(a) ( ) Let A, B be two languages, if both A and $A \cup B$ are regular, then B is definitely regular.

(b) ( ) Just as Turing Machine's encoding, DFAs "M" can also be encoded as strings "M". Let $D_{DFA} = \{\text{“}M\text{”} \mid \text{DFA } M \text{ rejects “}M\text{”}\}$, then D_{DFA} is recursively enumerable but not regular.

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

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

(e) ( ) Let $D_{TM} = \{\text{“}M\text{”} \mid \text{Turing machine } M \text{ does not halts on “}M\text{”}\}$, then D_{TM} is non-recursively enumerable.

(f) ( ) Let $H = \{\text{“}M\text{”}\text{“}w\text{”} \mid \text{Turing machine} M \text{ halts on } w\}$. If $H \leq L$ and $H \leq \overline{L}$, then L is recursive enumerable but not recursive.

(g) ( ) Let A, B, C be arbitrary languages. If $A \leq C$, $B \leq C$ and C is recursively enumerable, then $A \cup B$ is recursively enumerable.

(h) ( ) Let A, B, C be arbitrary languages. If $A \leq C$, $B \leq C$ and C is recursively enumerable, then $A \cap B$ is not recursively enumerable.

(i) ( ) For all languages L_1 and L_2, if L_1 is in $\mathcal{P}$ and $L_1 \leq_p L_2$, then L_2 is in $\mathcal{P}$.

(j) ( ) The class $\mathcal{NP}$ is closed under intersection and complementation.

(k) ( ) If L is polynomial time reducible to $\{a^n b^n c^n | n \in \mathbb{N}\}$, then L is in $\mathcal{P}$.

(l) ( ) If there is a polynomial time reduction from language L to SAT, then L is $\mathcal{NP}$-complete.

（a） F 例如 A 是{a*b*}，B 是{$a^n b^n$|n 是自然数}

（b） T 用通用图灵机 UTM，模拟 M（“M”），因为 M 是 DFA，所以必然会停机，如果 M 接受“M”那么给出 yes，否则死循环，D_{DFA} 是递归可枚举的。根据对角化定理，D_{DFA} 必然和每一个正则语言都不一样，因此其不是正则的。

（c） T 既然 x∈L 是可判定的，递归语言是补封闭的，那么自然其补也是可判定的。

（d） F M 在输入 x∉L 时也停机那么 L 才是可判定的。

（e） T 课本 5.3 节

（f） T 因为 H 不是递归的，H 可以归约到 L，所以 L 也不是递归的。

（g） T 因为 A 和 B 都可以归约到 C，C 是递归可枚举，所以 A 和 B 都是递归可枚举，递归可枚举对∪操作封闭，因此 A∪B 是递归可枚举。

（h） F 递归可枚举语言对∩封闭。

（i） F 表述反了。L1 多项式时间归约到 L2 只能说 L2 是 P 的那么 L1 也是 P 的。例如给定 x∈L1，多项式时间归约成 t（x）∈L2，然后 L2 的机器多项式时间计算了 t（x），得出结果，因此 L1 是 P 的。

（j） T 递归语言在并交补下都封闭。

（k） T {$a^n b^n c^n$}是 P，L 多项式时间归约到前者，所以 L 是 P 的。

（l） F 反了，如果 SAT 可以多项式时间归约到 L，且 L∈NP，那么 L 就符合 NPC 的定义，是 NPC 的。

2. **(12%)** Define $H(L)$ as the set of even-length strings in L. That is,

$$H(L) = \{w | w \in \{a, b\}^*, w \in L \text{ and } |w| = 2k \text{ for some } k \geq 0\}$$

(a) If L is a regular language, is $H(L)$ a regular language?

(b) If L is a context-free language, is $H(L)$ a context-free language?

State clearly "Yes" or "No", and support your answer with a convincing proof.

（a） 是的。设 L0={w|w∈{a，b}*，|w|=2k，k≥0}，显然 L0 是正则的，L 是正则的，H（L）=L∩L0，因为正则语言对交封闭，因此 H（L）是正则的。

（b） 是的。同样考虑 H（L）=L∩L0，上下文无关语言和正则语言的交是上下文无关的，所以 H(L)也是上下文无关的。

3. **(10%)** Using the pumping theorem to show that

$$L_1 = \{xcyczz^R | x, y, z \in \{a, b\}^*\}$$

is not regular.

证明：假设 L1 是正则的，那么对于任意 s∈L1，存在整数 k≥1，使得 s 满足泵定理。

现在考虑 $s=cca^{k-2}b^nb^na^{k-2}$，即原来的 x、y 都是空串，现在考虑当|s|≥k 时，分解 s=uvw，其中|uv|≤k，且 v≠e。

如果 v=c，u=e 或者 u=c，显然 $uv^iw=c^{i+1}a^{k-2}b^nb^na^{k-2}\notin L1$。

如果，v=cc，或者 v=ca，或者 $v=a^j$，显然都有 $uv^iw\notin L1$

因此，L1 不是正则的。

4. **(18%)**

(a) Construct a context-free grammar that generates language

$$L_2 = \{a^m b^n | m, n \in \mathbb{N}, \text{ and } m \neq n\}.$$

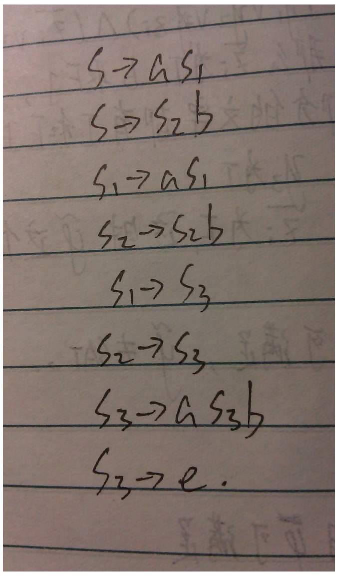
(b) Construct a pushdown automata that accepts language L_2.

（a）G=(V，Σ，R，S)

V={S，S1，S2，S3，a，b}

Σ={a，b}

R={


$S \rightarrow aS_1$
$S \rightarrow S_2b$
$S_1 \rightarrow aS_1$
$S_2 \rightarrow S_2b$
$S_1 \rightarrow S_3$
$S_2 \rightarrow S_3$
$S_3 \rightarrow aS_3b$
$S_3 \rightarrow e.$

}

（b）M=（K，Σ，Γ，Δ，s，F）

K={p，q}

Σ={S，S1，S2，S3，a，b}

Γ={S，S1，S2，S3，a，b}

Δ={

}

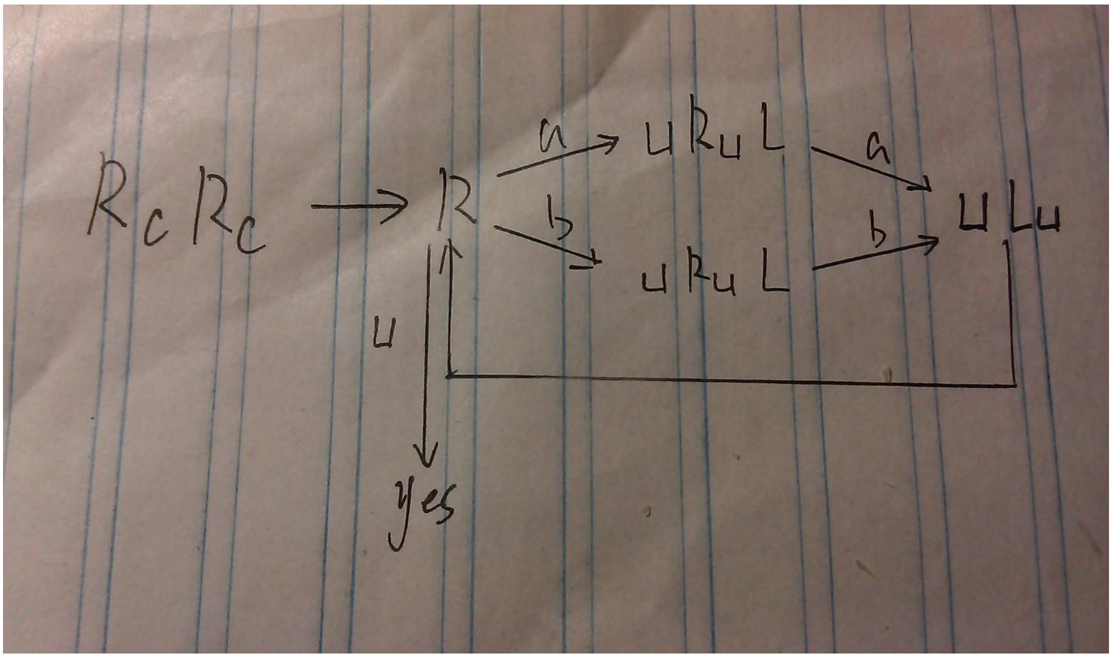
s=p

F={q}

5. **(12%)** Construct a Turing machine that decides the following language:

$$L_3 = \{xcyczz^R | x, y, z \in \{a, b\}^*\}$$

When describing the Turing machines above, you can use the elementary Turing machines described in textbook. Always assume that the Turing machines start computation from the configuration $\triangleright\underline{\sqcup}w$ where $w \in \{a, b, c\}^*$ is the input string.



6. **(12%)** Let

$$L_4 = \{\text{“}M_1\text{”}\text{“}M_2\text{”}\text{“}M_3\text{”} | M_1, M_2 \text{ and } M_3 \text{ are TMs}, M_1, M_2 \text{ and } M_3 \text{ halt on empty string}\}.$$

(a) Show that L_4 is recursively enumerable.

(b) Show that L_4 is not recursive.

An informal description suffices.

（a）构造通用图灵机 UTM，逐个模拟 M1、M2、M3 在空带输入上面的计算，如果都停机了那么 yes，否则不停机。

（b）把停机问题 H 归约到 L4 上。给定输入“M”“w”，我们构造“Mw1”，“Mw2”，“Mw3”，用通用图灵机模拟，对于 Mw1，则是接受空白输入，然后再带上写上 w，然后模拟 M 在 w 上的计算，Mw2 和 Mw3 同理。这样，当且仅当 M 在 w 上停机，Mw1，Mw2，Mw3 在空白带上停机。

7. **(12%)** A Boolean formula F in conjunctive normal form(CNF) is **One-Out-of-3SAT** if it contains a truth assignment to its variables such that exactly one of its clauses evaluate to false, and others of its clauses evaluate to true. Let

$$One\text{-}Out\text{-}of\text{-}3SAT = \{F | F \text{ is a Boolean formula in 3-CNF that is One-Out-of-3SAT}\}.$$

(a) Prove that $One\text{-}Out\text{-}of\text{-}3SAT$ is a $\mathcal{NP}$-problem.

(b) Prove that $One\text{-}Out\text{-}of\text{-}3SAT$ is $\mathcal{NP}$-complete.

Hint: *Use a reduction from the* $3SAT$ *Problem, which is to decide the language*

$$3SAT = \{F | F \text{ is a Boolean formula in 3-CNF that is satisfiable}\}.$$

（1）随机生成真值赋值验证

（2）原来的 3SAT 的式子合取一个（F∨F∨F），就变成 OOO3SAT 了

(a) ( ) Let L be a language, then $(L^+)^+ = L^+ \circ L^+$, where $L^+ = L \circ L^*$.

(b) ( ) Let L be a regular language, so is $\{w | w \in L \text{ and } w \text{ with even length}\}$.

(c) ( ) For 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.

(d) ( ) Just as Turing Machine's encoding, DFAs M can also be encoded as strings "M", then the language $\{$"M"$|$ DFA M rejects "M"$\}$ is not regular but recursive.

(e) ( ) Language $\{a^m b^n c a^{2n} b^{2m} | m, n \in \mathbb{N}\}$ is context-free.

(f) ( ) Let L be a context-free language, so is $\{w | w \in L \text{ and } |w| = 3k \text{ for some } k \in \mathbb{N}\}$.

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

(h) ( ) Language $\{$"M" : Turing machine M accepts at least 2011 distinct strings$\}$ is recursively enumerable.

(i) ( ) A language is recursive if and only if it is Turing-enumerable.

(j) ( ) Let L be a language, if there is a Turing machine M halts on x for every $x \in L$, then L is recursive.

(k) ( ) To simulate a computation of n steps for the nondeterministic Turing machine, it requires exponentially many steps in n for a deterministic Turing machine.

(l) ( ) If there is reduction τ from language A to $\{$"M"$|$ Turing machine M halts on empty string$\}$, where τ is a recursive function, then A is undecidable.

（a）T （L+）+ = L+^{o}（L+）* = L+^{o} L+

（b）T 偶数长度字符串集合是正则，合取 L 也是正则

（c）F 不一定，{ab} {$a^n b^n$} {a*b*}

（d）T 因为 DFA 最终会读入所有字符并且停机，所以递归；对角化定理可证明非正则

（e）T 可以构造 PDA 接受
（f）T 上下文无关和正则合取是上下文无关，{3 的倍数长度字符串}是正则的。
（g）F 原始递归语言包含于递归语言
（h）T 可以构造图灵机半判定，就是按字典序生成 n 个串，然后每个串模拟走 n 步，如果大于 2011 的时候停机那么 yes，否则不停机
（i）F 当且仅当字典序可枚举
（j）F 对于 x∉L 时也停机，才算是递归的
（k）T 对于非确定图灵机的 n 步，确定图灵机要用 n 的指数步来模拟。
（l）F 反了，应该是，停机问题归约到 A，那么 A 是不可判定的。

2. **(16%)** Decide whether the following languages are regular or not and provide a formal proof for your answer.

   (a) $L_1 = \{xcycz | x, y, z \in \{a, b\}^* \text{ and } y = y^R\}$.

（a）假设 L1 是正则的，那么对于任意字符串都有 k 使其满足泵定理。考虑字符串 ca^kba^kc，我们将其改写为 xyz，$|xy| \leqslant k$，$|y| \neq e$。考虑 y=c，或 $y=ca^n$，或者 $y=a^n$，那么都有 $xy^iz \notin$

   (b) $L_2 = \{xyz | x, y, z \in \{a, b\}^* \text{ and } y = y^R\}$.

（b）
也是泵定理来做。

3. **(20%) On Context-free Languages**

   (a) Construct a context-free grammar that generates the language

$$L_3 = \{xcycz | x, y, z \in \{a, b\}^* \text{ and } |x| = |z|\}.$$

   (b) Construct a pushdown automata that accepts L_3.

4. **(14%) On Turing machines**

   Design a single tape Turing machine M that decides the language L_4 on $\{a, b, c\}$:

$$L_4 = \{xcycz | x, y, z \in \{a, b\}^* \text{ and } z = z^R\}.$$

   When describing the Turing machines above, you can use the elementary Turing machines described in textbook. Always assume that your Turing machine starts from the configuration $\triangleright \underline{\sqcup} w$.