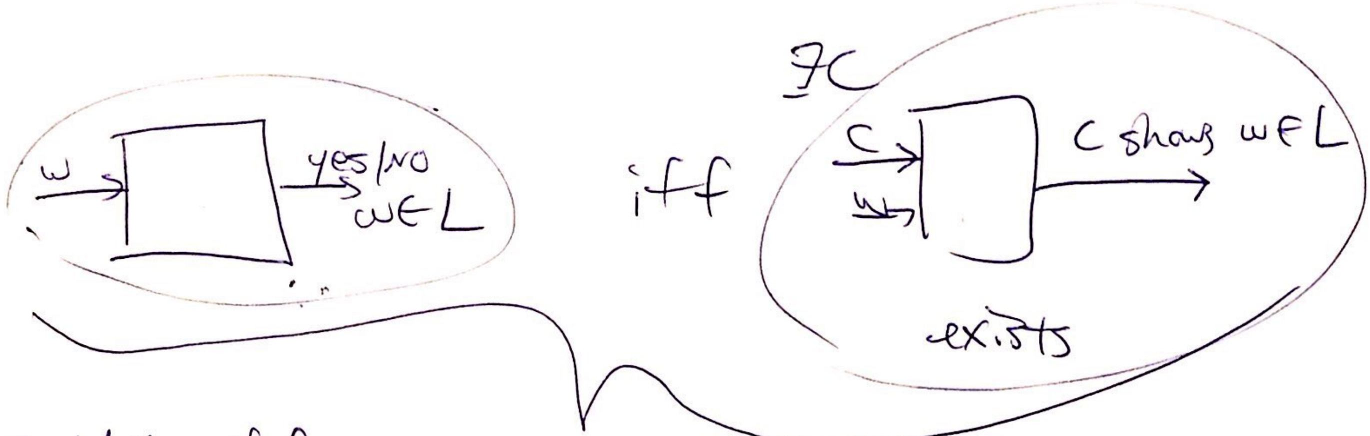

L ∈ NP iff

$\varphi \in$ Formula sat iff $\exists \rho$ $\varphi, \rho \in$ FORMULA_EVAL

$G, k \in$ CLIQUE iff $\exists C$ $G, k, C \in$ CLIQUE_EVAL

<u>Remember</u> : $P \subseteq NP$ because checker can ignore certificate.

3-dimensional matching.

3-DIM = { "S, T, U, R" | $R \subseteq S \times T \times U$ contains a subset that mentions each member of S, T, U exactly once }

The matching $M \subseteq R$ that uses each member of S, T, U ex. once is the certificate.

The problem check-match = { "S, T, U, R, M" | $M \in R$ solves the problem } $\in P$

"S, T, U, R" $\in$ 3-DIM iff $\exists M$ "S, T, U, R, M" $\in$ check_match

cert. checker

so 3-DIM $\in NP$

Non-determinism in programs.

Given a yes/no problem, allow a program to <u>guess</u> bits, costing <u>one time step</u> per bit.

```
int x = guess();    // x gets 0 or 1.
```

s.t guess returns an answer that allows the program to eventually say "yes" if such exists.

uses polynomial time using guess

```
int x = guess()
if (x)
    return yes
else return no;
```

```
FORMULA_SAT("φ")  / returns yes if φ satisfi[able]
```

guess the certificate

```
For i = 1 to n
    v[i] = guess();   // guess truth value
                      of variable i
```

check the certificate

```
if formula_check("φ", "v")  //check v makes φ true
    then return (yes)
    else return (no).
```

Alt (equivalent) def of NP:

$L \in NP$ iff there exists a polynomial-time bounded program P answering $w \in L$ questions, where guess() is allowed.

$L \in P$ iff there exists a polynomial-time bounded program P answering $w \in L$ questions.