C. The reason why cfg12a and cfg12b produce different values when using the f function to calculate the total probability is because of the initial probability values. The f function takes account of the initial probabilities while calculating the total probability. In cfg12a, the initial probability for VP is 1.0, which means that all strings start from VP, while in cfg12b, the initial probability for VP and NP are split at 0.5 each. This difference in initial probability affects the total probability computed by f, even though the inside values remain the same.

D. To convert a FSA to a CFG, first create non-terminal symbols for each state in FSA and for the transition symbols in FSA. Add a start symbol (S), a start marker (A), and an end marker which is represented by the end state. Then we create rules for transitions in the FSA, such as (X1) to (X2) on symbol (c) written as (X1 -> C X2), where C represents the non-terminal symbol of the terminal symbol c, and X1 and X2 are the non-terminal symbols for the states 1 and 2. Create another rule for the start symbol (S) to transition to initial state (X1) through the start marker (A) written as (S -> A X1). Lastly, create another rule for the start marker (A -> ⋊) and the end marker (X1 -> ⋉) to handle the beginning and end of strings.

E.

	…⋊	**…c**	**…v**	**…c**	**…⋉**
⋊…	S: 1 A: 1 X1: 1 X2: 0 X3: 0 C: 0 V: 0	S: 0 A: 0 X1: 0 X2: 0 X3: 0 C: 0 V: 0	S: 0 A: 0 X1: 0 X2: 0 X3: 0 C: 0 V: 0	S: 0 A: 0 X1: 0 X2: 0 X3: 0 C: 0 V: 0	S: 0 A: 0 X1: 0 X2: 0 X3: 0 C: 0 V: 0
c…		S: 0 A: 0 X1: 0 X2: 1 X3: 0 C: 1 V: 0	S: 0 A: 0 X1: 0 X2: 0 X3: 0 C: 0 V: 0	S: 0 A: 0 X1: 0 X2: 0 X3: 0 C: 0 V: 0	S: 0 A: 0 X1: 0 X2: 0 X3: 0 C: 0 V: 0
v…			S: 0 A: 0 X1: 1 X2: 0 X3: 1 C: 0 V: 1	S: 0 A: 0 X1: 0 X2: 0 X3: 0 C: 0 V: 0	S: 0 A: 0 X1: 0 X2: 0 X3: 0 C: 0 V: 0
c…				S: 0 A: 0 X1: 1 X2: 0 X3: 0 C: 1 V: 0	S: 0 A: 0 X1: 0 X2: 0 X3: 0 C: 0 V: 0
⋉…					S: 0 A: 0 X1: 1 X2: 0 X3: 0 C: 0 V: 0

This inside value table corresponds directly to the forward table since X1, X2, and X3 align with states 1, 2, and 3 in the FSA from (3). The inside values for X1 match the forward values for state 1, those for X2 match state 2, and those for X3 match state 3.

F.

- S → Z X1
- Z → ⩡
- X1 → ⩤
- C → c
- V → v

G. Both CFG in (4) and (5) generate the same strings. However, CFG (4) has rules that move forward, from state 1 to state 2 through a transition symbol (X1 -> C X2), while CFG (5) has rules that work backwards, moving from state 2 to state 1 with transition symbol (X1 -> X2 V). Because of this difference, the inside values of CFG (4) correspond to forward values in the FSA, while the inside values of CFG (5) correspond to backward values in the FSA, reflecting transitions in reverse order.