start(□)+(send+wait+show)\*□+stop - регулярное выражение

задание 3:

start, □,□,□send,show, □,□stop

start, □, λ, □, stop

start, □, □, send, wait, send, wait, □, stop - вот эту цепочку я буду разбирать.

start, □, wait, wait, □, stop

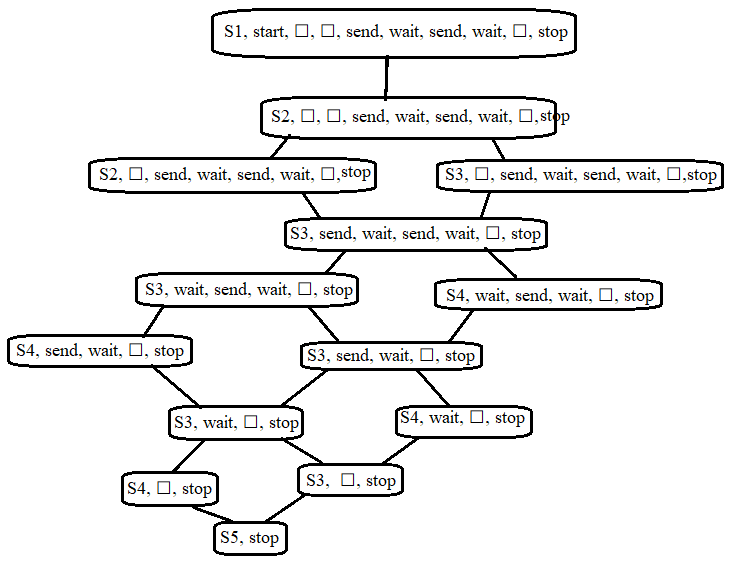
start, □, show, □,□,□, stop

start, □, □, □, □, □, □, □, □, □, λ, □, stop

start, □, show, wait, send, □, stop

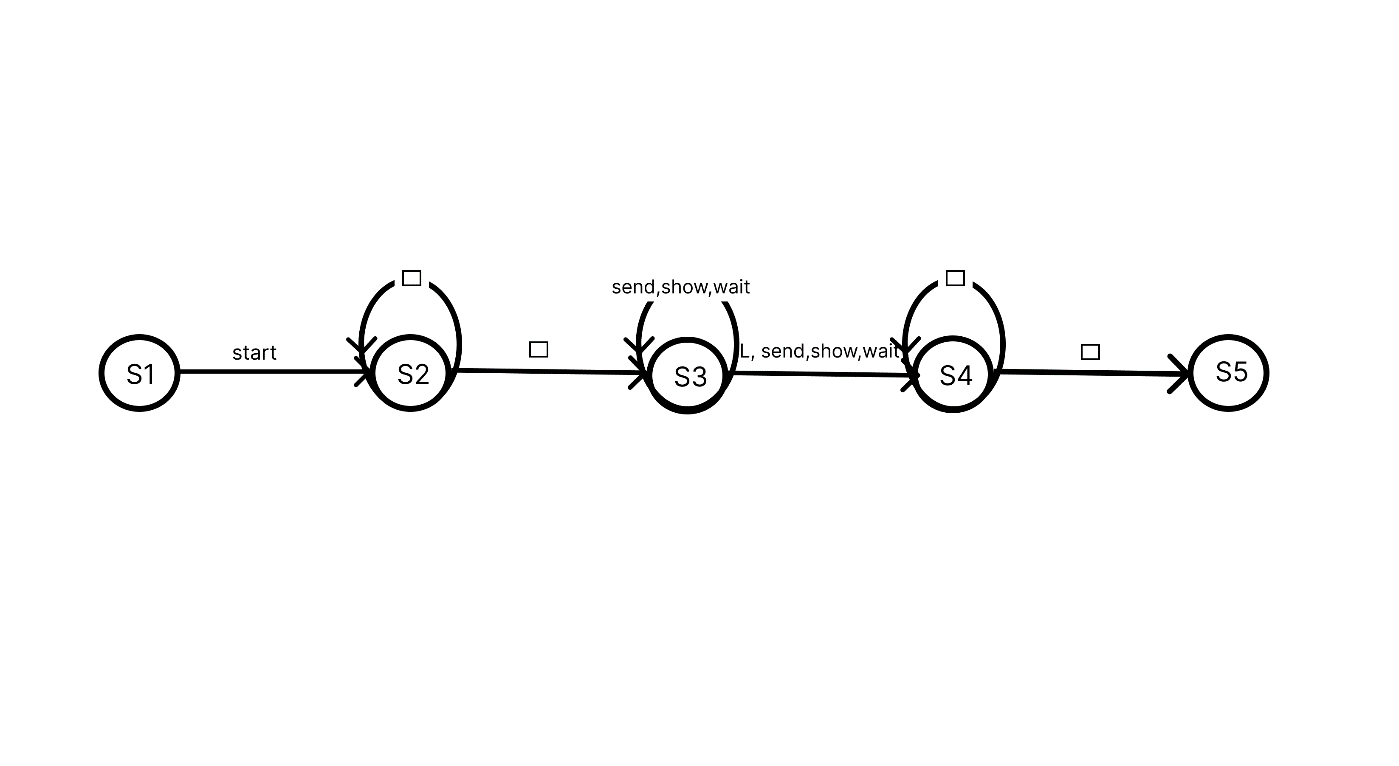
задание 4:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | start | □ | λ | send | show | wait | stop |
| S1 | {S2} |  |  |  |  |  |  |
| S2 |  | {S2, S3} |  |  |  |  |  |
| S3 |  |  | {S4} | {S3, S4} | {S3, S4} | {S3, S4} |  |
| S4 |  | {S4, S5} |  |  |  |  |  |
| S5 |  |  |  |  |  |  |  |



Задание 5:

start(□)+(send+wait+show)\*□+stop



Ответы на вопросы:

1. Алфавит – конечное непустое множество допустимых символов языка (букв языка).
2. λ- это пустота, I+ - цепочки повторяются от одного до бесконечности раз, I- - цепочка может повторяться или 0 или бесконечно раз.
3. Языком L(I) над алфавитом I называется произвольное подмножество цепочек из \* I , \* L(I) ⊆ I .
4. 2). Запись α ⇒ β \* предполагает n ≥ 0 шагов вывода β из α. В том случае, если α ⇒ β , то число шагов вывода n = 0. Запись α ⇒ β + предполагает n > 0 шагов вывода β из α
5. L(G) – язык, порождаемый грамматикой G. Язык L(G) содержит все терминальные цепочки, выводимые из S : ( ) { | \* } \* L G = α∈T S ⇒ α . Язык, порождаемый грамматикой – это множество всех выводимых из аксиомы грамматики терминальных цепочек

7. Форма Бэкуса — Наура (сокр. БНФ, Бэкуса — Наура форма) – формальная система описания синтаксиса, в которой одни синтаксические категории последовательно определяются через другие категории.

8. Грамматика языка описывает множество правильных цепочек симво лов над заданным алфавитом

9. Множество цепочек, описанных регулярным выражением, называется регулярным множеством (или регулярным языком)

10. Регулярное выражение описывает множество цепочек – формальный язык. Для записи регулярного выражения используются метасимволы.

11.Лексический анализ – первая (наиболее простая) фаза трансляции. Лексический анализ выполняется программой (входящей в состав транслятора), называемой лексическим анализатором (сканером)

12.**Лексический анализатор** ([англ.](https://dic.academic.ru/dic.nsf/ruwiki/6161) *lexical analyzer* или коротко *lexer*) — это программа или часть программы, выполняющая лексический анализ

***Лексический анализатор*** (или сканер) – это часть компилятора, которая читает литеры программы на исходном языке и строит из них слова (лексемы). На вход лексического анализатора поступает текст исходной программы, а выходная информация передается для дальнейшей обработки компилятором на этапе синтаксического анализа.

13.  На вход лексического анализатора поступает текст исходной программы, а выходная информация передается для дальнейшей обработки компилятором на этапе синтаксического анализа и разбора. Выходная-Протокол работы лексического анализатора

14. При последовательном варианте лексический анализатор просматривает весь текст исходной программы от начала до конца и преобразует его в структурированный набор данных.

При параллельном варианте лексический анализ исходного текста выполняется поэтапно так, что синтаксический анализатор, выполнив разбор очередной конструкции языка, обращается к сканеру за следующей лексемой.

16. 