Search Problems vs. Decision Problems

Decision Problem: problem with YES/NO answers

SAT: Given φ, does φ have a satisfying assignment?

VC: Given G, k, does G have a vertex cover S such that $|S| \le k$?

Search Problems:

FIND-SAT: Given Boolean formula φ, return a satisfying assignment (or NIL if none)

FIND-VERTEX-COVER: Given undirected G, return a minimum-size vertex cover

Search Problems vs. Decision Problems

Decision Problem: problem with YES/NO answers

SAT: Given φ, does φ have a satisfying assignment?

VC: Given G, k, does G have a vertex cover S such that $|S| \le k$?

Search Problems:

FIND-SAT: Given Boolean formula φ, return a satisfying assignment (or NIL if none)

FIND-VERTEX-COVER: Given undirected G, return a minimum-size vertex cover

SAT $\in$ P $\longleftrightarrow$ FIND-SAT $\in$ P

$\longleftarrow$: SAT(φ)
return FIND-SAT(φ) $\neq$ NIL

SAT $\le_p$ FIND-SAT

VC $\in$ P $\longleftrightarrow$ FIND-VC $\in$ P

$\longleftarrow$: VC(G, k)
$C \leftarrow$ FIND-VC(G)
return $|C| \le k$

VC $\le_p$ FIND-VC

Search Problems vs. Decision Problems

Decision Problem: problem with YES/NO answers

SAT: Given φ, does φ have a satisfying assignment?

VC: Given G, k, does G have a vertex cover S such that $|S| \le k$?

Search Problems:

FIND-SAT: Given Boolean formula φ, return a satisfying assignment (or NIL if none)

FIND-VERTEX-COVER: Given undirected G, return a minimum-size vertex cover

SAT $\in$ P $\longleftrightarrow$ FIND-SAT $\in$ P

$\longleftarrow$: SAT(φ)
return FIND-SAT(φ) $\neq$ NIL

SAT $\le_p$ FIND-SAT

VC $\in$ P $\longleftrightarrow$ FIND-VC $\in$ P

$\longleftarrow$: VC(G, k)
$C \leftarrow$ FIND-VC(G)
return $|C| \le k$

VC $\le_p$ FIND-VC

SAT is self-reducible

SAT $\in$ P $\longleftrightarrow$ FIND-SAT $\in$ P

SEARCH $\le_p$ DECISION

$\rightarrow$: We show FIND-SAT $\le_p$ SAT (SAT is $\overbrace{\text{self-reducible}}$)

FIND-SAT(φ) assume variables are $x_1, \dots, x_n$

```
FIND-SAT(φ)
    if SAT(φ)=NO then return NIL

    φ' ← φ
    A ← [NIL,...,NIL]
    for i=1 to n
        φ_T = φ' with x_i = T
        φ_F = φ' with x_i = F
        if SAT(φ_T)
            φ' ← φ_T
            A[i] = T
        else
            φ' ← φ_F
            A[i] = F
    return A
```

$\varphi = (x_1 \vee x_2 \vee x_3) \wedge (\sim x_1 \vee (\sim x_2 \wedge \sim x_3))$

$\varphi' = (x_1 \vee x_2 \vee x_3) \wedge (\sim x_1 \vee (\sim x_2 \wedge \sim x_3))$

$A \leftarrow [NIL, NIL, NIL]$

$\varphi_T \leftarrow (T \vee x_2 \vee x_3) \wedge (F \vee (\sim x_2 \wedge \sim x_3))$

$\varphi_F \leftarrow (F \vee x_2 \vee x_3) \wedge (T \vee (\sim x_2 \wedge \sim x_3))$

(can simplify)

$A \leftarrow [T, NIL, NIL]$

SAT is self-reducible

SAT $\in$ P $\longleftrightarrow$ FIND-SAT $\in$ P

SEARCH $\leq_p$ DECISION

$\rightarrow$: We show FIND-SAT $\leq_p$ SAT (SAT is self-reducible)

FIND-SAT(φ) assume variables are $x_1, \dots, x_n$

    if SAT(φ) = NO then return NIL

    $\varphi' \leftarrow \varphi$
    $A \leftarrow$ [NIL, ..., NIL]
    for $i = 1$ to n — n iterations
        $\varphi_T = \varphi'$ with $x_i = T$
        $\varphi_F = \varphi'$ with $x_i = F$ — poly
        if SAT(φ_T) — n calls to SAT in loop (+1 above)
            $\varphi' \leftarrow \varphi_T$
            $A[i] = T$
        else — poly
            $\varphi' \leftarrow \varphi_F$
            $A[i] = F$
    return A

$\varphi = (x_1 \vee x_2 \vee x_3) \wedge (\sim x_1 \vee (\sim x_2 \wedge \sim x_3))$

$\varphi' = (x_1 \vee x_2 \vee x_3) \wedge (\sim x_1 \vee (\sim x_2 \wedge \sim x_3))$
$A \leftarrow$ [NIL, NIL, NIL]

$\varphi_T \leftarrow (T \vee x_2 \vee x_3) \wedge (F \vee (\sim x_2 \wedge \sim x_3))$
$\varphi_F \leftarrow (F \vee x_2 \vee x_3) \wedge (T \vee (\sim x_2 \wedge \sim x_3))$
(can simplify)

$A \leftarrow$ [T, NIL, NIL]

INVARIANT: a) φ' is satisfiable
b) A + satisfying assignment for φ' satisfies φ
c) A has assignments for $x_1, \dots, x_{i-1}$

SAT is self-reducible

$SAT \in P \longleftrightarrow FIND\text{-}SAT \in P$

SEARCH $\leq_p$ DECISION

$\longrightarrow$: We show FIND-SAT $\leq_p$ SAT (SAT is $\overbrace{\text{self-reducible}}$)

$\varphi = (x_1 \vee x_2 \vee x_3) \wedge (\sim x_1 \vee (\sim x_2 \wedge \sim x_3))$

FIND-SAT(φ) ← assume variables are $x_1, \ldots, x_n$

```
FIND-SAT(φ)
    if SAT(φ) = NO then return NIL

    φ' ← φ
    A ← [NIL, ..., NIL]
    for i = 1 to n
        φ_T = φ' with x_i = T
        φ_F = φ' with x_i = F
        if SAT(φ_T)
            φ' ← φ_T
            A[i] = T
        else
            φ' ← φ_F
            A[i] = F
    return A
```

$\varphi' = (T \vee x_2 \vee x_3) \wedge (F \vee (\sim x_2 \wedge \sim x_3))$ $\quad i = 2$

What are φ_T and φ_F for the $i=2$ iteration?

Which one is satisfiable?

Vertex Cover is self-reducible

FIND-VERTEX-COVER $\le_P$ VERTEX-COVER:

FIND-VERTEX-COVER (G)

```
k ← n
while k > 0 and VERTEX-COVER(G, k-1)
    k ← k-1

C ← ∅
G' ← G
for each vertex v
    G'' ← G' with v and incident edges removed
    if VERTEX-COVER(G'', k-1)
        G' ← G''
        C ← C ∪ {v}
return C
```

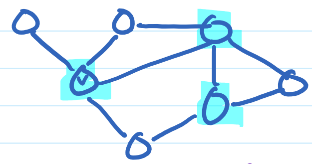
n calls to VERTEX-COVER

poly

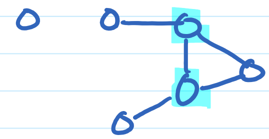
poly

n calls to VERTEX-COVER

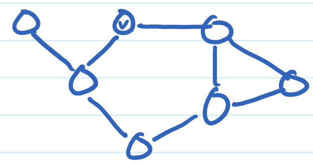
poly



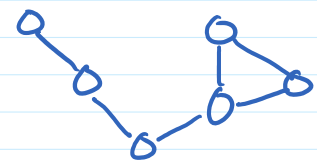
G has vertex cover of size 3 that includes v



G-v has a vertex cover of size 2



G has no vertex cover of size 3 that includes v



G-v has no vertex cover of size 2 (if it did, then that cover + v is a vertex cover of G of size 3)

DECISION ∈ P ⟷ SEARCH ∈ P