- Ch 7: NP-completeness (more examples)

→ Baker-Gill-Solovay: Methods that can't show $P \neq NP$

- Regular Languages

=

Last time:

- Surprised that a non-deterministic Turing machine can be "simulated" or "modelled" by a Boolean formula

⇝ Any $L \in NP$ has reduction to SAT (or 3SAT).

$$SAT = \left\{ w \in \Sigma_{SAT}^* \;\middle|\; w \text{ represents a satisfiable Boolean formula} \right\}$$

Similarly, 3SAT is NP-complete.

=

Today: SUBSET-SUM, PARTITION,

=

Eventually PSPACE. For NP, PSPACE, etc. there is a trick to produce complete languages...

=

$$L_{sneaky} = \left\{ \langle M, w, 1^t \rangle \;\middle|\; \begin{array}{l} M \text{ is an NTM,} \\ w \text{ input to } M \text{, and in time } t \\ M \text{ has at least one accepting path} \end{array} \right\}$$

SUBSET-SUM:

$$L = \left\{ \langle (n_1, \ldots, n_k ; m) \rangle \;\middle|\; \text{there is } I \subset \{1, \ldots, k\} \text{ s.t. } \sum_{i \in I} n_i = m \right\}$$

2#3#100##105 ∈ SUBSET-SUM

2#3#100##104 ∉ SUBSET-SUM

=

Claim: SUBSET-SUM is NP-complete

① SUBSET-SUM ∈ NP

idea: "guess" $I \in \{1, \ldots, k\}$

then check if $\sum_{i \in I} n_i = m$

Claim: L_{sneaky} is NP-complete.

$$1^t = \underbrace{11\cdots1}_{\text{length } t}$$

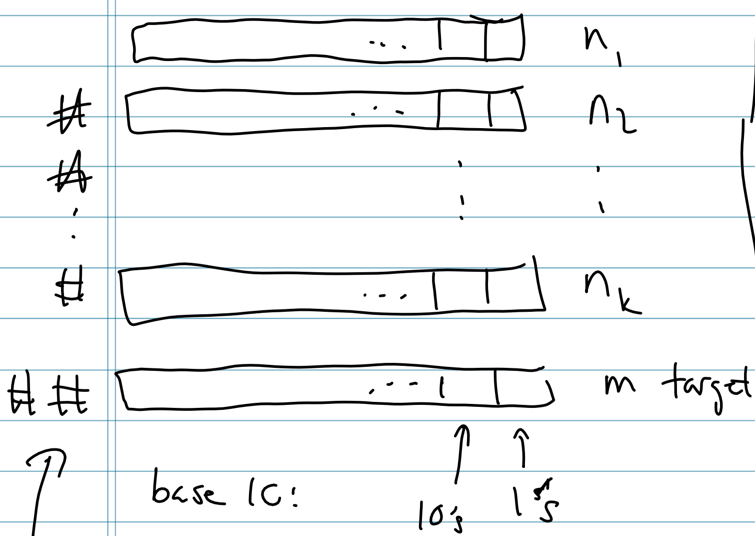
- $L_{sneaky} \in NP$
- $L \in NP$, $L \leq L_{sneaky}$

The above doesn't require anything except careful analysis

- Works in many contexts

=

has the answer "yes"



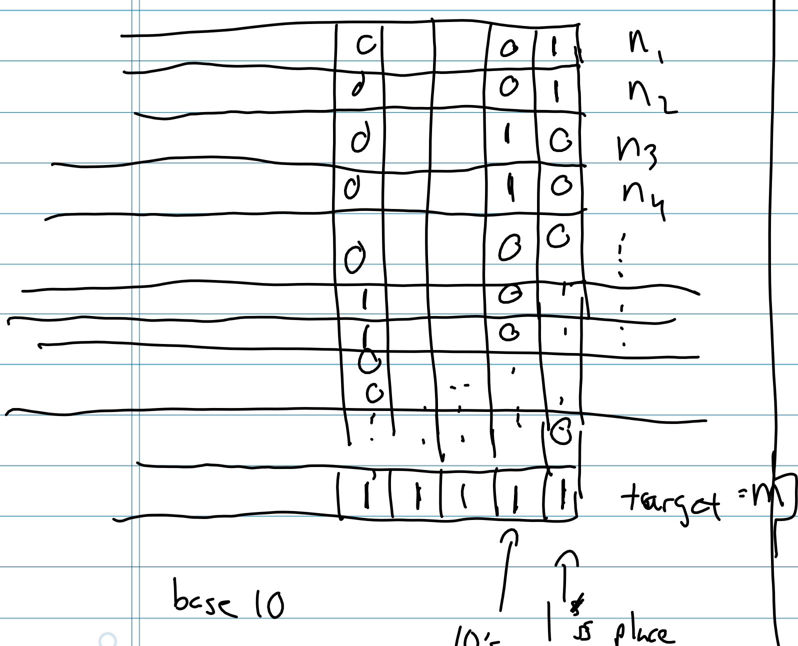
s.t. formula is true $\Leftrightarrow$

some subcollection of $n_1, \dots, n_k$ sum to m.

This entire string must have poly length

formula ⇝ 3CNF

Why a 3CNF



(2) If $L \in NP$ is there a reduction $L \leq$ SUBSET-SUM?

poly time

Usually $L \leq$ 3SAT so

3SAT $\leq$ SUBSET-SUM

we have a 3CNF formula

$(x_1 \wedge \neg x_2 \wedge x_3)$ OR $(x_2 \wedge \neg x_1 \wedge \neg x_3)$ OR( ) OR ... OR( )

Can we write a subset-sum question s.t. the formula is satisfiable iff the subset-sum question

Rough Idea!

Let's have n_1 represent x_1 is true

n_2 " x_1 is false

n_3 " x_2 is T

n_4 " x_2 is F

⋮

$f = (x_1$ OR x_2 OR $\neg x_3)$ AND

$(\neg x_1$ OR x_2 OR $x_3)$ AND

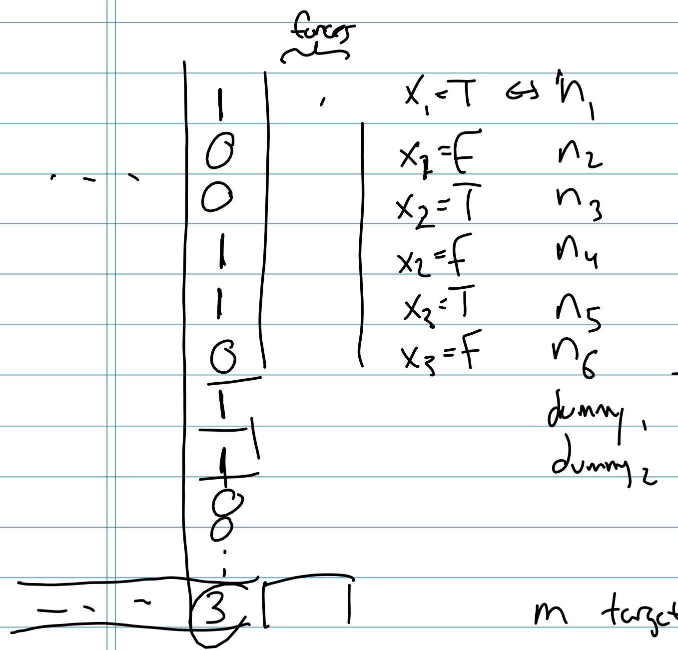
⋮

as string

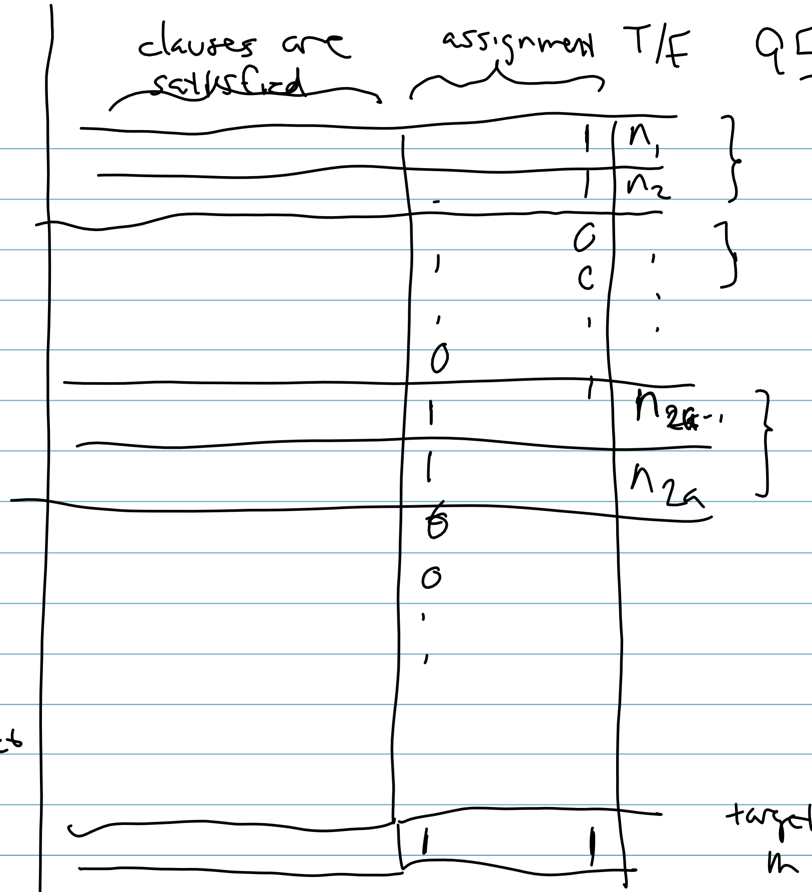
(X 1 OR X 2 OR ¬ X 3) AND . . .

95

x_1 OR $\neg x_2$ OR x_3 :



n_1 or n_4 or n_5 is in subset then sum: 1, 2, 3 can complete to target 3



formula involves $x_1, x_2, \ldots x_a$