7 Is there a set which is not decidable?

$\{ \langle p, d \rangle : p \text{ halts on } d \}$ is not decidable.

The set of all programs, $\{p : p \text{ always return } 0\}$ is not decidable.

Def: A set A is called recursively enumerable (r.e.) if $\exists$ an algorithm that eventually prints all elements of A.

```
n = 0;
while (true) { System.out.println (n);
               n++; }
```

This shows that {n} is recursively enumerable.

Tuesday 02.24.09

– A is decidable :–

we know : A is decidable.

~~means : there exists an algorithm that, given n, checks whether~~ $n \notin A$.

```
public static boolean inA (int n) { ... }
```

we want to prove :– – A is decidable.

n → [ ] → we want to have :

an algorithm inComplToA(n) :

$\text{inComplementToA}(n) = \begin{cases} \text{true if } n \in -A \iff n \notin A \\ \text{false otherwise} \end{cases}$

```
public static boolean inCompToA (int n) { return !inA(n); }
```

$n \in -A \iff n \notin A$.

A decidable $\iff \exists$ algorithm that, given n, check whether $n \in A$.

A r.e. $\iff \exists$ algorithm that eventually prints all elements of A.

↳ semi-decidable.

① $\mathbb{N}$ is r.e.

(2) ϕ is r.e.

(3) Every decidable set r.e.

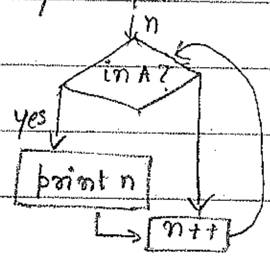
Let's say, A is decidable.

```
public static boolean inA(int n)
   n = 0;
   while (true) {  if (inA(n))
                       System.out.println(n);
                       n++;
   }
```

Thus we can print all $n \in A$.

$\therefore$ A is r.e.

Another way is by flowchart :-



(4) Every finite set is r.e. → since finite set is decidable

(5) Every co-finite (complement to finite) set is r.e. → since ¬(finite set) is decidable.

(6) If A is r.e. & B is r.e. then $A \cup B$ is r.e.?

we have inA & inB.

Proof: run inA for 1 hour
run inB for 1 hour
run inA for 1 more hr
run inB " " "
⋮

time-sharing algorithm

say A : odd nos B : even nos.

1
3
5
7
2
4
6
11
13
8
10
⋮

Thus we print $A \cup B$.

(7) if A is r.e & B is r.e then $A \cap B$ is r.e.?

run inA for 1 hr. (make partial list)
run inB for 1 hr. (")
~~print all common elements.~~
run inA for 1 more hr. (make longer list)
run inB " 1 more hr. (")
print all common elements.

$n \in A \cap B$

(7) t_A → appearance time in A of n

(11) t_B → time to appear n in B

thus, n's appearance time in $A \cap B$ is always $2 \cdot \max(t_A, t_B)$

here : $2 \cdot 11 = 22$ hrs.

[box] → prints all elements of A

[box] → " " " which are not in A

(8) If A is r.e & −A is r.e. then A is decidable.

run in A for 1 hr

run in −A for 1 hr

run in A for 1 more hr

run in −A for 1 more hr

⋮

Stops when n appears in one of the lists.

[decidable means, checks if any n ∈ set.]

(9) If A is r.e, is the −A is r.e.? → This is wrong hypothesis ~~for~~ becoz of (10)

* [we had an example of a set which is not decidable.

$H = \{(p,d) : p \text{ halts on } d\}$

H is r.e.

All rational numbers.

1/1 →	1/2 →	1/3 →	1/4	1/5
2/1	2/2	2/3 ↙	2/4	
3/1 ←	3/2 ↙ →	3/3 →	3/4	

irrespective of the order, we can print all $n \in \mathbb{R}$

run program #0 and #1 for 1 hr on data 0, 1

if halts print (p, d).

run programs 0, 1, and 2 for 2 hrs on data 0, 1, 2.

if halts print (p, d).

run programs 0, 1, 2, & 3 for 3 hr on data 0, 1, 2, 3

if halts print (p, d)

... for p = 7, d = 3

& t = 5

Thus halt-checker is r.e. A

∴ halt-checker is not decidable.

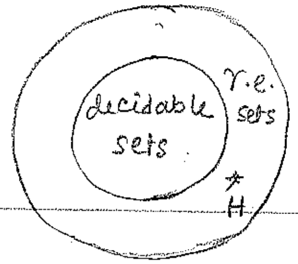
(10) Thus there exist a set which is r.e., but not decidable.

* there exist A which is r.e but −A is not r.e.

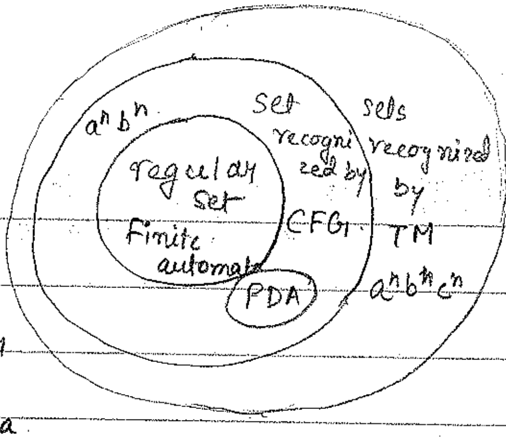
(11) Is every set r.e.?

No, there is a set which is not r.e. (−H)

− halt-checker



All sets.
* — H.



CFG:
context free grammar
PDA:
pushdown automata

Semi-decidable :-

$inA(n) = \begin{cases} \text{true if } n \in A \\ \text{false if } n \notin A \end{cases}$] decidable.

$inA(n) = \begin{cases} \text{true if } n \in A \\ \text{nothing if } n \notin A \end{cases}$ → half true

↳ runs indefinitely.

if A is r.e. then A is semi-decidable :-

suppose we have an algorithm that prints all elements of A.

inA waits & every hour checks whether n was printed.

If A is semi-decidable, run inA on 0 & 1 for 1 hr.

if it halts we print the corr #

run inA on 0, 1 & 2 for 2 hr.

if halts print

.....