Theorem. *For all lists* `l`*,* `even_parity l` $=$ `even_parity_tr l`*.*

Proof. By structural induction on `l`.

- **Base case**: `l = []`

$$\begin{aligned}
\texttt{even_parity l} &= \texttt{even_parity []} \\
&= \texttt{false} && \text{By definition of } \texttt{even_parity} \\
&= \texttt{parity false []} && \text{By definition of } \texttt{parity} \\
&= \texttt{even_parity_tr []} && \text{By definition of } \texttt{even_parity_tr} \\
&= \texttt{even_parity_tr l}
\end{aligned}$$

- **Step**: `l` $=$ `x :: xs` for some element `x` and list `xs`.
  Inductive Hypothesis: `even_parity xs` $=$ `even_parity_tr xs`

  - Case 1. `x` $=$ `true`

$$\begin{aligned}
&\texttt{even_parity_tr l} \\
&= \texttt{even_parity_tr true :: xs} \\
&= \texttt{parity false true :: xs} && \text{By definition of } \texttt{even_parity_tr} \\
&= \texttt{parity (false <> true) xs} && \text{By definition of } \texttt{parity} \\
&= \texttt{parity true xs} && \text{By definition of } \oplus \\
&= \texttt{not (not(parity true xs))} && \text{By definition of not} \\
(*) &= \texttt{not (parity false xs)} && \text{Since } \texttt{parity false l} = \texttt{not parity true l} \\
&= \texttt{not (even_parity_tr xs)} && \text{By definition of } \texttt{even_parity_tr} \\
&= \texttt{not (even_parity xs)} && \text{By IH} \\
&= \texttt{even_parity true :: xs} && \text{By definition of } \texttt{even_parity}
\end{aligned}$$

  - Case 2. `x` $=$ `false`.

$$\begin{aligned}
&\texttt{even_parity_tr l} \\
&= \texttt{even_parity_tr false :: xs} \\
&= \texttt{parity false false :: xs} && \text{By definition of } \texttt{even_parity_tr} \\
&= \texttt{parity (false <> false) xs} && \text{By definition of } \texttt{parity} \\
&= \texttt{parity false xs} && \text{By definition of } \oplus
\end{aligned}$$

$=$ `even_parity xs` By IH

$=$ `even_parity false :: xs` By definition of `even_parity`

Step (*) may not be correct/may require more expansion. $\square$