Homework 5
May 8, 2025

Group: August Wetterau, Brian Bowers, Thomas Rife, Walter Campos

Project: `https://github.com/thomas-rife/PANIC`

AW I read through all the assigned readings, watched all the required videos, worked through the course notes, read the book chapters, and browsed the provided articles.
BB I read through all the assigned readings, watched all the required videos, worked through the course notes, read the book chapters, and browsed the provided articles.
TR I read through all the assigned readings, watched all the required videos, worked through the course notes, read the book chapters, and browsed the provided articles.
WC I read through all the assigned readings, watched all the required videos, worked through the course notes, read the book chapters, and browsed the provided articles.

1. The answers to problem 1 can be found here: `https://github.com/Bbowers2/3802_homework_5/blob/main/src/regex_exercises.js`.

2. WebAssembly:

```
local.get 0 ;; get input number
i32.const 3 ;; push 3
i32.mul ;; n*3
i32.const 1 ;; push 1
i32.add ;; add it (3n + 1)
local.get 0 ;; get n again
i32.const 1 ;; push 1
i32.shr_s ;; shift right by 1 (same as n/2)
local.get 0 ;; get n again
i32.const 1 ;; push 1
i32.and ;; n & 1
i32.select ;; if odd use 3*n + 1 else use n/2
end_function
```

x86-64:

```
mov ecx, edi ;; copy lower 32 bits (int) of rdi into ecx
sar ecx ;; shift right by one, which is n/2
test dil, 1 ;; check if last bit is even (0) or odd (1)
lea eax, [rdi + 2*rdi + 1] ;; puts 3*n + 1 into eax
cmove eax, ecx ;; if n is even move n/2 (ecx) into eax otherwise keep 3*n+1
ret ;; return eax
```

3. Starting with any $\langle M, w\rangle$ from the undecidable acceptance problem, two machines can be constructed, $\langle M_1, M_2\rangle$, where M_1 runs M on input w, then accepts its own input if and only if that simulation accepts. Machine M_2 rejects every input unconditionally. If M accepts w, then $L(M_1) = \Sigma^*$ and $L(M_2) = \emptyset$. If M does not accept w, then both M_1 and M_2 accept no strings, so $L(M_1) = L(M_2) = \emptyset$. Thus, $L(M_1) = L(M_2)$ if and only if M does not accept w. Therefore, if we had a decider for testing language equivalence, we could use it to decide

whether M accepts w. Since the acceptance problem is undecidable, the language equivalence problem must also be undecidable.