ADM Exercise Session 31.01.

Announcements:
- 7.2.: Ex.Sess. about written exam / Modulprüfung & Repetition
- 12.2.: Last Ex. Sess. Q&A about ADM I & Exam
- today (evening): example chapter of exam available

Topic: NP-completeness
- 2SAT
- Circuit Satisfiability

NP-completeness:

- formulate each optimization problem as decision problem

eg: $\exists$ a b-transshipment in D, u, b, c with cost $\leq c$?

formally: $P = (X, Y)$ $\quad X \approx$ instances of P

$Y \approx Y \subseteq X$ yes-instances

Algorithm for P: input: $x \in X$

output: yes if $x \in Y$, no if $x \notin Y$

note: for the most problems in the discret opt. the yes answer / optimal solution can be validated easily

Class NP: (non-determ. polyn. solvable)

- for every yes instance exists a certificate:
  - polynomial size
  - checkable in polynomial time

  $\}$ of the size of the problem

s.t. we can verify it

∇ $\exists$ problems where a certificate can not be checked in polyn. time

eg: chess, Go, Checker (general. on $n \times n$ boards, q: can white win the game)

On the other side: P solvable in polyn. time

$\Rightarrow \quad P \subseteq NP$

NP-C: class of all problems P with
- $P \in NP$
- all problems $P' \in NP$ transformieren to $P \Rightarrow$ hardest possible in NP

NP-C: SAT, 3-SAT, Stable set, Clique, TSP, 3Partition, Steiner Tree, ILP, Ham. Path, Tetris, MineSweeper, ...

(2) SAT: $\cdot$ n variables (boolean) $\quad x_1, \dots, x_n$

$\cdot$ m clauses $\quad C_i := (y_{i1} \vee y_{i2} \vee y_{i3} \vee \dots y_{i\dots n_i})$

$\underbrace{y_{ij}}_{\text{Literals}} \in \{x_k, \bar{x}_k \mid 1 \leq k \leq n\}$

Question: $\exists$ values for $x_1, \dots, x_n$ such that all clauses are fulfilled?

Lecture: $\cdot$ SAT $\in$ NP-C

$\cdot$ 3-SAT $\in$ NP-C (each clause contains ≤ 3 literals)

1-SAT: trivial

2-SAT: polyn. solvable

3-SAT: hard

Observation: is a literal in a clause set to false

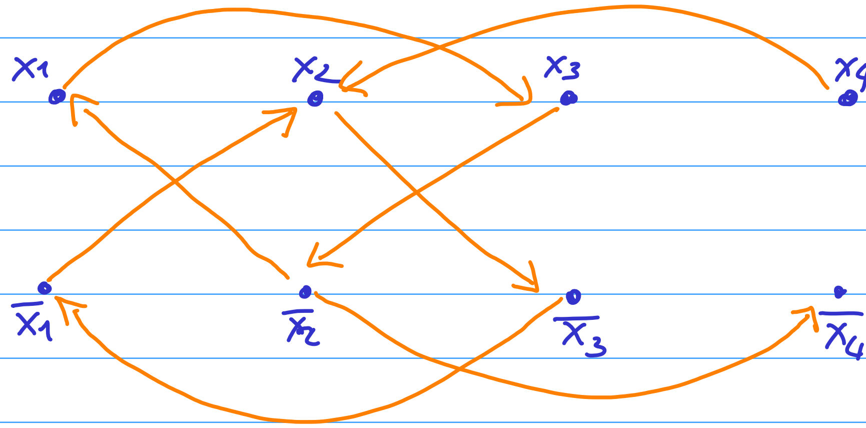
$\Rightarrow$ the other has to be true

$\leadsto$ model this as an implication graph

$V := \{x_i, \bar{x}_i \mid 1 \leq i \leq n\}$

$A := \{(v,w) : \exists \text{ clause } (\bar{v} \vee w)\}$ // implication for $v =$ true $\Leftrightarrow w =$ true

Example: $(x_1 \vee x_2) \wedge (x_2 \vee \overline{x_4}) \wedge (\overline{x_1} \vee x_3) \wedge (\overline{x_2} \vee \overline{x_3})$



Lemma 1: iff x_i and $\overline{x_i}$ are part of the same strongly connected component, then the 2SAT instance is not solvable

$\Rightarrow \exists$ $x_i \rightsquigarrow \overline{x_i}$ path and $\overline{x_i} - x_i$ path

Lemma 2: Let $C(y_i)$ be the strongly con. component of literal y_i, if no pair $x_i, \overline{x_i}$ is in the same strongly con. component, then the following values for x_i are:

a) well-defined

b) fulfill the 2SAT instance

$\exists$ $x_i - \overline{x_i}$ path $\Rightarrow$ $y_j = 0$ $\forall\ y_j \in C(x_i)$

$y_j = 1$ $\forall\ y_j \in C(\overline{x_i})$

$\exists$ $\overline{x_i} - x_i$ path $\Rightarrow$ $y_j = 1$ $\forall\ y_j \in C(x_i)$

$y_j = 0$ $\forall\ y_j \in C(\overline{x_i})$

all other variables:
- choose one without incoming edge
- set to true,
- follow implications

in example: no $x_i - \overline{x_i}$ path.

- x_4 = true $\Rightarrow$ x_2 = true $\Rightarrow$ x_3 = false $\Rightarrow$ x_1 = false

Variant: MAX SAT, fulfill as much clauses as possible
$\exists$ setting such that $\geq K$ clauses are fulfilled?

- MAX 1SAT: trivial, count x_i and $\bar{x}_i$ clauses
- MAX 2SAT: NP-C, 3SAT transforms to MAX 2SAT

3SAT $\longrightarrow$ MAX 2SAT:

- replace each clause $c = (\alpha \vee \beta \vee \gamma)$ with 10 clauses:

$$(\alpha) \wedge (\beta) \wedge (\gamma) \wedge (w_c) \wedge (\bar{\alpha} \vee \bar{\beta}) \wedge (\bar{\beta} \vee \bar{\gamma}) \wedge (\bar{\gamma} \vee \bar{\alpha}) \wedge (\alpha \vee \bar{w}_c) \wedge (\beta \vee \bar{w}_c) \wedge (\gamma \vee \bar{w}_c)$$

	(α)	(β)	(γ)	(w_c)	$(\bar{\alpha} \vee \bar{\beta})$	$(\bar{\beta} \vee \bar{\gamma})$	$(\bar{\gamma} \vee \bar{\alpha})$	$(\alpha \vee \bar{w}_c)$	$(\beta \vee \bar{w}_c)$	$(\gamma \vee \bar{w}_c)$	
• all 3 are true ($\alpha=\beta=\gamma$ =true)	1 +	1 +	1 +	?/1	0	0	0 +	1 +	1 +	1	= 7/10
• α = true $\beta=\gamma$ = false	1	0	0	0	1	1	1	1	1	1	= 7/10
• $\alpha=\beta$ = true γ = false	1	1	0	0	0	1	1	1	1	1	= 7/10
• $\alpha=\beta=\gamma$ = false	0	0	0	0	1	1	1	1	1	1	= 6/10

- $c = (\alpha \vee \beta \vee \gamma)$ is fulfilled iff $\geq$ 7/10 clauses are fulfilled w_c = "$\alpha=\beta=\gamma$ = true"
- c not fullfilled $\Rightarrow$ $\leq$ 6/10 clauses are fullfilled

$\Rightarrow$ Solving 3SAT per MAX 2SAT: 3SAT is yes Instance $\Leftrightarrow$ $\geq$ 7/10 clauses fulfillable in Max 2 SAT

Circuit Satisfiability

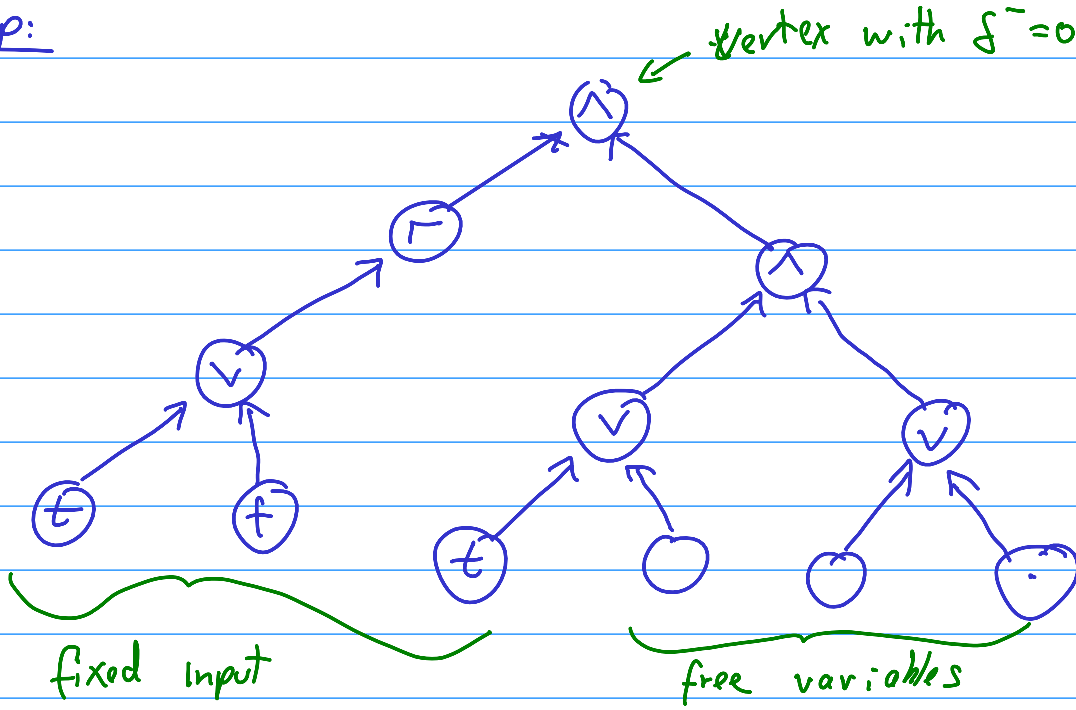
(first NP-c problem)

- digraph D, acyclic,
- Knotentypen: or, and, negation, boolean (variables), true, false

or	and	negation	boolean	true	false
$\vee$	$\wedge$	$\neg$	$\bigcirc$	t	f
$\delta^- = 2$		$\delta^- = 1$	$\delta^- = 0$		
$\delta^+ = 1$		$\delta^+ = 1$	$\delta^+ = 1$		

- Q: exists a setting (true/false) for boolean variables, such that the unique node with $\delta^- = 0$ has value of <u>true</u>

Bsp:



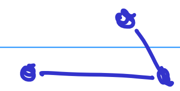
Lem: Ci. Sat. is NP-c

proof sketch:
- each $P_0 \in NP$ can be transformed to CS
- algorithm to check certificate as graph
- instance I is fixed input
- free variables is for the certificate

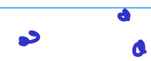
$\rightsquigarrow$ CS yes-instance $\Longleftrightarrow$ Z is a valid cert. for problem

Ex. Stable Set (set of nodes without edges/arcs)

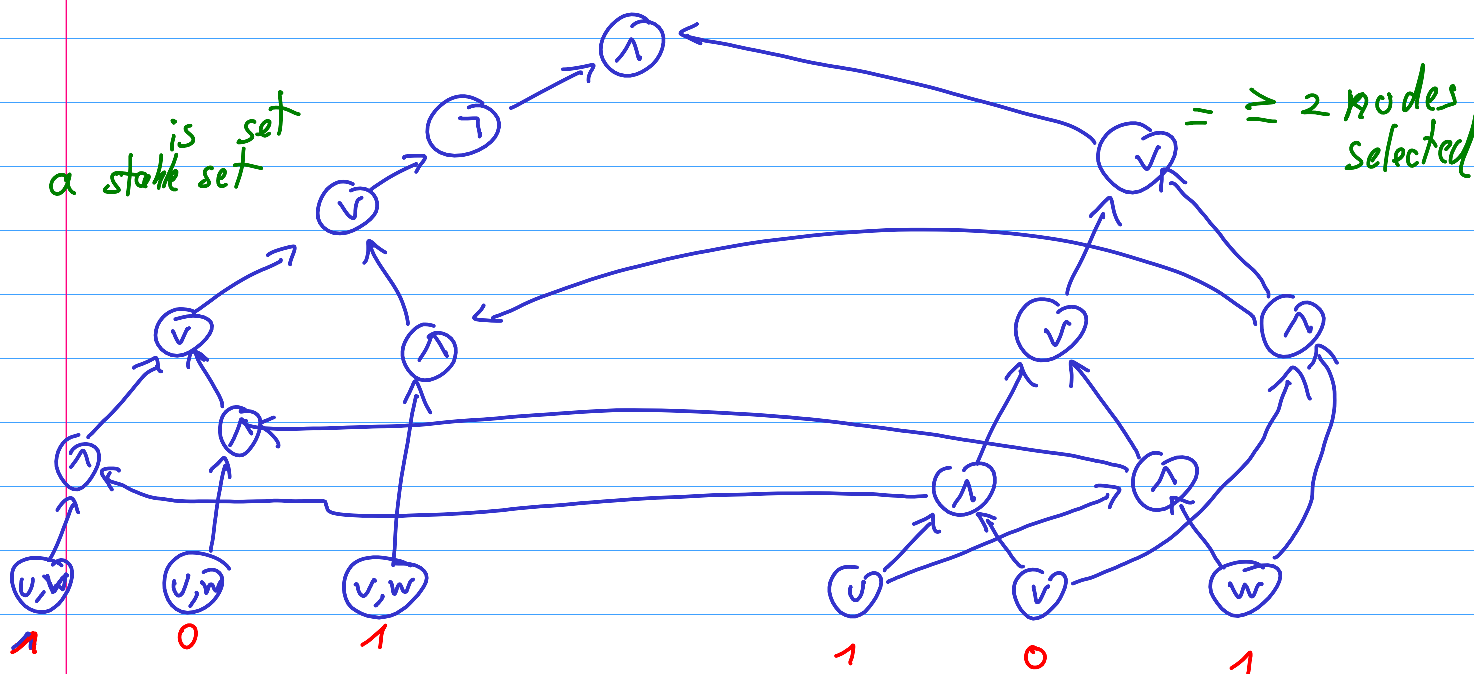
• $|V| = 3$

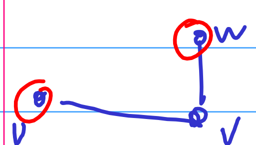


$\Rightarrow$ no for $k \geq 2$



$\Rightarrow$ yes for $k = 3$





stable set ≥ 2 nodes