15-453

FORMAL LANGUAGES, AUTOMATA AND COMPUTABILITY

THE ARITHMETIC HIERARCHY

THURSDAY, MAR 6

THE ARITHMETIC HIERARCHY

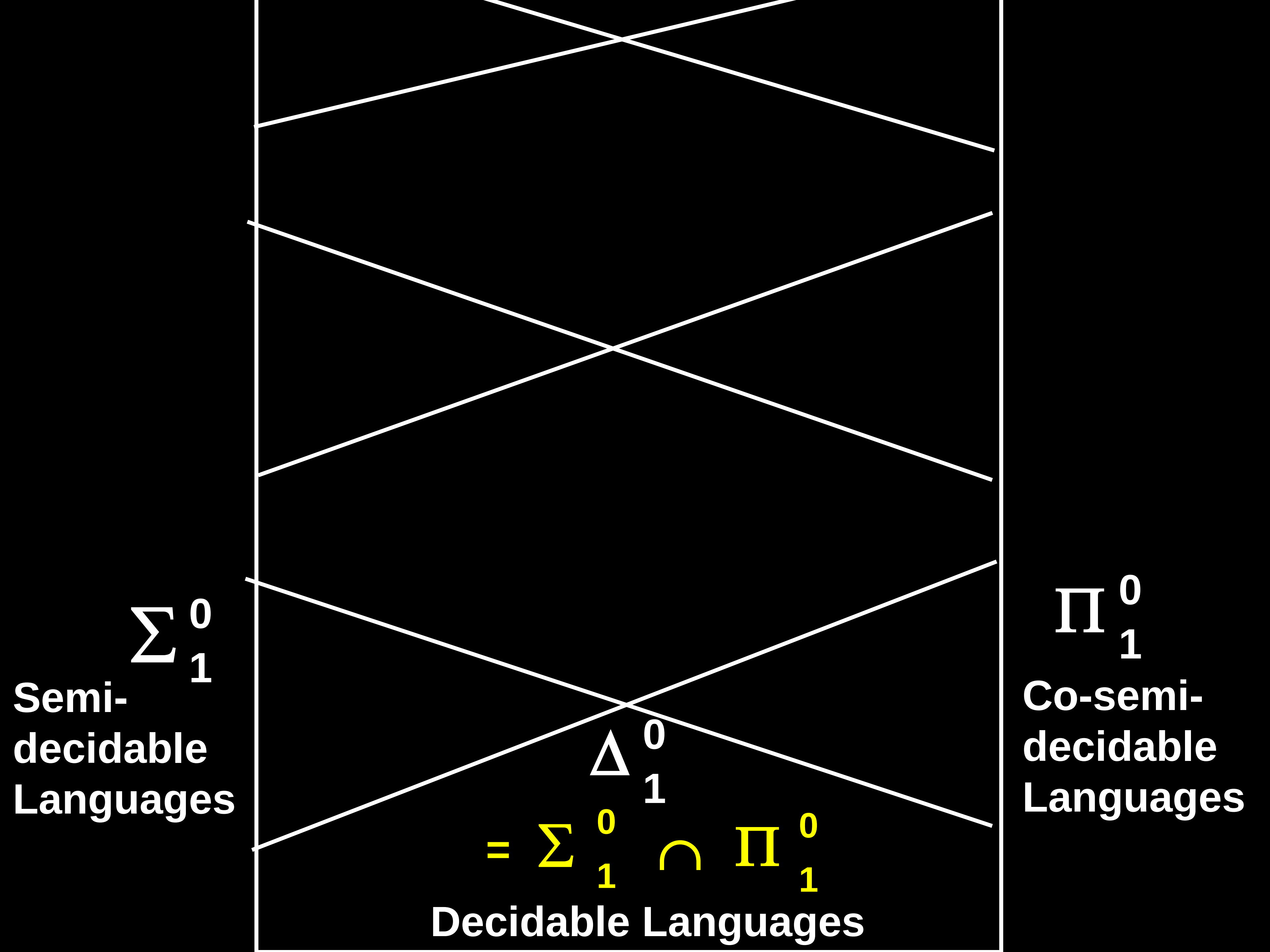
Δ^0_1 = { decidable sets } (sets = languages)

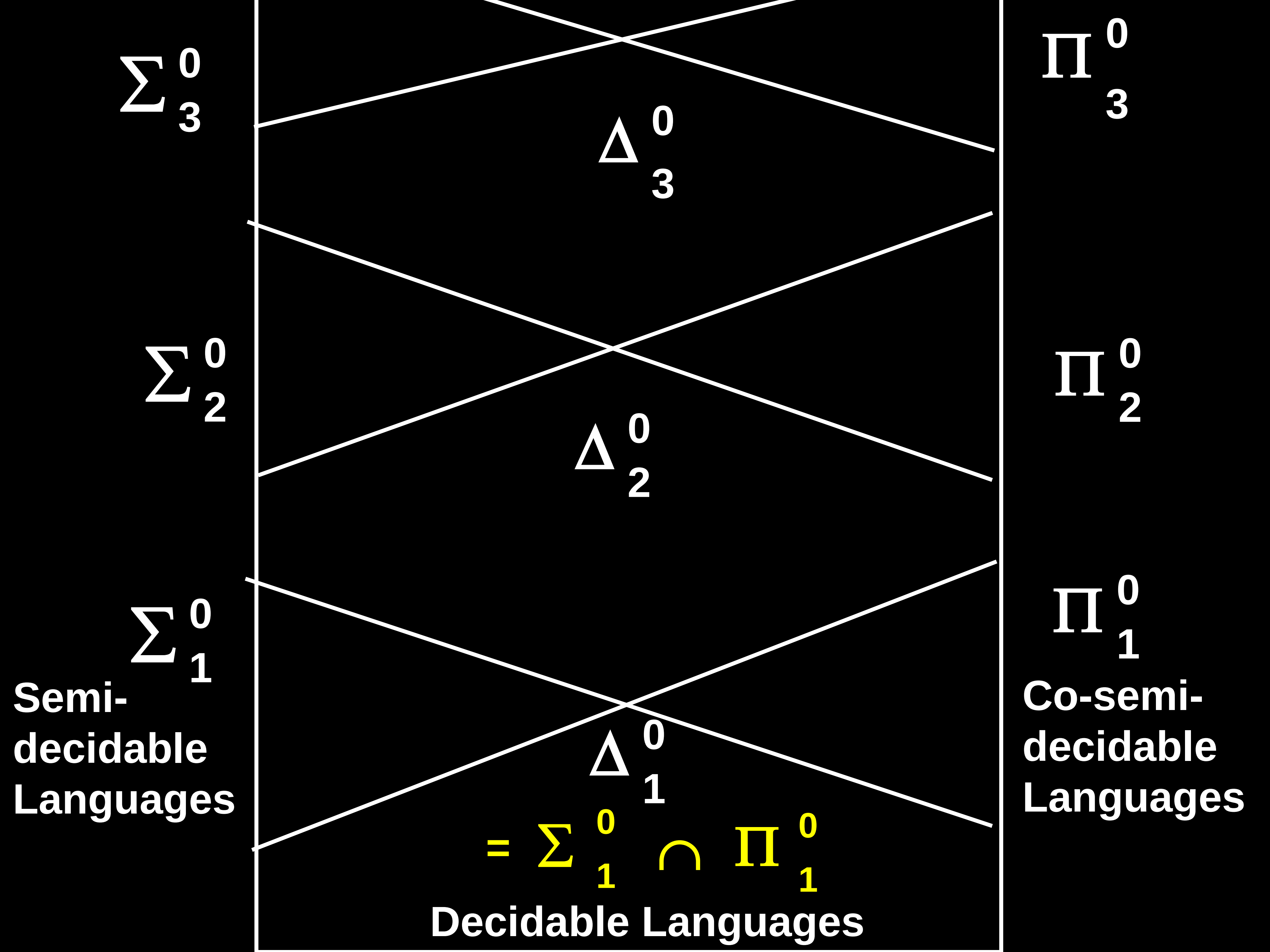
Σ^0_1 = { semi-decidable sets }

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 }


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

Definition: A **decidable predicate R(x,y)** is some proposition about **x** and **y**[1], 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**.

1. x, y are positive integers or elements of Σ^*

Definition: A **decidable predicate R(x,y)** is some proposition about x and y[1], 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) = "x + y is less than 100"

R(<N>,y) = "N halts on y in at most 100 steps"

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

1. x, y are positive integers or elements of Σ^*

Definition: A **decidable predicate R(x,y)** is some proposition about x and y[1], 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) = "x + y is less than 100"

R(<N>,y) = "N halts on y in at most 100 steps"

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

> **Note:** A is decidable $\Leftrightarrow$ A = {x | R(x,ε)}, for some decidable predicate R.

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:

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 $A = \{ x \mid \exists y\ R(x,y) \}$

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, $A = \{ x \mid \exists y\, T(\langle M \rangle, x, y) \}$ (Here M is fixed.)

where

Kleene's T predicate, T($\langle M \rangle$, x, y): M accepts x in y steps.

Theorem

Σ^0_1 = { semi-decidable sets }

= languages of the form { x | ∃y R(x,y) }

Π^0_1 = { complements of semi-decidable sets }

= languages of the form { x | ∀y R(x,y) }

Δ^0_1 = { decidable sets }

$= \Sigma^0_1 \cap \Pi^0_1$

Where R is a decidable predicate

Theorem

Σ^0_2 = { sets semi-decidable in some semi-dec. B }
= languages of the form { x | $\exists y_1 \forall y_2$ R(x,y_1,y_2) }

Π^0_2 = { complements of Σ^0_2 sets}
= languages of the form { x | $\forall y_1 \exists y_2$ R(x,y_1,y_2) }

$\Delta^0_2 = \Sigma^0_2 \cap \Pi^0_2$

Where R is a decidable predicate

Theorem

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

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

$\Delta^0_n = \Sigma^0_n \cap \Pi^0_n$

Where R is a decidable predicate

Example

Decidable predicate

Σ^0_1 **= languages of the form { x | ∃y R(x,y) }**

We know that A_{TM} is in Σ^0_1 Why?

Show it can be described in this form:

Example

Decidable predicate

Σ^0_1 **= languages of the form { x | ∃y R(x,y) }**

We know that A_{TM} is in Σ^0_1 Why?

Show it can be described in this form:

A_{TM} = { <(M,w)> | ∃t [M accepts w in t steps] }

decidable predicate

Example

Decidable predicate

Σ^0_1 **= languages of the form { x | ∃y R(x,y) }**

We know that A_{TM} is in Σ^0_1 Why?

Show it can be described in this form:

A_{TM} = { <(M,w)> | ∃t [M accepts w in t steps] }

decidable predicate

A_{TM} = { <(M,w)> | ∃t T (<M>, w, t }

Example

Decidable predicate

Σ^0_1 **= languages of the form { x | ∃y R(x,y) }**

We know that A_{TM} is in Σ^0_1 Why?

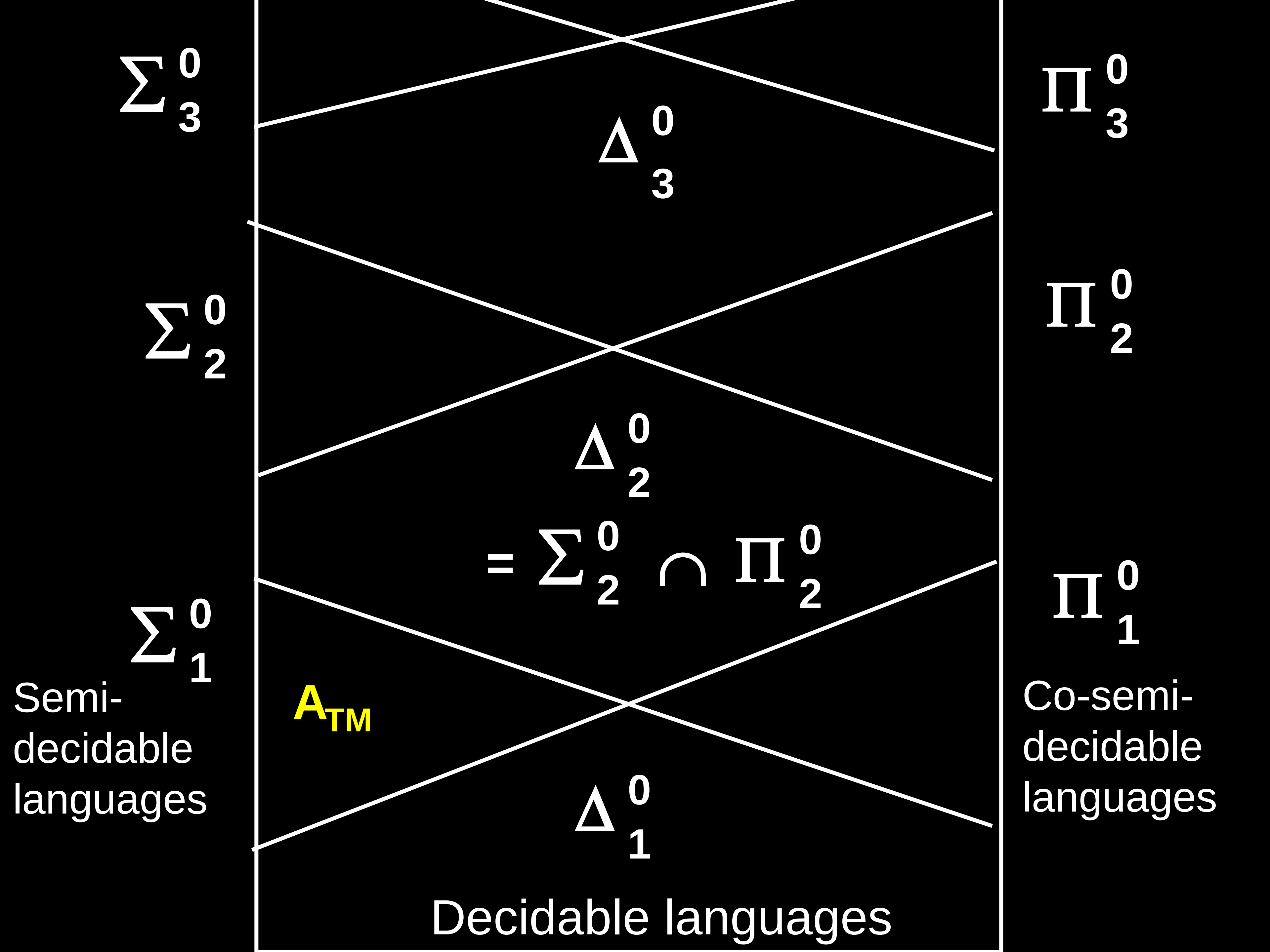
Show it can be described in this form:

A_{TM} = { <(M,w)> | ∃t [M accepts w in t steps] }

decidable predicate

A_{TM} = { <(M,w)> | ∃t T (<M>, w, t }

A_{TM} = { <(M,w)> | ∃v (v is an accepting computation history of M on w}



Σ^0_3
Π^0_3
Δ^0_3
Σ^0_2
Π^0_2
Δ^0_2
$= \Sigma^0_2 \cap \Pi^0_2$
Σ^0_1
Π^0_1
Semi-decidable languages
A_{TM}
Co-semi-decidable languages
Δ^0_1
Decidable languages

Π^0_1 = languages of the form { x | ∀y R(x,y) }

Show that EMPTY (ie, E_{TM}) = { M | L(M) = ∅ } is in Π^0_1

Π^0_1 = languages of the form { x | ∀y R(x,y) }

Show that EMPTY (ie, E_{TM}) = { M | L(M) = ∅ } is in Π^0_1

EMPTY = { M | ∀w∀t [M doesn't accept w in t steps] }

decidable predicate

Π^0_1 = languages of the form { x | ∀y R(x,y) }

Show that EMPTY (ie, E_{TM}) = { M | L(M) = ∅ } is in Π^0_1

EMPTY = { M | ∀w∀t [¬T(<M>, w, t)] }

decidable predicate

Π^0_1 = languages of the form $\{ x \mid \forall y\, R(x,y) \}$

Show that EMPTY (ie, E_{TM}) = $\{ M \mid L(M) = \varnothing \}$ is in Π^0_1

EMPTY = $\{ M \mid \forall w \forall t\, [\neg T(\langle M \rangle, w, t)] \}$

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 π_1 and $\pi_2 : \Sigma^* \rightarrow \Sigma^*$ such that

$$z = \langle w, t \rangle \Rightarrow \pi_1(z) = w \text{ and } \pi_2(z) = t$$

THE PAIRING FUNCTION

Theorem. There is a 1-1 and onto computable function $< , >: \Sigma^* \times \Sigma^* \rightarrow \Sigma^*$ and computable functions π_1 and $\pi_2 : \Sigma^* \rightarrow \Sigma^*$ such that

$$z = \langle w, t\rangle \Rightarrow \pi_1(z) = w \text{ and } \pi_2(z) = t$$

EMPTY = { M | ∀w∀t [M doesn't accept w in t steps] }

EMPTY = { M | ∀z[M doesn't accept π_1 (z) in π_2(z) steps]}

EMPTY = { M | ∀z[¬T(<M>, π_1 (z) , π_2(z))] }

THE PAIRING FUNCTION

Theorem. **There is a 1-1 and onto computable function $< , >: \Sigma^* \times \Sigma^* \rightarrow \Sigma^*$ and computable functions π_1 and $\pi_2 : \Sigma^* \rightarrow \Sigma^*$ such that**

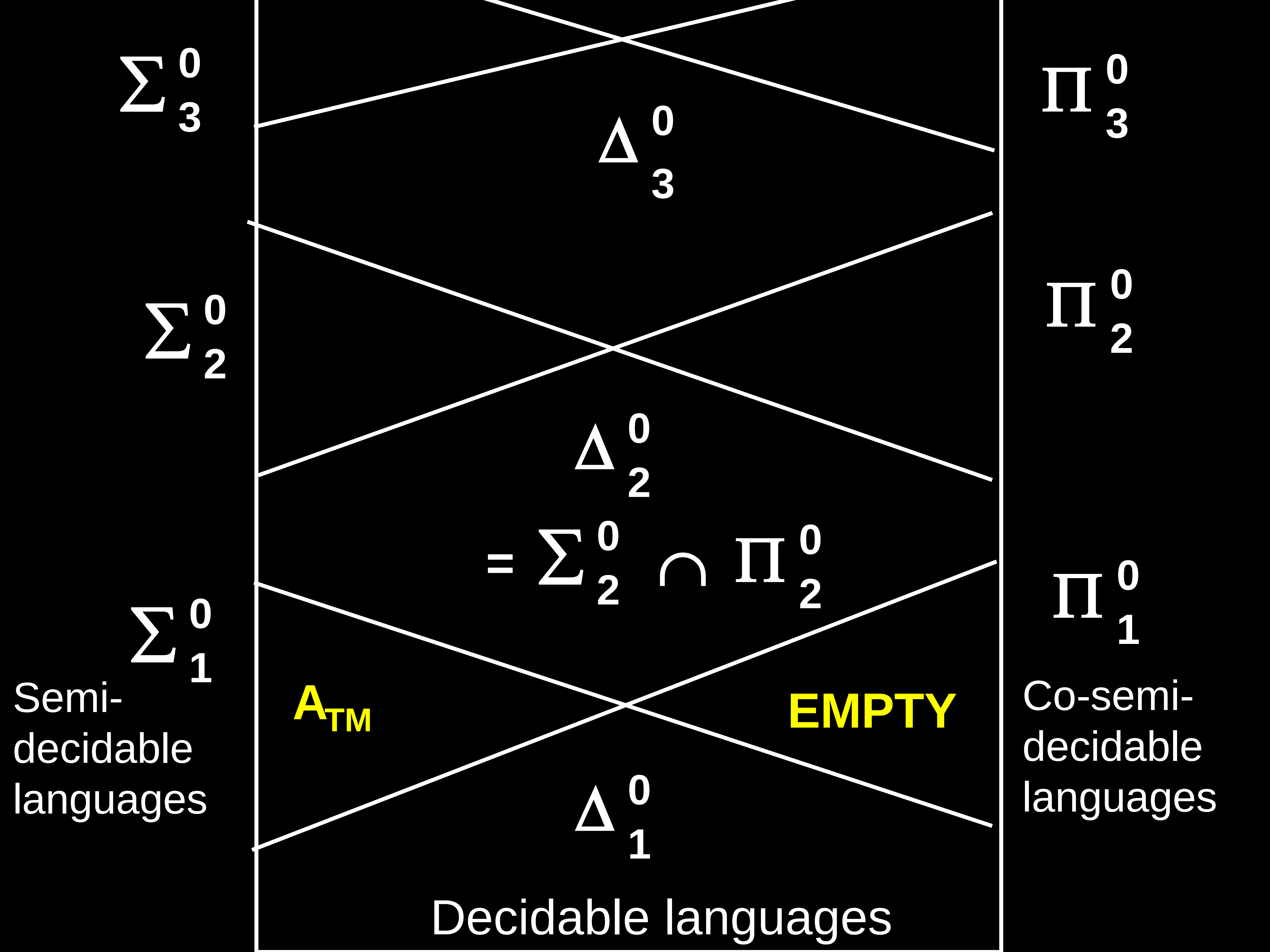
$$z = \langle w, t \rangle \Rightarrow \pi_1(z) = w \text{ and } \pi_2(z) = t$$

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

$$\langle w, t \rangle := a\, w_1 \ldots a\, w_n\, b\, t$$

$\pi_1(z)$:= "if z has the form a w_1… a w_n b t, then output w_1… w_n, else output ε"

$\pi_2(z)$:= "if z has the form a w_1… a w_n b t, then output t, else output ε"


Σ_3^0
Δ_3^0
Π_3^0
Σ_2^0
Π_2^0
Δ_2^0
$= \Sigma_2^0 \cap \Pi_2^0$
Σ_1^0
Π_1^0
Semi-decidable languages
A_{TM}
EMPTY
Co-semi-decidable languages
Δ_1^0
Decidable languages

Π^0_2 = languages of the form { x | ∀y∃z R(x,y,z) }

Show that TOTAL = { M | M halts on all inputs }

is in Π^0_2

Π^0_2 = languages of the form { x | ∀y∃z R(x,y,z) }

Show that TOTAL = { M | M halts on all inputs } is in Π^0_2

TOTAL = { M | ∀w ∃t [M halts on w in t steps] }

decidable predicate

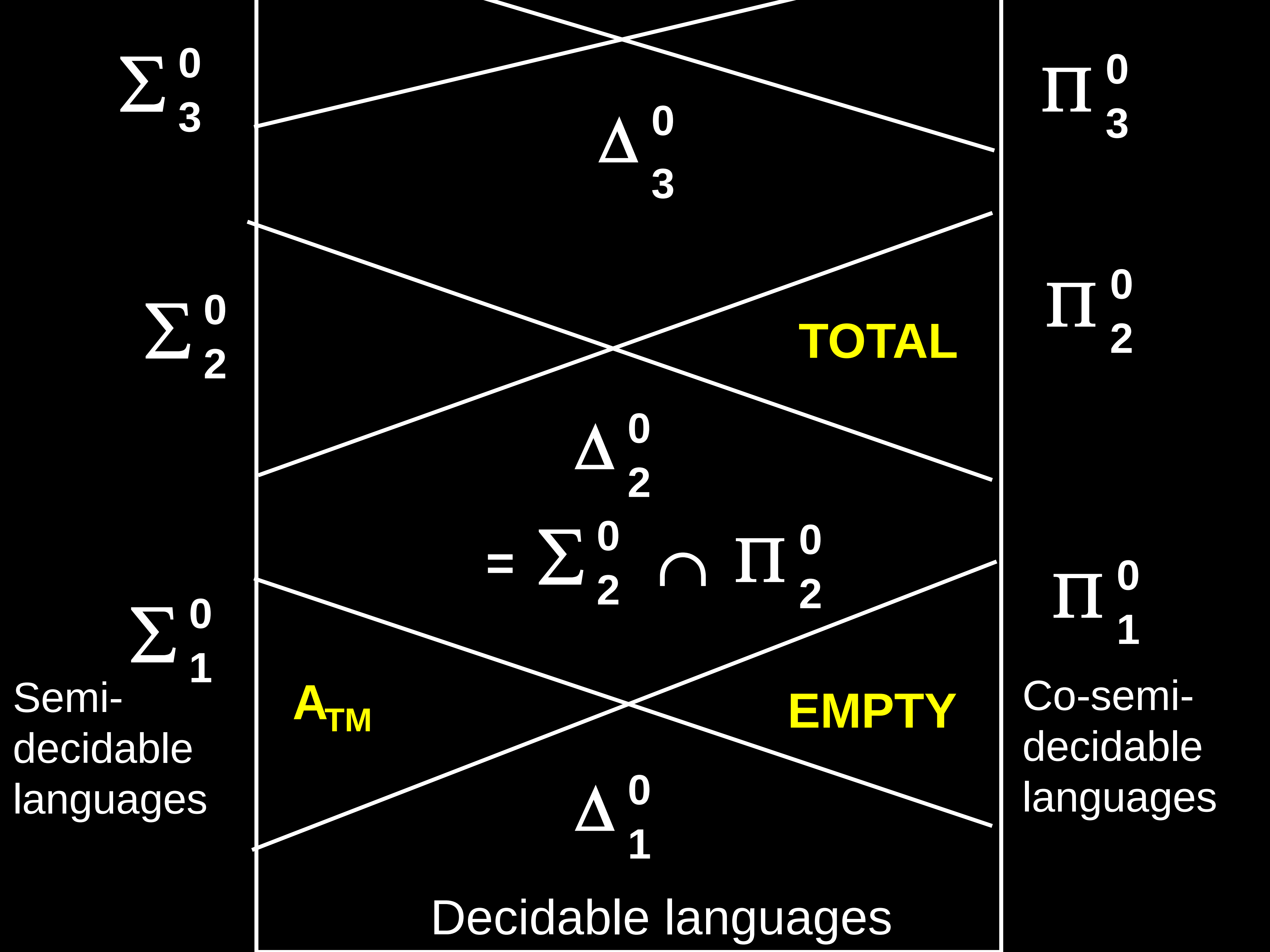
Π^0_2 = languages of the form { x | ∀y∃z R(x,y,z) }

Show that TOTAL = { M | M halts on all inputs }

is in Π^0_2

TOTAL = { M | ∀w ∃t [T(<M>, w, t)] }

decidable predicate


Σ^0_3
Δ^0_3
Π^0_3
Σ^0_2
TOTAL
Π^0_2
Δ^0_2
$= \Sigma^0_2 \cap \Pi^0_2$
Σ^0_1
Π^0_1
Semi-decidable languages
A_{TM}
EMPTY
Co-semi-decidable languages
Δ^0_1
Decidable languages

Σ_2^0 = languages of the form { x | ∃y∀z R(x,y,z) }

Show that FIN = { M | L(M) is finite } is in Σ_2^0

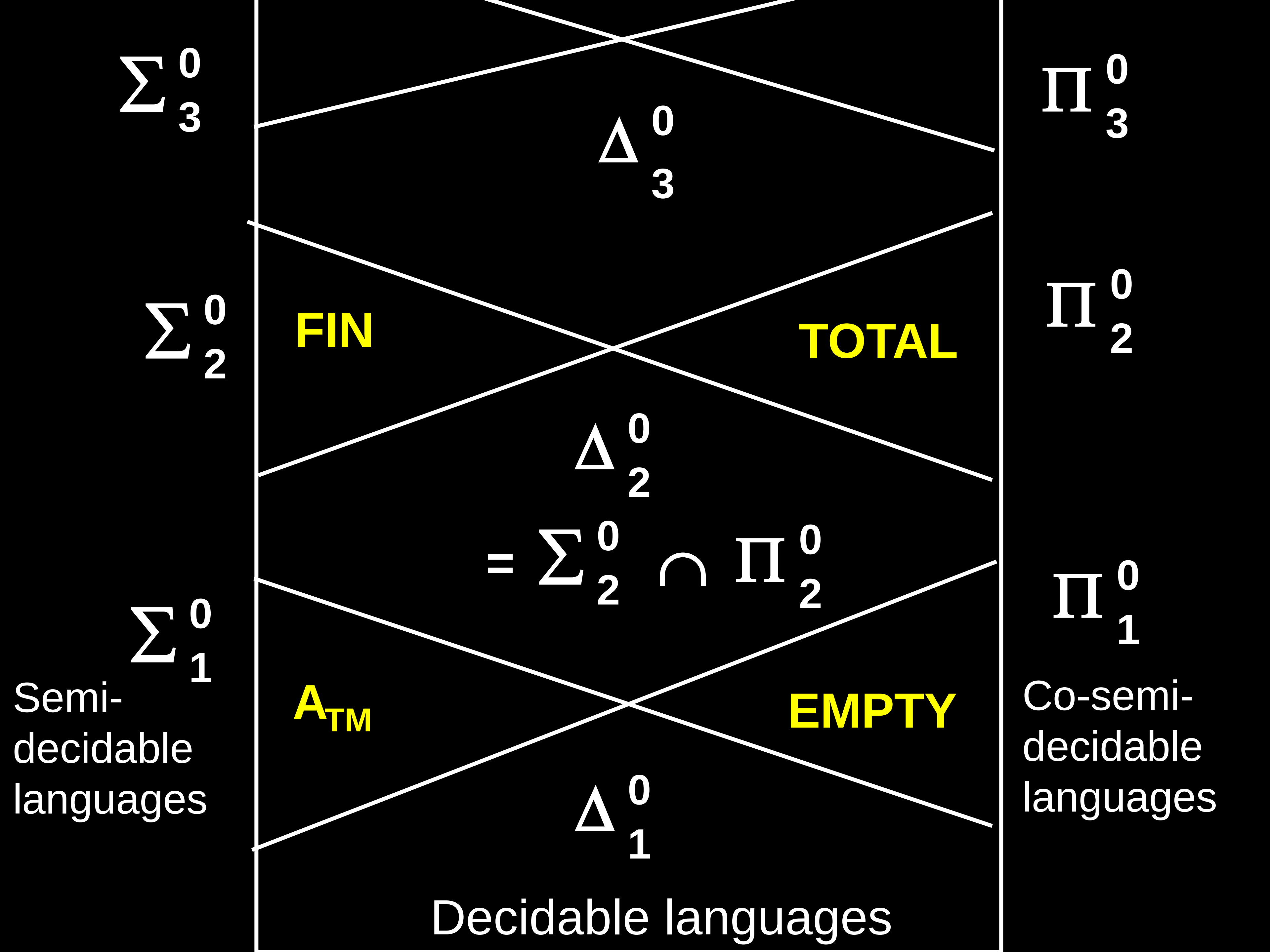
Σ_2^0 = languages of the form { x | ∃y∀z R(x,y,z) }

Show that FIN = { M | L(M) is finite } is in Σ_2^0

FIN = { M | ∃n∀w∀t [Either |w| < n, or
M doesn't accept w in t steps] }

FIN = { M | ∃n∀w∀t (|w| < n ∨ ¬ T(<M>,w, t))}

decidable predicate


Σ^0_3
Π^0_3
Δ^0_3
Σ^0_2
FIN
TOTAL
Π^0_2
Δ^0_2
$= \Sigma^0_2 \cap \Pi^0_2$
Σ^0_1
Π^0_1
Semi-decidable languages
A_{TM}
EMPTY
Co-semi-decidable languages
Δ^0_1
Decidable languages

Σ_3^0 = languages of the form { x | ∃y∀z∃u R(x,y,z,u) }

Show that COF = { M | L(M) is cofinite } is in Σ_2^0

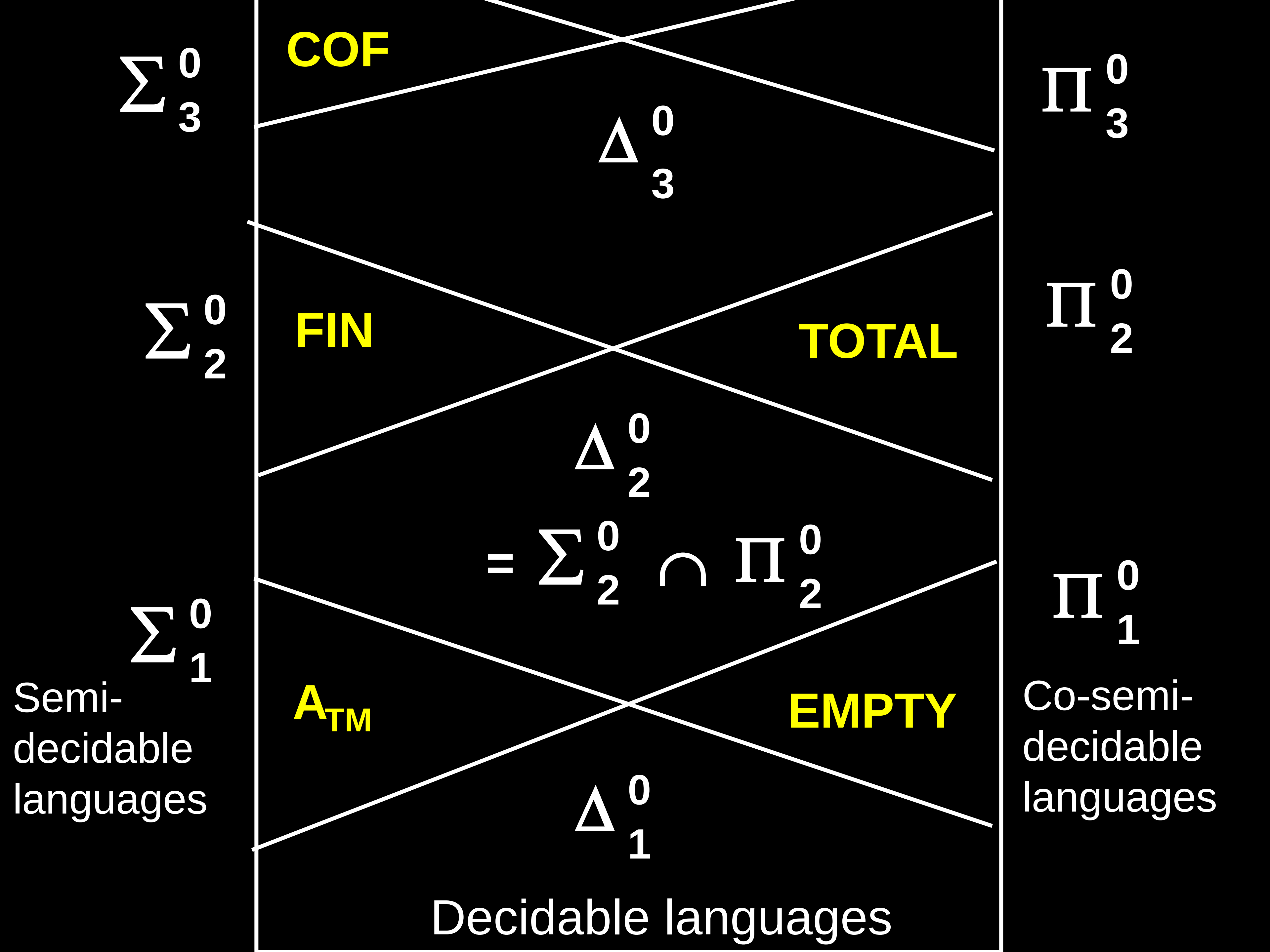
Σ_3^0 = languages of the form { x | ∃y∀z∃u R(x,y,z,u) }

Show that COF = { M | L(M) is cofinite } is in Σ_2^0

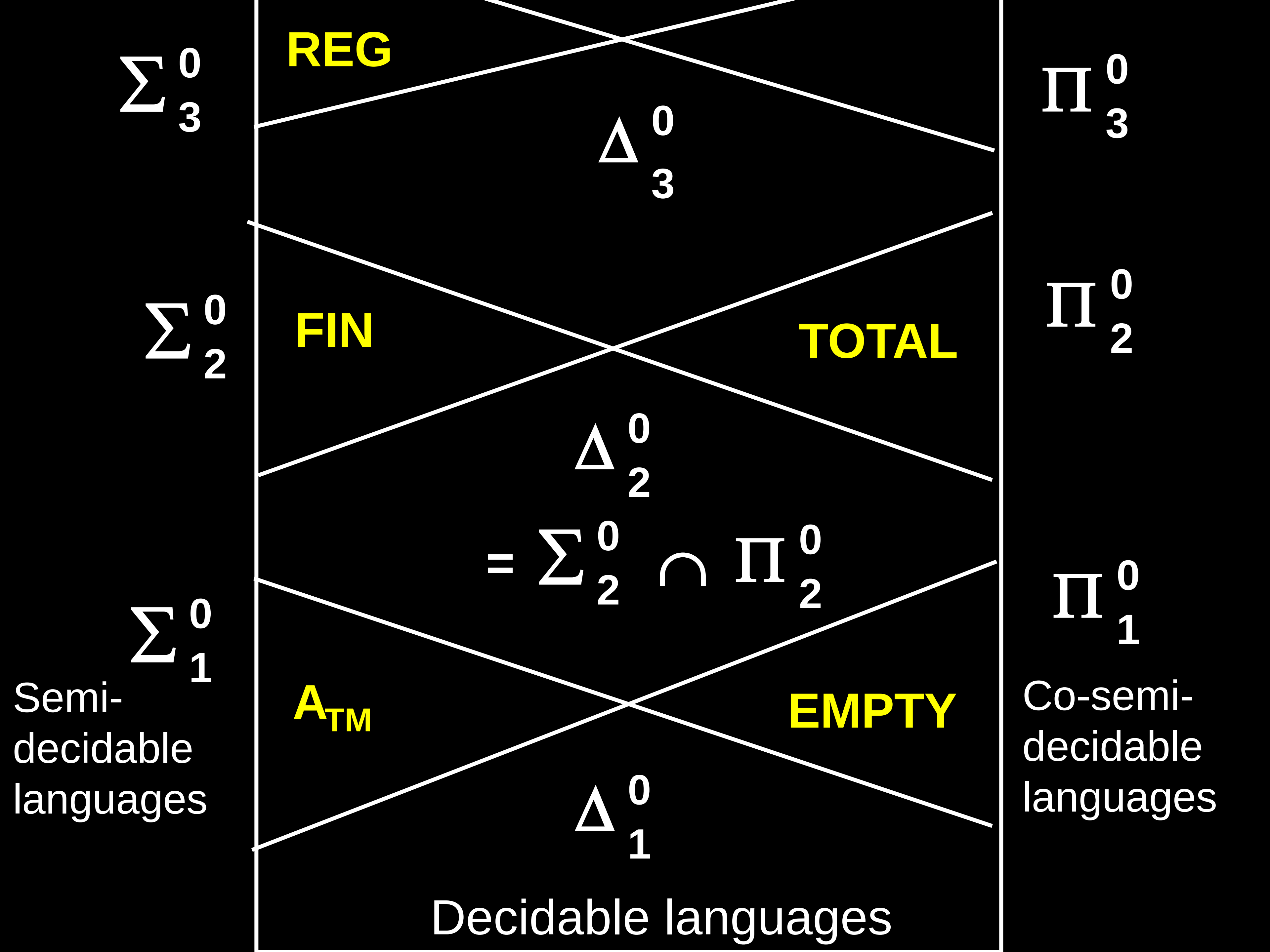
COF = { M | ∃n∀w∃ t [|w| > n ⇒ M accept w in t steps] }

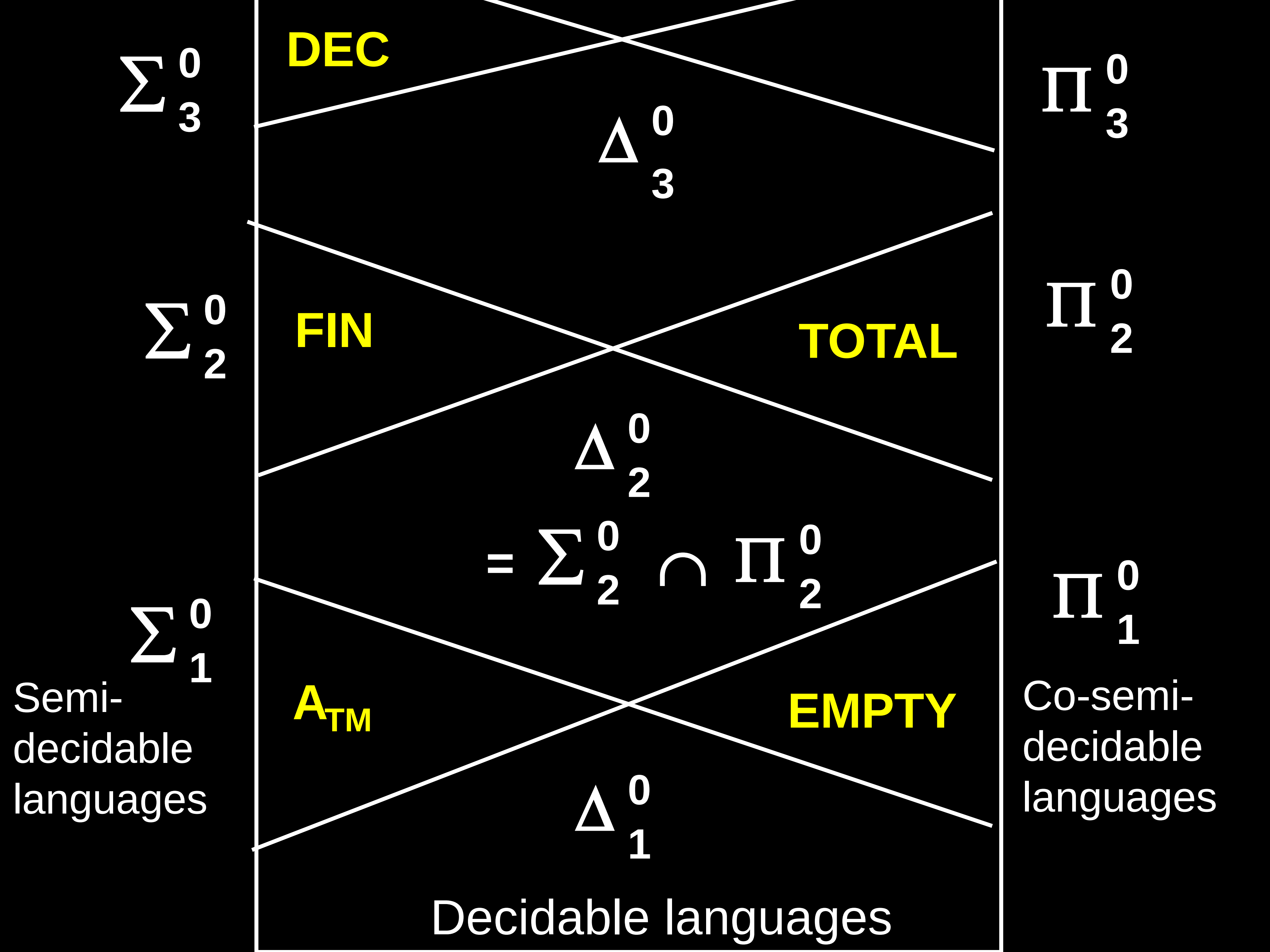
COF = { M | ∃n∀w∃ t (|w| ≤ n ∨T(<M>,w, t))}

decidable predicate

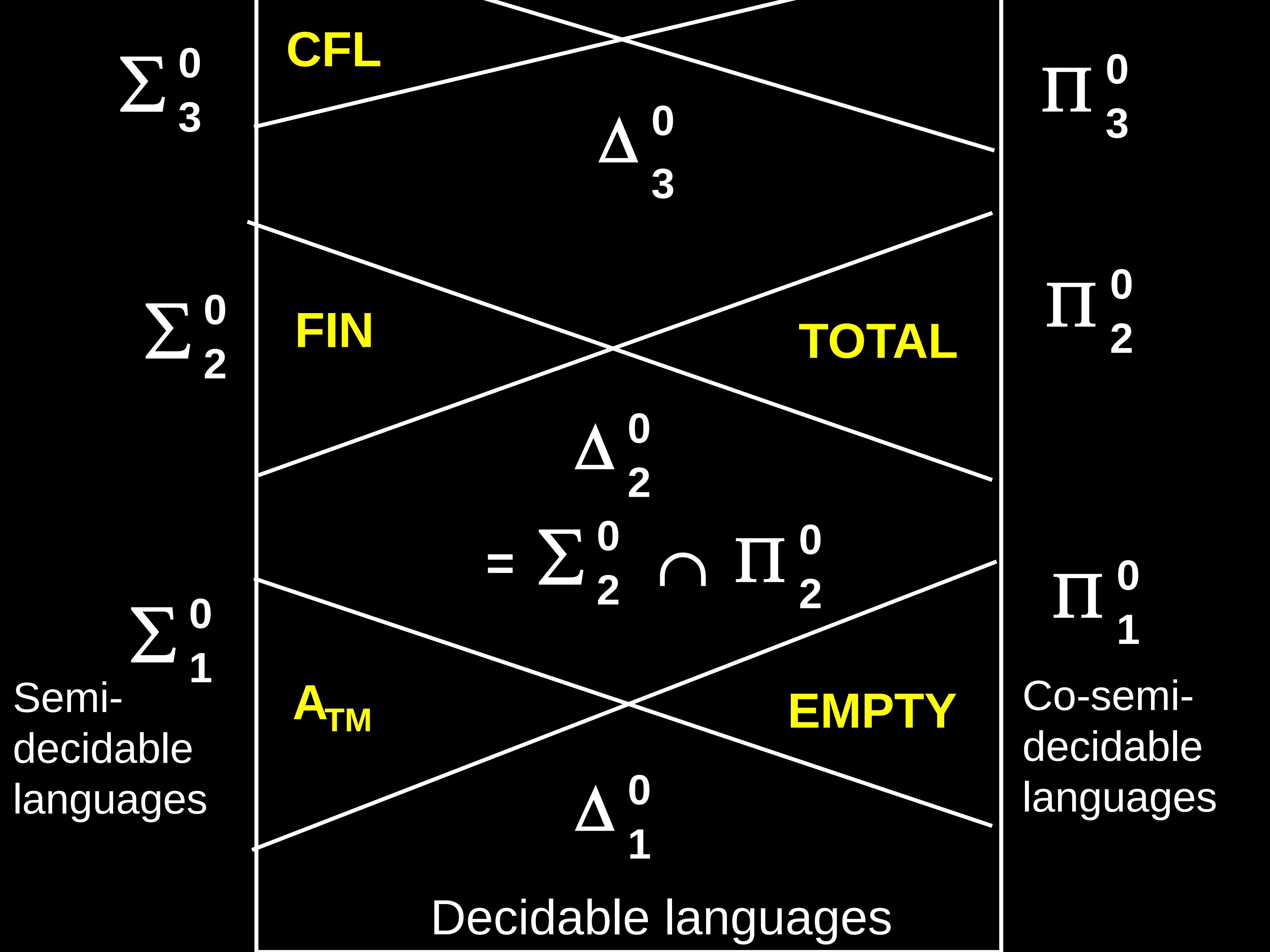


Σ_3^0
COF
Π_3^0
Δ_3^0
Σ_2^0
FIN
TOTAL
Π_2^0
Δ_2^0
$= \Sigma_2^0 \cap \Pi_2^0$
Σ_1^0
Π_1^0
Semi-decidable languages
A_{TM}
EMPTY
Co-semi-decidable languages
Δ_1^0
Decidable languages


REG
Σ_3^0
Δ_3^0
Π_3^0
Σ_2^0
FIN
TOTAL
Π_2^0
Δ_2^0
$= \Sigma_2^0 \cap \Pi_2^0$
Σ_1^0
Π_1^0
Semi-decidable languages
A_{TM}
EMPTY
Co-semi-decidable languages
Δ_1^0
Decidable languages



Σ^0_3
DEC
Π^0_3
Δ^0_3
Σ^0_2
FIN
TOTAL
Π^0_2
Δ^0_2
$= \Sigma^0_2 \cap \Pi^0_2$
Σ^0_1
Π^0_1
Semi-decidable languages
A_{TM}
EMPTY
Co-semi-decidable languages
Δ^0_1
Decidable languages



Σ_3^0
CFL
Δ_3^0
Π_3^0
Σ_2^0
FIN
TOTAL
Π_2^0
Δ_2^0
$= \Sigma_2^0 \cap \Pi_2^0$
Σ_1^0
Π_1^0
Semi-decidable languages
A_{TM}
EMPTY
Co-semi-decidable languages
Δ_1^0
Decidable languages

Each is m-complete for its level in hierachy and cannot go lower (by the SuperHalting Theorem, which shows the hierarchy does not collapse).

Each is m-complete for its level in hierachy and cannot go lower (by the SuperHalting Theorem, which shows the hierarchy does not collapse).

L is m-complete for class C if

i) $L \in C$ and

ii) L is m-hard for C,

ie, for all $L' \in C$, $L' \leq_m L$

A_{TM} is m-complete for class $C = \Sigma^0_1$

i) $A_{TM} \in C$

ii) A_{TM} is m-hard for C,

A_{TM} is m-complete for class $C = \Sigma^0_1$

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

$\Sigma^* \rightarrow \Sigma^*$

where $w \rightarrow (M, w)$.

Then, $w \in L \Leftrightarrow (M,w) \in A_{TM}$ 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

Supose $L \in \Sigma^0_2$ ie $L = \{ w \mid \exists y \forall z \; R(w,y,z) \}$ where R is decided by some TM D

Show: $L \le_m FIN$

Supose $L \in \Sigma_2^0$ ie $L = \{ w \mid \exists y \forall z\ R(w,y,z) \}$
where R is decided by some TM D

Show: $L \leq_m FIN$

Map $\Sigma^* \rightarrow \Sigma^*$
where $w \rightarrow N_{D,w}$

Supose $L \in \Sigma_2^0$ ie $L = \{ w \mid \exists y \forall z\ R(w,y,z) \}$
where R is decided by some TM D

Show: $L \leq_m FIN$

Map $\Sigma^* \rightarrow \Sigma^*$
where $w \rightarrow N_{D,w}$

Define $N_{D,w}$ On input s:

1. Write down all strings y of length |s|
2. For each y, try to find a z such that
$\neg R(w, y, z)$ and accept if all are successful
(here use D and w)

So, $w \in L \Leftrightarrow N_{D,w} \in FIN$

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, with an oracle for the Halting Problem, halts on x}

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, with an oracle for the Halting Problem, halts on x}

Can use diagonalization here!

Suppose H decides SUPERHALT (with oracle)

Define **D(X) = "if H(X,X) accepts (with oracle) then LOOP, else ACCEPT."**

D(D) halts ⇔ H(D,D) accepts ⇔ 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.

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 halts on x}.

$SUPERHALT^1$ = { (M,x) | M, with an oracle for the Halting Problem, halts on x}

$SUPERHALT^n$ = { (M,x) | M, with an oracle for $SUPERHALT^{n-1}$, halts on x}

WWW.FLAC.WS

Read Chapter 6.4 for next time