• By original assumption, terms[startB].expon > terms[i].expon > terms[finishB].expon for $\forall i$; $\boxed{startB < i < finishB}$ (Eqn. 1)

$\therefore$ terms[x'].expon + terms[startB].expon > terms[x'].expon + terms[~~start~~ i].expon > terms[finishB].expon + terms[x'].expon for $\forall i$; startB < i < finishB

We know, by law of exponents, $x^m \times x^n = x^{m+n}$, $m, n \in N$. (Eqn. 2)

So, comparing eqn(1) and (2), we see that the no. of terms in $B(x)$ is the same as no. of terms in terms[x'] * $B(x)$

$\boxed{\therefore \text{finishy} - \text{starty} = \text{finishB} - \text{startB}}$. We know, $\boxed{\text{finishy} = \text{avail} - 1}$ justified before

$\therefore$ starty = (avail − 1) − finishB + startB ~~= avail~~
= avail − (finishB − startB + 1). We have done just that in lines 13–14.

We know that $\boxed{\sum_{K=startA}^{x'-1} \text{terms}[K] \cdot B(x) + \text{terms}[x'] \cdot B(x) = \sum_{K=startA}^{x'} \text{terms}[K] \cdot B(x)}$

Distributive law.

• By correctness of padd function, on line 15, we have passed the polynomials x and y (startx, finishx and starty, finishy) to padd and it correctly computes $\boxed{D = x + y}$. It returns the pointers to the starting (&startD) and ending (&finishD) to the polynomial D in the global terms[ ] array.

• Now, we initialize this sum as polynomial x on lines 16–17.

• The loop now gets updated to x'+1. It's easy to see that the loop invariant holds.

$\boxed{\text{Termination:}}$ The loop terminates when i = finishA + 1. By the loop invariant, the polynomial $\sum_{K=startA}^{(finishA+1)-1} \text{terms}[K] \cdot B(x)$ has been correctly computed and stored in the terms array between indices startx and finishx as polynomial x.

We know, by distributive law and mathematical induction that