \end{array}$ = There is a polynomial reduction from A to B

2.4. The Web of Reductions 51


∀L ∈ NP
Theorem 2.10 (Lemma 2.11)
SAT
Theorem 2.10 (Lemma 2.14)
Theorem 2.17
dHAMPATH
Ex 2.18
HAMPATH
Ex 2.18
TSP
HAMCYCLE
Ex 2.11
THEOREMS
Ex 2.22
Ex 2.19
QUADEQ
COMBINATORIAL AUCTION
Theorem 2.16
INTEGERPROG
Theorem 2.15
INDSET
Ex 2.15
CLIQUE
VERTEXCOVER
Ex 2.16
MAXCUT
3SAT
Ex 2.17
Exactone3SAT
Ex 2.17
SUBSETSUM
Ex 2.21
3COL

⇐ Original Proof using def. of NP-Hard

Figure 2.4. Web of reductions between the **NP**-completeness problems described in this chapter and the exercises. Thousands more are known.

(From Arora + Barak)