15-453

FORMAL LANGUAGES, AUTOMATA AND COMPUTABILITY

ORACLE TURING MACHINES AND TURING REDUCIBILITY

TUESDAY, MAR 4

ORACLE TMs

Oracle

FINITE STATE CONTROL

I	N	P	U	T					

INFINITE TAPE

Computational Theology?

ORACLE TMs

Oracle for A_{TM}

Is (M,w) in A_{TM}?

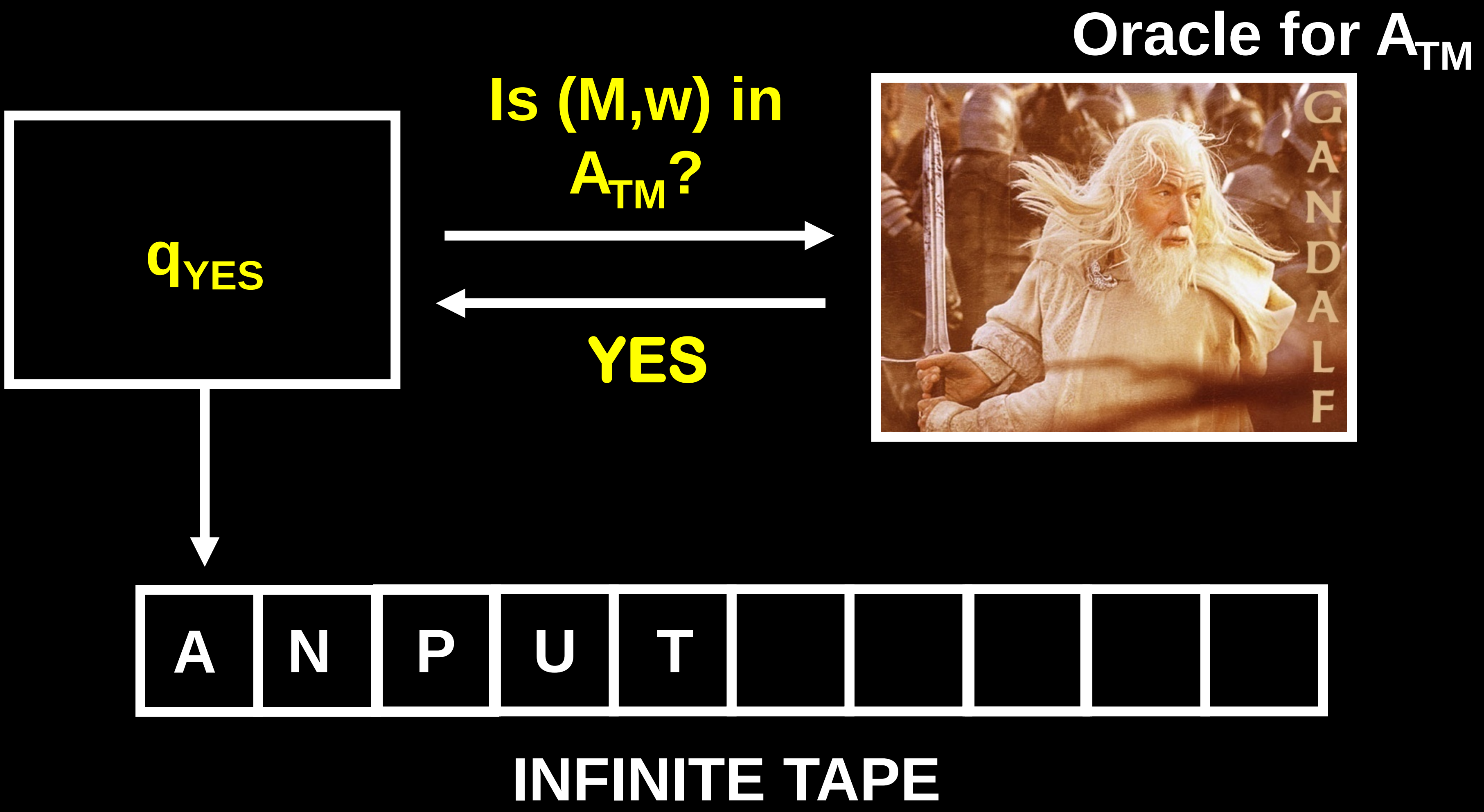
$q_?$

YES

I	N	P	U	T					

INFINITE TAPE

ORACLE TMs


Oracle for A_{TM}
Is (M,w) in A_{TM}?
q_{YES}
YES
GANDALF
A
N
P
U
T
INFINITE TAPE

ORACLE MACHINES

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

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

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

$HALT_{TM}$ is DECIDABLE in A_{TM}

On input (M,w), decide if M halts on w as follows:

1. Ask the oracle for A_{TM} whether M accepts w. If yes, then ACCEPT

2. Switch the accept and reject states of M to get M′. Ask the oracle for A_{TM} whether M′ accepts w. If yes, then ACCEPT

3. REJECT

A_{TM} is DECIDABLE in $HALT_{TM}$

On input (M,w), decide if M accepts w as follows:

Ask the oracle for $HALT_{TM}$ whether M halts on w.
If yes, then run M(w) and output its answer.
If no, then REJECT.

Language A "Turing Reduces" to Language B

if A is decidable in B, ie if there is an oracle TM M with oracle B that decides A

$$A \leq_T B$$

$\leq_T$ VERSUS $\leq_m$

Theorem: If $A \leq_m B$ then $A \leq_T B$

Proof:

If $A \leq_m B$ then there is a computable function $f : \Sigma^* \rightarrow \Sigma^*$, where for every w,

$$w \in A \Leftrightarrow f(w) \in B$$

We can thus use an oracle for B to decide A

Theorem: $\neg HALT_{TM} \leq_T HALT_{TM}$

Theorem: $\neg HALT_{TM} \not\leq_m HALT_{TM}$

THE ARITHMETIC HIERARCHY

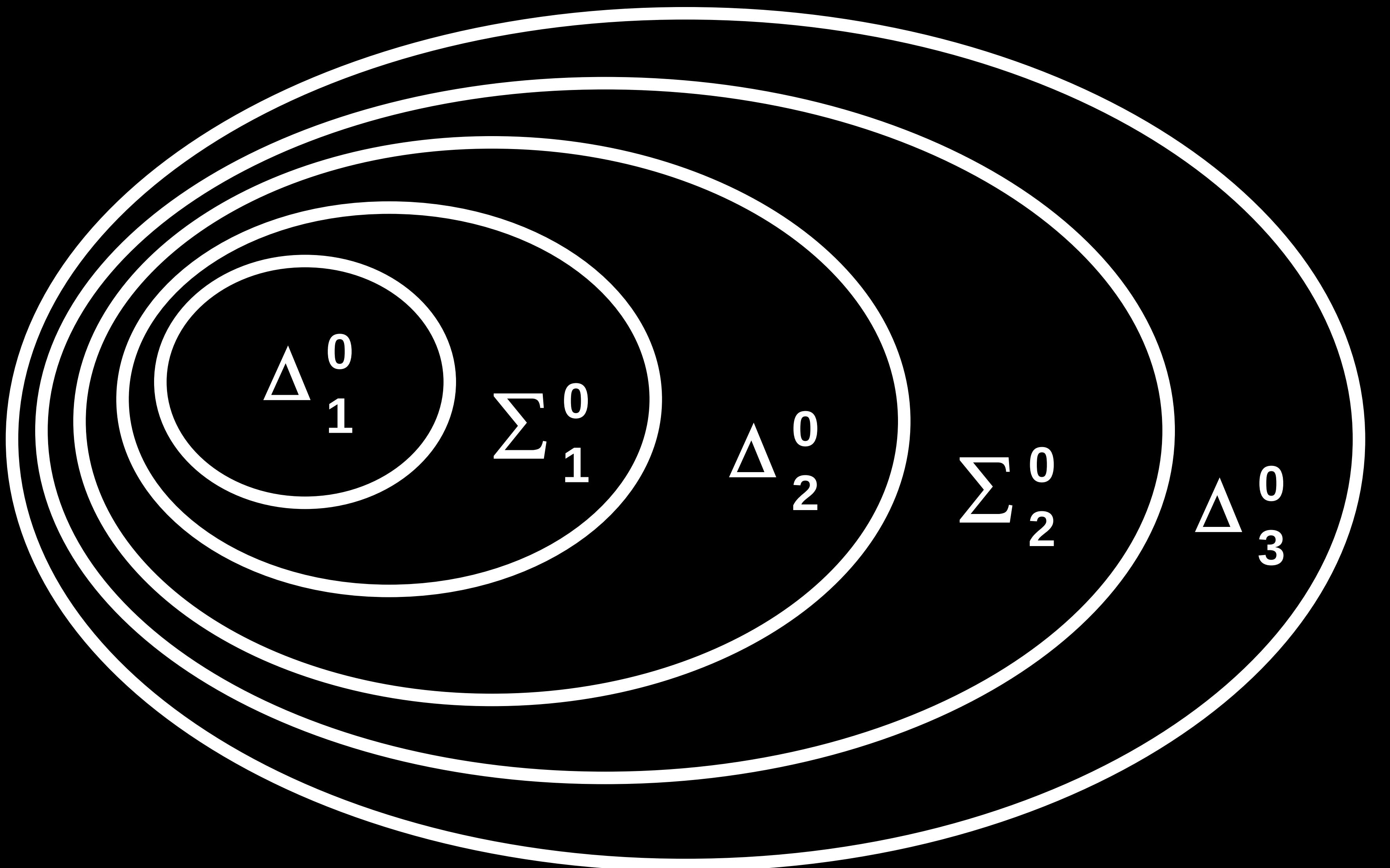
Δ^0_1 = { decidable sets }   (sets = languages)

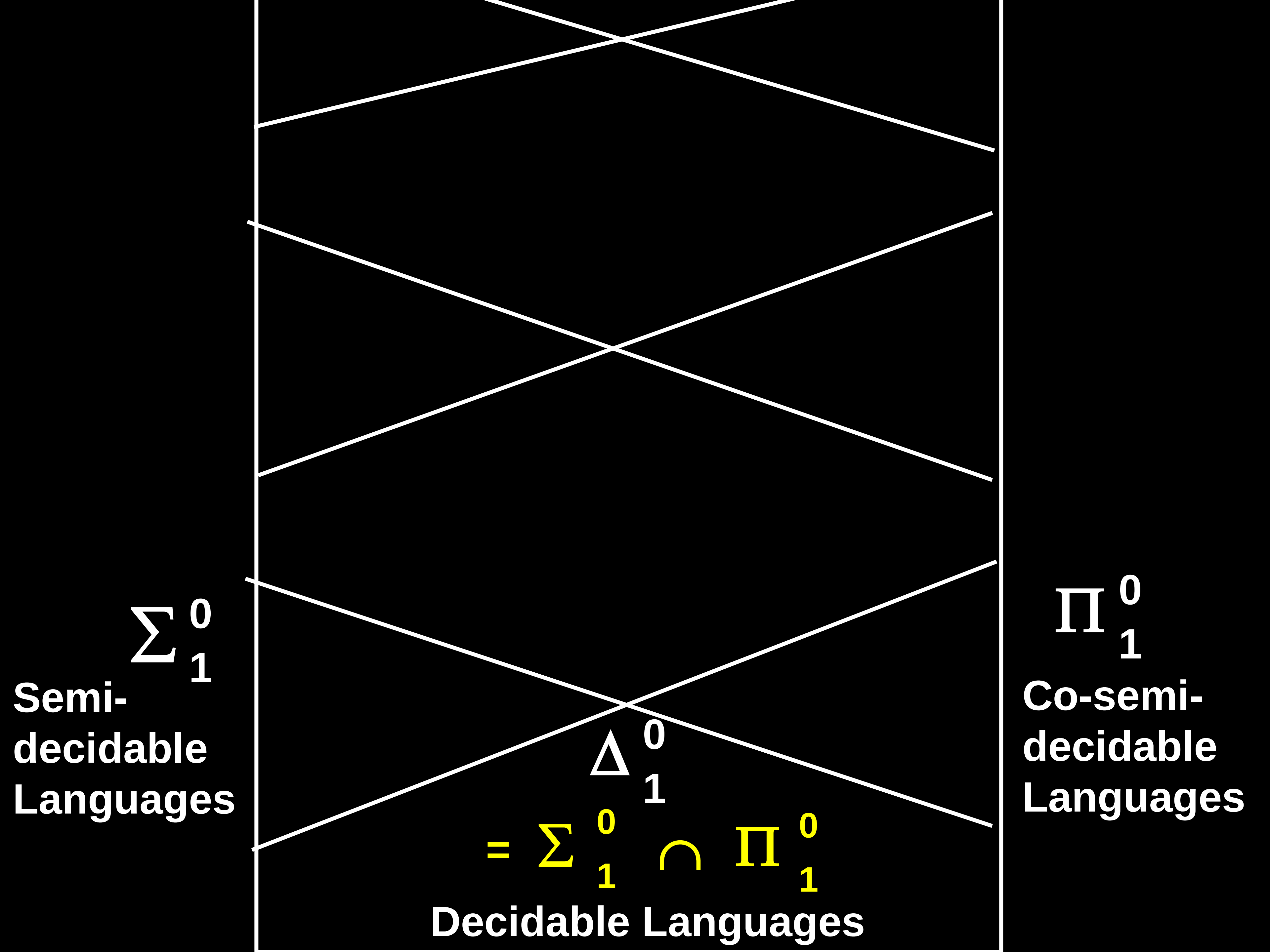
Σ^0_1 = { semi-decidable sets }

Σ^0_{n+1} = { sets semi-decidable in some B $\in \Sigma^0_n$ }

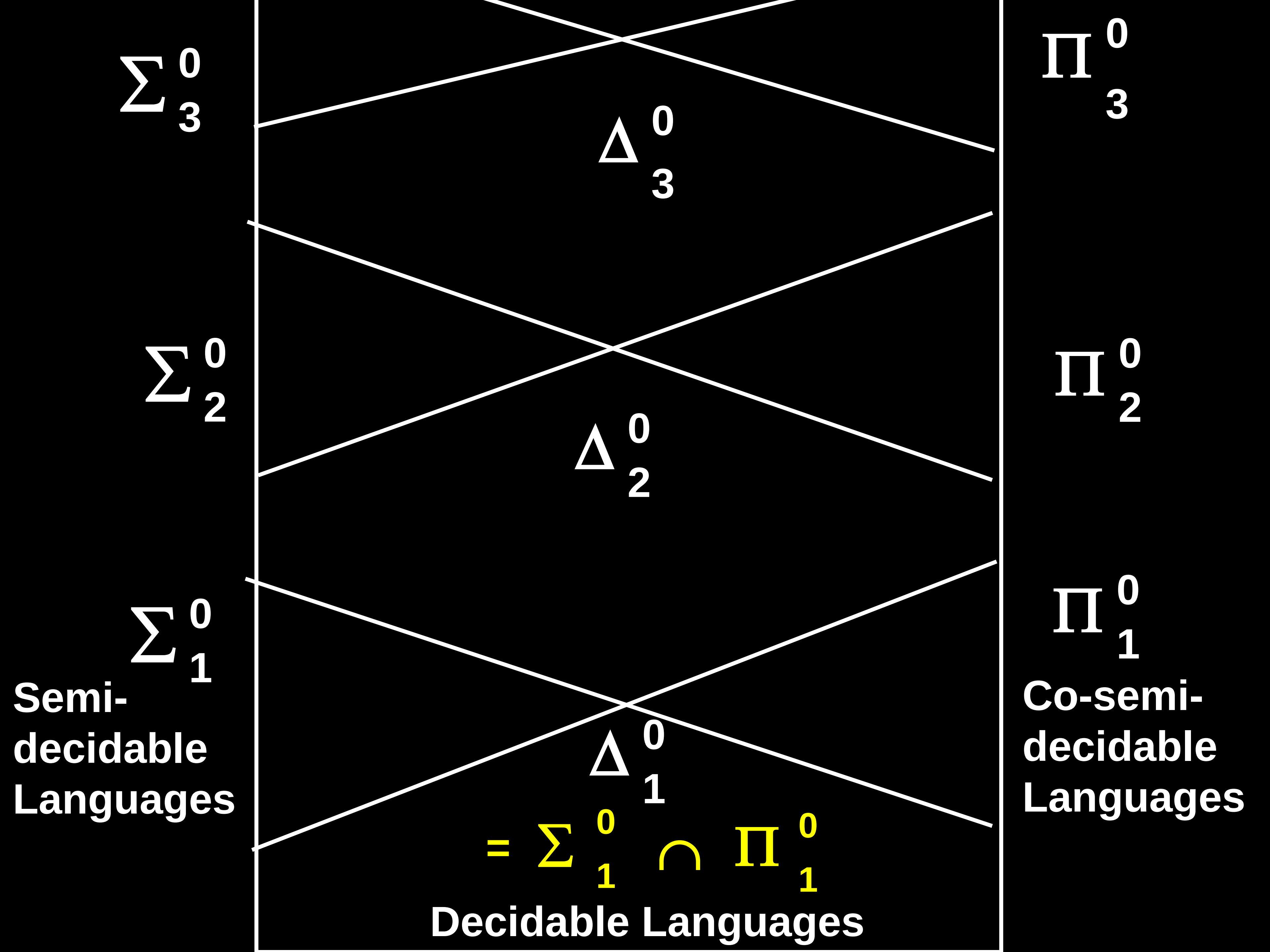
Δ^0_{n+1} = { sets decidable in some B $\in \Sigma^0_n$ }

Π^0_n = { complements of sets in Σ^0_n }


Δ^0_1
Σ^0_1
Δ^0_2
Σ^0_2
Δ^0_3



Σ^0_1
Semi-decidable Languages
Π^0_1
Co-semi-decidable Languages
Δ^0_1
$= \Sigma^0_1 \cap \Pi^0_1$
Decidable Languages



Σ^0_3
Π^0_3
Δ^0_3
Σ^0_2
Π^0_2
Δ^0_2
Σ^0_1
Π^0_1
Semi-decidable Languages
Co-semi-decidable Languages
Δ^0_1
$= \Sigma^0_1 \cap \Pi^0_1$
Decidable Languages

WWW.FLAC.WS

Read Chapter 6.4 for next time