Reduction from **CIRCUIT SAT** to **3-SAT**

Let an arbitrary instance of **CIRCUIT SAT** be given by a Boolean circuit C. We construct the following instance $\varphi(C)$ of **SAT** (φ is in CNF with some clauses smaller than 3. The transformation into 3-CNF is obvious):

The formula $\varphi(C)$ uses all variables of C. Moreover, for each gate g of C, $\varphi(C)$ has a new variable g and the following clauses.

1. If g is a variable gate x: $(g \vee \neg x), (\neg g \vee x)$. $[g \leftrightarrow x]$
2. If g is a **true** (resp. **false**) gate: g (resp. $\neg g$).
3. If g is a NOT gate with a predecessor h: $(\neg g \vee \neg h), (g \vee h)$. $[g \leftrightarrow \neg h]$
4. If g is an AND gate with predecessors h, h': $(\neg g \vee h), (\neg g \vee h'), (g \vee \neg h \vee \neg h')$. $[g \leftrightarrow (h \wedge h')]$
5. If g is an OR gate with predecessors h, h': $(\neg g \vee h \vee h'), (g \vee \neg h'), (g \vee \neg h)$. $[g \leftrightarrow (h \vee h')]$
6. If g is also the output gate: g.

NAESAT

Not-all-equal SAT (NAESAT)

INSTANCE: Boolean formula φ in 3-CNF

QUESTION: Does there exist a truth assignment T appropriate to φ, such that the 3 literals in each clause do not have the same truth value?

Remark. Clearly **NAESAT** $\subset$ **3-SAT**.

Theorem

NAESAT *is* NP-*complete.*

NAESAT

Proof of NP-Hardness

Recall the Boolean formula $\varphi(C)$ resulting from the reduction of **CIRCUIT SAT** to **3-SAT**. For all one- and two-literal clauses in the resulting CNF-formula $\varphi(C)$, we add the same literal z (possibly twice) to make them 3-literal clauses.

The resulting formula $\varphi_z(C)$ fulfills the following equivalence:

$$\varphi_z(C) \in \textbf{NAESAT} \Leftrightarrow C \in \textbf{CIRCUIT SAT}.$$

"$\Rightarrow$" If a truth assignment T satisfies $\varphi_z(C)$ in the sense of **NAESAT**, so does the complementary truth assignment $\overline{T}$.
Thus, z is **false** in either T or $\overline{T}$ which implies that $\varphi(C)$ is satisfied by either T or $\overline{T}$. Thus C is satisfiable.

NAESAT

Proof of NP-Hardness (continued)

"$\Leftarrow$" If C is satisfiable, then there is a truth assignment T satisfying $\varphi(C)$. Let us then extend T for $\varphi_z(C)$ by assigning $T(z) =$ **false**.

By assumption, T is a satisfying truth assignment of $\varphi(C)$ and, therefore, also of $\varphi_z(C)$. Hence, in no clause of $\varphi_z(C)$ all literals are **false**.
It remains to show that in no clause of $\varphi_z(C)$ all literals are **true**:

(i) Clauses for **true**/**false**/NOT/variable gates contain z that is **false**.

(ii) For AND gates the clauses are: $(\neg g \vee h \vee z)$, $(\neg g \vee h' \vee z)$, $(g \vee \neg h \vee \neg h')$ where in the first two z is **false**, and in the third all three cannot be **true** as then the first two clauses would be **false**.

(iii) For OR gates the clauses are: $(\neg g \vee h \vee h'), (g \vee \neg h' \vee z)$, $(g \vee \neg h \vee z)$ where in the last two z is **false**, and in the first all three cannot be **true** as then the last two clauses would be **false**.