Proving evlselv

Steven Nguyen (icecream17)

March 12, 2025

Abstract

Trying to prove that evaluating some variables, then the rest, is the same as evaluating all those variables at once, and succeeding after a few mistakes in explaining that I haven't cleaned up. Probably the hardest proof I've done.

1 Introduction

Facing a gap in understanding, I tried asking AIs to give me an overview of how one would prove this. The proof generally goes as follows: substitute some variables, and substitute the rest, tada!

The problem is, in metamath, the intermediate mathematical something that results when you only substitute some variables is very complicated.

1.1 Definition rodeo

A polynomial is a sum of several terms.

Each term is a coefficient times a "bag of variables" ($b \in \text{bag}_v$).

A bag of variables maps variables in the set V to their powers (in $\mathbb{N}_0$).

A bag $b \in \text{bag}_V$ is a function $b : V \to \mathbb{N}_0$.

All in all, the polynomial ring in the set of variables V over a ring R is denoted $R[V]$, where a polynomial $p \in R[V]$ is a function from bags to coefficients: $p : \text{bag}_V \to R$

To evaluate a polynomial $p \in R[V]$, we need assignments from variables to values: a function $A : V \to R$. So, $\text{eval}(P) : (V \to R) \to R$, where

$$\text{eval}(P)(A) = \sum_{b_v \in \text{bag}_V} \left(P(b_v) \times \prod_{v \in V} A(v)^{b_v(v)} \right) \tag{ev}$$

This can be generalized to evaluation in a subring pretty trivially by ~ evlsevl. Even more complicated is when it comes to selecting certain variables for evaluation. First, observe that we can lift $r \in R$ to a constant polynomial $p \in R[V]$.

$$\begin{tikzcd}
R \arrow[r] \arrow[d, tail, "\text{lift}"] & S \arrow[d, tail, "\text{lift}"] \\
R[V] \arrow[r] & S[V]
\end{tikzcd}$$

So given a ring homomorphism from R to S, we get the corresponding map from $R[V]$ to $S[V]$. In particular, the lift from R to $R[V]$ is itself a ring homomorphism. Substituting S with $R[V]$ we get a map from $R[V]$ to $R[V][W] = (R[V])[W]$. And we can chain infinitely: lifting R to $R[V]$ to $R[V][W]$ and so on.

We can also lift a variable to the corresponding polynomial of a single term whose bag just has the variable and whose coefficient is one $= 1$. The polynomial of a variable v will be denoted $\text{polyVar}(v)$. This lift isn't really a homomorphism, but it can still be composed with homomorphisms.

So, here's how we select certain variables J out of the index set of variables I (I is used more often instead of V). Let $p \in R[I]$. Selecting J out will result in an element of $R[I - J][J]$, as follows:

$$\begin{aligned}
\text{select}(p)(J) = \sum_{b_i \in \text{bag}_I} \Bigg[& (p(b_i) \in R \text{ lifted to } R[I-J][J]) \times \\
& \left(\left(\prod_{k \in I-J} \text{polyVar}(k) \right) \in R[I-J] \text{ lifted to } R[I-J][J] \right) \times \\
& \prod_{j \in J} \text{polyVar}(j) \Bigg]
\end{aligned}$$

...or, with creative use of scalar multiplication:

$$\text{select}(p)(J) = \sum_{b_i \in \text{bag}_I} \left[\left(p(b_i) \cdot \prod_{k \in I-J} \text{polyVar}(k) \right) \cdot \prod_{j \in J} \text{polyVar}(j) \right]$$

Finally, we have a concrete equation. The starting definition is actually more complicated since it actually lifts p to $R[I-J][J][I]$ and evaluates that, but this is equivalent. Note that the assignments of a lifted polynomial $R[V][W]$ will map variables W to $R[V]$ not R, so we can't just restrict the assignments A but also lift them for the inner evaluation. Thus, given a polynomial $P \in R[I]$ and a set of assignments $A : I \to R$, we want to prove:

$$\text{eval}(\text{eval}(\text{select}(P)(J))(\text{lift}(A\restriction_J)))(A\restriction_{I-J}) = \text{eval}(P)(A) \tag{goal}$$

where everything expands into oblivion.

2 First steps

Let's first look at the overall structure of (goal). Upon expanding the evaluations using (ev) we get:

$$\sum_{b_k \in \text{bag}_{I-J}} \left[\left(\sum_{b_j \in \text{bag}_J} (\text{select}(P)(J)(b_j) \times \dots) \right) (b_k) \times \dots \right] = \sum_{b_i \in \text{bag}_I} (p(b_i) \cdot \dots)$$

where each ellipsis is the evaluation of the bag given the corresponding assignments.

An immediate issue is that the index sets of the summations are not equal. However, any $b_i \in \text{bag}_I$ (where $b_i : I \to \mathbb{N}_0$) can be decomposed into subset functions $b_j = b\restriction_J : J \to \mathbb{N}_0$ and $b_k = b\restriction_{I-J} : I - J \to \mathbb{N}_0$. As such the right hand side becomes

$$\begin{aligned}
\text{eval}(P)(A) &= \sum_{b_i \in \text{bag}_I} \left(P(b_i) \times \prod_{i \in I} A(i)^{b_i(i)} \right) \\
&= \sum_{b_k \in \text{bag}_{I-J}} \sum_{b_j \in \text{bag}_J} \left(p(b_k \cup b_j) \times \prod_{i \in I} A(i)^{(b_k \cup b_j)(i)} \right) \\
&= \sum_{b_k \in \text{bag}_{I-J}} \sum_{b_j \in \text{bag}_J} \left(p(b_k \cup b_j) \times \prod_{k \in I-J} A(k)^{b_k(k)} \times \prod_{j \in J} A(j)^{b_j(j)} \right) \\
&= \sum_{b_k \in \text{bag}_{I-J}} \left[\prod_{k \in I-J} A(k)^{b_k(k)} \times \sum_{b_j \in \text{bag}_J} \left(p(b_k \cup b_j) \times \prod_{j \in J} A(j)^{b_j(j)} \right) \right]
\end{aligned}$$

We ignore the restrictions of A since they don't affect the result. The outside product $\prod_{k \in I-J} A(k)^{b_k(v)}$ corresponds to the outside ellipsis on the left hand side! So now our goal is:

$$\begin{aligned}
&\sum_{b_k \in \text{bag}_{I-J}} \left[\left(\sum_{b_j \in \text{bag}_J} \left(\text{select}(P)(J)(b_j) \times \prod_{j \in J} \text{lift}\,(A(j))^{b_j(j)} \right) \right) (b_k) \right] \\
&\qquad = \sum_{b_k \in \text{bag}_{I-J}} \sum_{b_j \in \text{bag}_J} \left(p(b_k \cup b_j) \times \prod_{j \in J} A(j)^{b_j(j)} \right) \qquad (1)
\end{aligned}$$

3 Dozens of steps in already

It doesn't seem like that last equation can be manipulated much more, so we'll have to expand $\text{select}(P)(J)(b_j)$.

$$\text{select}(P)(J)(b_j)$$

$$= \left(\sum_{b_i \in \text{bag}_I} \left[\left(P(b_i) \cdot \prod_{k \in I-J} \text{polyVar}(k) \right) \cdot \prod_{j \in J} \text{polyVar}(j) \right] \right) (b_j)$$

By ~ gsummhm, if we apply a (monoid) homomorphism to every addend of a sum, it is the same as applying the homomorphism to the whole sum. So by showing

$$(p \in R[I-J][J] \mapsto p(b_j))$$

is a monoid homomorphism, we can move the function application $(\sum \ldots)(b_j)$ inside: $\sum(\ldots(b_j))$. Since a product sums over the multiplicative group, and a ring homomorphism provides a monoid homomorphism over the multiplicative groups (by ~ rhmmhm), we can also do this with products. The proof that the function above is a ring homomorphism is hand-waved by referencing ~ evls1maprhm.

So clearly, equation 1 can be manipulated after all. Firstly,

$$\prod_{j \in J} \text{lift}\,(A(j))^{b_j(j)} = \prod_{j \in J} \text{lift}\left(A(j)^{b_j(j)}\right) = \text{lift}\left(\prod_{j \in J} A(j)^{b_j(j)}\right)$$

So the left hand side becomes:

$$\begin{aligned}
&\sum_{b_k \in \text{bag}_{I-J}} \left[\left(\sum_{b_j \in \text{bag}_J} \left(\text{select}(P)(J)(b_j) \times \prod_{j \in J} \text{lift}\,(A(j))^{b_j(j)} \right) \right) (b_k) \right] \\
&= \sum_{b_k \in \text{bag}_{I-J}} \sum_{b_j \in \text{bag}_J} \left[\left(\text{select}(P)(J)(b_j) \times \prod_{j \in J} \text{lift}\,(A(j))^{b_j(j)} \right) (b_k) \right] \\
&= \sum_{b_k \in \text{bag}_{I-J}} \sum_{b_j \in \text{bag}_J} \left[\left(\text{select}(P)(J)(b_j) \times \text{lift}\left(\prod_{j \in J} A(j)^{b_j(j)} \right) \right) (b_k) \right] \\
&= \sum_{b_k \in \text{bag}_{I-J}} \sum_{b_j \in \text{bag}_J} \left(\text{select}(P)(J)(b_j)(b_k) \times \text{lift}\left(\prod_{j \in J} A(j)^{b_j(j)} \right) (b_k) \right)
\end{aligned}$$

The last part is a property of being a ring homomorphism. And now, equating that last expression with the right hand side, if we can show:

$$\text{select}(P)(J)(b_j)(b_k) = p(b_k \cup b_j), \text{ and}$$

$$\text{lift}\left(\prod_{j \in J} A(j)^{b_j(j)} \right) (b_k) = \prod_{j \in J} A(j)^{b_j(j)}$$

then we have done it!!

4 Library of Alexandria

This section title is not because valuable information is lost but because I feel on fire finally figuring it out.

$$\text{lift}\left(\prod_{j\in J} A(j)^{b_j(j)}\right)(b_k) = \prod_{j\in J} A(j)^{b_j(j)}$$

Ok, the second equation isn't directly true for all b_k, but it is true when we sum over b_k. A lifted constant only has a nonzero coefficient for the bag of all variables raised to the power of zero, so setting b_k to that bag, we get the result.

Similarly, for the first equation, $\text{lift}(P(b_i))$ is a lifted constant, so its coefficient is only nonzero when setting b_j and b_k to identity-1-bags. So we get:

$$\begin{aligned}
&\text{select}(P)(J)(b_j)(b_k) \\
&= \left(\sum_{b_i\in\text{bag}_I}\left[\left(P(b_i)\cdot\prod_{k\in I-J}\text{polyVar}(k)\right)\cdot\prod_{j\in J}\text{polyVar}(j)\right]\right)(b_j)(b_k) \\
&= \sum_{b_i\in\text{bag}_I}\left[\text{lift}(P(b_i))(b_j)(b_k)\times\ldots\right] \\
&= \sum_{b_i\in\text{bag}_I}\left[\text{lift}(P(b_i))(1)(1)\times\ldots\right] \\
&= \sum_{b_i\in\text{bag}_I} P(b_i)\times\text{lift}\left(\prod_{k\in I-J}\text{polyVar}(k)\right)(1)(1)\times\prod_{j\in J}\text{polyVar}(j)(1)(1) \\
&= \sum_{b_i\in\text{bag}_I} P(b_i)\times 1\times 1 \\
&= \sum_{b_k\in\text{bag}_{I-J}}\ \sum_{b_j\in\text{bag}_J} p(b_k\cup b_j)
\end{aligned}$$

which was what we wanted. (Note: variables have coefficient 1)