Alternate Terms

- Turing-decidable: decidable, recursive, rec
- Turing-recognizable: recognizable, semi-decidable, recursively enumerable, RE, re
- $\leq_T$: Turing-reducible, reducible
- $\leq_m$: mapping-reducible, many-one-reducible

Definitions

1. A set $B$ is used as an oracle if we allow a computation to ask questions of the form “$y \in B$” as a basic step. Think of it as allowing the computation as having a sub-routine that determines membership in $B$.
2. A set $A$ is $B$-recursive if $A$ can be decided using a $B$-oracle. Similarly, $A$ is $B$-re if it can be recognized using a $B$-oracle.
3. We say $A$ reduces to $B$ ($A \leq_T B$) if $A$ is $B$-recursive.
4. $A$ m-reduces to $B$ ($A \leq_m B$) if there is a Turing-computable string function $f : \Sigma^* \rightarrow \Sigma^*$ such that for all strings $w \in \Sigma^*$ we have
   \[ w \in A \iff f(w) \in B \]
5. A set $K$ is RE-hard if, for all recognizable sets $A$, $A \leq_m K$.
6. $K$ is RE-complete if $K$ is both recognizable and RE-hard. Alternate terms:
   - m-complete for RE
   - complete for the recognizable sets under $\leq_m$

“Obvious?” Facts

- Both $\leq_T$ and $\leq_m$ are reflexive ($A \leq_m A$) and transitive ($A \leq_m B$ and $B \leq_m C$ implies $A \leq_m C$)
- $A \leq_m B$ implies $A \leq_T B$
- $A \leq_T \overline{A}$
- \( A \leq_m B \) iff \( \bar{A} \leq_m \bar{B} \)
- the following are equivalent: \( A \leq_T B, A \leq_T \bar{B}, \bar{A} \leq_T B, \bar{A} \leq_T \bar{B} \)
- \( A_{TM} \) is RE-complete (proof below)
- If
  - \( K \) is RE-complete,
  - \( K \leq_m K' \), and
  - \( K' \) is RE

then \( K' \) is RE-complete (follows from definition of complete and transitivity of \( \leq_m \)).
- If both \( A \) and \( \bar{A} \) are RE, then \( A \) is recursive. (Theorem 4.22 of text says it better.)

More Facts, Definitions

\( A_{TM} \) is RE-complete

(proof) First, \( A_{TM} \) is RE, thanks to the existence of a universal TM \( U \). Now we have to show that any RE set \( A \) m-reduces to \( A_{TM} \). Since \( A \) is RE, it is recognized by some TM \( M \). By defining \( f(w) = \langle M, w \rangle \) we get

\[
w \in A \iff M \text{ accepts } w \iff f(w) = \langle M, w \rangle \in A_{TM}\]

\( A_{TM} \leq_T HALT_{TM} \)

(proof) pretty easy, done in text

\( A_{TM} \leq_m HALT_{TM} \), so therefore HALT_{TM} is also RE-complete

(proof) Also sort of easy, but subtle. We show a computable \( f \) such that on input \( \langle M, w \rangle \), \( f(\langle M, w \rangle) = \langle M', w \rangle \) so that

\[
M \text{ accepts } w \iff M' \text{ halts on } w
\]

What \( M' \) will do is simulate \( M \) on its input: \( M' \) halts if \( M \) accepts but it goes into \( \infty \)-loop if \( M \) rejects. (Obviously, if \( M \) loops then so will \( M' \)) Notably, the constructor for \( f \) does not run \( M \), instead it wraps the “code” for \( M \) with “code” that will simulate it and behave as described.

Closure properties for \( \leq_m \) and \( \leq_T \)

1. The recursive sets are closed under \( \leq_m \): if \( A \leq_m B \) and \( B \) is recursive, then \( A \) is recursive.
2. The RE sets are closed under \( \leq_m \): if \( A \leq_m B \) and \( B \) is RE, then \( A \) is RE.
3. The recursive sets are closed under \( \leq_T \): if \( A \leq_T B \) and \( B \) is recursive, then \( A \) is recursive.
4. The RE sets are not closed under \( \leq_T \): for example \( A_{TM} \) is RE and \( \bar{A}_{TM} \leq_T A_{TM} \), but \( \bar{A}_{TM} \) is not RE (if it were, then \( A_{TM} \) would be recursive).

Arithmetic Hierarchy
• $\Sigma_0 = \Delta_0 = \Delta_1 = \Pi_0 = \text{recursive}$
• $\Sigma_1 = \text{recursively-enumerable}$
• $\Pi_1 = \text{co-RE} = \{ A \mid \overline{A} \in \Sigma_1 \}$
• in general $\Pi_k = \text{co-}\Sigma_k = \{ A \mid \overline{A} \in \Sigma_k \}$
• $\Delta_{k+1} = \{ A \mid A \text{ is } B\text{-recursive for some } B \in \Sigma_k \} = \{ A \mid A \leq_T B \text{ for some } B \in \Sigma_k \}$
• $\Sigma_{k+1} = \{ A \mid A \text{ is } B\text{-RE for some } B \in \Sigma_k \}$

**Basic AH Facts**

• $\Delta_k = \Sigma_k \cap \Pi_k$
• $\Delta_k = \text{co-}\Delta_k$
• $\Delta_k \neq \Sigma_k$ (follows from ex 7, hw 7), so $\Sigma_k \neq \Pi_k$

**Post’s Theorem**

$A$ is $\Sigma_k$ if it can be characterized as

$$A = \{ x \mid \exists y_1 \forall y_2 \exists y_3 \ldots Q_k y_k \langle x, y_1, y_2, y_3, \ldots, y_k \rangle \in B \}$$

where $B$ is decidable and quantifier $Q_k = \exists$ if $k$ is odd and $\forall$ if it is even. (Here $x$ and all the $y_i$’s are strings over the same alphabet.) Similarly, $A$ is $\Pi_k$ if we can write

$$A = \{ x \mid \forall y_1 \exists y_2 \forall y_3 \ldots \overline{Q_k} y_k \langle x, y_1, y_2, y_3, \ldots, y_k \rangle \in B \}$$

where $B$ is decidable and quantifier $\overline{Q_k} = \forall$ if $k$ is odd and $\exists$ if it is even.

**Friedberg-Muchnik Theorem**

There are two RE sets $A, B$ such that $A \not\leq_T B$ and $B \not\leq_T A$. 