Show that the class of languages that belong to $\text{DTIME}[O\left(n^{2-\varepsilon}\right)]$ for some $\varepsilon > 0$ is recursively presentable.

We may assume that for all rational $\varepsilon > 0$, the function $n^{2-\varepsilon}$ is fully time constructible, so that in $O\left(n^{2-\varepsilon}\right)$ steps a TM $T$ on any input of length $n$ can write down the value $m = \lfloor n^{2-\varepsilon} \rfloor$ on a tape, so that it can count down in real time from it.

Let us begin with a recursive presentation $[P_i]$ of all poly-time bounded machines. (We may presume that every $P_i$ occurs infinitely often in this enumeration as $P_{i'}$ for infinitely many $i' > i$.)

Nomenclature: For fixed $\varepsilon$, our machines are $Q_{i\varepsilon} = "P_i with the n^{2-\varepsilon} clock attached."$ What if I want to allow $kn^{2-\varepsilon}$ time for any fixed constant $k$? What is the nomenclature now? $Q_{i\varepsilon,k} = "P_i with kn^{2-\varepsilon} clock attached."$ Note that each $Q_{i\varepsilon,k}$ runs in $O\left(n^{2-\varepsilon}\right)$ time. For any language $L$ that is accepted in $O\left(n^{2-\varepsilon}\right)$ time, some $P_i$ accepts $L$ in that time, and for some $k$ large enough to be the constant in the $O$, when we attach the $kn^{2-\varepsilon}$ clock, it does not disturb the output of $P_i$ on any input, so that $L(Q_{i\varepsilon,k}) = L(P_i) = L$. Hence $[Q_{i\varepsilon,k}]_{i,k=1}^\infty$ is a recursive presentation of $\text{DTIME}[O\left(n^{2-\varepsilon}\right)]$ for that $\varepsilon$.

Now we want to define a recursive presentation of the class of languages that belong to $\text{DTIME}[O\left(n^{2-\varepsilon}\right)]$ for some $\varepsilon > 0$. It is: $[Q_{i\varepsilon,k}]_{i,j,k=1}^\infty$. All we need is some (any) effectively definable sequence $\varepsilon_j$ that goes to 0, so $\varepsilon_j = 1/j$ is good enough. One this is understood, fine to just write $[Q_{i\varepsilon,k}]$ without the outer subscripts.

Great question: what about time $O\left(n^{2+\varepsilon}\right)$ for "some" $\varepsilon > 0$? Time $O\left(n^{2+\varepsilon(1)}\right)$ is probably what you really meant. What it means is that there is a function $f(n) = \varepsilon_n$ that $\to 0$ as $n \to \infty$ such that the language is in $\text{DTIME}[n^{2+f(n)}]$. Note: DQL is contained in $\text{DTIME}[n^{1+1/n}]$ (?).
Show that \#P is closed under the operations \(f + g\) and \(f \cdot g\) with additive overhead in a related sense. This also serves as a work-in for the next problem.

**Set-Up:**

By \(f, g \in \#P\), we can take predicates \(R(x, y), S(x, z)\) with bounding polynomials \(p(n), q(n)\) respectively such that for all \(x, f(x) = \#^ry. R(x, y)\) and \(g(x) = \#^rS(x, z)\). We need to show that \(h(x) = f(x) + g(x)\) belongs to \#P, which means we need to find a poly-time decidable predicate \(T(x, w)\) and a bounding polynomial \(r(n)\) such that \(h(x) = \#^rw. T(x, w)\).

Moreover, we need \(T\) to have small overhead, in the sense that given circuits \(C_n\) for \(R(x, y)\) and \(D_n\) for \(S(x, z)\), we get a circuit \(E_n\) for \(T(x, w)\) with overhead [...].

**Execution:** Build \(E_n\) as: \(E_n\) adds an bottom OR gate connected from the outputs of \(C_n\) and \(D_n\), giving the idea that \(E(x, w) = "R(x, y) OR S(x, z)"\). But how does \(w\) relate to \(y\) and \(z\)?

\[
\begin{align*}
C_n(x, y) & \quad D_n(x, z) \\
E_n(x, w) & \quad w = \ldots \quad (w = 0y \text{ for some } y \text{ and } R(x, y)) \text{ or } (w = 1z \text{ for some } z \text{ and } S(x, z)) \\
\end{align*}
\]

Let \(Y\) be the set of good \(y\)'s and \(Z\) the set of good \(z\)'s. What set adds both together? Not \(Y \times Z\) which is what we did below—that multiplies them. Not \(Y \cup Z\) though that is closer, because there might be overlaps. Use the join \(Y \oplus Z = \{0y : y \in Y\} \cup \{1z : z \in Z\}\).

Remaining technical niggle: what if \(q(n) \neq p(n)\), so what is \(r(n)\)? Assuming \(q(n) > p(n)\) without loss of generality, we can make the first case read \(w = 0\left(0^{q(n) - p(n)}\right)y\) for some \(y\), so that \(w\) always has length exactly \(r(n) = q(n) + 1\).

Try: \(E(x, w) \equiv w = yz \text{ and } R(x, y) \text{ and } S(x, z)\). What function is \(\#^rE_n(x, w)\)? It is \(h(x) = f(x)\cdot g(x)\) --- so we have actually solved the second problem first! For multiplication, we got \(E_n(x, yz) = C_n(x, y) \&\& D_n(x, z)\), so \(s(E_n) = s(C_n) + s(D_n) + 1\) counting wires. \(\text{When } g = f\) this is linear: \(2s + 1\). What happens to the size for addition? Still has \(s(C_n) + s(D_n)\) in general because \(E_n\) needs to include both \(R\) and \(S\). But when \(R = S\), can we save...?

**Verification:** (can be folded in with the above)