**Задание 2**

Привести пример цепочки, для которой существует два левых вывода. Какие символы используются в записи вывода? Номер правила в выводе можно записывать нижним индексом.

**Ответ на задание 2**

Пример цепочки с двумя левыми выводами

Для заданной грамматики:

1. S→AB
2. A→a
3. B→b

Цепочка: ab

Два левых вывода:

1. Первый вывод:
2. Второй вывод:

Для цепочки **ab** существуют два левых вывода, которые различаются последовательностью применения правил, но приводят к одной и той же терминальной цепочке.

Символы, используемые в записи вывода:

* Большие буквы (например S, A, BS, A, B) — это нетерминалы.
* Маленькие буквы (например a,ba, b) — это терминалы.
* Нижние индексы (например ,) — обозначают номер примененного правила.

**Задание 3**

Результат работы программы не понятен.

**Ответ на задание 3**

Программа проверяет возможность применения заданной последовательности правил для выполнения левого вывода цепочки в заданной КС-грамматике. Левый вывод означает, что на каждом шаге программа заменяет самый левыйнетерминал по указанному правилу.

*Этапы работы программы*

1. Инициализация начальной цепочки
   * Программа начинает с начального символа грамматики, обычно это S.
   * intermediateChain — это промежуточная цепочка, которая будет изменяться по ходу выполнения.
   * Массив commands содержит последовательность номеров правил, которые необходимо применить.
   * commandIndex отслеживает текущее правило из массива commands.
2. Цикл выполнения

Программа выполняет цикл до тех пор, пока в цепочке остаются нетерминалы (заглавные буквы).

В каждой итерации:

* + Программа просматривает текущую цепочку символ за символом.
  + Если встречается нетерминал (заглавная буква), программа:
    - Ищет все правила из грамматики, которые можно применить к этому нетерминалу.
    - Берет следующее правило из последовательности команд (массив commands).
    - Проверяет, применимо ли это правило.
      * Если правило применимо, программа заменяет нетерминал на правую часть этого правила.
      * Если правило неприменимо, программа завершает выполнение и возвращает false.
  + Если символ в цепочке не является нетерминалом, программа просто переносит его в новую цепочку без изменений.

1. Завершение работы
   * Когда в цепочке не остается нетерминалов, программа проверяет:
     + Все ли правила из последовательности команд были использованы?
   * Если все команды использованы и цепочка полностью терминальная, программа возвращает true.
   * Если остались неиспользованные команды или возникли ошибки при применении правил, программа возвращает false.

*Пример работы программы*

Грамматика:

1. S → AB
2. A → a
3. B → b

Входные данные:

* Начальная цепочка: S
* Последовательность правил: 1, 2, 3

Ход работы программы:

1. Шаг 1:
   * Текущая цепочка: S
   * Применяем правило 1 (S → AB)
   * Новая цепочка: AB
2. Шаг 2:
   * Текущая цепочка: AB
   * Самый левый нетерминал: A
   * Применяем правило 2 (A → a)
   * Новая цепочка: aB
3. Шаг 3:
   * Текущая цепочка: aB
   * Самый левый нетерминал: B
   * Применяем правило 3 (B → b)
   * Новая цепочка: ab
4. Завершение:
   * В цепочке ab нет нетерминалов.
   * Все правила из последовательности 1, 2, 3 использованы.
   * Программа возвращает true.

**Результаты работы программы**

* true: если все правила из последовательности были успешно применены и цепочка стала терминальной.
* false: если возникли ошибки:
  + Правило неприменимо для текущего символа.
  + Остались неиспользованные правила из последовательности.

**Задание 4**

В грамматике заменить 1. S→Ssa на S→SSa и выполнить задание.

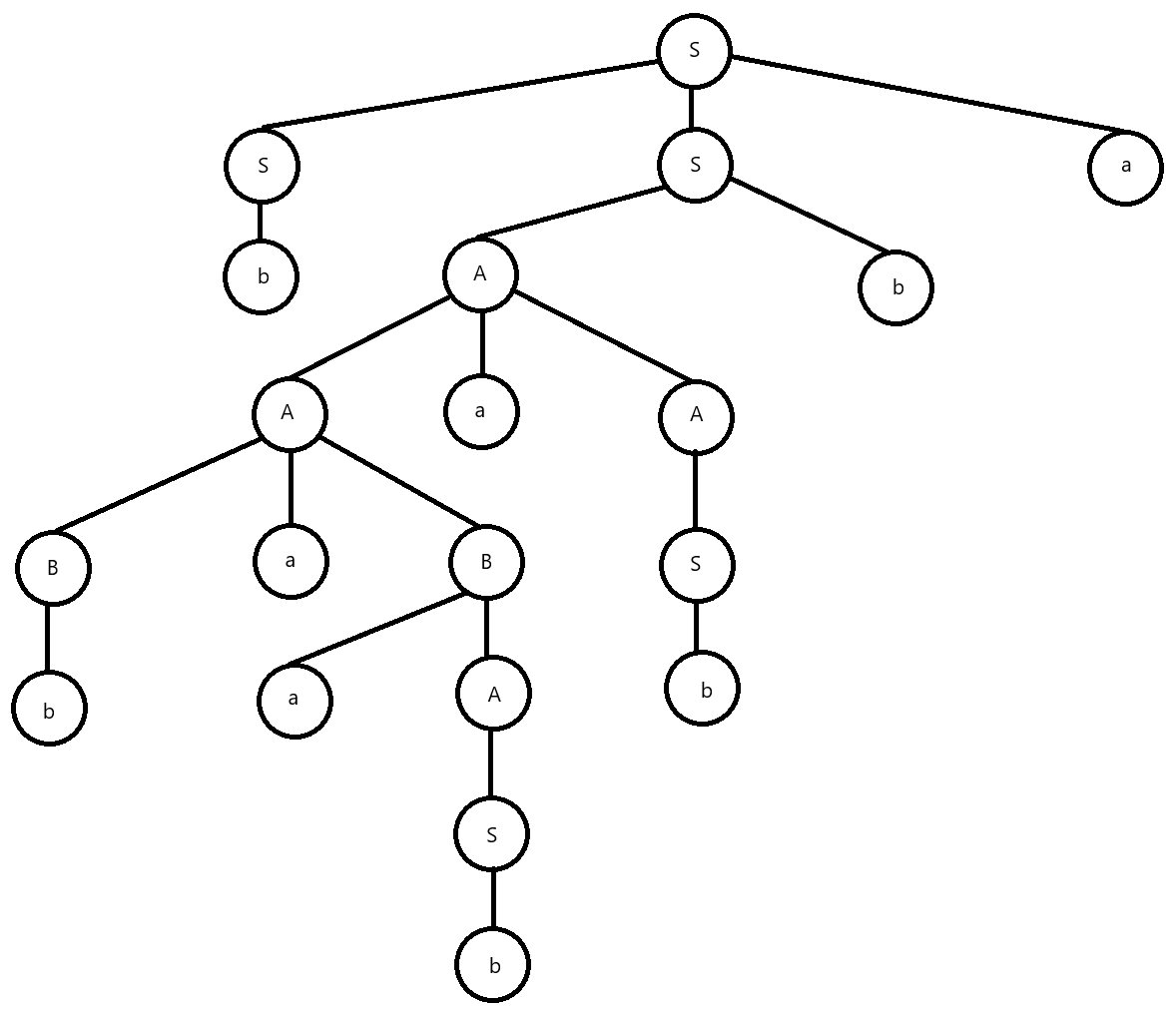
**Ответ на задание 4**

Заменяем правило: 1.S→ Ssa на S→SSa

**Левый вывод:**

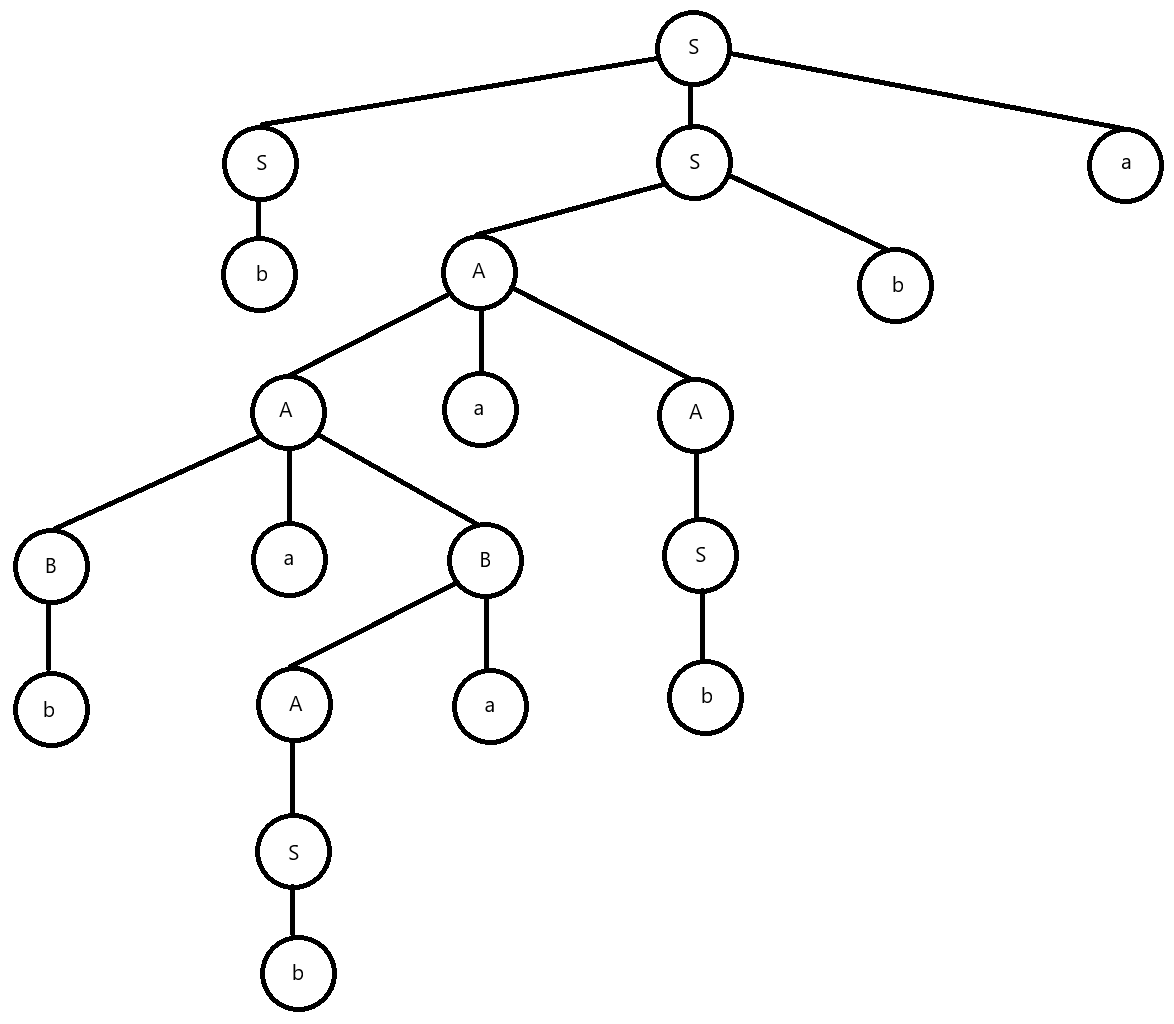
**Последовательность правил 1**: 1,2,3,4,5,6,7,8,6,2,2

ЛСФ ДВ: S(S(b)S(A(A(B(b)aB(aA(S(b))))aA(S(b)))b)a)



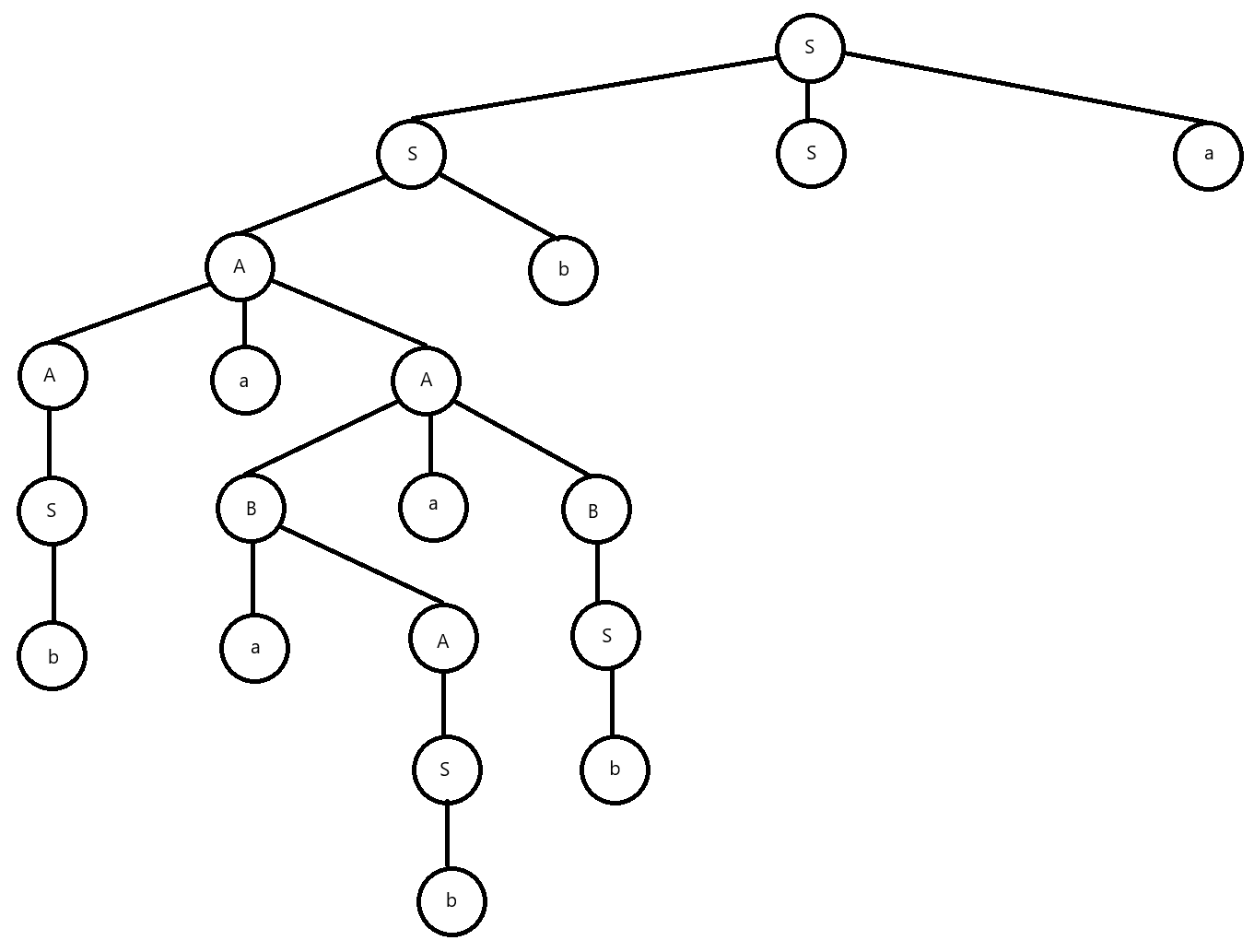
**Последовательность правил 2**: 1,2,3,4,5,7,8,6,2,6,2

ЛСФ ДВ: S(S(b)S(A(A(B(b)aB(aA(S(b))))aA(S(b)))b)a)



**Последовательность правил 3**: 1,3,4,6,2,5,8,6,2,7,2

ЛСФ ДВ: S(S(A(A(S(b))aA(B(aA(S(b)))aB(b)))b)S(b)a)



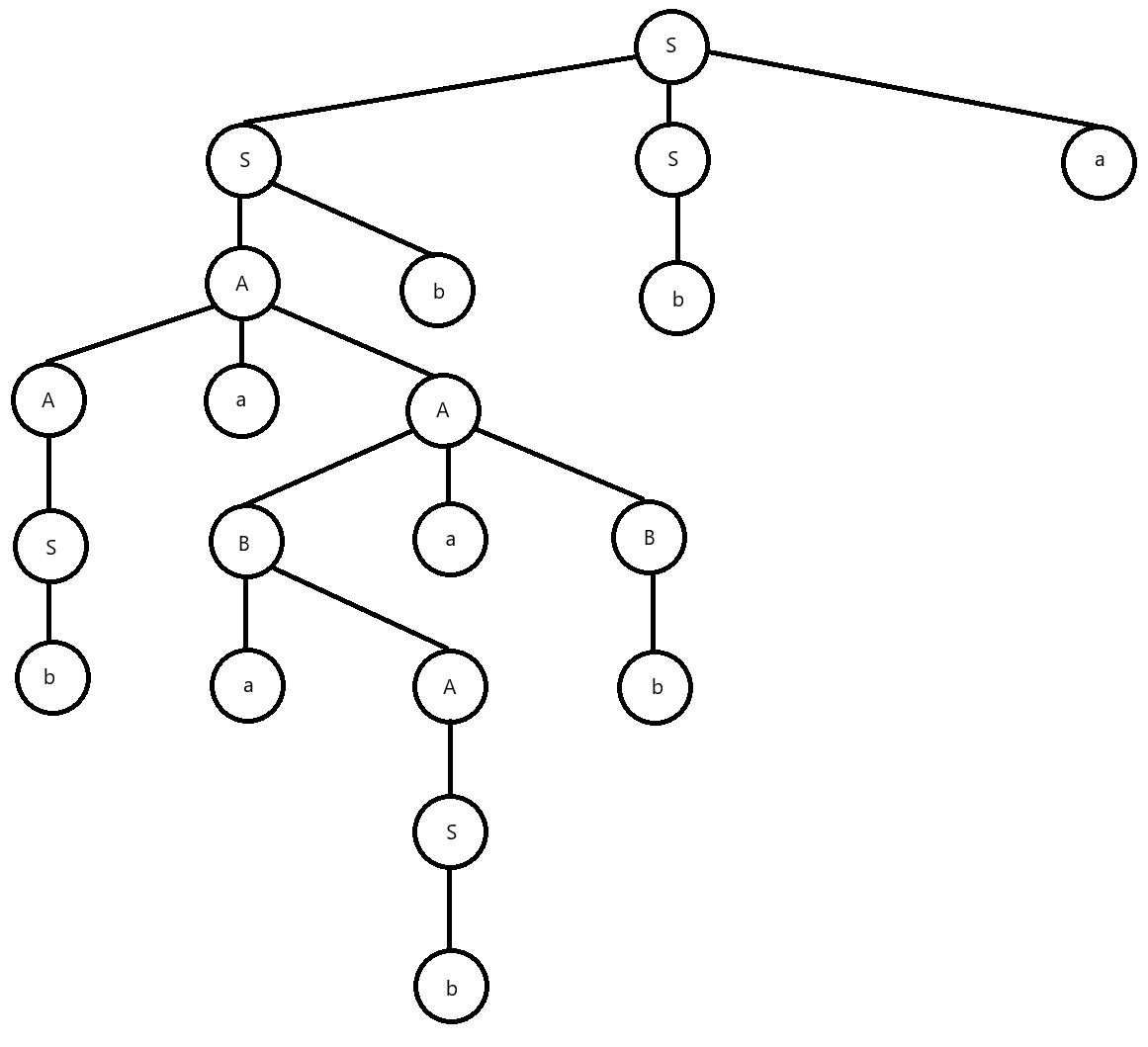
**Последовательность 4:** 1,2,3,4,6,6,2,2,7,8,6

нельзя использовать данную последовательность при левом выводе, тк после применения второго правила получили терминальную цепочку.

**Правый вывод:**

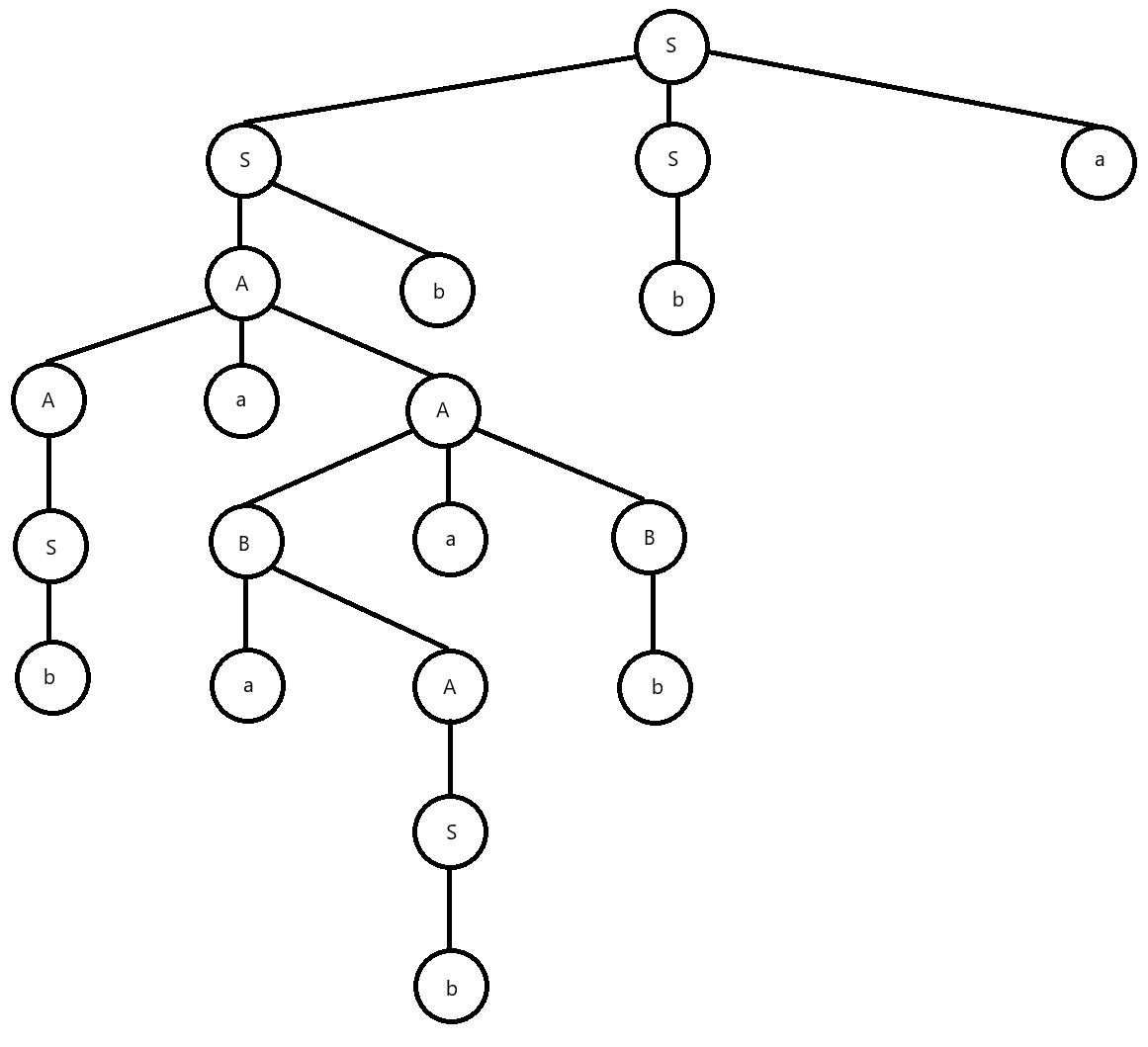
**Последовательность правил 1**: 1,2,3,4,5,6,7,8,6,2,2

ПСФ ДВ: S(S(A(A(S(b))aA(B(aA(S(b)))aB(b)))b)S(b)a)



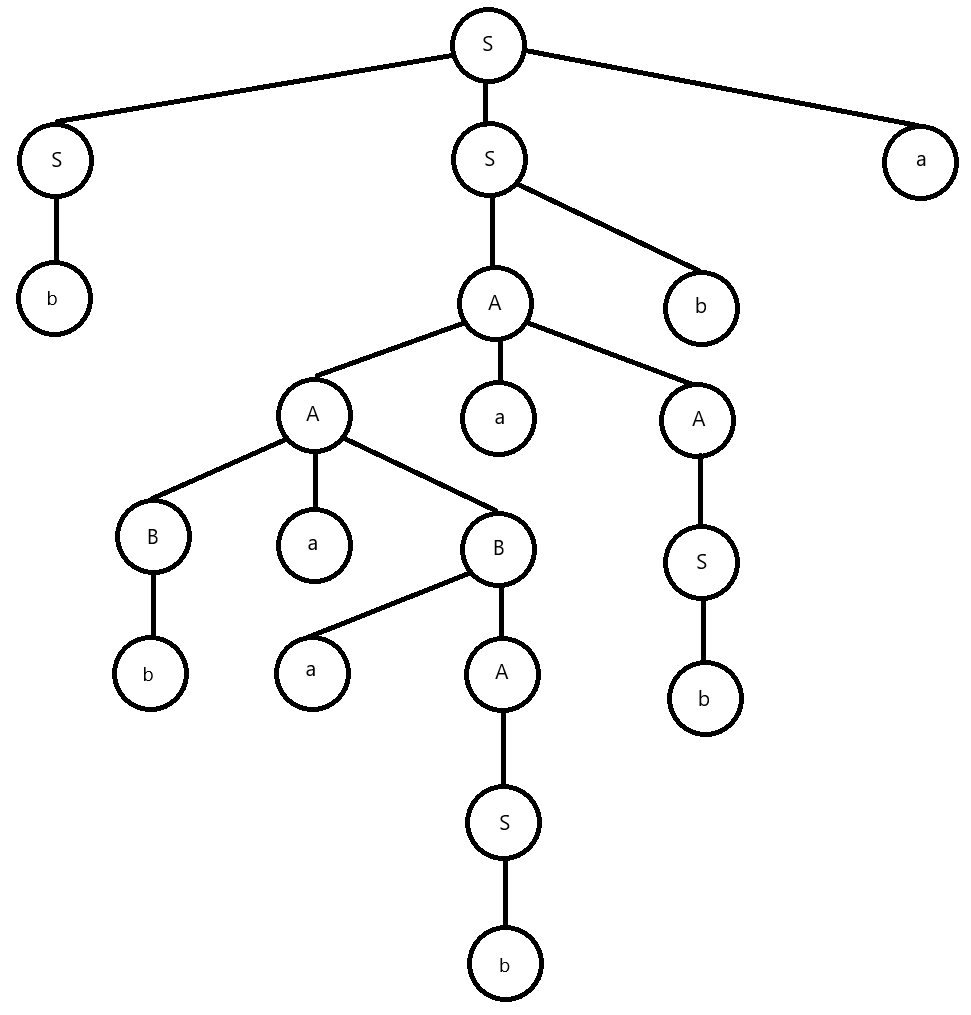
**Последовательность правил 2**: 1,2,3,4,5,7,8,6,2,6,2

ПСФ ДВ: S(S(A(A(S(b))aA(B(aA(S(b)))aB(b)))b)S(b)a)



**Последовательность правил 3**: 1,3,4,6,2,5,8,6,2,7,2

ПСФ ДВ: S(S(b)S(A(A(B(b)aB(aA(S(b))))aA(S(b)))b)a)



**Последовательность 4:** 1,2,3,4,6,6,2,2,7,8,6

нельзя использовать данную последовательность при правом выводе, тк после применения второго правила получили терминальную цепочку.

**Задание 6**

В грамматике заменить 1. S→Ssa на S→SSa и выполнить задание.

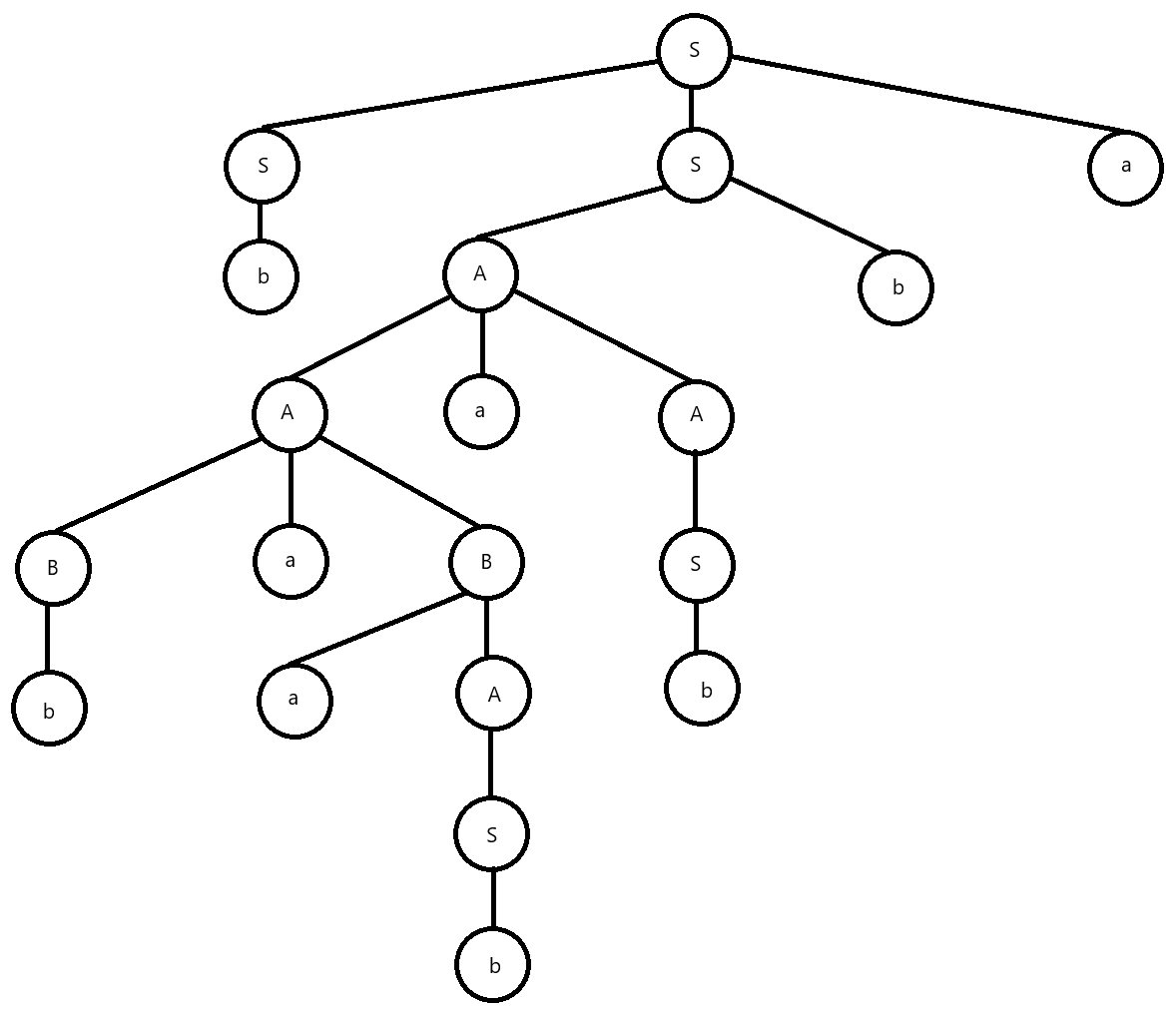
**Ответ на задание 4**

Заменяем правило: 1.S→ Ssa на S→SSa

**Левый вывод:**

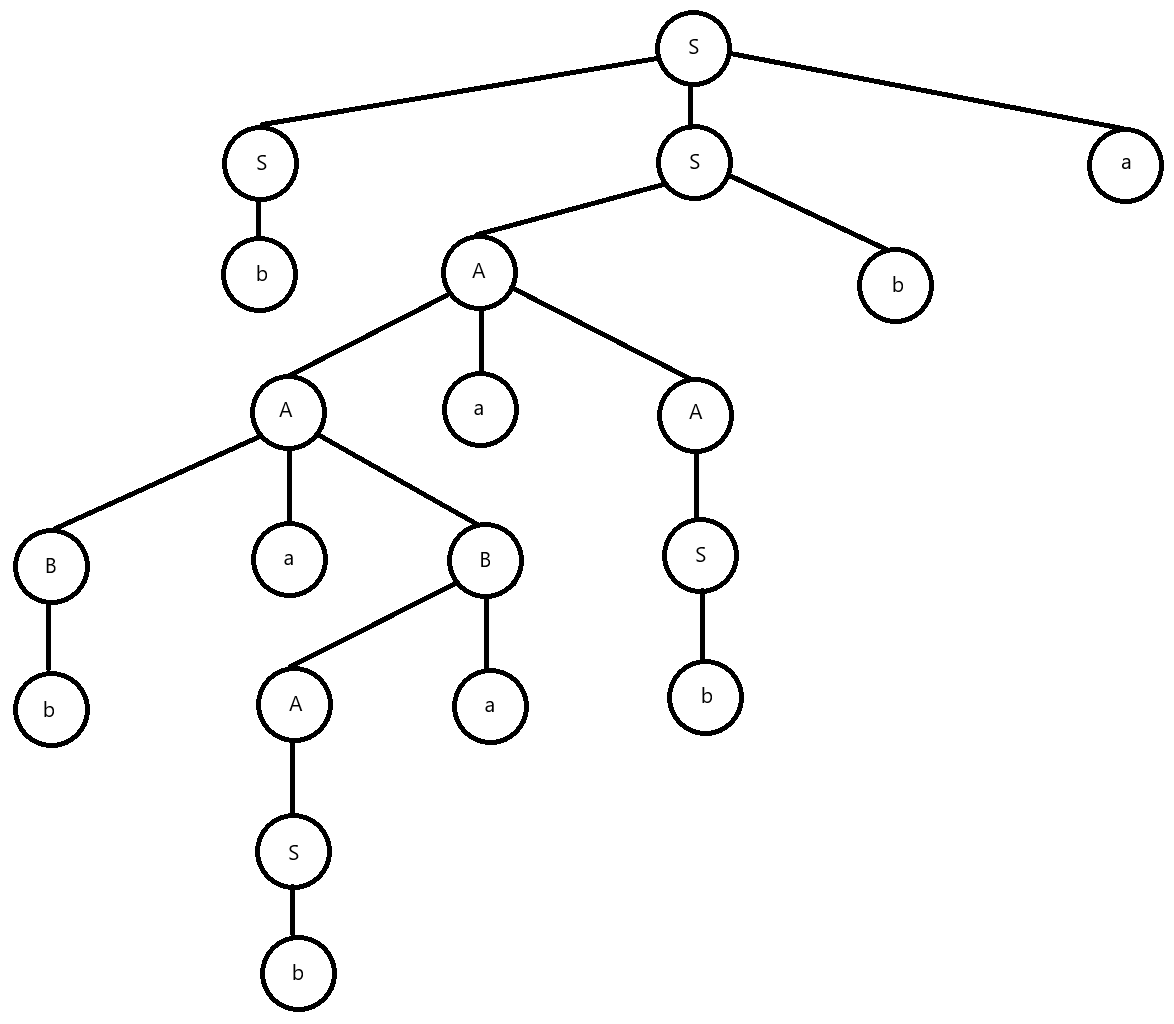
**Последовательность правил 1**: 1,2,3,4,5,6,7,8,6,2,2

ЛСФ ДВ: S(S(b)S(A(A(B(b)aB(aA(S(b))))aA(S(b)))b)a)



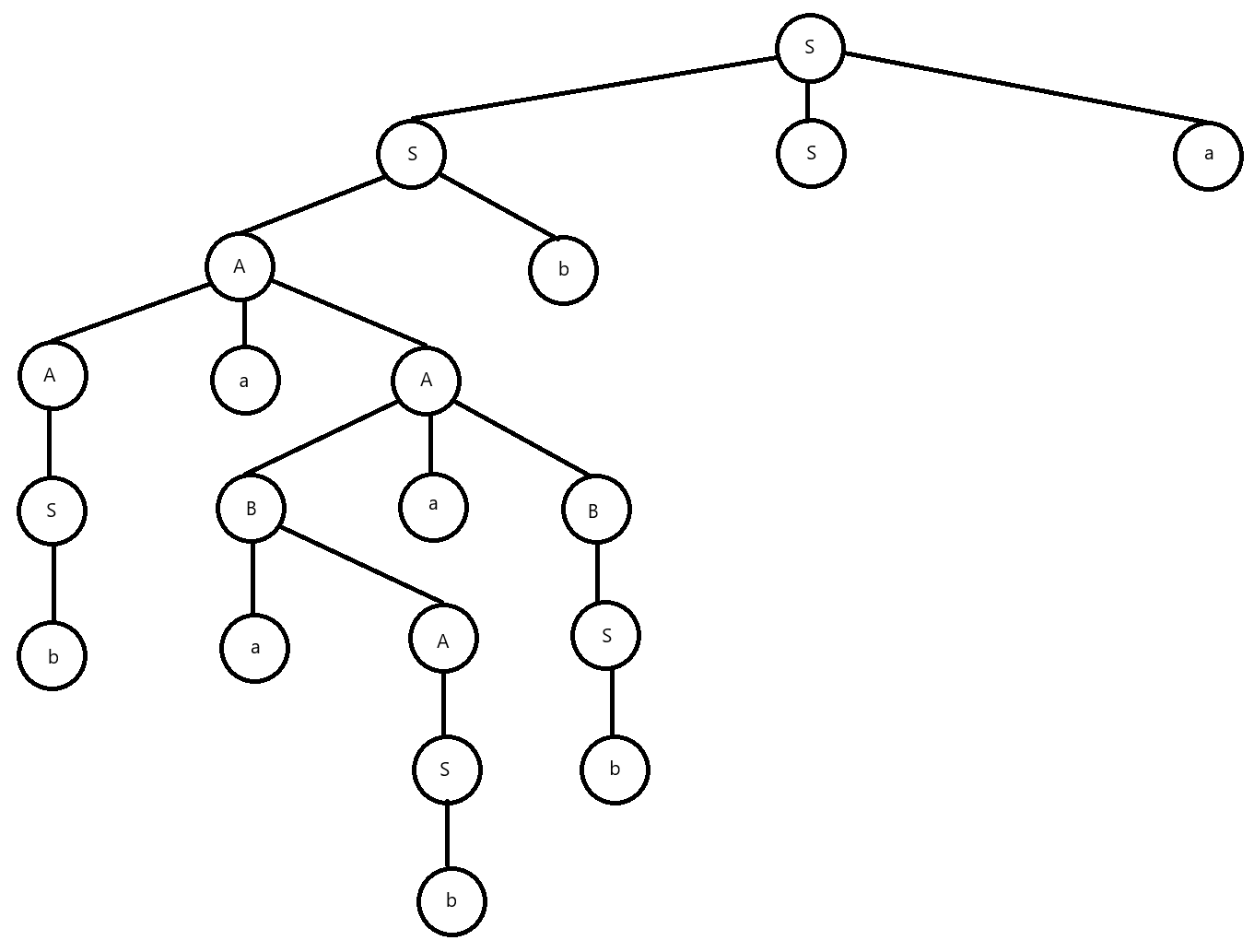
**Последовательность правил 2**: 1,2,3,4,5,7,8,6,2,6,2

ЛСФ ДВ: S(S(b)S(A(A(B(b)aB(aA(S(b))))aA(S(b)))b)a)



**Последовательность правил 3**: 1,3,4,6,2,5,8,6,2,7,2

ЛСФ ДВ: S(S(A(A(S(b))aA(B(aA(S(b)))aB(b)))b)S(b)a)



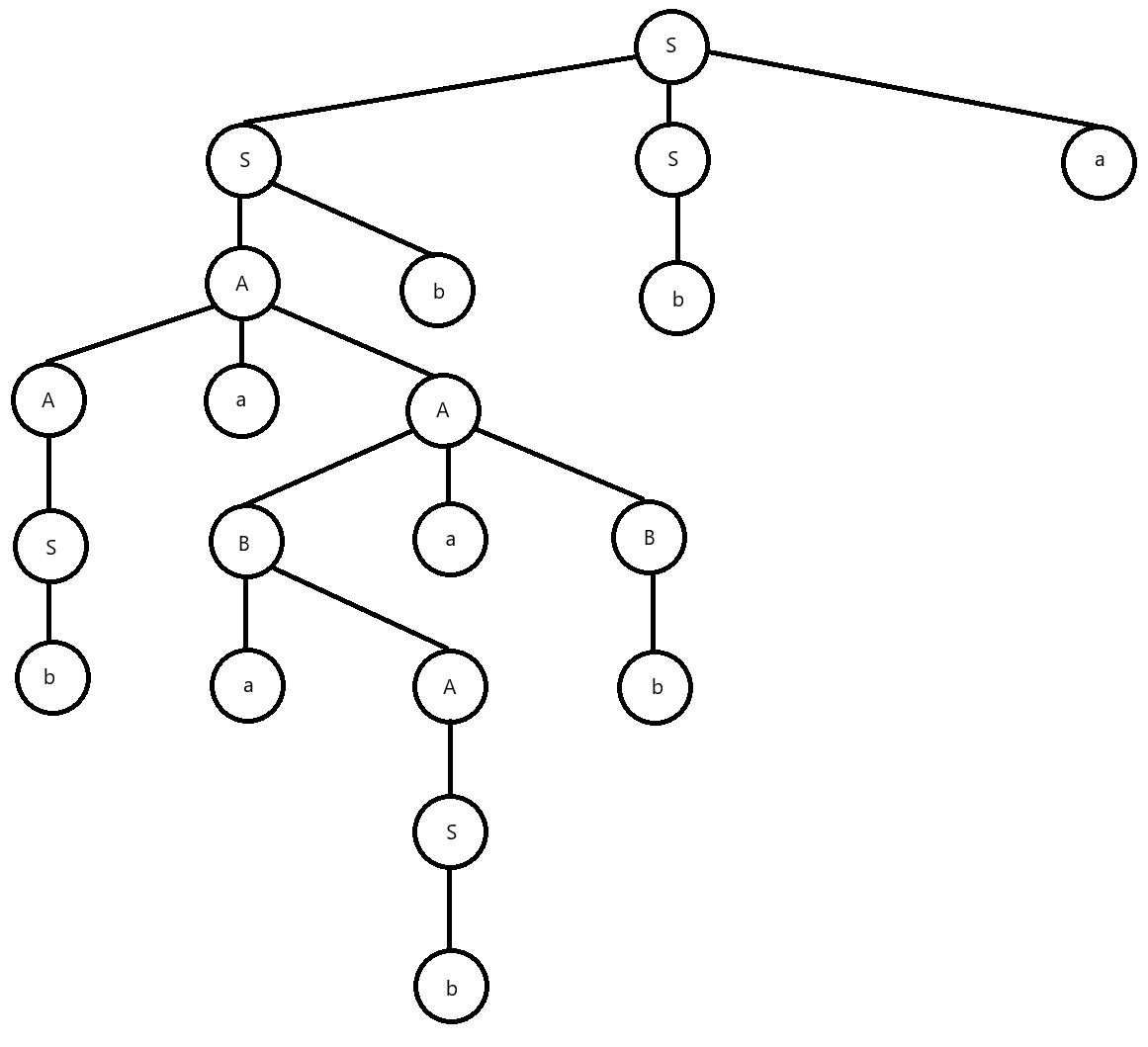
**Последовательность 4:** 1,2,3,4,6,6,2,2,7,8,6

нельзя использовать данную последовательность при левом выводе, тк после применения второго правила получили терминальную цепочку.

**Правый вывод:**

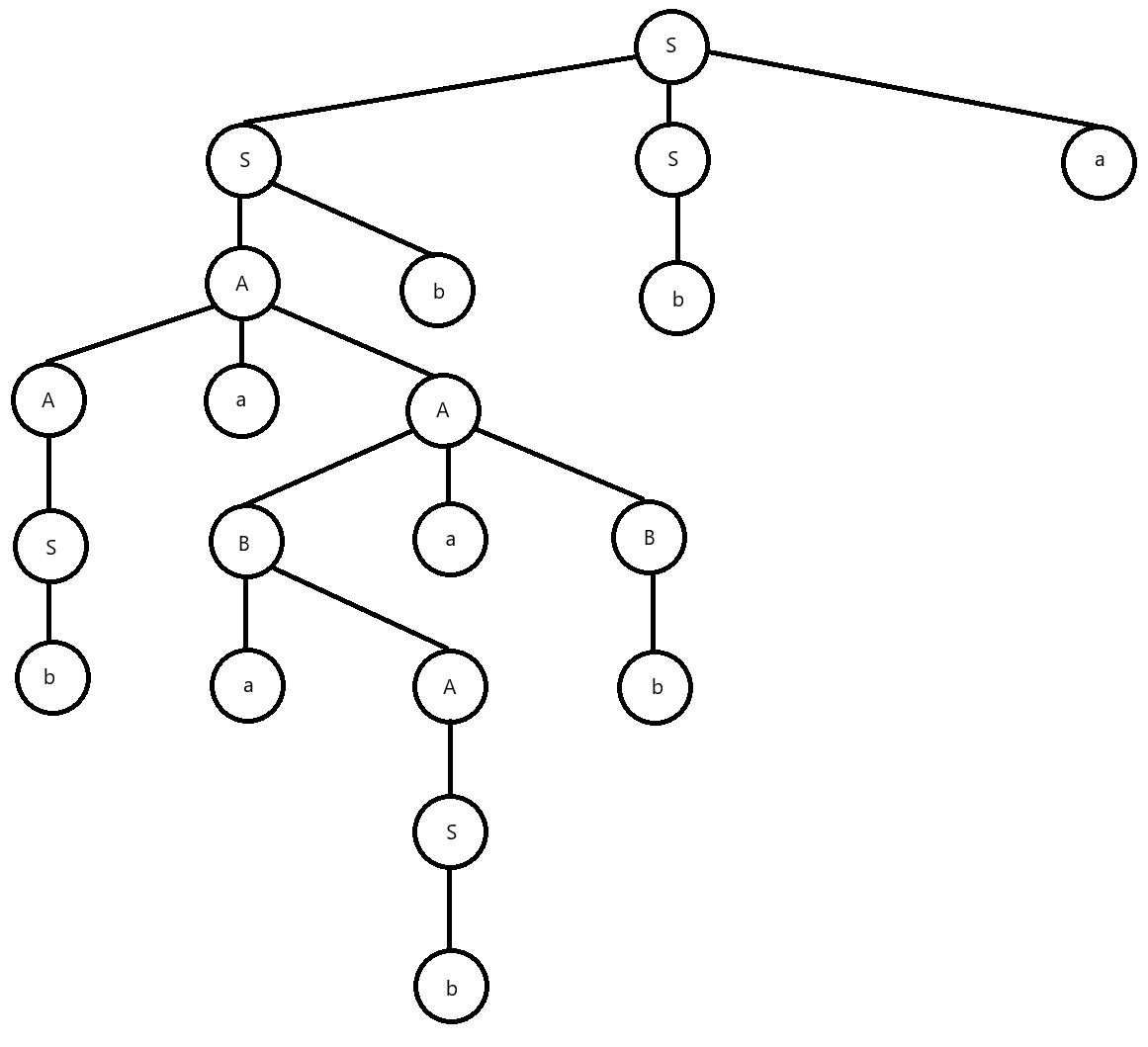
**Последовательность правил 1**: 1,2,3,4,5,6,7,8,6,2,2

ПСФ ДВ: S(S(A(A(S(b))aA(B(aA(S(b)))aB(b)))b)S(b)a)



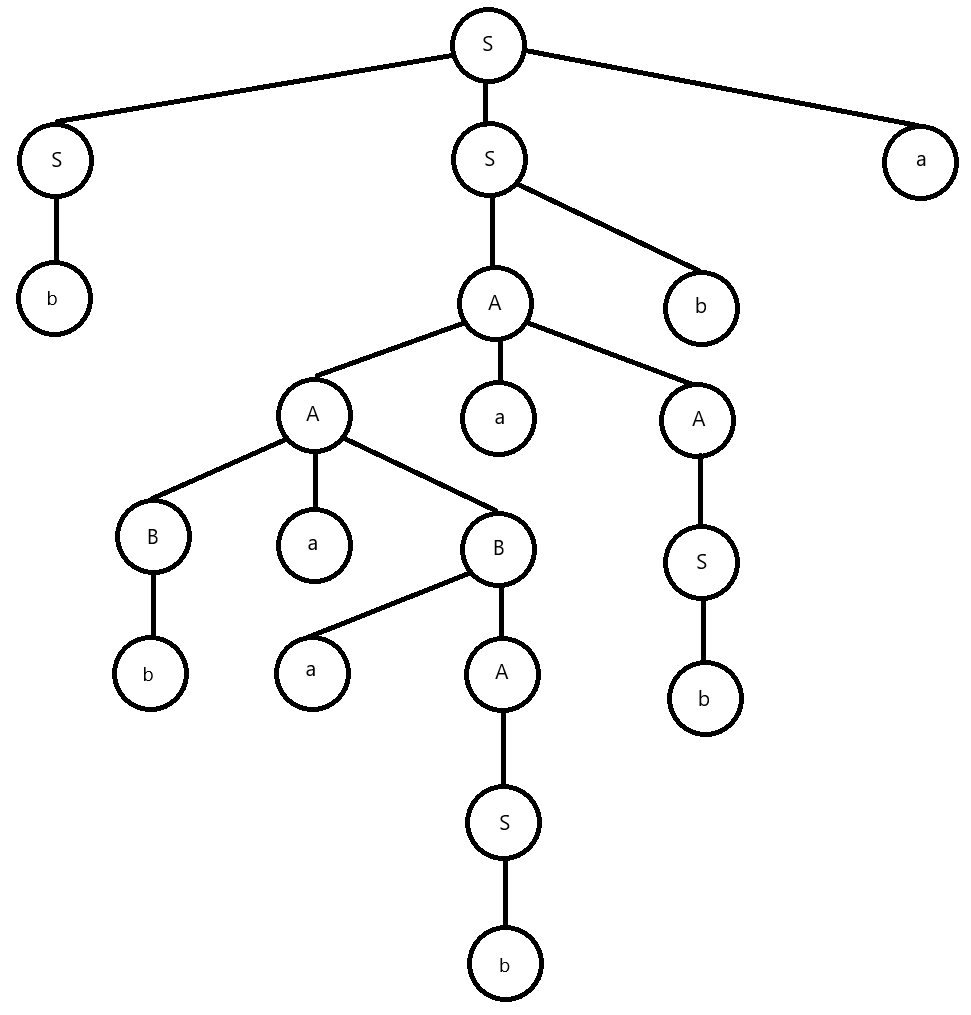
**Последовательность правил 2**: 1,2,3,4,5,7,8,6,2,6,2

ПСФ ДВ: S(S(A(A(S(b))aA(B(aA(S(b)))aB(b)))b)S(b)a)



**Последовательность правил 3**: 1,3,4,6,2,5,8,6,2,7,2

ПСФ ДВ: S(S(b)S(A(A(B(b)aB(aA(S(b))))aA(S(b)))b)a)



**Последовательность 4:** 1,2,3,4,6,6,2,2,7,8,6

нельзя использовать данную последовательность при правом выводе, тк после применения второго правила получили терминальную цепочку.