18 Pumping Lemma : U versus Y

The Pumping Lemma provides a concrete setting to understand adversarial arguments. Consider proving, *directly using the Pumping Lemma*, that the language

$$L = \{0^i 1^j \mid i \neq j\}$$

is not regular. Here is how the proof goes as an adversarial argument. Suppose an adversary (Y) claims that this is a regular language. You (U) want to prove it is not. Here is how you can argue and win:

1. U: "OK if L is regular, you have a DFA D with you right?"
2. Y: "Yes."
3. U: "How many states in it?"
4. Y: "n".
5. U: "OK, describe to me the sequence of states that D goes through upon seeing the first n symbols of the string $0^n 1^{(n+n!)}$." Here, n is chosen to be the number of states in D. Since $n \neq (n + n!)$, this string surely must be in L. (The choice of $(n + n!)$ as the exponent of 1s is rather purposeful — and *very astute* on the part of U — as we shall see momentarily).
6. Y (Straight-faced): "It visits s_0, s_1, ..., *all of which are different from one another*."
7. U: (Red-faced): "Lie! If there are n states in your DFA, then seeing n symbols, the DFA must have traversed a loop, and hence you must have listed two states that are the same. Don't you know that this follows from the pigeon-hole principle?"
8. Y: (Blue-faced): "OK, you are right, it is "$s_0, s_1, \ldots, s_i, s_{i+1}, s_{i+2}, \ldots,$ $\ldots, s_i, s_j \ldots$." Notice that s_i is repeating in such a sequence.
9. U: "Aha! I'm going to call the pieces of the above sequence as follows."
   (a) the piece that leads up to the loop, $s_0, s_1, \ldots, s_i$, will be called x,
   (b) the piece that traverses the loop, $s_i, s_{i+1}, s_{i+2}, \ldots, s_i$, will be called y, and
   (c) the piece that exits the loop and visits the final state, $s_i, s_j \ldots$, will be called z.

   "You may pick any such x, y, z and I'm going to confound you."
10. Y: "How?"
11. U: "Watch me!" (private thoughts of U now follow...)
    (a) Since I have no idea what $|y|$ is, I must ensure that by pumping y, no matter what its length, I should be able to create a string of 0s equal in length to $(n + n!)$.

(b) So, by pumping, if I can create an overall string $0^{(n+n!)}1^{(n+n!)}$, I would have created the desired contradiction.

(c) The initial distribution of 0s along the path xyz is as follows:

i. x has $|x|$ 0s,

ii. y has $|y|$ 0s, and

iii. z has $(n - |y| - |x|)$ 0s.

(d) Hence, by pumping y k-times, for integral k, we must be able to attain $n + n! = |x| + k \times |y| + (n - |y| - |x|)$.

(e) Simplifying, we should be able to satisfy $n! = (k-1)|y|$. Since $|y| \leq n$, such a k exists!

12. U now begins his animated conversation: "See the above argument. I can now pump up the y of your string k times where $k = n!/|y| + 1$. Then you get a string $0^{n+n!}1^{n+n!}$ that is not in L. This path also exists in your DFA. So your DFA cannot be designed exactly for L— it also accepts illegal strings. Admit defeat!"

13. Y: Tries for an hour, furiously picking all possible x, y, z and goes back to step 8. For each such choice, U defeats Y[5] in the same fashion. Finally Y admits defeat.

14. U: "Thank you. Next victim please."

[5] I promise to make Y win in my next *two* books—and meanwhile, offer to put replacement pages on my web-page for the benefit of anyone wishing that Y trounce U in this very book!