Evaluation of SDT

Example:-

S → AS {printf(1)}
S → AB {printf(2)}
A → a {printf(3)}
B → bC {printf(4)}
B → dB {printf(5)}
C → c {printf(6)}

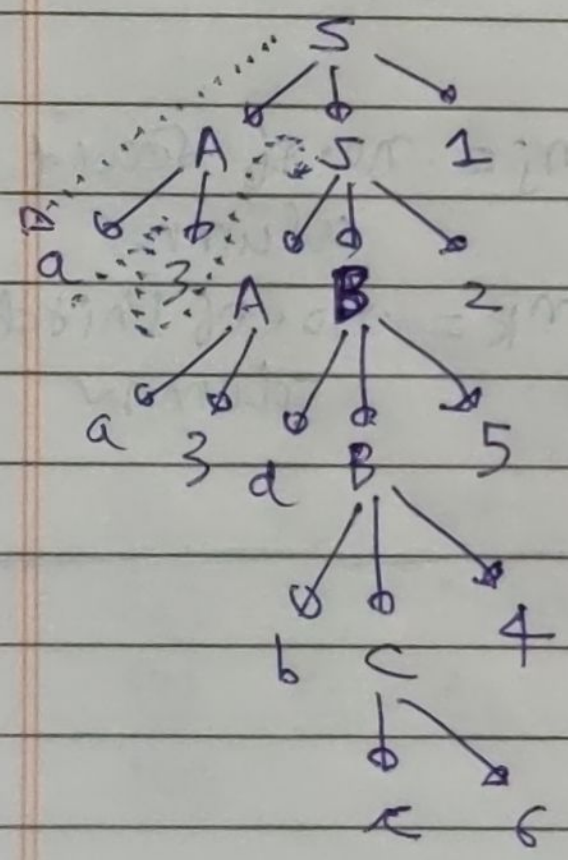
What is the output when the following is parsed using the given SDD.
I/P = aadbc

Solution:- There are two methods to evaluate SDT:

(a) top-down
(b) bottom-up.

(a) top-down:-

- {...} will be treated as a non-terminal
- Parse tree will be constructed
- Depth-First traversal.

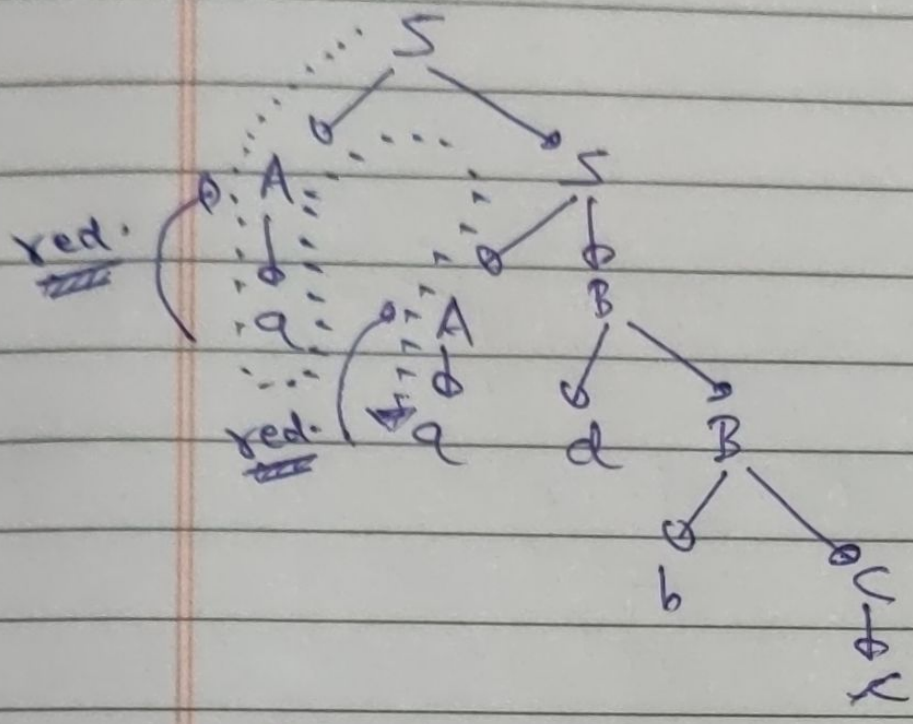


Output: 3364521 Ans

(depth first traversal).

(b) Bottom-up :-

- Don't consider {...} while building parse tree.
- When there is a reduction, then evaluate {...} it.



- Firstly, when (·) move ahead in bottom up parsing, we will have a reduction in the Stack : a → A in the depth first order.
- Output : 3364521

Ans.

- Go by bottom-up parsing algorithm.