ORACLE MACHINES

An ORACLE is a set $B$ to which the TM may pose membership questions “Is $w$ in $B$?” (formally: TM enters state $q_i$) and the TM always receives a correct answer in one step (formally: if the string on the “oracle tape” is in $B$, state $q_i$ is changed to $q_{\text{YES}}$, otherwise $q_{\text{NO}}$).

This makes sense even if $B$ is not decidable! (We do not assume that the oracle $B$ is a computable set!)

THE ARITHMETIC HIERARCHY

We say $A$ is semi-decidable in $B$ if there is an oracle TM $M$ with oracle $B$ that semi-decides $A$.

We say $A$ is decidable in $B$ if there is an oracle TM $M$ with oracle $B$ that decides $A$.

\[ \Delta^0_1 = \{ \text{decidable sets} \} \quad (\text{sets = languages}) \]
\[ \Sigma^0_1 = \{ \text{semi-decidable sets} \} \]
\[ \Sigma^0_{n+1} = \{ \text{sets semi-decidable in some } B \in \Sigma^0_n \} \]
\[ \Delta^0_{n+1} = \{ \text{sets decidable in some } B \in \Sigma^0_n \} \]
\[ \Pi^0_n = \{ \text{complements of sets in } \Sigma^0_n \} \]
Definition: A decidable predicate $R(x,y)$ is some proposition about $x$ and $y$, where there is a TM $M$ such that

for all $x, y$, $R(x,y)$ is TRUE $\Rightarrow$ $M(x,y)$ accepts
$R(x,y)$ is FALSE $\Rightarrow$ $M(x,y)$ rejects

We say $M$ “decides” the predicate $R$.

**EXAMPLES:**

$R(x,y) = \text{“} x + y \text{ is less than 100} \text{”}$

$R(<N>,y) = \text{“} N \text{ halts on } y \text{ in at most 100 steps} \text{”}$

Kleene’s T predicate, $T(<M>, x, y): M$ accepts $x$ in $y$ steps.

1. $x, y$ are positive integers or elements of $\Sigma^*$

Definition: A decidable predicate $R(x,y)$ is some proposition about $x$ and $y$, where there is a TM $M$ such that

for all $x, y$, $R(x,y)$ is TRUE $\Rightarrow$ $M(x,y)$ accepts
$R(x,y)$ is FALSE $\Rightarrow$ $M(x,y)$ rejects

We say $M$ “decides” the predicate $R$.

**EXAMPLES:**

$R(x,y) = \text{“} x + y \text{ is less than 100} \text{”}$

$R(<N>,y) = \text{“} N \text{ halts on } y \text{ in at most 100 steps} \text{”}$

Kleene’s T predicate, $T(<M>, x, y): M$ accepts $x$ in $y$ steps.

1. $x, y$ are positive integers or elements of $\Sigma^*$

Theorem: A language $A$ is semi-decidable if and only if there is a decidable predicate $R(x, y)$ such that $A = \{ x \mid \exists y \ R(x,y) \}$

Proof:

(1) If $A = \{ x \mid \exists y \ R(x,y) \}$ then $A$ is semi-decidable

(2) If $A$ is semi-decidable, then $A = \{ x \mid \exists y \ R(x,y) \}$
Theorem: A language $A$ is semi-decidable if and only if there is a decidable predicate $R(x, y)$ such that: $A = \{ x \mid \exists y R(x, y) \}$

Proof:

1. If $A = \{ x \mid \exists y R(x, y) \}$ then $A$ is semi-decidable
   Because we can enumerate over all $y$'s

2. If $A$ is semi-decidable, then $A = \{ x \mid \exists y R(x, y) \}$
   Let $M$ semi-decide $A$
   Then, $A = \{ x \mid \exists y T(<M>, x, y) \}$ (Here $M$ is fixed.)
   where
   Kleene's $T$ predicate, $T(<M>, x, y)$: $M$ accepts $x$ in $y$ steps.

---

Theorem

$$\sum_1^0 = \{ \text{semi-decidable sets} \}$$

= languages of the form $\{ x \mid \exists y R(x, y) \}$

$$\Pi_1^0 = \{ \text{complements of semi-decidable sets} \}$$

= languages of the form $\{ x \mid \forall y R(x, y) \}$

$$\Delta_1^0 = \{ \text{decidable sets} \}$$

= $\sum_1^0 \cap \Pi_1^0$

Where $R$ is a decidable predicate

---

Theorem

$$\sum_2^0 = \{ \text{sets semi-decidable in some semi-dec. } B \}$$

= languages of the form $\{ x \mid \exists y_1 \forall y_2 R(x, y_1, y_2) \}$

$$\Pi_2^0 = \{ \text{complements of } \sum_2^0 \text{ sets} \}$$

= languages of the form $\{ x \mid \forall y_1 \exists y_2 R(x, y_1, y_2) \}$

$$\Delta_2^0 = \sum_2^0 \cap \Pi_2^0$$

Where $R$ is a decidable predicate

---

Theorem

$$\sum_n^0 = \text{languages } \{ x \mid \exists y_1 \forall y_2 \exists y_3 \ldots \exists y_n R(x, y_1, \ldots, y_n) \}$$

$$\Pi_n^0 = \text{languages } \{ x \mid \forall y_1 \exists y_2 \forall y_3 \ldots \exists y_n R(x, y_1, \ldots, y_n) \}$$

$$\Delta_n^0 = \sum_n^0 \cap \Pi_n^0$$

Where $R$ is a decidable predicate

---

Example

Decidable predicate

$$\sum_1^0 = \text{languages of the form } \{ x \mid \exists y R(x, y) \}$$

We know that $A_{TM}$ is in $\sum_1^0$
Why?
Show it can be described in this form:
\[ \Pi_1^0 = \text{languages of the form } \{ x | \forall y \text{ R}(x,y) \} \]

Show that \( \text{EMPTY} \) (ie, \( E_{TM} = \{ M | L(M) = \emptyset \} \)) is in \( \Pi_1^0 \)

\[ \text{EMPTY} = \{ M | \forall z \left[ \neg T(<M>, w, t) \right] \} \]

two quantifiers??
decidable predicate

---

**THE PAIRING FUNCTION**

Theorem. There is a 1-1 and onto computable function \( <, > : \Sigma^* \times \Sigma^* \rightarrow \Sigma^* \) and computable functions \( \pi_1 \) and \( \pi_2 : \Sigma^* \rightarrow \Sigma^* \) such that

\[ z = <w, t> \implies \pi_1(z) = w \text{ and } \pi_2(z) = t \]

**THE PAIRING FUNCTION**

Theorem. There is a 1-1 and onto computable function \( <, > : \Sigma^* \times \Sigma^* \rightarrow \Sigma^* \) and computable functions \( \pi_1 \) and \( \pi_2 : \Sigma^* \rightarrow \Sigma^* \) such that

\[ z = <w, t> \implies \pi_1(z) = w \text{ and } \pi_2(z) = t \]

Proof: Let \( w = w_1...w_n \in \Sigma^* \), \( t \in \Sigma^* \).
Let \( a, b \in \Sigma \), \( a \neq b \).

\(<w, t> := a w_1...a w_n b t\)

\[ \pi_1(z) := \text{"if } z \text{ has the form } a w_1...a w_n b t, \text{ then output } w_1...w_n \text{, else output } \varepsilon \” \]

\[ \pi_2(z) := \text{"if } z \text{ has the form } a w_1...a w_n b t, \text{ then output } t, \text{ else output } \varepsilon \” \]
\[ \Pi_2^0 = \text{languages of the form } \{ x | \forall y \exists z R(x,y,z) \} \]

Show that \( \text{TOTAL} = \{ M | M \text{ halts on all inputs} \} \) is in \( \Pi_2^0 \)

\[ \Sigma_2^0 = \text{languages of the form } \{ x | \exists y \forall z R(x,y,z) \} \]

Show that \( \text{FIN} = \{ M | L(M) \text{ is finite} \} \) is in \( \Sigma_2^0 \)

\[ \Sigma_3^0 = \text{languages of the form } \{ x | \exists y \forall z \exists u R(x,y,z,u) \} \]

Show that \( \text{COF} = \{ M | L(M) \text{ is cofinite} \} \) is in \( \Sigma_2^0 \)
Each is m-complete for its level in hierarchy and cannot go lower (by the SuperHalting Theorem, which shows the hierarchy does not collapse).

L is m-complete for class C if
i) L ∈ C and
ii) L is m-hard for C,
i.e., for all L’ ∈ C, L’ ≤m L

A_{TM} is m-complete for class \( C = \sum_1^0 \)
i) \( A_{TM} \in C \)
ii) \( A_{TM} \) is m-hard for \( C \),

Suppose \( L \in C \). Show: \( L \leq_m A_{TM} \)

Let M semi-decide L. Then Map
\( \Sigma^* \to \Sigma^* \)

where \( w \to (M, w) \).

Then, \( w \in L \iff (M, w) \in A_{TM} \) QED
**FIN** is m-complete for class \( C = \Sigma_2^0 \)

i) **FIN** \( \in \) \( C \)

ii) **FIN** is m-hard for \( C \):

Suppose \( L \in C \). Show: \( L \leq_m \) FIN

So suppose \( L = \{ w | \exists y \forall z \ R(w, y, z) \} \)

where \( R \) is decided by some TM \( D \)

Map \( \Sigma^* \rightarrow \Sigma^* \)

where \( w \rightarrow N_{D,w} \)

---

**ORACLES not all powerful**

The following problem cannot be decided, even by

a TM with an oracle for the Halting Problem:

SUPERHALT = \( \{ (M,x) | M, \text{ with an oracle for the} \)

\text{Halting Problem, halts on } x \} \)

Can use diagonalization here!

Suppose \( H \) decides SUPERHALT (with oracle)

Define \( D(X) = \text{“if } H(X,X) \text{ accepts (with oracle)} \)

\text{then LOOP, else ACCEPT.”} \)

\( D(D) \) halts \( \Leftrightarrow H(D,D) \) accepts \( \Leftrightarrow D(D) \) loops…

---

**ORACLES not all powerful**

Theorem: The arithmetic hierarchy is strict.

That is, the nth level contains a language

that isn’t in any of the levels below n.

Proof IDEA: Same idea as the previous slide.

SUPERHALT\(^0\) = HALT = \( \{ (M,x) | M \text{ halts on } x \} \).

SUPERHALT\(^1\) = \( \{ (M,x) | M, \text{ with an oracle for the} \)

\text{Halting Problem, halts on } x \} \)

SUPERHALT\(^n\) = \( \{ (M,x) | M, \text{ with an oracle for} \)

SUPERHALT\(^{n-1}\), halts on } x \} \)

---

**WWW.FLAC.WS**

Read Chapter 6.4 for next time