COMP250 -

More on decision problems

satisfiability:

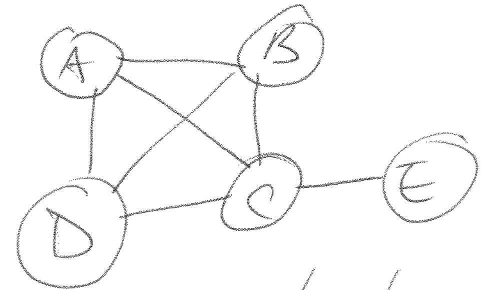
$(x_1 \vee x_2) \wedge (\neg x_1 \vee x_2)$

Conjunctive normal form

Is there an assignment of values to vars which makes formula true

3SAT (all OR's have 3 literals)

K-clique



Does the graph have a clique of size K?

what is max clique size in a graph G => is there a clique of size 3, 4, ... K, ... n

n vertices

SAT → Clique

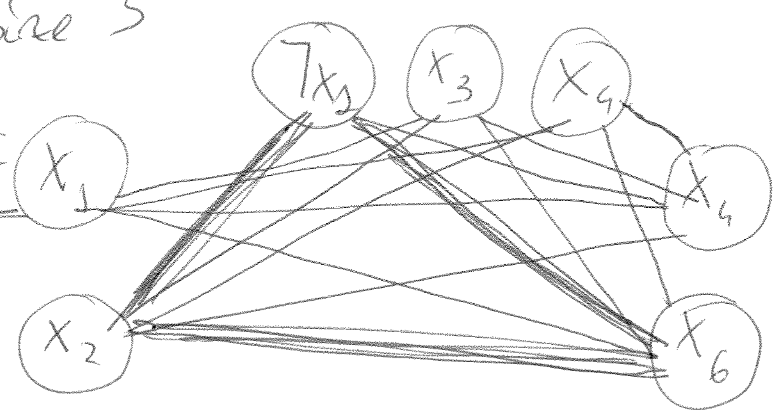
$\underbrace{(x_1 \vee x_2)}_{t} \wedge \underbrace{(\neg x_1 \vee x_3 \vee x_4)}_{t} \wedge \underbrace{(x_4 \vee x_6)}_{t}$ clause: C clauses = 3

clique of size 3

$x_2 = t$

$\neg x_1 = t \Rightarrow x_1 = f$

$x_6 = t$



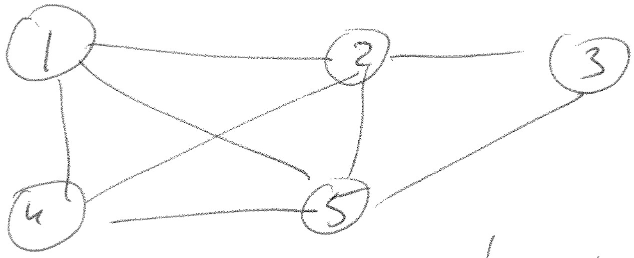
For every node → draw edges to all nodes corresp to vars in other clauses except its negation

No edge between nodes for same clause

clique of size C => satisfying assignment (otherwise - no)

~~lecture 35~~

Clique $\longrightarrow$ Sat



$\Rightarrow$ formula

$i, j \longrightarrow$ ith, jth vertex in the graph

Is there a clique of size K?

$X_{i,r} = \begin{cases} \text{true if ith node in the graph is in the clique of size } K \text{, in position } r \\ \text{false otherwise} \end{cases}$

- A node cannot be in 2 clique pos at same time

For all i, $r < s$, $r, s \in \{1, \dots K\}$ $\quad \neg X_{i,r} \vee \neg X_{i,s}$

$\neg(X_{i,r} \wedge X_{i,s})$

- If there is a clique, there is a node in pos $\underline{r}$

$\forall r \in \{1, \dots K\}$ $\quad X_{1r} \vee X_{2r} \vee \dots \vee X_{nr}$

- If there is <u>no edge</u> between 2 nodes i, j then they cannot both be in the clique.

For all $i < j$, $r \neq s$, $r, s \in \{1 \dots K\}$

$$\neg(X_{ir} \wedge X_{js}) = \neg X_{ir} \vee \neg X_{js}$$

$\rightarrow O(K^2 n) + O(K) + O(n^2 K^2) \Rightarrow O(n^2 K^2)$ clauses

over $O(nK)$ variables

Suppose we have an algorithm H s.t.

Program P ⟹ [H] ⟹ true if P stops on I, false otherwise
Input I ⟹

Turing 1950s ⟹ (P, I) ⟶ string of bits in a "canonical" way

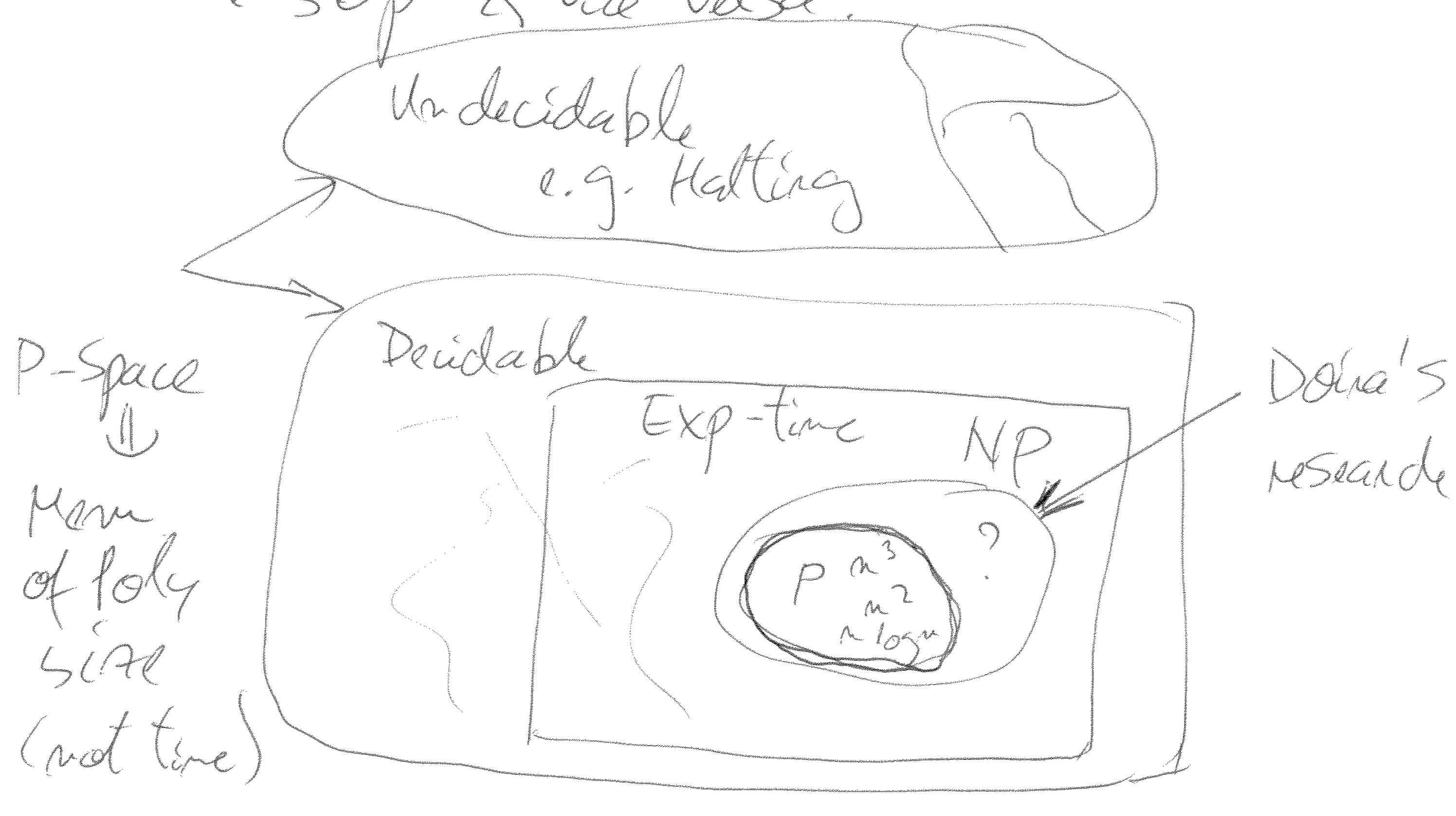
Write an algorithm D which calls H

Run D on input D

(Programs and inputs are strings of bits, so we can do this)

if (H(D, D)) <u>run forever</u> else exit

<u>Contradiction!</u> If H says program runs forever, we stop & vice versa.



Decision problems (the range)

- Is a number in an array? → Poly-time
- Is the mean of elems of an array < X? → Poly time
- Sat
- K-clique
- Hamilton cycle

] NP problems ⇒ Guess & check

Guess: space is huge!

Check: Poly time

- Is a Java program syntactically correct? (easy-ish)
- Does a Java program ever stop ~~on~~ (always)?

⇓

impossible! (undecidable)

Decision problem is decidable if there is an algorithm that will return right answer in finite time

Sat, compiling, ... ⇒ decidable

Not Halting problem