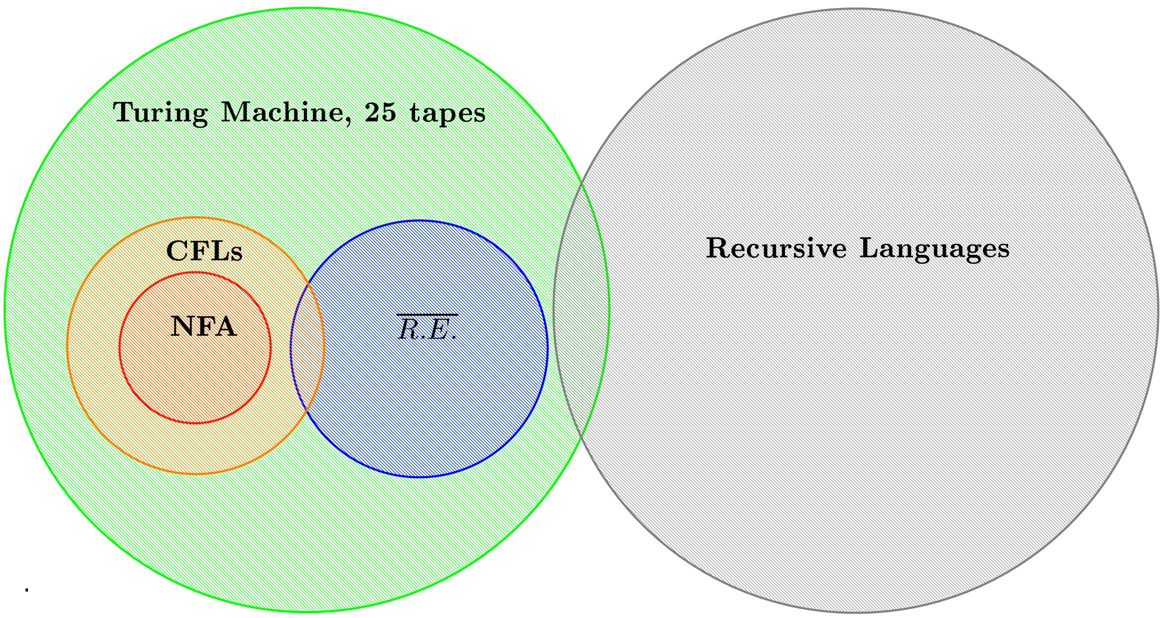
Introduction to the Theory of Computation
Homework #8
Brian Gianforcaro

$\boxed{1}$ Venn Diagram:



.

$\boxed{2}$ Show the following is decidable:

Given a CFG G, is $L(G)$ non empty?

Proof:

$\boxed{3}$ Sipser, exercise 4.6

Assume B is countable and that $f\colon \mathbb{N} \rightarrow \mathbb{B}$ is a functional correspondence. Let $f_i(j)$ denote the i^{th} symbol in $f(j)$. Consider the string s whose i^{th} element differs from $f_i(i)$, then the i^{th} symbol in s is 1 and vice versa. Then s cannot be in the image of f since it differs from every string in the image of f by at least one symbol. Furthermore, s is in B since it is an infinite sequence over $\{0,1\}$. Therefore, B is uncountable.

$\boxed{4}$ Sipser, exercise 4.12

Definition of the given Turing machine:

$M =$ "On input $< R, S >$ where R and S are regular expressions

1. Convert R to DFA A and S to DFA B
2. Construct DFA C such that $L(C) = L(B)?L(A)$
3. Submit $< A, C >$ to the decider for EQDFA

4. If it accepts, *accept*.

5. If it rejects, *reject*."

M is a decidable since steps 1, 2, 4, and 5 will not create and infinite loops and step 2 calls a decider. Also, M accepts $< R, S >$ if and only if $L(R) = L(R)$? $L(S)$. Therefore, M is a decider for A so A is decidable.

$\boxed{5}$ **Problem:** Given a CFG G, is $L(G) = \Sigma^*$

(a) CFGs are strictly more powerful than REs, this problem is in both RE and $\overline{RE}$.

(b)

$\boxed{6}$ Sipser, exercise 5.9

Proof:

We show that the language T is undecidable by showing that $H_{TM} \geq T$. The reduction function f takes as input (M, w) where M is a Turing Machine and w is a string, and outputs $f(M, w) = (N_{M,w})$, where $N_{M,w}$ is the Turing Machine that does the following:

- On input x:
  1. If $x = 01$, then *accept*.
  2. Else if $x \neq 10$, then *reject*.
  3. Else then simulate M on w. If M halts on w, then *accept*.

Clearly, the function f is computable. If $(M, w) \in H_{TM}$ or M halts on w then $L(N_{M,w}) = \{01, 10\}$, thus $(N_{M,w}) \notin T$. We conclude that $(M, w) \in H_{TM}$ if and only if $f(M, w) \in T$. Therefore f is a reduction from H_{TM} to T. It follows that T is undecidable as H_{TM} is undecidable.
$\square$

$\boxed{7}$ Sipser, exercise 5.13

Proof:

We show that $USELESS_{TM}$ is undecidable by showing that $\overline{H_{TM}} \geq USELESS_{TM}$. The reduction function f takes as input (M, w) where M is a Turing Machine and w is a string, and outputs $f(M, w) = (N_{M,w})$, where $N_{M,w}$ is the Turing Machine that does the following:

- On input symbol 0, $N_{M,w}$ simply enters all it's states except for a special state $\overline{q}$.
- On input symbol 1, $N_{M,w}$ simulates M on w, if M halts on w, then enters state $\overline{q}$.

Clearly, the function f is computable. From the behavior of $N_{M,w}$ on input symbol 0, it is clear that only the special state $\overline{q}$ can be useless, and from the behavior of $N_{M,w}$ on input symbol 1, it is clear that $\overline{q}$ is useless if and only if M does not halt on w. Therefore $(M, w) \in \overline{H_{TM}}$ if and only if $f(M, w) \in USELESS_{TM}$, and hence f is a reduction from $\overline{H_{TM}}$ to $USELESS_{TM}$. It follows that $USELESS_{TM}$ is undecidable as H_{TM} is undecidable.

□