Exercise (Witness indistinguishability of disjunctive composition). Let $\mathbb{G}$ be a discrete logarithm group with a prime number $q$ elements. Use the Schnorr protocol

\[
\begin{array}{c}
P
\end{array}
\begin{array}{c}
r \leftarrow \mathbb{Z}_q
\end{array}
\begin{array}{c}
\alpha = g^r
\end{array}
\begin{array}{c}
\beta \leftarrow \mathbb{Z}_q
\end{array}
\begin{array}{c}
\gamma = x\beta + r
\end{array}
\begin{array}{c}
\left[ g^\gamma = \alpha y^\beta \right]
\end{array}
\]

\[
\begin{array}{c}
V
\end{array}
\begin{array}{c}
\left[ \exists \beta_1, \beta_2 = \beta \right]
\end{array}
\begin{array}{c}
\left[ g^{\gamma_1} = \alpha_1 y^{\beta_1} \right] \land \left[ g^{\gamma_2} = \alpha_2 y^{\beta_2} \right]
\end{array}
\]

to construct a proof of knowledge POK[$\exists x_1, x_2 : y_1 = g^{x_1} \lor y_2 = g^{x_2}$]. Give a complete description of the prover’s behavior if it knows only $x_1$, only $x_2$ or both secrets. Use game rewriting to show that the output distribution of a verifier does not depend which of those provers it interacts.

Solution. Recall that the disjunctive composition is a three-move protocol, where the prover sends two commitment messages $\alpha_1$ and $\alpha_2$. After that the verifier sends a challenge $\beta$, which is then freely decomposed into sub-challenges $\beta_1$ and $\beta_2$ and augmented with corresponding responses $\gamma_1$ and $\gamma_2$. Next, the verifier checks that $\beta_1 + \beta_2 = \beta$ and that individual protocol transcripts $(\alpha_1, \beta_1, \gamma_1)$ and $(\alpha_2, \beta_2, \gamma_2)$ are valid. The overall structure of disjunctive composition is depicted below.

\[
\begin{array}{c}
P
\end{array}
\begin{array}{c}
\alpha_1, \alpha_2
\end{array}
\begin{array}{c}
\beta
\end{array}
\begin{array}{c}
\beta_1, \beta_2, \gamma_1, \gamma_2
\end{array}
\begin{array}{c}
\left[ \beta_1 + \beta_2 = \beta \right]
\end{array}
\begin{array}{c}
\left[ g^{\gamma_1} = \alpha_1 y^{\beta_1} \right] \land \left[ g^{\gamma_2} = \alpha_2 y^{\beta_2} \right]
\end{array}
\]

If the prover knows both secret exponents $x_1$ and $x_2$, then it can carry out both protocols as usual and we get the following detailed description of a protocol execution:

\[
\begin{array}{c}
P
\end{array}
\begin{array}{c}
r_1 \leftarrow \mathbb{Z}_q
\end{array}
\begin{array}{c}
r_2 \leftarrow \mathbb{Z}_q
\end{array}
\begin{array}{c}
\alpha_1 \leftarrow g^{r_1}
\end{array}
\begin{array}{c}
\alpha_2 \leftarrow g^{r_2}
\end{array}
\begin{array}{c}
\alpha_1, \alpha_2
\end{array}
\begin{array}{c}
\beta \leftarrow \mathbb{Z}_q
\end{array}
\begin{array}{c}
\beta_2 \leftarrow \mathbb{Z}_q
\end{array}
\begin{array}{c}
\beta_1 \leftarrow \beta - \beta_2
\end{array}
\begin{array}{c}
\gamma_1 \leftarrow x_1\beta_1 + r_1
\end{array}
\begin{array}{c}
\gamma_2 \leftarrow x_2\beta_2 + r_2
\end{array}
\begin{array}{c}
\beta_1, \beta_2, \gamma_1, \gamma_2
\end{array}
\begin{array}{c}
\left[ \beta_1 + \beta_2 = \beta \right]
\end{array}
\begin{array}{c}
\left[ g^{\gamma_1} = \alpha_1 y^{\beta_1} \right] \land \left[ g^{\gamma_2} = \alpha_2 y^{\beta_2} \right]
\end{array}
\]
**Prover knows only** $x_1$. When the prover does not know $x_2$, it cannot create the response for the second sub-challenge $\beta_2$ without creating a simulated protocol transcript first. As a result, the prover must fix the value $\beta_2$ before the challenge $\beta$ is generated. The corresponding protocol execution is depicted below.

\begin{align*}
P &\quad \begin{aligned}
r_1 &\leftarrow \mathbb{Z}_q \\
\beta_2 &\leftarrow \mathbb{Z}_q \\
\gamma_2 &\leftarrow \mathbb{Z}_q \\
\alpha_1 &\leftarrow g^{\gamma_1} \\
\alpha_2 &\leftarrow g^{\gamma_2} \\
\end{aligned} \\
\beta &\leftarrow \mathbb{Z}_q \\
\beta_1 &\leftarrow \beta - \beta_2 \\
\gamma_1 &\leftarrow x_1\beta_1 + r_1 \\
\beta_1, \beta_2, \gamma_1, \gamma_2 &\rightarrow \\
[\beta_1 + \beta_2 \overset{?}{=} \beta] \\
\left[ g^{\gamma_1} \overset{?}{=} \alpha_1y^{\beta_1} \right] \land \left[ g^{\gamma_2} \overset{?}{=} \alpha_2y^{\beta_2} \right]
\end{align*}

**Prover knows only** $x_2$. When the prover does not know $x_1$, it cannot create the response for the second sub-challenge $\beta_1$ without creating a simulated protocol transcript first. As a result, the prover must fix the value $\beta_1$ before the challenge $\beta$ is generated. The corresponding protocol execution is depicted below.

\begin{align*}
P &\quad \begin{aligned}
r_2 &\leftarrow \mathbb{Z}_q \\
\beta_1 &\leftarrow \mathbb{Z}_q \\
\gamma_1 &\leftarrow \mathbb{Z}_q \\
\alpha_1 &\leftarrow g^{\gamma_1} \\
\alpha_2 &\leftarrow g^{\gamma_2} \\
\end{aligned} \\
\beta &\leftarrow \mathbb{Z}_q \\
\beta_2 &\leftarrow \beta - \beta_1 \\
\gamma_2 &\leftarrow x_2\beta_2 + r_2 \\
\beta_1, \beta_2, \gamma_1, \gamma_2 &\rightarrow \\
[\beta_1 + \beta_2 \overset{?}{=} \beta] \\
\left[ g^{\gamma_1} \overset{?}{=} \alpha_1y^{\beta_1} \right] \land \left[ g^{\gamma_2} \overset{?}{=} \alpha_2y^{\beta_2} \right]
\end{align*}

**Witness indistinguishability.** In the following we show that the verifier cannot distinguish which of those three modes the honest prover runs. As the first step, we show that the verifier cannot distinguish whether the prover knows $x_1$ and $x_2$ or only $x_1$. For that, we construct games for both execution modes where the output is determined by the verifier $V$, and then show that both games produce identical output distributions. The game $G_0$ corresponds to the case where the prover knows $x_1$ and $x_2$ and game $G_3$ to the
case where prover knows only $x_1$:

$$G_0$$

\[
\begin{align*}
r_1 & \leftarrow \mathbb{Z}_q \\
r_2 & \leftarrow \mathbb{Z}_q \\
\alpha_1 & \leftarrow g^{r_1} \\
\alpha_2 & \leftarrow g^{r_2} \\
\beta & \leftarrow V_s(\alpha_1, \alpha_2) \\
\beta_2 & \leftarrow \mathbb{Z}_q \\
\beta_1 & \leftarrow \beta - \beta_2 \\
\gamma_1 & \leftarrow x_1 \beta_1 + r_1 \\
\gamma_2 & \leftarrow x_2 \beta_2 + r_2 \\
\text{return } & V_s(\beta_1, \beta_2, \gamma_1, \gamma_2)
\end{align*}
\]

$$G_3$$

\[
\begin{align*}
r_1 & \leftarrow \mathbb{Z}_q \\
\beta_2 & \leftarrow \mathbb{Z}_q \\
\gamma_2 & \leftarrow \mathbb{Z}_q \\
\alpha_1 & \leftarrow g^{r_1} \\
\alpha_2 & \leftarrow g^{\gamma_2} \\
\beta & \leftarrow V_s(\alpha_1, \alpha_2) \\
\beta_1 & \leftarrow \beta - \beta_2 \\
\gamma_1 & \leftarrow x_1 \beta_1 + r_1 \\
\gamma_2 & \leftarrow x_2 \beta_2 + r_2 \\
\text{return } & V_s(\beta_1, \beta_2, \gamma_1, \gamma_2)
\end{align*}
\]

Since the sampling of $\beta_2$ does not depend on $\beta$ in $G_0$ we can move it towards the beginning of the game and get a new game with the same output distribution:

$$G_1$$

\[
\begin{align*}
r_1 & \leftarrow \mathbb{Z}_q \\
\beta_2 & \leftarrow \mathbb{Z}_q \\
r_2 & \leftarrow \mathbb{Z}_q \\
\alpha_1 & \leftarrow g^{r_1} \\
\alpha_2 & \leftarrow g^{r_2} \\
\beta & \leftarrow V_s(\alpha_1, \alpha_2) \\
\beta_1 & \leftarrow \beta - \beta_2 \\
\gamma_1 & \leftarrow x_1 \beta_1 + r_1 \\
\gamma_2 & \leftarrow x_2 \beta_2 + r_2 \\
\text{return } & V_s(\beta_1, \beta_2, \gamma_1, \gamma_2)
\end{align*}
\]

Let us now concentrate on the variables $r_2, \alpha_2, \beta_2, \gamma_2$. Note that even for fixed $\beta_2$ the value $\gamma_2$ must be uniformly distributed as $r_2$ is uniformly distributed. Hence, we can pick $\gamma_2$ and then calculate $r_2$ from it:

$$G_2$$

\[
\begin{align*}
r_1 & \leftarrow \mathbb{Z}_q \\
\beta_2 & \leftarrow \mathbb{Z}_q \\
\gamma_2 & \leftarrow \mathbb{Z}_q \\
r_2 & \leftarrow \gamma_2 - x_2 \beta_2 \\
\alpha_1 & \leftarrow g^{r_1} \\
\alpha_2 & \leftarrow g^{r_2} \\
\beta & \leftarrow V_s(\alpha_1, \alpha_2) \\
\beta_1 & \leftarrow \beta - \beta_2 \\
\gamma_1 & \leftarrow x_1 \beta_1 + r_1 \\
\gamma_2 & \leftarrow x_2 \beta_2 + r_2 \\
\text{return } & V_s(\gamma_1, \gamma_2, \beta_1, \beta_2)
\end{align*}
\]

Now note that $\alpha_2 = g^{r_2} = g^{\gamma_2 - x_2 \beta_2} = \frac{g^{\gamma_2}}{g^{x_2 \beta_2}}$ and thus $G_2$ is equivalent to $G_3$.

The sequence of game transformations that show that knowledge of $x_2$ is indistinguishable from the
knowledge of $x_1$ and $x_2$ is analogous, except for a minor detail how $\beta_1$ and $\beta_2$ are generated:

\[ \mathcal{G}_0 \]

\[
\begin{align*}
r_1 &\leftarrow \mathbb{Z}_q \\
r_2 &\leftarrow \mathbb{Z}_q \\
\alpha_1 &\leftarrow g^{r_1} \\
\alpha_2 &\leftarrow g^{r_2} \\
\beta &\leftarrow \mathcal{V}_+(\alpha_1, \alpha_2) \\
\beta_2 &\leftarrow \mathbb{Z}_q \\
\beta_1 &\leftarrow \beta - \beta_2 \\
\gamma_1 &\leftarrow x_1 \beta_1 + r_1 \\
\gamma_2 &\leftarrow x_2 \beta_2 + r_2 \\
&\text{return } \mathcal{V}_+(\beta_1, \beta_2, \gamma_1, \gamma_2)
\end{align*}
\]

\[ \mathcal{G}_3 \]

\[
\begin{align*}
r_2 &\leftarrow \mathbb{Z}_q \\
\beta_1 &\leftarrow \mathbb{Z}_q \\
\gamma_1 &\leftarrow \mathbb{Z}_q \\
\alpha_1 &\leftarrow g^{\gamma_1} \\
\gamma_1 &\leftarrow \mathbb{Z}_q \\
\alpha_2 &\leftarrow g^{r_2} \\
\beta &\leftarrow \mathcal{V}_+(\alpha_1, \alpha_2) \\
\beta_2 &\leftarrow \beta - \beta_1 \\
\gamma_1 &\leftarrow x_1 \beta_1 + r_1 \\
&\text{return } \mathcal{V}_+(\beta_1, \beta_2, \gamma_1, \gamma_2)
\end{align*}
\]

It is easy to see that it does not matter if we first pick $\beta_1$ and then calculate $\beta_2 = \beta - \beta_1$ in $\mathcal{G}_0$ or if we first pick $\beta_2$ and then calculate $\beta_1 = \beta - \beta_2$. After doing this switch, we will arrive at the setting where the games are analogously aligned as in the first equivalence proof and we can do analogous proof that games $\mathcal{G}_0$ and $\mathcal{G}_3$ are identical. As a result, we have shown

\[ \mathcal{G}_3^{V} \equiv \mathcal{G}_0^{V} \equiv \mathcal{G}_3^{V} \]

and thus by transitivity the verifier will not be able to distinguish between any of these games. Note that the proof holds also for malicious adversary who could create $\beta$ as it wishes, since during the game rewriting we made no assumption about $\beta$. 

4