1. **(16%)** Suppose there are four languages A, B, C and D. Each of these languages may or may not be recursively enumerable. However, we know the following about them:

   i. There is a reduction from A to B

   ii. There is a reduction from B to C

   iii. There is a reduction from D to C

   Below are eight statements indicate whether each is:

   - CERTAIN to be true: regardless of what problems$A$, B, C and D are
   - MAYBE true: depending on what A, B, C and D are
   - NEVER true: regardless of what A, B, C and D are

   (a) ( ) A is recursively enumerable but not recursive, and C is recursive.

   (b) ( ) A is not recursive and D is not recursively enumerable.

   (c) ( ) The complement of A is not recursively enumerable, but the complement of B is recursively enumerable.

   (d) ( ) The complement of B is not recursive, but the complement of C is recursive.

   (e) ( ) If A is recursive, then the complement of B is recursive.

   (f) ( ) If C is recursive, then the complement of D is recursive.

   (g) ( ) If C is recursively enumerable, then the union of B and D is recursively enumerable.

   (h) ( ) If C is recursively enumerable, then the intersection of B and D is recursively enumerable.

解析：这是一条变相的选择题。已知存在 A 到 B、B 到 C、D 到 C 的归约，显然也有 A 到 C 的归约。然后选项有三个选项：必然是真的 CERTAIN、可能是真的 MAYBE（视 ABCD 的具体情况）、肯定不真 NEVER。

（a）NEVER 定理 5.4.1，L1 非递归，存在 L1 到 L2 的归约，则 L2 也非递归。因此 A 不是递归的 C 肯定非递归。

（b）MAYBE A 非递归那么 C 非递归。D 不是递归可枚举的那么 C 也应该不是递归可枚举。然后这就视 C 的具体情况，如果 C 是递归可枚举的（例如 C 是停机问题）那么就不符合后者，C 不是递归可枚举的那么都符合。

（c）MAYBE 前半句可推断 A 不是递归的，因为递归语言对补封闭，因此 B 也不是递归的。后半句推断，B 要么是递归的，要么连递归可枚举都不是。前者与前半句相悖，后者与前半句相符。因此 MAYBE。

（d）NEVER 前半句推断 B 不是递归的，因为递归对补封闭，也因此 C 也不是递归。后半句推断 C 递归。

（e）MAYBE B 可能是递归的也可能是递归可枚举但非递归，这两种情况 A 都可以归约到 B，但是不一定能推出 B 是递归的，自然 B 的补不一定递归。

（f）CERTAIN C 是递归，那么 D 是递归的，D 的补也是递归的。

（g）CERTAIN C 是递归可枚举，那么 B 和 D 都是递归可枚举，因此 B 和 D 的并都是递归可枚举。

（h）CERTAIN B 和 D 的交也是递归可枚举。

2. **(14%)** Suppose there are four languages A, B, C and D. Each of these languages may or may not be in the class $\mathcal{NP}$. However, we know the following about them:

   i. There is a polynomial-time reduction from A to B
   ii. There is a polynomial-time reduction from B to C
   iii. There is a polynomial-time reduction from D to C

   Below are seven statements Indicate whether each is:

   • CERTAIN to be true, regardless of what problems A, B, C and D are and regardless of the resolution of unknown relationships among complexity classes of "which is $\mathcal{P} = \mathcal{NP}$" is one example.

   • MAYBE true, depending on what A, B, C and D are and/or depending on the resolution of unknown relationships such as $\mathcal{P} = \mathcal{NP}$?

   • NEVER true, regardless of what A, B, C and D are and regardless of the resolution of unknown relationships such as $\mathcal{P} = \mathcal{NP}$?

   (a) ( ) If A is $\mathcal{NP}$-complete then C is $\mathcal{NP}$-complete.

   (b) ( ) A is $\mathcal{NP}$-complete and C is in $\mathcal{P}$.

   (c) ( ) B is $\mathcal{NP}$-complete and D is in $\mathcal{P}$.

   (d) ( ) If A is $\mathcal{NP}$-complete and B is in $\mathcal{NP}$ then B is $\mathcal{NP}$-complete.

   (e) ( ) If C is $\mathcal{NP}$-complete then D is in $\mathcal{NP}$.

   (f) ( ) C is in $\mathcal{P}$ and the complement of D is not in $\mathcal{P}$.

   (g) ( ) B is not in $\mathcal{P}$ and A is not in $\mathcal{NP}$.

（a） MAYBE 定理 7.1.1，两次多项式时间归约还是多项式时间归约。因此，A 可以多项式时间归约到 C。如果 C 是 NPC 那么 A 是 NPC，但是逆命题不一定成立。

（b） MAYBE 如果 C 是 P 那么 A 也是 P 的。如果 NP=P=NPC，那么可以是真的。

（c） MAYBE 如果 NP=P=NPC，那么无论 C 是 NPC 还是 P 都成立。否则不成立。

（d） CERTAIN 定义 7.1.2，NPC 的语言是每个 NP 的语言都可以多项式归约到的。A 是 NPC，A 可以多项式时间归约到 B，那么 B 自然也是 NP 的。

（e） CERTAIN 根据上述定义 7.1.2 可知。

（f） NEVER 显然 P 语言对补运算封闭。C 是 P 那么 D 也是 P，D 的补自然也是 P。

（g） MAYBE 如果 NP 是 EXP 的真子集，A 和 B 都在 EXP 而不属于 NP，那么命题成立。如果 B 是 NPC，那么 A 至少是 NP 的。因此 MAYBE。

3. **(12%)** Consider the binary operator $\circ$ on languages as follows: given two languages L_1 and L_2 over Σ, $L_1 \circ L_2$ consists of words of the form uv such that $u \in L_1$, $v \in L_2$ and $|u| = |v|$.

   (a) Prove that if L_1 and L_2 are regular languages, then $L_1 \circ L_2$ is context-free.

   (b) Give a counter-example to disprove that if L_1 is a regular language and L_2 is a context-free language, then $L_1 \circ L_2$ is context-free.

（a）通过构造接受 L1°L2 的下推自动机 M 来证明其是上下文无关的。

设 DFA M1=（K1，Σ1，δ1，s1，F1）接受 L1，M2=（K2，Σ2，δ2，s2，F2）接受 L2，

那么 M=（K，Σ，Γ，Δ，s，F），其中 K=K1∪K2，Σ=Σ1∪Σ2，Γ=Σ1，s=s1，F=F2
对于任意δ（p，a）=q ∈ δ1，则有（（p，a，e），（q，a））∈Δ；
对于任意δ（p，a）=q ∈ δ2，则有（（p，a，b），（q，b））∈Δ，b≠e；
并且任意 f∈F1，都有（（f，e，e），（s2，e））∈Δ。
这样，M 就接受了 L1°L2，当输入 uv∈L1°L2，首先 M 模拟 M1 的操作并且把 u 的所有字符压栈，然后 M 会模拟 M2 的操作并没读到一个 v 的字符就出栈一次。如果最后读完了栈为空，那么接受这个字符串，否则拒绝。
（b）L1={w|w 中 a 和 b 的个数模 2 不同余}，L2={w|w 中 a 和 b 的个数相等}，因此 L1°L2 依然是{w|w 中 a 和 b 的个数模 2 不同余}。

4. **(20%)**

(a) Give a context-free grammar for the language

$$L_3 = \{a^m b^m c a^{2n} b^{2n} \mid m, n \in \mathbb{N}\}.$$

(b) Design a PDA $M = (K, \Sigma, \Gamma, \Delta, s, F)$ accepting the language L_3.

(a) G=（V，Σ，R，S）
V={S，S1，S2，a，b，c}
Σ={a，b，c}
R={
S→S1cS2
S1→aS1b
S2→aaS2bb
S1→e
S2→e
}
不要忘了最后把非终结符变 e

(b) M=（K，Σ，Γ，Δ，S，F）
K={p，q}
Σ={a，b，c，S，S1，S2}
Γ={a，b，c，S，S1，S2}
Δ={
((p,e,e),(q,S))
((q,e,S),(q,S1cS2))
((q,e,S1),(q,aS1b))
((q,e,S2),(q,aaS2bb))
((q,e,S1),(q,e))
((q,e,S2),(q,e))
((q,a,a),(q,e))
((q,b,b),(q,e))
((q,c,c),(q,e))
}

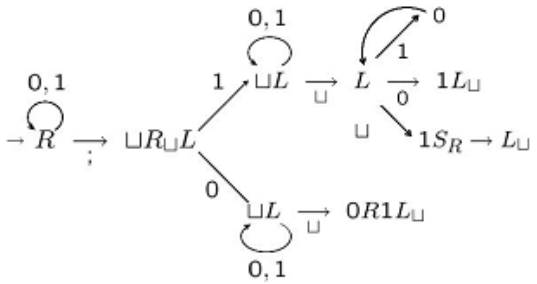
S={p}
F={q}

方法请参照课本 88 页最上方

5. **(10%)** Show that the following language
   $\{\text{“}M_1\text{”}\,\text{“}M_2\text{”} \mid M_1, M_2 \text{ are Turing machines and both } M_1 \text{ and } M_2 \text{ halt on blank tape}\}$
   is recursively enumerable. An informal description suffices.

解析：用一个多带的通用图灵机可以半判定这个语言。
具体就是对于输入的图灵机编码分别复制到两条带上面，然后通用图灵机开始在另外两条带上模拟两个图灵机在空输入上的操作，M1 做一步的同时 M2 就做一步。如果两个图灵机都停机了，那么通用图灵机就停机了，否则不停机。
因此，通用图灵机半判定这个语言，因此这个语言是递归可枚举的。

6. **(16%)**Let the following Turing machine M compute function $f(x, y)$, where x and y are represented by binary strings respectively and separated with the symbol “;”, i.e. the initial configuration in form of $\triangleright\underline{\sqcup}x; y$.



(a) Describe the key configurations when M started from the configuration $\triangleright\underline{\sqcup}10111; 10$.
(b) Try to give the function $f(x, y)$ computed by Turing Machine M.

Y 是奇数则 x+1，y 是偶数则 4x+1
（a）
▷_10111;10
▷ 10111;10
▷ 10111 10
▷ 10111_
▷_1011101
（b）
当 y 是奇数，f(x,y)=x+1
当 y 是偶数，f(x,y)=4x+1
奇偶数的判定很明显，上面的通路，遇 0 加 1，遇 1 变 0 则是进位，全 1 那么在最左边添 1 然后字符串右移一格，然后向左边找空格。下面的通路，相当于添两个 0 再加 1.

7. **(12%)** The **non-tautology, NT** problem is defined as follows: given a Boolean expression E, does there exist a truth-assignment for the variables of E that makes E false.
   (a) Prove NT is in $\mathcal{NP}$.
   (b) Describe a polynomial-time reduction from SAT to NT and show that NT problem is $\mathcal{NP}$-complete.

NT 的问题是，给定布尔表达式 E，是否存在真值赋值，使得表达式 E 为假。这是变相的可满足问题。

（a）存在非确定图灵机，可以随机生成各个布尔变量的真值，然后在多项式时间内代入验证 E 是否为 false。因此存在非确定图灵机在多项式时间内计算 NT，因此 NT∈NP。

（b）这个归约就是求反。SAT 是找真值赋值使得布尔表达式为真，求反则为假，变成了 NT 问题。求反当然是多项式时间内完成的。因为 SAT 是 NPC 问题，所有 NP 问题可以多项式时间归约到 SAT，SAT 又可以多项式时间归约到 NT，因此所有 NP 问题都可以多项式时间归约到 NT，因此根据定义 7.1.2，NT 是 NPC 的。