G 版

1. **(20%) Determine whether the following statements are true or false. If it is true write a T otherwise a F in the bracket before the statement.**

   (a) (   ) For all languages L_1, L_2 and L_3, if $L_1 \subseteq L_2 \subseteq L_3$ and both L_1 and L_3 are regular, then L_2 is also regular.

   (b) (   ) For a given context-free language L and a string x, the decision problem for whether $x \in \overline{L}$ is decidable.

   (c) (   ) The complement of every recursive enumerable language is necessarily nonrecursive enumerable.

   (d) (   ) Languages {"M" : Turing machine M accepts at least 2009 distinct inputs} is recursive enumerable.

   (e) (   ) Let L be a language and there is a Turing machine M halts on x for every $x \in L$, then L is decidable.

   (f) (   ) For all languages L_1 and L_2, if L_1 is in $\mathcal{P}$ and $L_1 <_p L_2$, then L_2 is in $\mathcal{NP}$.

   (g) (   ) The class $\mathcal{NP}$ is closed under complementation.

   (h) (   ) Checking equivalence of two propositional formulas is in $\mathcal{NP}$.

   (i) (   ) If L is polynomial time reducible to a finite language, then L is in $\mathcal{P}$.

   (j) (   ) If SAT reduces to a language L, then L is $\mathcal{NP}$-complete.

(a) F 假设 L1={aa}，L2={a^m|m 是素数}，L3={a*}，虽然满足题目条件，但是 L2 不是正则的。

(b) T 给定 L 和 x 以及接受 L 的 M，我们用通用图灵机来模拟 M 在 x 上的计算。如果 x∈L，那么 M 就接受 x，这是我们返回 no 表示 x∉L 补，反之 M 拒绝 x 就返回 yes。因此通用图灵机判定了这个问题。

(c) F 定理 5.7.1，递归语言其本身和补都是递归可枚举的。

(d) T 同 06 年的（g）

(e) F 这样只能说 M 半判定 L。当 x∈L 时 M 停机接受 x，且 x∉L 时 M 停机拒绝 x，那么才能说 M 判定 L。

(f) F L2 可以多项式时间归约到 L1，那么 L1 是 P 则 L2 是 NP。

(g) T NP 是递归的，递归对补封闭，因此 NP 对补也封闭。

(h) T 计算两个命题的等价性。显然需要对每个输入变量赋值的结果一一验证才可以，所以是 NP 的。

(i) F 有限的语言不一定是 P 的。例如我们重新定一个 SAT 问题的子集，例如三个布尔变量 x、y、z 和∪、∩不重复出现地组成布尔表达式，这样的有限个字符串的集合作为一个语言。可是每个字符串的可满足性验证依然是一个 NP 问题。所以有限语言不一定是 P 的，题目表示错误。

(j) F 多项式时间归约表述才成立。

2. **On Regular Languages**

(12%) Consider two deterministic finite automata $M_1 = (K_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (K_2, \Sigma, \delta_2, p_1, F_2)$, where

$$\Sigma = \{a, b\}$$

$$K_1 = \{q_1, q_2\} \text{ and } K_2 = \{p_1, p_2, p_3\}$$

$$F_1 = \{q_2\} \text{ and } F_2 = \{p_3\}$$

and δ_1 and δ_2 are the functions tabulated below.

δ_1	a	b
q_1	q_2	q_1
q_2	q_1	q_2

δ_2	a	b
p_1	p_1	p_3
p_2	p_3	p_2
p_3	p_2	p_1

Use the Cartesian product(笛卡尔积) construction to produce deterministic automata accepting the union of the two languages accepted by these automata.

解析：M=（K，Σ，δ，s，F）

Σ={a，b}

K=K1×K2

F=F1×K2∪F2×K1

	a	b
(q1,p1)	(q2,p1)	(q1,p3)
(q1,p2)	(q2,p3)	(q1,p2)
(q1,p3)	(q2,p2)	(q1,p1)
(q2,p1)	(q1,p1)	(q2,p3)
(q2,p2)	(q1,p3)	(q2,p2)
(q2,p3)	(q1,p2)	(q2,p1)

3. **On Context-free Languages**

(18%)

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

$$L_1 = \{a^m b^n c w w^R | m, n \in \mathbb{N}, n \le m \le 2n, w \in \{a, b\}^*\}$$

(b) Design a PDA $M = (K, \Sigma, \Gamma, \Delta, s, F)$ accepting the language L_1.

Solution: (b) The PDA $M = (K, \Sigma, \Gamma, \Delta, s, F)$ is defined below:

	(q, σ, β)	(p, γ)
$K =$______		
$\Sigma = \{a, b, c\}$		
$\Gamma =$______		
$s =$______		
$F =$______		

(a)

G=（V，Σ，R，S）’

V={a,b,S.S1.S2}

Σ={a，b}

R={

S→S1cS2，

S1→aS1b，

S1→aaS1b，

S1→e，

S2→aS2a，

S2→bS2b，

S2→e

}

(b)

$K = \{p, q\}$

$\Sigma = \{a, b, c\}$

$\Gamma = \{a, b, c, S, S_1, S_2\}$

$s = \{p\}$

$F = \{q\}$

$\Delta = \{ ((p, e, e), (q, S)),\ ((q, e, S), (q, S_1 c S_2)),$

$((q, e, S_1), (q, aS_1b)),\ ((q, e, S_1), (q, aaS_1b)),$

$((q, e, S_1), (q, e)),\ ((q, e, S_2), (q, aS_2a)),$

$((q, e, S_2), (q, bS_2b)),\ ((q, e, S_2), (q, e)),$

$((q, a, a), (q, e)),\ ((q, b, b), (q, e)),$

$((q, c, c), (q, e))$

$\}$

一条线段与前一个顶点连接，第一个顶点是根，而最后一个顶点是一片树叶。

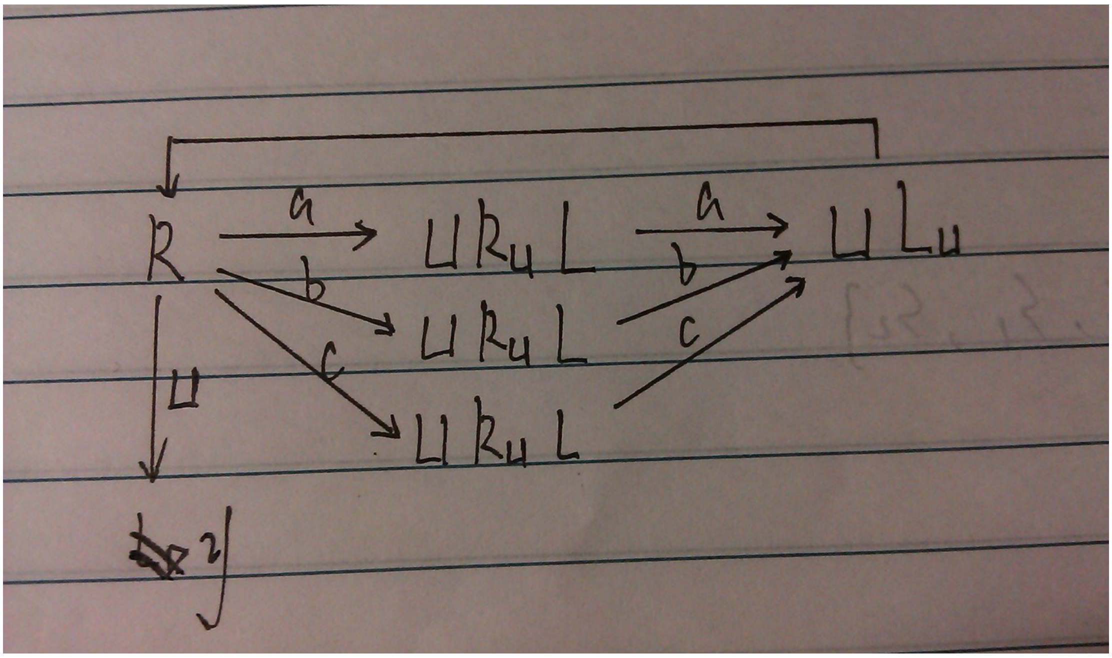
是它里面的线段数。语法分析树的高度是树中最长通路的长度。

4. **On Turing Machines**

(10%) Design a Turing Machine to decide the following language:

$$L_2 = \{ww^R | w \in \{a, b, c\}^*\}$$

where the initial configuration in form of $\triangleright\underline{\sqcup}ww^R$.



5. **On Undecidability**

(16%) Consider the language $L_3 = \{$"M"$|M$ when started on a blank tape eventually writes a 1 somewhere on the tape$\}$.

(a) Show that L_3 is recursively enumerable.

(b) Show that L_3 is not recursive by a reduction from the halting problem.

（a）可以用通用图灵机来半判定。给定输入 w，将其交给通用图灵机运算，如果最后停机了，并且通用图灵机检测带上某处是否有一个 1（假设带长已知而不是无限长），如果是那么给出 yes，如果没有可以给出 no。但是因为“M”不一定在空白的带上停机，所以通用图灵机只能半判定 L3。

（b）题目意思是，通过把停机问题归约到 L3，因为停机问题不是递归的，那么 L3 也不是递归的。我们构造图灵机 M0，

6. **On Primitive Recursive Functions**
(10%) Let $g(x, y)$ be a primitive recursive function. Show that the function

$$e(x, y) = \begin{cases} 1, & \text{if } \exists t_{(0 \le t < y)}(g(x, t) = 0) \\ 0, & \text{otherwise} \end{cases}$$

is primitive recursive.

由题意，$e(x, y) = \sum_{t-0}^{y-1}(g(x,t) = 0)$

因为 g(t,t)是原始递归的，iszero 的判定也是原始递归的，最后的析取运算也是原始递归的，所以 e(x,y)也是原始递归的。

7. **On $\mathcal{P}$ and $\mathcal{NP}$ Problems**
(14%) The **clique** problem is defined as follows: given a graph $G = (V, E)$ with n vertices, is there a set of k vertices such that there is edge between any two vertices in the set?
   (a) Prove that **clique** problem is $\mathcal{NP}$ Problem.
   (b) Prove that **clique** problem is $\mathcal{NP}$-complete.

   For showing hardness, you can assume that the VERTEX-COVER problem is $\mathcal{NP}$-complete.

（a）可以构造 NTM，随机选取 V 中的几个结点，然后在多项式时间内验证其是否是团。因此团问题是 NP 问题。
（b）如果顶点覆盖问题有解，那么可以求 G 的补图，顶点覆盖问题的解对 V 求补就是点集就是图 G 团问题的解。由此，顶点覆盖问题归约到团问题，因此团问题是 NP 完全的。

U 版

(a) ( ) For all languages L_1 and L_2, if L_1 is context free and $L_1 \subseteq L_2$, then L_2 cannot be regular.

(b) ( ) There exists a language L such that L is not regular but L^* is regular.

(c) ( ) If L_1 is regular and L_2 is non-regular, then $L_1 \circ L_2$ must be non-regular.

(d) ( ) If L is a language over alphabet Σ and there is an integer k such that any string $w \in L$ with $|w| > k$ can be written as $w = uvxyz$ such that for each $i \geq 0$, $vy \neq e$, $|vxy| \leq k$ and $uv^ixy^iz \in L$, then L is context-free.

(e) ( ) The context-free languages are closed under complement.

(f) ( ) Let $\mathcal{F} = \{f : f$ is a primitive recursive function from $\mathbb{N}$ to $\mathbb{N}\}$, then $2^{\mathcal{F}}$ (Power set of $\mathcal{F}$) is uncountable infinite.

(g) ( ) Every computable function is primitive recursive.

(h) ( ) Turing Machines with k tapes can accept more languages than Turing Machines with one tape.

(i) ( ) There exists a language L such that L is recursively enumerable and $\overline{L}$ is not recursively enumerable.

(j) ( ) There exists a language L such that L is recursively enumerable and $\overline{L}$ is recursive.

(k) ( ) There is a reduction from A to B, A is recursively enumerable, but not recursive, and B is recursive.

(l) ( ) There are only countably many Turing machines, and there are uncountably many languages, so most languages are not recursive.

（a）F 设 L1={a^nb^n},L2={a*b*}，符合题目条件但是 L2 是正则的。

（b）T 例如 L={a^m|m 是素数}，L*={a*}-{a，e}，是正则的。

（c）F L1={a*}，L2={a^m|m 是素数}，两者连接是{a*}-{a，e}

（d）F 这是上下文无关版和正则版的泵定理的合体，并且泵定理本来是个性质，这里的表述成了个判断，个中必然有诈。在下水平有限看不出来，所以只能谨慎地蒙个 F 吧……

（e）F 定理 3.5.4 上下文无关语言在交和补下不封闭。

（f）T 显然，F 是可数无穷的，可数无穷的幂集是不可数无穷的。

（g）F 原始递归是μ递归的真子集，μ递归的函数才是所有递归函数，才全是可计算的。

（h）F 多带图灵机和单带图灵机的计算能力一样，因此可以接受的语言一样多。

（i）T H={“M”|图灵机 M 在输入”M”上停机}，这个问题是递归但其补连递归可枚举都不是，详见课本 164 页。

（j）T 这样表述有点奇怪但还是能接受的。L 的补是递归的说明 L 是递归的，因此 L 也是递归可枚举的。

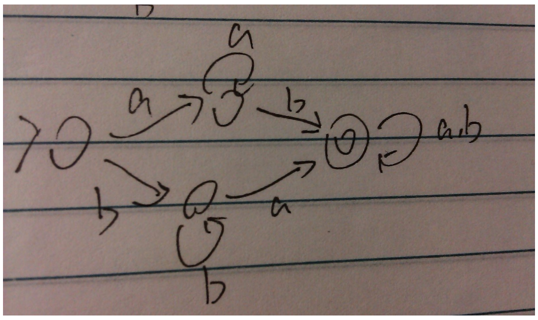
（k）F 定理 5.4.1，L1 非递归，L1 到 L2 归约，L2 也非递归。所以 B 应该也是非递归的。

（l）T 这好像是老师课堂上的原话……首先，图灵机是可以编码的，因此某程度图灵机本身就是个语言，这个语言的字母表是有限的，所以图灵机肯定是有可数无穷多个，这是木有问题的。然后语言，因为这个世界上的字母是可数无穷多个，每个语言只能包含有限多个字母，然后我们可以用对角化原理证明，语言是不可数无穷的。然后每个图灵机能接受的字母是有限的，所以只能判定可数多个语言，因此能够本图灵机判定的语言也只有可数多个。因而大量的语言都不能被判定，是非递归的。

2. **(13%)** Let $\Sigma = \{a, b\}$, consider following language: $L_1 = \{w :$ all symbols in Σ must appear in $w\}$.

(a) Design a finite automaton that accepts L_1.

(b) Give a regular expression for the language L_1.

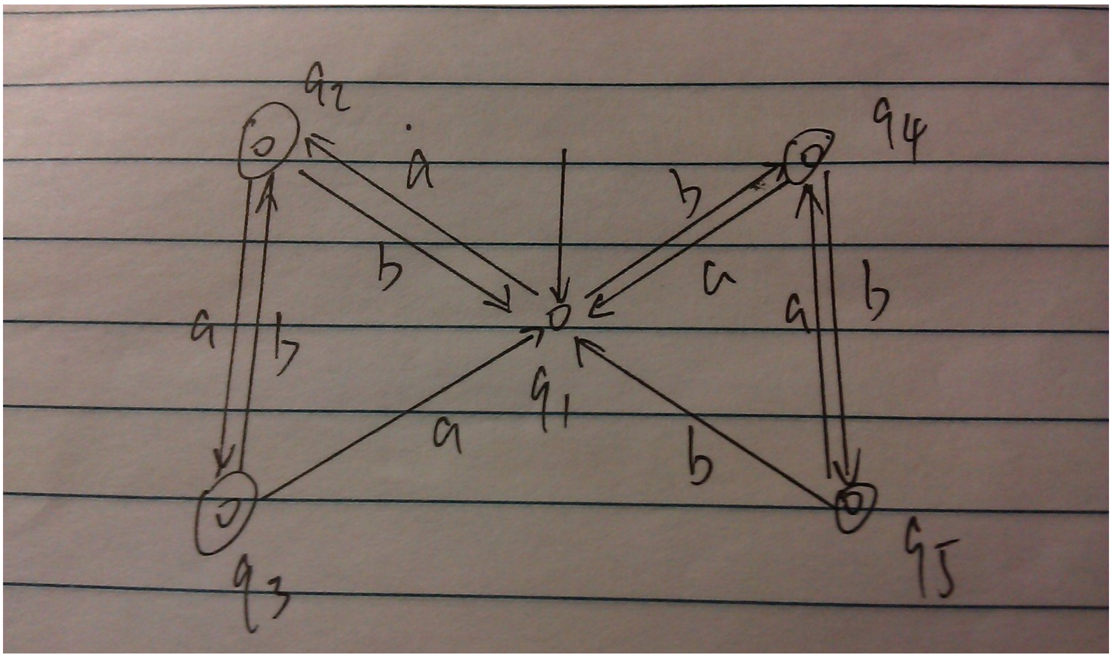
（a）

（b）(aa*b∪bb*a){a,b}*

3. **(14%)** Decide whether the following languages are regular or not and provide a formal proof for your answer.

(a) $L_2 = \{a^m b^n | (m - n) \bmod 3 \neq 0\}$.

解析：（m-n）模 3 不等于 0，即 m 和 n 模 3 不同余。我们可以构造如下 DFA 来接受他。



每个状态都有其含义。Q1 表示 a 和 b 模 3 同余。Q2 表示 a 模 3 比 b 模 3 大 1。Q3 表示 a 模 3 比 b 模 3 大 2.q4 和 q5 如此类推。那么 q2、q3、q4、q5 都是终结状态。因此 L2 是正则的。

(b) $L_3 = \{w | w \in \{a, b\}^*, w \neq w^R\}$.

解析：假设 L3 是正则的，那么 L4={w|w∈{a，b}*，w=w^R}也是正则的，因为正则语言在补运算下封闭。

又因为正则语言在交运算下封闭，所以 L4∩{a*bba*}也是正则的，即{w|w=a^kbba^k}也是正则的。因为|w|≥k，让 w=xyz，且|xy|小于等于 k，y≠e，那么 y=a^i，i>0，然后 xy^nz=$a^{k+(n-1)i}$bba^k，显然已经不对称了，所以 L4 不是正则的，所以 L3 不是正则的。

4. **(10%)** Give Context-Free grammars that generate the following language:

$$L_4 = \{w : w \in \{a, b\}^* \text{ and } w \text{ contains two more } b's \text{ than } a's\}.$$

S→aSb

S→bSa

S→Sb

S→e

5. **(15%)**Consider the pushdown automaton $P = \{K, \Sigma, \Gamma, \Delta, s, F\}$ where $K = \{s, f\}$, $\Sigma = \{a, b\}$, $\Gamma = \{b\}$, $F = \{f\}$ and Δ is given by the following table

$(p, a, \beta), (q, \gamma)$
$((s, a, e), (f, e))$
$((s, b, e), (s, b))$
$((s, a, b), (s, b))$
$((s, e, e), (f, e))$
$((f, a, e), (f, e))$
$((f, b, e), (s, b))$

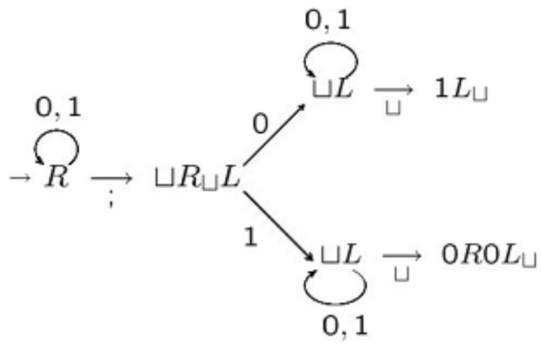
(a) Can PDA P accept string $aaababa$?

(b) Describe the language accepted by P;

(c) Give a Turing machine that decides the same language.

这个就是那个只能输入 a 不能输入 b 的题目。

6. **(12%)** 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.

(a) $\triangleright\underline{\sqcup}10111;10 \vdash_M \triangleright\sqcup10111\underline{;}10 \vdash_M \triangleright\sqcup10111\sqcup1\underline{0}$
$\vdash_M \triangleright\sqcup10111\underline{\sqcup} \vdash_M \triangleright\underline{\sqcup}10111$

(b) $f(x,y) = \begin{cases} 2x+1 & y\text{偶} \\ 4x & y\text{奇} \end{cases}$

7. **(12%)** Explain that why the following language

$$H = \{\text{“}M\text{”} : \text{Turing machine } M \text{ halts on at least two input strings}\}$$

is recursively enumerable. An informal description suffices.

通用图灵机随机生成两个输入丢个“M”，都停机就 yes，否则就 no。