1 Negligible Functions

1.1 Negligibility Proof

As has been included in lecture [1], the definition for negligible function can be:
A function $\nu(n)$ is negligible if $\forall c, \exists n_0$ such that $\forall n > n_0, \nu(n) \leqslant \frac{1}{n^c}$.

By definition of little ω, we have:

$$\omega(\log n) > c \cdot \log n$$

for arbitrary real constant c.
And it further shows

$$\omega(\log n) > c \cdot \log n \tag{1}$$
$$2^{\omega(\log n)} > 2^{c \cdot \log n} = n^c \tag{2}$$
$$2^{-\omega(\log n)} < n^{-c} = \frac{1}{n^c} \tag{3}$$

Therefore, we proved the negligibility of the function.

1.2 Example

$f(n) = 2^{-n}$ and $g(n) = 4^{-n}$. Then we have $\frac{f(n)}{g(n)} = 2^n$, which is not negligible.

2 Hybrid Lemma

No, it is not.
As is also pointed out in Chapter 4.2 from The Joy of Cryptography [2], transitivity only applies on a polynomial number of times and it is not allowed to build a series of hybrids on exponential times.
By definition [1], we have $X^1, \ldots, X^m$ distribution ensembles for $m = \textbf{poly}\ (n)$. If for every $i \in [m-1], X^i$ and X^{i+1} are computationally indistinguishable, then X^1 and X^m are computationally indistinguishable.
While in the problem setting, $U_{0,2^n-1} \approx U_{2^n,2^{n+1}-1}$ on exponential times is concluded based on the condition that for every $i, H_i \approx H_{i+1}$.
To prove this, we need to show that $\left|\Pr\left[\mathcal{D}^{U_{0,2^n-1}} = 1\right] - \Pr\left[\mathcal{D}^{U_{2^n,2^{n+1}-1}} = 1\right]\right|$ with polynomial-time (PPT) distinguisher $\mathcal{D}$ may not be negligible.

$$\left|\Pr\left[\mathcal{D}^{U_{0,2^n-1}} = 1\right] - \Pr\left[\mathcal{D}^{U_{2^n,2^{n+1}-1}} = 1\right]\right| \leq \sum_{i=1}^{2^n} \left|\Pr\left[\mathcal{D}^{\mathcal{H}_i}\right] - \Pr\left[\mathcal{D}^{\mathcal{H}_{i-1}}\right]\right| \tag{4}$$
$$= 2^n \cdot negl. \text{ because } (H_i \approx H_{i+1}) \tag{5}$$

And the result $2^n \cdot negl.$ may not be negligible as we also proved in Q1.2 and thus the proof does not hold.

3 Pseudorandom Generators

3.1 Proof

No, it is not since randomness implies we know nothing about the next bit based on first part of the string but here we know every bit in the second half with probability 1.
In this question I am inspired by what has been discussed in Chapter 5.2 from The Joy of Cryptography [2]. We now consider the case by G_1 and G_2 are identical where $G(s) = G'(s)||G'(s)$. We can prove

this with an efficient attack by

$$\begin{aligned}
&\underline{\mathcal{A}}: && (6)\\
&\quad x||y := QUERY() && (7)\\
&\quad \text{return } x \overset{?}{\leftarrow} y && (8)\\
& && (9)
\end{aligned}$$

The first line obtains the result of query and set its first half to be the string x and its second half to be y. This calling program simply checks whether the output of query has equal halves.
Now we need to show this calling program can distinguish between the PRG constructed and random.

$$\begin{aligned}
&\underline{\mathcal{L}^G_{PRG-REAL}}: && (10)\\
&\quad QUERY(): && (11)\\
&\quad\quad s \leftarrow \{0,1\}^\lambda && (12)\\
&\quad\quad \text{return } G'(s)||G'(s) && (13)\\
& && (14)
\end{aligned}$$

and

$$\begin{aligned}
&\underline{\mathcal{L}^G_{PRG-RANDOM}}: && (15)\\
&\quad QUERY(): && (16)\\
&\quad\quad r \leftarrow \{0,1\}^{2\lambda} && (17)\\
&\quad\quad \text{return } r && (18)\\
& && (19)
\end{aligned}$$

Since the calling program always outputs matching halves, we can have

$$\Pr\left[\mathcal{A} \diamond \mathcal{L}^G_{\text{PRG-REAL}} \Rightarrow 1\right] = 1$$

In view of random, we will have the matching halves the same as matching half length since here we just double the length by duplicating in $G(s)$ and thus we will obtain

$$\Pr\left[\mathcal{A} \diamond \mathcal{L}^G_{\text{PRG-RANDOM}} \Rightarrow 1\right] = \frac{1}{2^\lambda}$$

Finally we can show the disadvantage by the difference between two probabilities, which is $1 - \frac{1}{2^\lambda}$, a non-negligible, showing $G(s)$ is NOT computationally indistinguishable from the uniform distribution and thus is not a secure PRG.

3.2 Proof

Yes, it is a PRG.
Intuitively, $G(s)$ on uniform s is computationally indistinguishable from the uniform distribution and thus s_1 and s_2 are also computationally indistinguishable from the uniform distribution. Further, we know $G(s_1)$ and $G(s_2)$ are also computationally indistinguishable from the uniform distribution and so is their concatenation.
We can prove with hybrid argument.
Firstly, we need to construct $H_1(s) = s_1$ and $H_2(s) = s_2$ to show that $G(H_1(s))||G(H_2(s))$ is indistinguishable from $G(r_1)||G(r_2)$, for some random r_1 and r_2, we have the following hybrids:

$$\mathcal{H}_0 : \left\{G(H_1(s))||G(H_2(s)); s \overset{\$}{\leftarrow} \{0,1\}^n, H_1(s)||H_2(s) = G(s)\right\} \tag{20}$$

$$\mathcal{H}_1 : \left\{G(r_1)||G(H_2(s)); r_1 \overset{\$}{\leftarrow} \{0,1\}^n, s \overset{\$}{\leftarrow} \{0,1\}^n, H_1(s)||H_2(s) = G(s)\right\} \tag{21}$$

$$\mathcal{H}_2 : \left\{G(r_1)||G(r_2); r_1, r_2 \overset{\$}{\leftarrow} \{0,1\}^n\right\} \tag{22}$$

From the security of PRG, we know that

$$\left\{G(s); s \xleftarrow{\$} \{0,1\}^n\right\} \approx_c \left\{r; r \xleftarrow{\$} \{0,1\}^{2n}\right\}$$

From closure property of computational indistinguishability and $H_1(s)$ represents for the first half in $G(s)$, we get

$$\left\{H_1(s); s \xleftarrow{\$} \{0,1\}^n\right\} \approx_c \left\{r_1; r_1 \xleftarrow{\$} \{0,1\}^n\right\}$$

and thus we have

$$\mathcal{H}_0 \approx_c \mathcal{H}_1$$

Similarly, we have

$$\mathcal{H}_1 \approx_c \mathcal{H}_2$$

Secondly, we need to show that $G(r_1)||G(r_2)$ is indistinguishable from $r'_1||r'_2$, for some random r'_1 and r'_2, we have the following hybrids:

$$\mathcal{H}_2 : \left\{G(r_1)||G(r_2); r_1, r_2 \xleftarrow{\$} \{0,1\}^n\right\} \tag{23}$$

$$\mathcal{H}_3 : \left\{r'_1||G(r_2); r'_1 \xleftarrow{\$} \{0,1\}^{2n}, r_2 \xleftarrow{\$} \{0,1\}^n\right\} \tag{24}$$

$$\mathcal{H}_4 : \left\{r'_1||r'_2; r'_1, r'_2 \xleftarrow{\$} \{0,1\}^{2n}\right\} \tag{25}$$

From the security of PRG, we know that

$$\left\{G(r_1); r_1 \xleftarrow{\$} \{0,1\}^n\right\} \approx_c \left\{r'_1; r'_1 \xleftarrow{\$} \{0,1\}^{2n}\right\}$$

From closure property of computational indistinguishability, we get

$$\mathcal{H}_2 \approx_c \mathcal{H}_3$$

Similarly, we have

$$\mathcal{H}_3 \approx_c \mathcal{H}_4$$

Now we have shown that $\mathcal{H}_0 \approx_c \mathcal{H}_1 \approx_c \mathcal{H}_2 \approx_c \mathcal{H}_3 \approx_c \mathcal{H}_4$. By hybrid argument, we prove that $H(\cdot)$ is a secure PRG.

3.3 Proof

Consider the following hybrids[1]:

- $\mathcal{H}_0 : \{s_1||x; s_1, s_2 \xleftarrow{\$} \{0,1\}^n, x = G(s_2)\}$
- $\mathcal{H}_1 : \{s_1||r_2; s_1 \xleftarrow{\$} \{0,1\}^n, r_2 \xleftarrow{\$} \{0,1\}^{3n}\}$,
- $\mathcal{H}_2 : \{r_1||r_2; r_1 \xleftarrow{\$} \{0,1\}^n, r_2 \xleftarrow{\$} \{0,1\}^{3n}\}$,

In order to show that $H(s)$ is a PRG, it suffices to show that $\mathcal{H}_0$ is indistinguishable from $\mathcal{H}_2$. Assume that there exists an adversary $\mathcal{A}$ who can distinguish between $\mathcal{H}_0$ and $\mathcal{H}_2$ with some non-negligible advantage $\mu(n)$. From hybrid lemma, it follows that there must exist $i \in \{0,1\}$, such that $\mathcal{A}$ can distinguish between $\mathcal{H}_i$ and $\mathcal{H}_{i+1}$ with non-negligible advantage at least $\mu(n)/4$. Now we need to show there exists another adversary $\mathcal{B}$ that can break the security of PRG G by giving a proof via reduction for each $i \in \{0,1\}$.

[1]Inspired by lecture notes: https://github.com/heldridge/ModernCryptography-Fall2022/blob/main/notes/reduction˙example.pdf

1. Let $\mathcal{A}$ distinguish between $\mathcal{H}_0$ and $\mathcal{H}_1$ non-negligible advantage at least $\mu(n)/4$. We now construct another adversary $\mathcal{B}$ that breaks the security of G as follows:

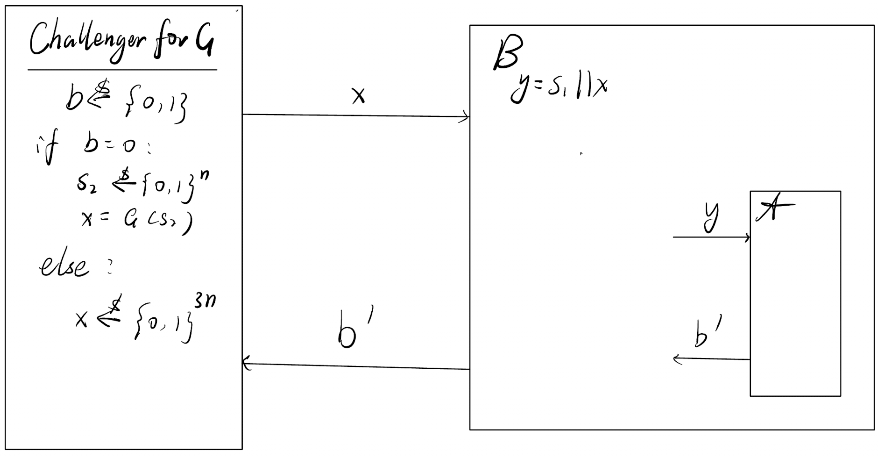

Figure 1: PRG Reduction 1

From the above reduction, it is clear that $\mathcal{B}$ has the same advantage in breaking G as the advantage that $\mathcal{A}$ has in distinguishing between $\mathcal{H}_0$ and $\mathcal{H}_1$, which is at least $\mu(n)/4$. Since $\mu(n)/4$ is non-negligible, this would mean $\mathcal{B}$ can break G. However, we know that G is a secure PRG, and hence, no such adversary can exist. Therefore our assumption was incorrect and $\mathcal{H}_0 \approx \mathcal{H}_1$.

2. Let $\mathcal{A}$ distinguish between $\mathcal{H}_1$ and $\mathcal{H}_2$ non-negligible advantage at least $\mu(n)/4$. We now construct another adversary $\mathcal{B}$ that breaks the security of G as follows:

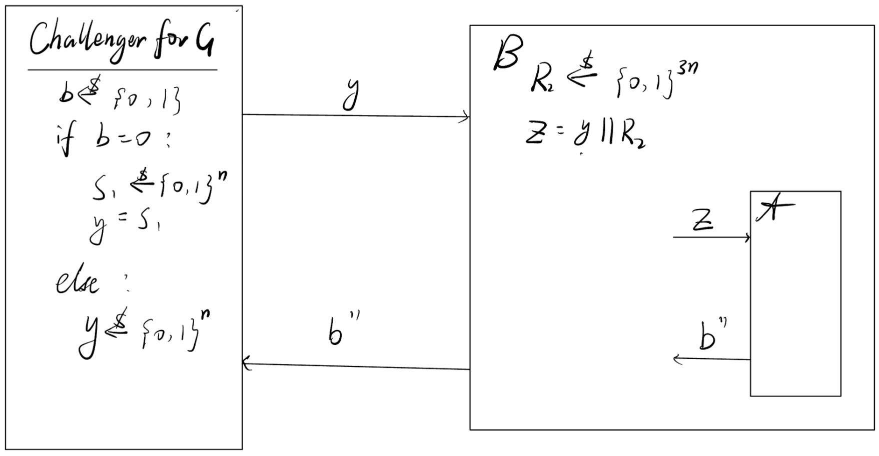
Challenger for G
$b \xleftarrow{\$} \{0,1\}$
if $b = 0$:
$s_1 \xleftarrow{\$} \{0,1\}^n$
$y = s_1$
else :
$y \xleftarrow{\$} \{0,1\}^n$
y
b''
$\mathcal{B}$
$R_2 \xleftarrow{\$} \{0,1\}^{3n}$
$z = y || R_2$
z
$\mathcal{A}$
b''

Figure 2: PRG Reduction 2

From the above reduction, it is clear that $\mathcal{B}$ has the same advantage in breaking G as the advantage that $\mathcal{A}$ has in distinguishing between $\mathcal{H}_1$ and $\mathcal{H}_2$, which is at least $\mu(n)/4$. Since $\mu(n)/4$ is non-negligible, this would mean $\mathcal{B}$ can break G. However, we know that G is a secure PRG, and hence, no such adversary can exist. Therefore our assumption was incorrect and $\mathcal{H}_1 \approx \mathcal{H}_2$
We have shown that $\mathcal{H}_0 \approx \mathcal{H}_1 \approx \mathcal{H}_2$. which contradicts the assumption and thus there does not

exist any adversary $\mathcal{A}$ who can distinguish between $\mathcal{H}_0$ and $\mathcal{H}_2$ with non-negligible advantage $\mu(n)$. Therefore, $H(\cdot)$ is a secure PRG.

4 One-Way Functions

4.1 Proof

Yes, it still is.
Proof via reduction: Suppose f' is not an OWF, then there exists a non-uniform PPT adversary $\mathcal{A}$ that inverts f' with non-negligible probability. Then we construct another adversary $\mathcal{B}$ to invert f with non-negligible probability. Now we show

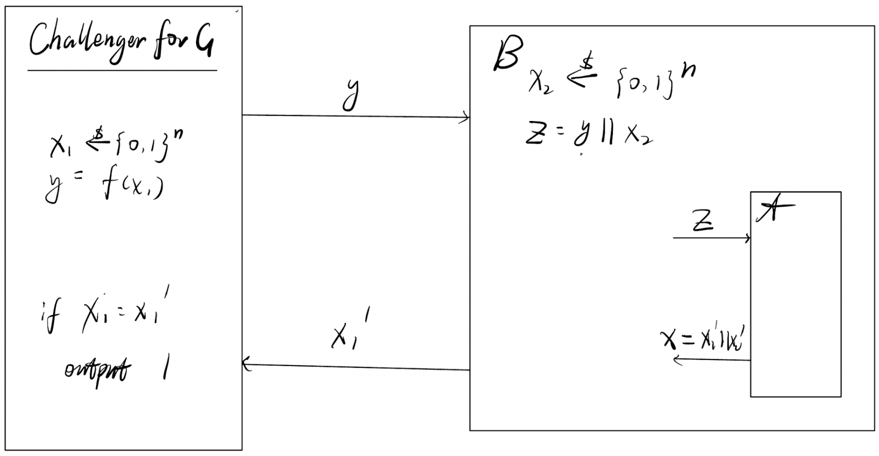

Figure 3: OWF Reduction

$\mathcal{A}$ is always able to give x_2 since it is exposed. Now we need to show that $\mathcal{B}$ has the same advantage of breaking f as $\mathcal{A}$ is able to break f'.

$$\begin{aligned}
Pr[\mathcal{B} \text{ breaks } f] &= Pr[x_1 = x_1'] && (26)\\
&= Pr[f(x_1) = f(x_1')] && (27)\\
&= Pr[f(x_1) + x_2 = f(x_1') + x_2] && (28)\\
&= Pr[f'(x_1||x_2) = f(x_1'||x_2)] && (29)\\
&= Pr[\mathcal{A} \text{ breaks } f'] && (30)
\end{aligned}$$

As in the assumption, there exists non-negligible probability that A breaks f' so that there exists non-negligible probability that B breaks f. However, f is an OWF and this contradicts the conclusion above. Therefore, f' is an OWF.

4.2 Proof

$f'(x_1||x_2)$ is not an OWF.
Given the value $y = f'(x_1||x_2) = f(x_1) \oplus x_2$ to invert $x_1||x_2$, we can construct the OWF by generating random x_1' and calculate $f(x_1')$. Then we can obtain the value x_2' by $x_2' = y \oplus f(x_1')$.
Now that we obtain $x_1'||x_2'$ by $x_1'||(y \oplus f(x_1'))$ where $f'(x_1'||x_2') = y$ and the probability for such efficient invert is constant 1, not negligible, and thus $f'(x_1||x_2)$ is not an OWF.

References

[1] Computational Security and Pseudorandomness I, https://github.com/heldridge/ModernCryptography-Fall2022/blob/main/L03_CompSec.pdf

[2] The Joy of Cryptography, https://joyofcryptography.com/