1. The + operator is typically used to indicate union (|, or) in academic regular expressions, not "one or more" as it typically means in non-academic settings (such as more regular implementations)

So, a+b means [ab] or a|b, thus (a+b)* means any string of length 0 or more, containing any number of a's and b's in any order.

Likewise, (a*b*)* also means any string of length 0 or more, containing any number of a's and b's in any order.

The two expressions are different ways of expressing the same language.

2. Turing-recognizable = recursively enumerable

co-Turing-recognizable = co-recursively enumerable

3. decidable = recursive*

4. 构建图灵机时记得声明 turning machine 起始 position (at the tape)

5. recursively enumerable languages is semi-decidable.

A language L is decidable if there is a Turing machine M such that L(M) = L and M halts on every input.

Thus, if L is decidable then L is recursively enumerable.

但不代表所有的 REL 都是 decidable 的。

6.Closure Properties of Regular languages:

Closure properties express the idea that when one (several) languages are regular, then certain language obtained by some operation is also regular.

1)The difference of two regular languages is regular.

2)Union: If L1 and If L2 are two regular languages, their union L1 ∪ L2 will also be regular. For example,

$L1 = \{a^n \mid n \geq 0\}$ and $L2 = \{bn \mid n \geq 0\}$

$L3 = L1 \cup L2 = \{a^n \cup b^n \mid n \geq 0\}$ is also regular.

3)Intersection: If L1 and If L2 are two regular languages, their intersection L1 ∩ L2 will also be regular. For example,

$L1= \{a^m b^n \mid n \geq 0 \text{ and } m \geq 0\}$ and $L2= \{a^m b^n \cup b^n a^m \mid n \geq 0 \text{ and } m \geq 0\}$

$L3 = L1 \cap L2 = \{a^m b^n \mid n \geq 0 \text{ and } m \geq 0\}$ is also regular.

4) The reversal of a regular languages is regular.

5)Concatenation: If L1 and If L2 are two regular languages, their concatenation L1.L2 will also be regular. For example,

$L1 = \{a^n \mid n \geq 0\}$ and $L2 = \{b^n \mid n \geq 0\}$

$L3 = L1.L2 = \{a^m . b^n \mid m \geq 0 \text{ and } n \geq 0\}$ is also regular.

6)Kleene Closure: If L1 is a regular language, its Kleene closure L1* will also be regular. For example,

$L1 = (a \cup b)$

$L1^* = (a \cup b)^*$

7)Complement: If L(G) is regular language, its complement L'(G) will also be regular. Complement of a language can be found by subtracting strings which are in L(G) from all possible strings. For example,

$L(G) = \{a^n \mid n > 3\}$

$L'(G) = \{a^n \mid n <= 3\}$

PS: Apply Closure Properties to Prove a language is not regular:

To prove a language L, is not regular, we can find a regular language L' and operate it using (Union, difference, etc...) with L, if the resulted language is not regular, then L is not regular.

Context-free Language

Context-free languages are **closed** under −

- Union
- Concatenation
- Kleene Star operation

Union

Let L_1 and L_2 be two context free languages. Then $L_1 \cup L_2$ is also context free.
Example:
Let $L_1 = \{ a^nb^n , n > 0\}$. Corresponding grammar G_1 will have P: $S1 \rightarrow aAb|ab$
Let $L_2 = \{ c^md^m , m \geq 0\}$. Corresponding grammar G_2 will have P: $S2 \rightarrow cBb| \varepsilon$
Union of L_1 and L_2, $L = L_1 \cup L_2 = \{a^nb^n\} \cup \{c^md^m\}$
The corresponding grammar G will have the additional production $S \rightarrow S1 | S2$

Concatenation

If L_1 and L_2 are context free languages, then L_1L_2 is also context free.
Example:
Union of the languages L_1 and L_2, $L = L_1L_2 = \{a^nb^nc^md^m\}$
The corresponding grammar G will have the additional production $S \rightarrow S1\ S2$

Kleene Closure

If L is a context free language, then L* is also context free.
Example:
Let $L = \{a^nb^n , n \geq 0\}$. Corresponding grammar G will have P: $S \rightarrow aAb| \varepsilon$
Kleene Star $L_1 = \{a^nb^n \}^*$
The corresponding grammar G_1 will have additional productions $S1 \rightarrow SS_1 | \varepsilon$

Context-free languages are **not closed** under −

- **Intersection** − If L1 and L2 are context free languages, then L1 ∩ L2 is not necessarily context free.
- **Complement** − If L1 is a context free language, then L1’ may not be context free.

Intersection and complementation:
For example,
$L1 = \{ a^nb^nc^m \mid n >= 0 \text{ and } m >= 0 \}$ and $L2 = (a^mb^nc^n \mid n >= 0 \text{ and } m >= 0 \}$
$L3 = L1 \cap L2 = \{ a^nb^nc^n \mid n >= 0 \}$ need not be context free. (Proved by Pumping Lemma)
Similarly, complementation of context free language L1 which is $\sum^* – L1$, need not be context free.

Intersection with Regular Language − If L1 is a regular language and L2 is a context free language, then L1 ∩ L2 is a context free language.

7. Languages recognized can be accepted by a PDA it is a context free language and if it can be accepted by a DPDA it is a deterministic context-free language (DCFL). Not all context-free languages are deterministic. This makes the DPDA a strictly weaker device than the PDA.

8. A TM accepts a language if it enters into a final state for any input string w. A language is recursively enumerable (generated by Type-0 grammar) if it is accepted by a Turing machine.

A TM decides a language if it accepts it and enters into a rejecting state for any input not in the language. A language is recursive if it is decided by a Turing machine.

There may be some cases where a TM does not stop. Such TM accepts the language, but it does not decide it.

8.

a) Using Pumping Lemma to prove the Language A is not regular.

Assume that A is regular, it has to have a Pumping Length P.
All string longer than P can be pumped |S| >= P
Find a string S in A such that |S| >= P
Divide S in to three parts x y z
If A is regular language the following condition should be true:

1) xy^iz 属于 A,
2) $|y| > 0$
3) $|xy| <= P$

Show that $x\ y^i\ z$ 不属于 A for some i
Then consider all ways that S can be divided into xyz
Show that none of them these can satisfy all the 3 pumping conditions at the same time.
S cannot be Pumped = A is not regular language.