Computable enumerability + semidecidability

Recall what decidable meant.

Algorithm M - input w

$M(w)$ accepts iff $w \in R$ (R - relation)

$M(w)$ rejects iff $w \notin R$

So $M(w)$ halts on all inputs w.

Now allow M may run forever + not produce an answer.

Def'n Let R be a k-ary relation. Then R is semidecidable if there is an algorithm M s.t. for all $w_1, \ldots w_k \in \Sigma^*$,

$M(w_1 \ldots w_k)$ accepts iff $\langle w_1 \ldots w_k \rangle \in R$
i.e. $R(w_1 \ldots w_k)$

For other $w_1 \ldots w_k \notin R$, $M(w_1 \ldots w_k)$ might reject or might run forever.

<u>Example</u>: $R = \{ \langle n, m \rangle : \exists i \geq n$ s.t. i and $i+m$ are both prime $\}$.

$R \subseteq \mathbb{N}^2$.

An algorithm that semidecides is:

<u>Input</u>: n, m

<u>Algorithm</u>

For $i = n, n+1, n+2, \ldots$

If i and $i+m$ are prime,

accept (and halt)

End-for

<u>Theorem</u> If R is decidable, then R is semidecidable

<u>Pf</u> Obvious.

Allow a ~~machine~~ algorithm to output successively arbitrarily many things.

Called an "enumerator".

Outputs may be k-tuples $v_1, v_2, v_3, \ldots$ for $(\Sigma^*)^k$

Could be finitely many v_i's, infinitely many, even zero

Algorithm does not take any input.

<u>Defn</u> Let R be a k-ary relation. R is <u>computably enumerable</u> iff there is an enumerator algorithm s.t. for all $v \in (\Sigma^*)^k$,

$v \in R$ iff v is enumerated as one of the outputs $v_1, v_2, \ldots$ of the enumerator.

We don't require any order of enumeration.

Theorem: R is computably enumerable if and only if R is semi-decidable.

Example $\{\langle n, m\rangle : \exists i \geq n,\ i \text{ and } i+m \text{ are prime}\}$

Algorithm to enumerate this relation:

```
For k = 2, 3, 4, ...
    For l = 2, 3, 4, ..., k-2        (So l < k)
        If l and k are prime,
            For n = 0, 1, 2, ..., l
                Output <n, k-l>  (and keep running)
            End-for
        End-if
    End for
End-for.
```

OK to output same thing more than once.

Proof: Suppose R is computable enumerable by algorithm M. Then R is semidecided by:

Input $v \in (\Sigma^*)^k$

Algorithm

Run M, watch what it outputs.

If v is ever output, accept (+ halt).

Suppose R is semidecidable by algorithm N.

Here is an algorithm to computably enumerate R:

Algorithm:

Let $v_1, v_2, v_3, v_4, \ldots$ enumerate all members of $(\Sigma^*)^k$ in order of increasing length

v_i - i-th member of $(\Sigma^*)^k$

For $j = 1, 2, 3, \ldots$

For $\ell = 1, 2, 3, \ldots j$

Let u_ℓ be the ℓ-th member of $(\Sigma^*)^k$,

Run $N(u_\ell)$ for up to j steps.

If $N(u_\ell)$ accepts in $\leq j$ steps, then output u_ℓ.

End-for End-for

"Dovetailing algorithm"

How to enumerate Σ^*: Let $\Sigma = \{a_1 \dots a_\ell\}$

$$\underbrace{\varepsilon}_{\substack{\text{length}\\ 0}}, \quad \underbrace{a_1, \dots, a_\ell}_{\substack{\ell \text{ many}\\ \text{of length } 1}} \quad \underbrace{a_1a_1 \quad a_1a_2 \quad \cdot \quad a_\ell a}_{\substack{\ell^2 \text{ many}\\ \text{of length } 2}} \quad \underbrace{\qquad}_{\substack{\ell^3 \text{ many}\\ \text{of length } 3.}}$$

To enumerate $(\Sigma^*)^k$: If $\langle w_1 \dots w_k \rangle \in (\Sigma^*)^k$,

let $|\langle w_1 \dots w_k \rangle| = \sum_i |w_i|$

Enumerate members of $(\Sigma^*)^k$ in increasing order of total length $\sum_i |w_i|$.

<u>Thm</u> R is decidable iff R and $\bar{R}$ are computably enumerable.

<u>Proof</u>: Already showed:
- R decidable $\Rightarrow$ R is semidecidable
- R decidable $\Rightarrow$ $\bar{R}$ is decidable

$\bar{R} = (\Sigma^*)^k \setminus R$

Proof - continued

Suppose R is computably enumerable by algorithm M

" $\overline{R}$ " " " " " N.

Here is an algorithm that <u>decides</u> R:

<u>Input</u> $v \in (\Sigma^*)^k$

<u>Algorithm</u>

Run M and N in parallel.

If M outputs ω, accept

If N outputs ω, reject

One of these two possibilities eventually happens.

So the algorithm always halts

<u>Terminology</u> computable enumerable is also called "c.e"

A relation is often called a "set"

A unary relation is often called a "language".

A Language L is a subset of Σ^*.

"M semidecides R", often instead, say

"M recognizes R" or

"M accepts R".

<u>Defn</u> A <u>partial</u> <u>computable</u> function f is a function with domain a subset of $(\Sigma^*)^k$ and range Σ^*, and s.t. there is an algorithm M that on input $v \in (\Sigma^*)^k$

$M(v) = w$ if $f(v) = w$.

$M(v)$ never halts if $f(v)$ is undefined

<u>Notation</u> $f(v)\downarrow$, $f(v)$ <u>converges</u> if $v \in \text{domain}(f)$

$f(v)\uparrow$, $f(v)$ <u>diverges</u> if $v \notin \text{dom}(f)$

$$f(n,m) = \begin{cases} \text{the least } i \geq n, \text{ s.t. } i \text{ and } i+m \text{ are prime} \\ \text{undefined if no such } i \text{ exists} \end{cases}$$

$$f(n,m) = (\mu i)[\, i \geq n \text{ and } i \text{ and } i+m \text{ are prime} \,]$$

"the least i such that ..."