

Question I.1) *Définition inductive de la concaténation en Ocaml*

Pour toutes listes l et q , et tout élément e :

$$\begin{cases} [] @ l = l \\ (e :: q) @ l = e :: (q @ l) \end{cases}$$

Question I.2) *Rédaction de démonstration par induction structurelle*

Soit l_2 une liste. Montrons par récurrence structurelle sur l_1 que $|l_1 @ l_2| = |l_1| + |l_2|$ pour toutes listes l_1 et l_2 .

- Si $l_1 = []$, $|l_1 @ l_2| = |l_2| = |l_1| + |l_2|$
- Si $l_1 = e :: q$, supposons que $|q @ l_2| = |q| + |l_2|$, pour toute suite l_2 (*hypothèse d'induction*)

$$\begin{aligned} |l_1 @ l_2| &= |(e :: q) @ l_2| \\ &= |e :: (q @ l_2)| && \text{par définition de @} \\ &= 1 + |q @ l_2| && \text{par définition de } l \mapsto 1 \\ &= 1 + |q| + |l_2| && \text{par hypothèse d'induction} \\ &= |e :: q| + |l_2| \\ |l_1 @ l_2| &= |l_1| + |l_2| \end{aligned}$$

Ainsi, par induction structurelle, $|l_1 @ l_2| = |l_1| + |l_2|$, pour toutes listes l_1 et l_2 , et ce, indépendamment du choix de l_2