Summary of Algebraic Proof

Result: $\text{NPC} \subseteq \text{PCP}[O(\log n), O(1)] \subseteq \text{polylog} n$

Toolbox: "The low deg poly toolbox"

1. Low degree extension - work with low degree poly instead of arbitrary strings.
   - Low degree poly = good code ("Reed-Muller code")
     - (i) good rate ($\Rightarrow$ encoding doesn't increase the length too much)
     - (ii) good distance ($\Rightarrow$ two codewords differ almost everywhere)
     - (iii) self-correction
     - (iv) local testing

Disadvantages:
   - (i) large alphabet $\mathbb{F}$ which is large if $|\mathbb{F}| = \text{polylog} n$
   - Note the alphabet of PCP construction is $\subseteq \text{polylog} n$ (which is not $O(1)$, but not that bad either).
     - The reason is (ii):
   - (ii) large locality & test locality are $|\mathbb{F}|$ (or poly$|\mathbb{F}|$) points (corr. to line or plane inside $\mathbb{F}^m$).
   - (iii) non-optimal rate: length of encoding is polynomial rather than linear in length of message.

Choosing $|\mathbb{F}|$ and $m$ differently can make almost linear...
Proof Outline: Want to decide whether \( \Phi \) is sat.

1. **Sum-Check**

   \[
   \begin{align*}
   \text{Comp} & \quad \Phi \text{ is sat } \Rightarrow \exists \text{ proof that consists of poly}
   \text{ that we always accept.} \\
   \text{Sound} & \quad \Phi \text{ is not sat } \Rightarrow \forall \text{ proof that consists of poly}
   \text{ we accept with probability } \leq \frac{1}{2}.
   \end{align*}
   \]

2. **New proof consists of:**
   - I. Proof from sum-check.
   - II. Proof needed for self-correction (restrictions to lines)
   - III. Proof needed for low deg testing (restrictions to planes)

3. **New verification**

   Simulate sum-check verifier. Replace each query to poly by:
   - I. Low degree test.
   - II. Self correction to answer query.
The argument Completeness is easy. Assume $e$ is not sat.

Three BAD things can happen:

I For some poly, the prover in fact gives a table of a func. that is far from poly.
   $\rightarrow$ Will catch with const. prob.

II For some poly, the prover gives a table of a func. that is close to poly, but on some query it gives a value which is not consistent with the close poly.
   $\rightarrow$ Will catch with const. prob.

III For all poly, the prover gives to all queries values that are consist. with the close low deg poly
   $\rightarrow$ The Sum-Check verifier on the proof that contains for every poly, the low deg poly that is close to the given func.
   Should reject with const. prob.
Next Decrease alphabet from $o(1)$ to $O(1)$. - Via composition.