$O(T \log T)$ **Universal Turing Machine**

* Suppose we want to simulate an arbitrary Turing machine M with its input string x.
* Assume that M has k tapes, its alphabet is Γ and its running time on input x is T.

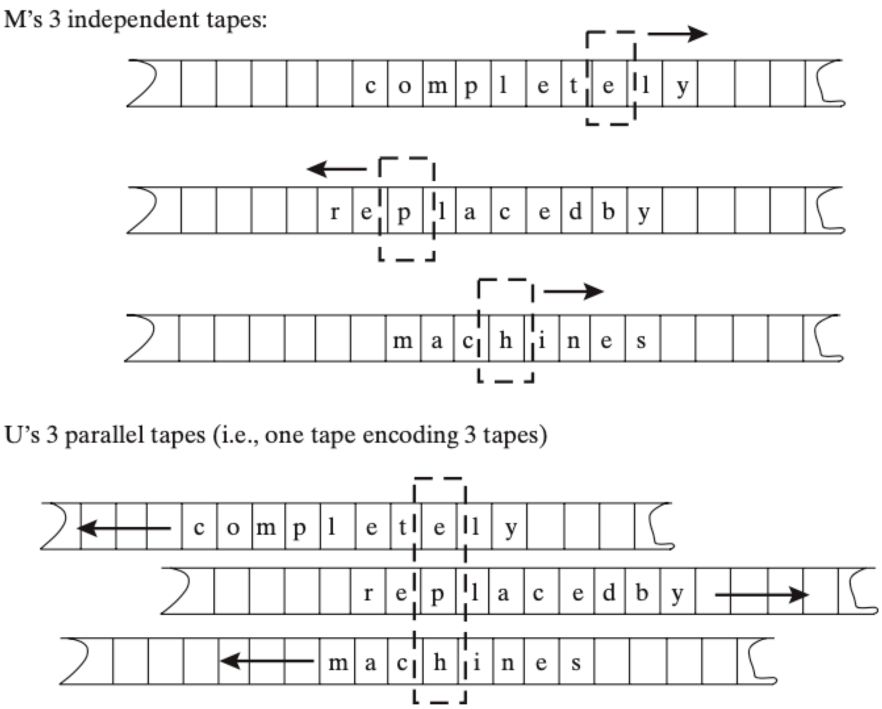
A universal Turing machine $\mathscr{U}$ is constructed as follows:
* $\mathscr{U}$ use only one tape to simulate M's tapes.
* $\mathscr{U}$ only use constant more tapes to do other things, such as simulate M for one step
* $\mathscr{U}$'s alphabet is Γ^k.
  * So it would be convenient for $\mathscr{U}$ to simulate k tapes.
  * Transform Γ^k to $\{0,1\}$ only has $\log|\Gamma^k|$ overhead, which could be seen as constant.

The problem now becomes:

How to simulate k tapes using only one tape.

A naive approach

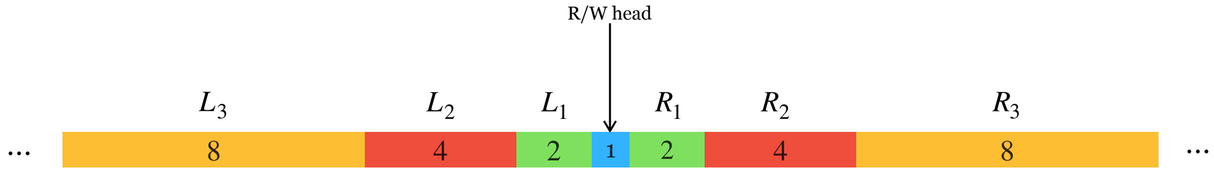
* *Let $\mathscr{U}$'s tape (the one to simulate M's tapes) be indexed from $-\infty$ to $+\infty$*
* *Let $\mathscr{U}$'s read/write head always stays at point* 0.
* *The content on $\mathscr{U}$'s tape now becomes a tuple as $(u_1, u_2, \cdots, u_k)$ such that u_i represents the content on the i-th tape of M.*



In this approach, when one read/write head of M moves, all the contents on $\mathscr{U}$'s tape should be modified.

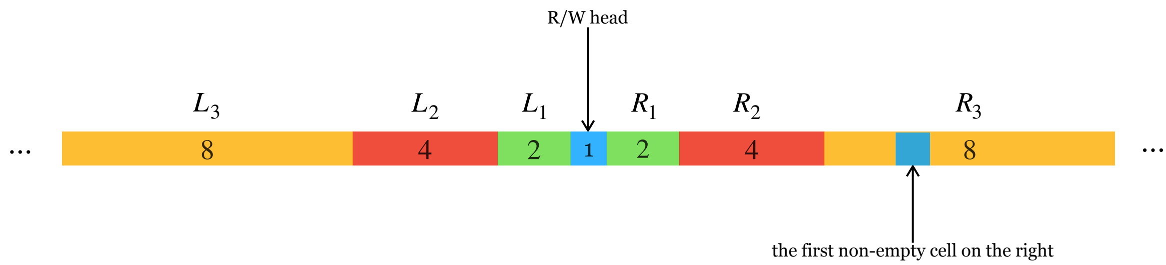
A potential approach

* *The main idea is to add a new symbol which represents "empty" to* Γ.
* *We add "empty spaces" between the original symbols, so that in the most cases, when a read/write head of M moves, we do not need to modify all the contents on $\mathscr{U}$'s tape.*

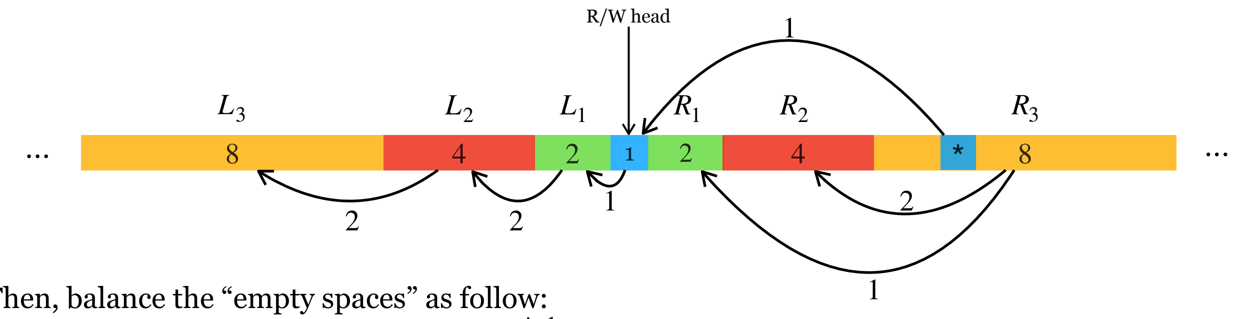


The “empty spaces” is inserted as follow.
* The cell on $\mathscr{U}$’s tape is split into several intervals. $(L_1, R_1, L_2, R_2, \cdots)$
* $|L_i| = |R_i| = 2^i$
* In each operation, the following invariants are maintained:
    * The number of non-empty cells in R_i or L_i could be $\{0,\ 2^{i-1},\ 2^i\}$.
    * In any time, the number of non-empty cells in L_i and R_i should be exactly 2^i in total.
    * The cell under read/write head should not contain “empty space”.

* In the beginning, we could insert all the “empty spaces” in $\mathcal{O}(T)$. Ensure the number of non-empty cells in L_i and R_i be exactly 2^{i-1} and 2^{i-1} respectively.
* Then, when some read/write head of M moves one cell to the right, we maintain the invariants as follows:



First, find the first non-empty cell x on right. Suppose $x \in R_i$.



Then, balance the “empty spaces” as follow:
* $x \in R_i \Rightarrow \#\{\text{non-empty cells in } R_i\} \geq 2^{i-1} = 1/2\,|R_i|$
* move x to the cell under the read/write head.
* move the other $2^{i-1} - 1$ non-empty cells in R_i to $R_1, R_2, \cdots R_{i-1}$ so that for any $j \in [1, i-1]$, $\#\{\text{non-empty cells in } R_j\}$ becomes $1/2\,|R_j|$.
* move the original cell under the read/write head to L_1
* for any $j \in [1, i-1]$, move all the non-empty cells of L_j to L_{j+1}. (R_j is empty $\Rightarrow$ L_j is full)

Note that
* If $x \in R_i$, then including the current move, there must be at least 2^i moves till now.
* If $x \in R_i$, then the cost of the simulation of the current $\propto 2 \cdot 2^{i+1}$.
* If the total move of M is T, then there are at most $T/2^i$ moves that could cost $2 \cdot 2^{i+1}$ steps of simulation.

So, the total time cost is

$$\sum_{i=1}^{\log T} \frac{T}{2^i} \cdot 4 \cdot 2^i = \sum_{i=1}^{\log T} 4T = 4T \log T.$$

The $\log T$ comes from the fact that T could be split in at most $\log T$ intervals. So for any R_i, L_i , we have $i \leq \log T$.