Remember that EQ_{CFG} is co-Turing-recognizable language if and only if its complement $\overline{EQ_{CFG}}$ is a Turing-recognizable language.

Now, $\overline{EQ_{CFG}} = A \cup B$, where

$$A = \{ w \mid w \textit{ does not have the form } \langle G1, G2 \rangle \textit{ for some CFGs } G1 \textit{ and } G1 \},$$

$$B = \{ \langle G1, G2 \rangle \mid \text{G1 and G2 are CFGs and } L(G1) \neq L(G2) \}.$$

• *A* contains strings that defy the syntax for encoding $\langle G1, G2 \rangle$) is simple to accept.

• The set B is realized in the following way:

• Transform the G1, G2 to Chomsky Normal Form (CNF).

• Then begin numbering strings in Σ^* lexicographically, where Σ is the group of terminals for G1, G2.

• For every string *w* numbered, check if it is produced by G1 and by G2.

• If the 2 Context Free Grammars or both of them cannot produce *w*, then TM goes on to generate the next string in the lexicographic order.

• Else, precisely one of the CFGs produces the string, and the TM accepts.

• Therefore, B is a Turing recognizable language.

• It is already proved that Turing-recognizable languages are closed under union, so EQ_{CFG} is Turing-recognizable language.

• Let the list of strings that are listed in lexicographic order are as follows:

s1, s2, s3 . . . over the input Σ^* .

Now EQ_{CFG} is realized by the following *TM M*:

M = "On input $\langle G1, G2 \rangle$, where G1, G2 are Context Free Grammars:

1. Examine if G1, G2 are valid Context Free Grammars. If atleast 1 does not, accept.

2. Transform G1, G2 into corresponding Context Free Grammars G'_1 , G'_2 .both into CNF.

3. Replicate the below step 4 for j = 1, 2, 3 . . .

4. Examine if G'_1 and G'_2 produce s_j if precisely one of them does, *accept*."

Hence it is proved that EQ_{CFG} is co-Turing-recognizable language.