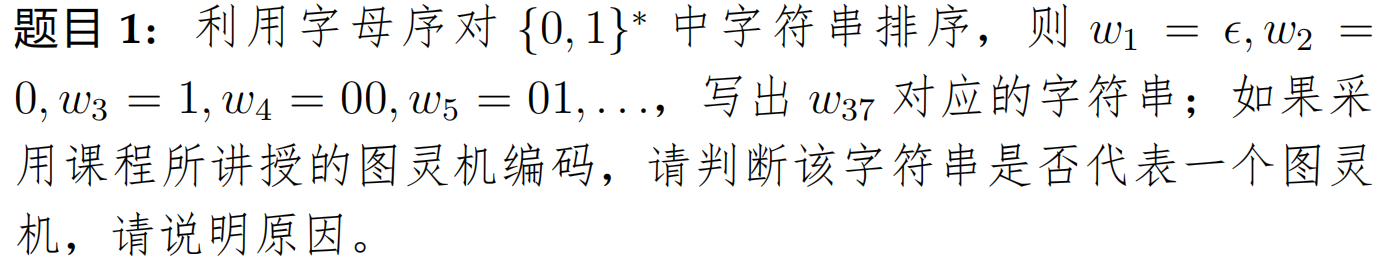
（一）图灵机相关的题目



题目 1：

解：为。该字符串不代表一个图灵机，因为任意一条转移规则对应编码，而即的完整编码连一条转移规则的编码都不包含。或者，更简单地说，由于一个图灵机的编码的最后一位一定是，故一定不代表一个图灵机。



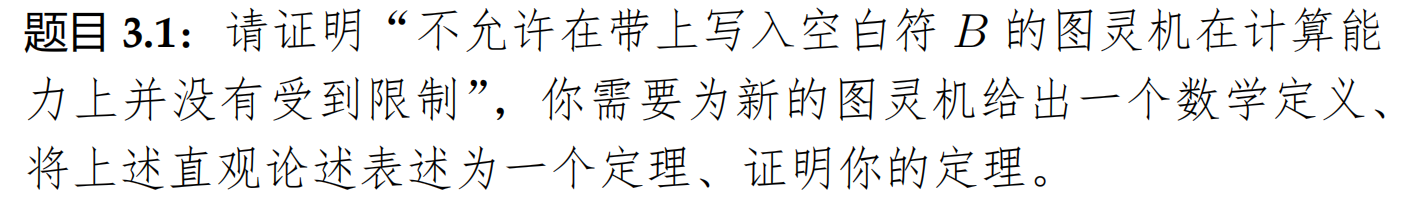
题目 2：

证明：设任意单带TM ，则可以构造onestopTM 使得，构造方法如下：

令，定义如下：



接受当且仅当接受，所以。



题目 3.1：

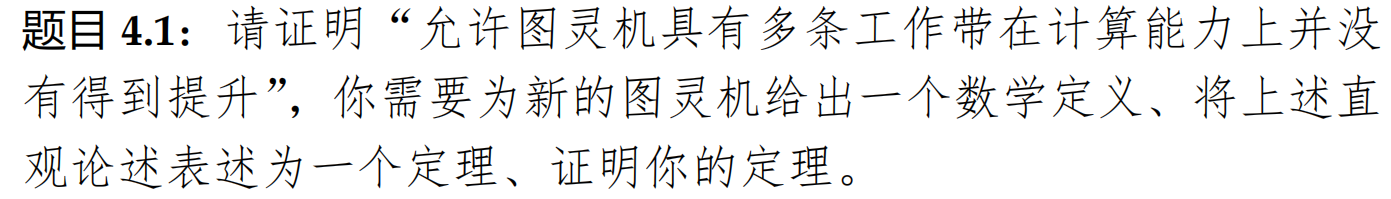
将“不允许在带上写入空白符的图灵机在计算能力上并没有受到限制”表述为如下定理：

存在一个具有限制“永远不写空白符”的TM 接受TM 所接受的语言。

证明：设。新增不是空白符但作用像的新的带符号，构造，定义如下：



则恰好当进入接受状态之一时才接受，因此，。



题目 4.1：

将“允许图灵机具有多条工作带在计算能力上并没有得到提升”表述为如下定理：

多带TM接受的每个语言都是递归可枚举的。

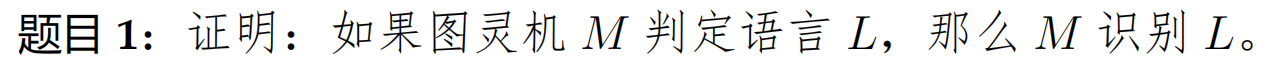
或：

令是k带图灵机，是所识别的语言，则存在单带图灵机满足。

证明：假如k带TM 接受语言，用认为具有2k道的单带TM 来模拟。用两道模拟的一带，一道放置带上的内容，另一道只包含表示该带带头目前位于何处的一个标记。为了模拟的一个动作，的带头从左至右访问k个带头标记，再从右至左更新每个带头标记处的表示相应带的道中的符号，并更新带头标记的位置。每当被模拟的接受时，也接受；否则，不接受。所以。所以任意一个多带TM都可以被一个单带TM模拟，即多带TM接受的每个语言都是递归可枚举的。TM 的数学定义如下：

设，其中。构造，其中即个组成的有序对。的转移如下：

1. 首先用标记的每一带带头位于最左侧：对于任意属于，。
2. 再将的带头返回到输入最左侧：对于任意属于，。
3. 开始模仿的任意一个动作，首先从左至右访问所有带头标记：对于任意属于，有；对于任意属于使得任意属于的有序对中有个，有。
4. 访问了所有带头标记后，调转方向：对于任意属于，及任意属于，有。
5. 接着从右至左更新每个带头标记处的表示相应带的道中的符号：对于任意属于，有；对于任意属于，属于，有，其中；若，则 ；若，则；若，则。然后。
6. 然后更新带头标记的位置：，然后；，然后；，然后。对于同理。
7. 每模拟了上一条带的动作后，减1：。
8. 减至0时，模拟完了上所有带的动作，即模拟完了的一次完整动作：。
9. 图灵机语言相关的题目



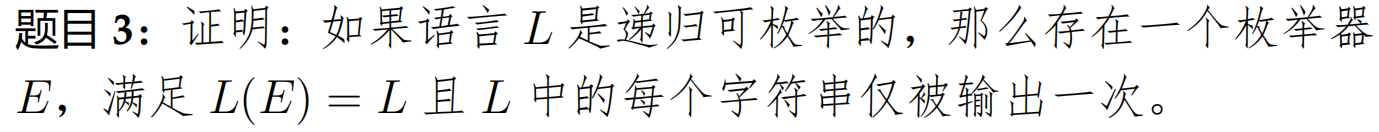
题目 1：

证明：如果图灵机判定语言，则，则根据可判定的定义，总是会停机，即如果，则接受并停机，如果 , 则不接受并停机。如果图灵机识别语言，则当, 则接受并停机，如果, 则不接受，但可能不停机。因此，如果图灵机判定，则该图灵机也识别，区别在于当，图灵机是否一定停机。



题目 2：

证明：若语言是图灵可判定的，则存在一个总是停机的TM ，，如果，则接受；如果，则拒绝。而若语言是可识别的，则存在一个TM ， ，如果，则接受；如果，则不接受，但可能不停机。因此如果语言是可判定的，则语言也是可识别的。



题目 3：

证明：因为语言是递归可枚举的，所以存在一个TM 使得。则可构造一个非确定型TM枚举器，满足且中的每个字符串仅被输出一次；又根据非确定型TM与确定型TM的等价性，可以把转化为确定型TM。的构造方法如下：

无输入；

1. 对i=1,2,3,…，重复下列步骤。
2. 使用非确定性能力，猜测可能接受的输入；
3. 接着让以作为输入运行步，若在这个计算过程中接受，则将复制到的工作带，同时要确保将复制到的工作带上后工作带上只包含，若后面的单元中还有非空白字符，要将它们全部扫描并替换成空白字符，然后再将带头重新定位到的左端；
4. 在将输出到输出带之前，要检查是否已经出现在输出带中。用另一条中间带保存输出带的中间结果，该带上内容与输出带上内容一致，但在中间带更新之后，输出带才更新；中间带从左至右检查输出内容中是否含有，若不含有，则将复制到中间带上，字符串之间#字符分隔，否则不复制到中间带上，最后把中间带的带头重新定位到带的左侧；
5. 若检查中间带时发现输出结果中不含有，则进入状态，将工作带上的内容即复制到输出带上，利用#字符分隔，否则不复制到输出带上；
6. 完成上述操作后，机器返回常规状态继续运行，重复上述过程。



题目 4：

证明：



是图灵可判定的，则也是图灵可判定的，因此与都是图灵可识别的。



若与都是图灵可识别的，则可以构造一个图灵机，其内部由识别的图灵机和识别的图灵机构成。对于任意的输入，将复制，一份传递给，一份传递给，的工作流程如下：

1. 若，则图灵机接受，此时不考虑的工作情况，直接令接受。
2. 若，则图灵机接受，此时不考虑的工作情况，直接令拒绝。

于是可判定语言。



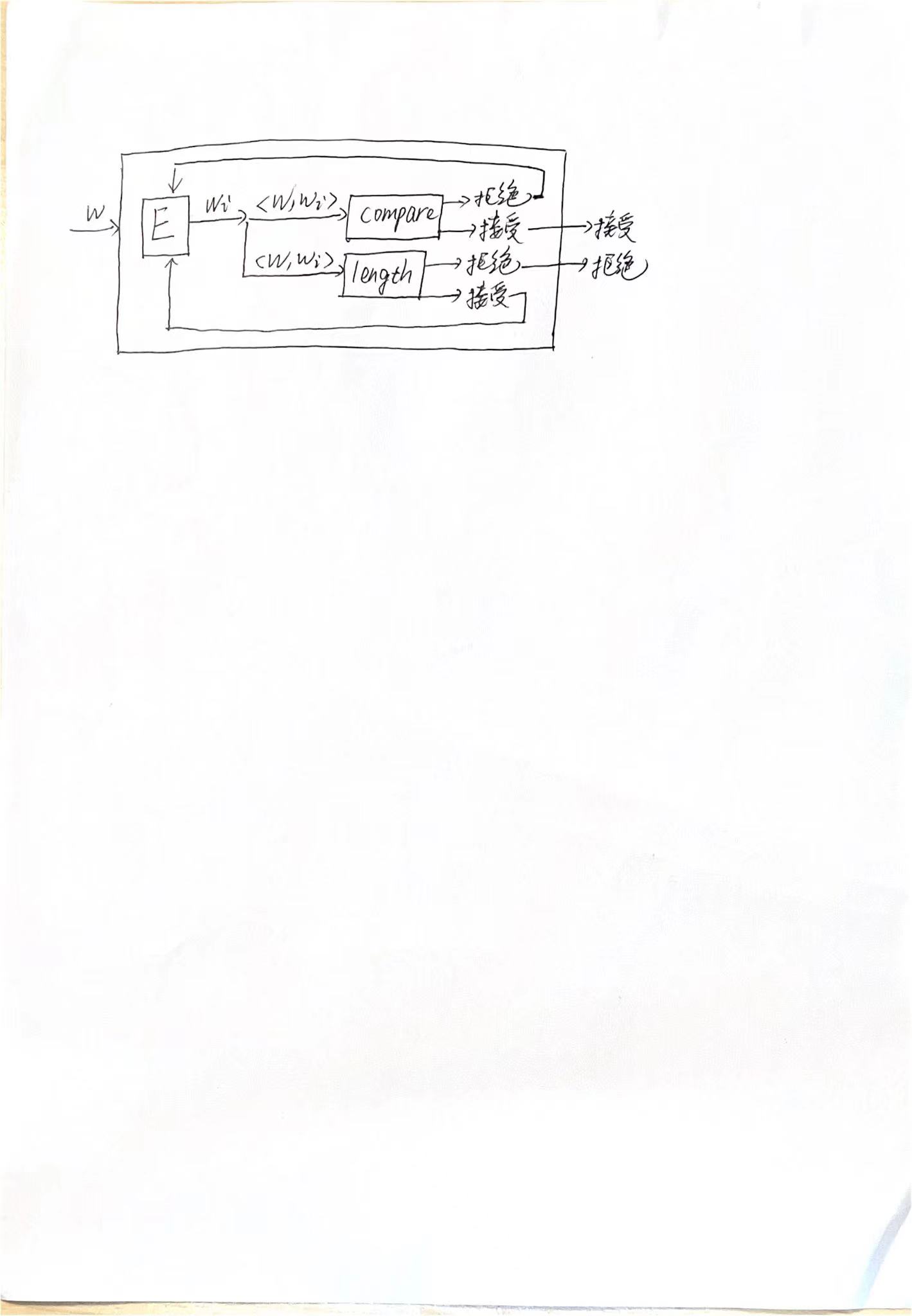
题目 5.1：

证明：



不妨规定一个良顺序为：按照字符串长度由小到大打印。因为语言是递归的，则存在一个枚举图灵机，满足，并且以该良顺序打印。

据此构造图灵灵机判定语言。如下图所示：给定输入，枚举中字符串，将分别传递给图灵机compare和图灵机length，compare判断和是否相同，若相同则接受，若不同，则等待新的输入；length判断，若是，则等待新的输入，若不是，则拒绝。



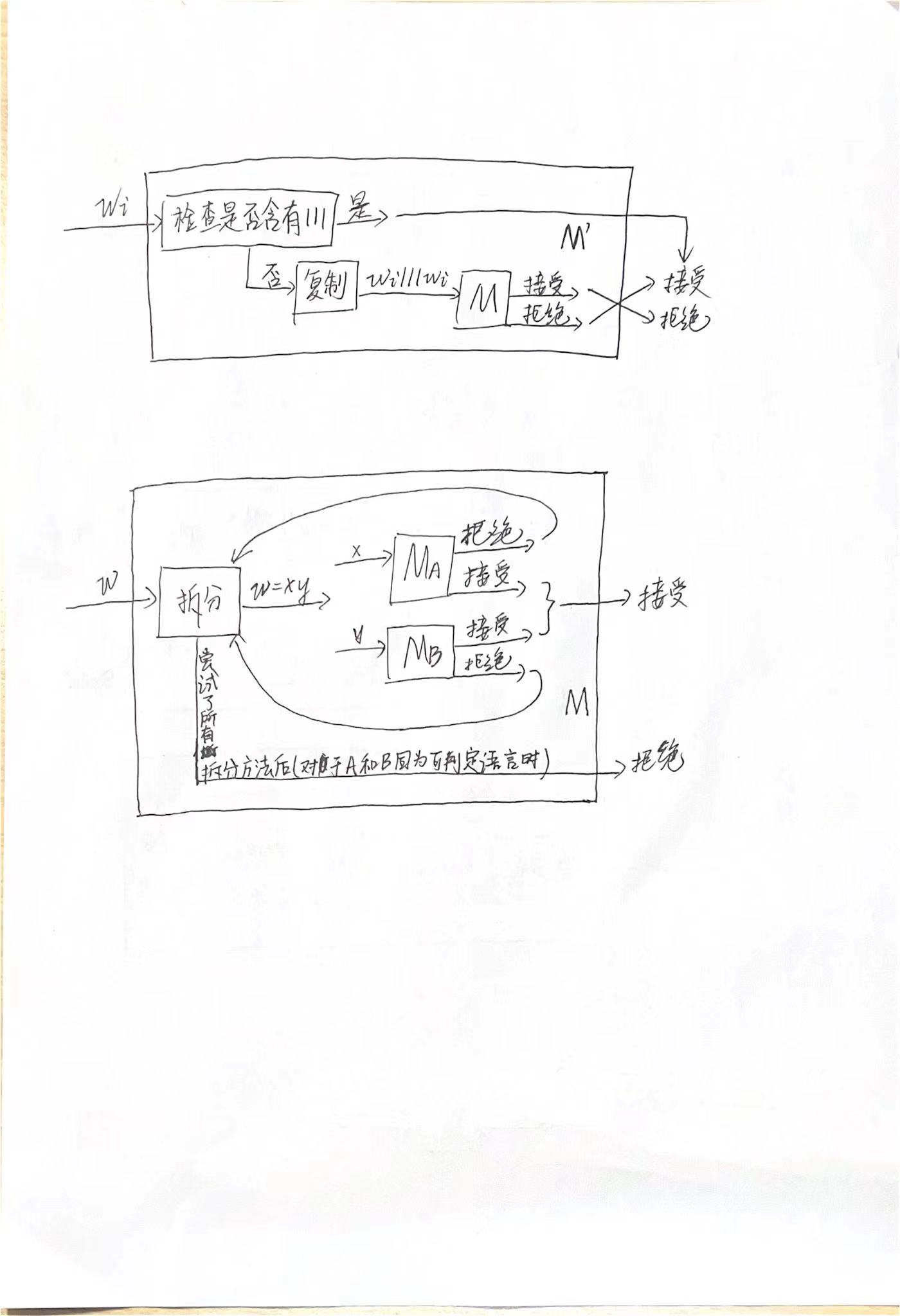


若语言是图灵可判定的，则存在总是停机的TM ，，使得当，接受，当时，拒绝。



题目 6.2：

证明：假设和同为图灵可判定（或图灵可识别）语言，和分别判定（或识别）和。，则可构造判定（或识别），的构造方法如下：



输入为，由于长度有限，将拆分和并使得的方法有穷，若，则有种拆分的方法；对于每一种将拆分和的方法，在输入上模拟，在输入上模拟，若接受且接受，则接受；若每一种拆分的方法都不能使接受且接受，则不接受（若和同为可判定语言，则不接受或不接受时都停机，也停机，此时也为可判定语言）。

通过这种方式，若，则一定接受；否则不接受（若和同为可判定语言，拒绝并停机）。所以当和同为可判定（或可识别）语言时，也是可判定（或可识别）语言。

1. 不可计算证明相关的题目



题目 7：

证明：假设是可判定的，则也可判定，且可证（下面给出证明），所以也是可判定的，但非递归可枚举，一定不是可判定的，矛盾，故假设错误，是不可判定的。

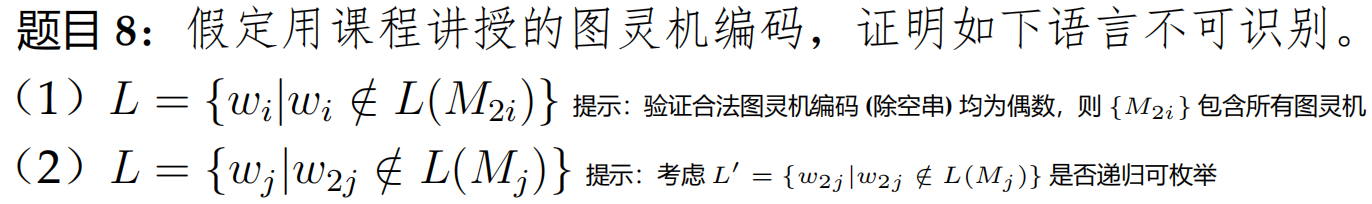
下证：

构造使得：

如果不是有效的图灵机编码，则认为对应一个在任何输入上都立即停机的图灵机，，则，所以，令。

如果是有效的图灵机编码，同样令。

若，则，图灵机不接受输入，所以；若，则，图灵机接受输入，所以。所以。



题目 8：

1. 证明：由于转移规则编码最后一位一定是0，图灵机编码为用11隔开的转移规则编码，故图灵机编码最后一位也一定是0。故将二进制图灵机编码解释为十进制数时，其一定为偶数。故包含所有图灵机。假设存在识别，则，那么考虑：若，则由定义，；若，则，则由定义，。和都矛盾，故假设不成立，不可识别。
2. 证明：假设识别，，考虑：若，则由定义，；若，则由定义，。和都矛盾，故假设不成立，不是图灵可识别的。若，那么由于不是图灵可识别的，则也不是图灵可识别的。

下证：

构造使得：

如果，其中为奇数，令，其中。

如果，其中为偶数，令。

若，则且，则。若，若，则显然；若且，则。所以。



题目 9.1：

证明：先证：

构造使得：

如果不是合法编码，令，其中。

如果是合法编码，令，的构造如下：

首先模拟在上运行，若接受，则接受一切输入；若不接受，则让无限循环、永不接受任何输入。

若，则，，所以。若，则、或、，所以。所以。

因为是不可判定的，则由可知是不可判定的，则也是不可判定的。