Little-oh (o) Notations

⇒ Big-oh (O) is used as a tight upper bound on the growth of an algorithm's effort (this effort is described by the function $f(n)$)

⇒ Little-oh (o) notation is used to describe an upper bound that cannot be tight (loose upper bound)

Note

$n^2 + 2n \overset{Big}{=} O(n^3)$ is a correct statement by the eqn. But according to the rules of Big-oh, the smallest function on n that will satisfy the requirement should be chosen as $g(n)$ [i.e. its represents tight upper bound.]

So $n^2 + 2n = O(n^2)$, not $O(n^3)$.

But when we use Little-oh (o) notation, its represents loose upper bound so $n^2 + 2n = o(n^3)$ is correct