S → Aa
A → Bb
B → Sc | c

Follow the algorithm on Slide 10. The key is to transfer indirect recursion to direct (immediate) recursion through replacement.

First, add S' → S

Work on all nonterminals in an arranged order. Any order is fine. Let's follow S, A, and then B. (I also did B, A, S in class.)

For S, no replacement and no direct recursion.

For A, no replacement and no direct recursion.

For B, note that B → Sc and S appears before B in the ordering. Replacing S, we get
    B → Aac | c

We see A is a nonterminal in the leftmost position and A appears before B. So we need replace A. After replacement of A, we get
    B → Bbac | c

We just exposed a direct left recursion for B. Remove this recursion using the algorithm on Slide 9. We have

    B → cB'
    B' → bacB' | ε

The original grammar is converted to the grammar below.

S' → S
S → Aa
A → Bb
B → cB'
B' → bacB' | ε