Exercise: Reduction from 3-SAT to CSAT

(see textbook 10.3.4)

Given a CNF formula $E = C_1 \cdot C_2 \cdots C_k$
with each $C_i = \sum_{j=1}^{k_i} l_{ij}$,

we construct a 3-CNF formula F as follows.

For each clause C_i of E

1) if $C_i = (l)$ (i.e., a single literal)
introduce two new variables u, v, and replace
C_i by 4 clauses

$$(l + u + v) \cdot$$
$$(l + u + \bar{v}) \cdot$$
$$(l + \bar{u} + v) \cdot$$
$$(l + \bar{u} + \bar{v})$$

Since u, v appear in all 4 combinations, the 4 clauses can be satisfied only if l is true

2) if $C_i = (l_1 + l_2)$
introduce a new variable z, and replace
C_i by 2 clause

$$(l_1 + l_2 + z) \cdot$$
$$(l_1 + l_2 + \bar{z})$$

as in 1

3) if $C_i = (l_1 + l_2 + l_3)$, just leave it

4) if $C_i = (l_1 + l_2 + \cdots + l_m)$ with $m \geq 4$
introduce $y_1, y_2, \ldots, y_{m-3}$ and replace C_i by

$$(l_1 + l_2 + y_1) \cdot (l_3 + \bar{y}_1 + y_2) \cdot (l_4 + \bar{y}_2 + y_3) \cdots$$
$$+ (l_{m-2} + \bar{y}_{m-4} + y_{m-3}) \cdot (l_{m-1} + l_m + \bar{y}_{m-3})$$

- An assignment T satisfying E makes at least one literal of C_i true. Let it be l_j.

  Then, by making $y_1, \dots, y_{j-2}$ true and

  $y_{j-1}, \dots, y_{m-3}$ false,

  we satisfy all clauses replacing C_i.

  Thus we can extend T to satisfy F.

- Conversely, if T makes all l_j of C_i false, then not all new clauses can be satisfied.

  Why? each y_j can make at most 1 clause true, but there are $m-2$ clauses and

  $m-3$ y_j's.

The 3-CNF formula F is linear in E and can be constructed in linear time

We get: CSAT $<_{poly}$ 3-SAT

$\Rightarrow$ from CSAT NP-hard, we get 3-SAT NP-hard

We also know 3-SAT $\in$ P (since SAT $\in$ P)

$\Rightarrow$ 3-SAT is NP-complete

Exercise: 2SAT is in P

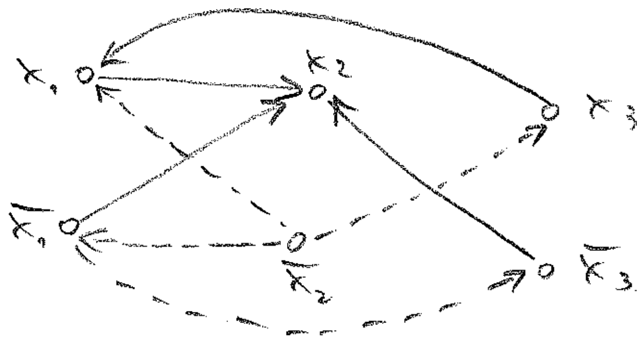
Idea: we show that 2SAT can be encoded as a graph reachability problem, and then use an algorithm for graph reachability

1) Encoding of 2SAT as a directed graph reachability problem

Let ϕ be an instance of 2SAT. We define a graph $G(\phi)$ as follows:
- one node for each variable and for each negated variable
- for each clause $\alpha \vee \beta$ two edges $\bar{\alpha} \to \beta$, $\bar{\beta} \to \alpha$

(note: $\alpha \vee \beta \equiv \bar{\alpha} \to \beta \equiv \bar{\beta} \vee \alpha$)

Example: $(x_1 + x_2) \cdot$
$(x_1 + \bar{x}_3) \cdot$
$(\bar{x}_1 + x_2) \cdot$
$(x_2 + x_3)$



Then ϕ is unsatisfiable iff there is a variable x such that $G(\phi)$ contains two paths $x \to \cdots \to \bar{x}$ and $\bar{x} \to \cdots \to x$

"$\Leftarrow$" Suppose that ϕ has a satisfying truth assignment T. Assume that $T(x) = true$ (a similar argument holds for $T(x) = false$)

Since $T(x) = true$ and $T(\bar{x}) = false$, and there is a path $x \to \cdots \to \bar{x}$, there must be an edge $\alpha \to \beta$ along this path with $T(\alpha) = true$ and $T(\beta) = false$.

However, since $\alpha \to \beta$ is an edge of $G(\Phi)$, it follows that $\bar{\alpha} + \beta$ is a clause of Φ. This clause is not satisfied by T_1 which is a contradiction.

"$\Rightarrow$" Let $G(\Phi)$ be a graph that does not contain any node x with
$$\begin{tikzcd} x \arrow[r] & \cdots \arrow[r] & \bar{x} \arrow[ll, bend left, dashed] \end{tikzcd}$$

We construct from such a graph a satisfying truth assignment T.
Repeat the following step as often as possible:
Choose a node α such that
- $T(\alpha)$ is not yet defined, and
- there is no path $\alpha \xrightarrow{*} \bar{\alpha}$

For every node β that is reachable from α

1) set $T(\beta) = \text{true}$
2) set $T(\bar{\beta}) = \text{false}$

(Note: 2) means to assign false to all predecessors of $\bar{\alpha}$) $\alpha \cdot \xrightarrow{*} \cdot \beta$, $\bar{\alpha} \cdot \xleftarrow{*} \cdot \bar{\beta}$

Observe: 1) the truth assignment T is well defined, i.e., we never have both $T(\beta) = \text{true}$ and $T(\bar{\beta}) = \text{true}$ or $T(\beta) = \text{false}$ and $T(\bar{\beta}) = \text{false}$

T would not be well defined, if we had both
$$\begin{tikzcd} \alpha \arrow[r, "*"] \arrow[dr, "*"'] & \beta \\ & \bar{\beta} \end{tikzcd}$$
(for some β)

But this cannot happen, since we would have
$$\begin{tikzcd} \alpha \arrow[r, "*"] \arrow[dr, "*"] & \beta \arrow[dl, dashed] \\ \bar{\alpha} & \bar{\beta} \arrow[l, dashed] \end{tikzcd}$$
Hence, we would have $\alpha \xrightarrow{*} \bar{\alpha}$

2) We assign to all nodes a truth value, since there is no path
$$\begin{tikzcd} \alpha \arrow[r, "*"] & \bar{\alpha} \arrow[l, bend left, "*"] \end{tikzcd}$$

3) The truth assignment satisfies all clauses of F, since each clause corresponds to an implication, and there is no $\alpha \rightleftarrows \bar{\alpha}$.