University of Virginia **cs3102: Theory of Computation** 28 December 2020

Response by: **Benjamin (aqn9yv, dlb2ru, ht6xd, iad4de, jmn4fms, lw7jz)**

Week 10: Have we been reduced to this?

Problem 1 *Entranced by Re-Entrances*

Proof.

Assume REENTER is decidable. Then we can construct a Pythonic function REENTER that returns 1 if the machine passed in as input re-enters its start state on input ε at some point in its execution and 0 otherwise.

Construct a function mPrime as follows:

```
def mPrime(x):
    M(w, x)
    if x == "":
        transition to start state
```

Since we assumed that REENTER is computable, we may call it on our constructed mPrime. The FSA associated with function mPrime re-enters its start state if and only if M(w, x) halts, and so calling REENTER on mPrime will accordingly return 1 if and only if M(w,x) halts.

Thus we have an algorithm to compute HALT(M, w):

```
if REENTER(mPrime(x)):
    accept
else:
    reject
```

Thus we have demonstrated that HALT is reducible to REENTER, and because HALT is uncomputable REENTER is uncomputable as well.

□