Paper Discussion; DDE (Doctrine of Double Effect) => DDE* (for self-sacrifice) => DTripleE

Selmer Bringsjord

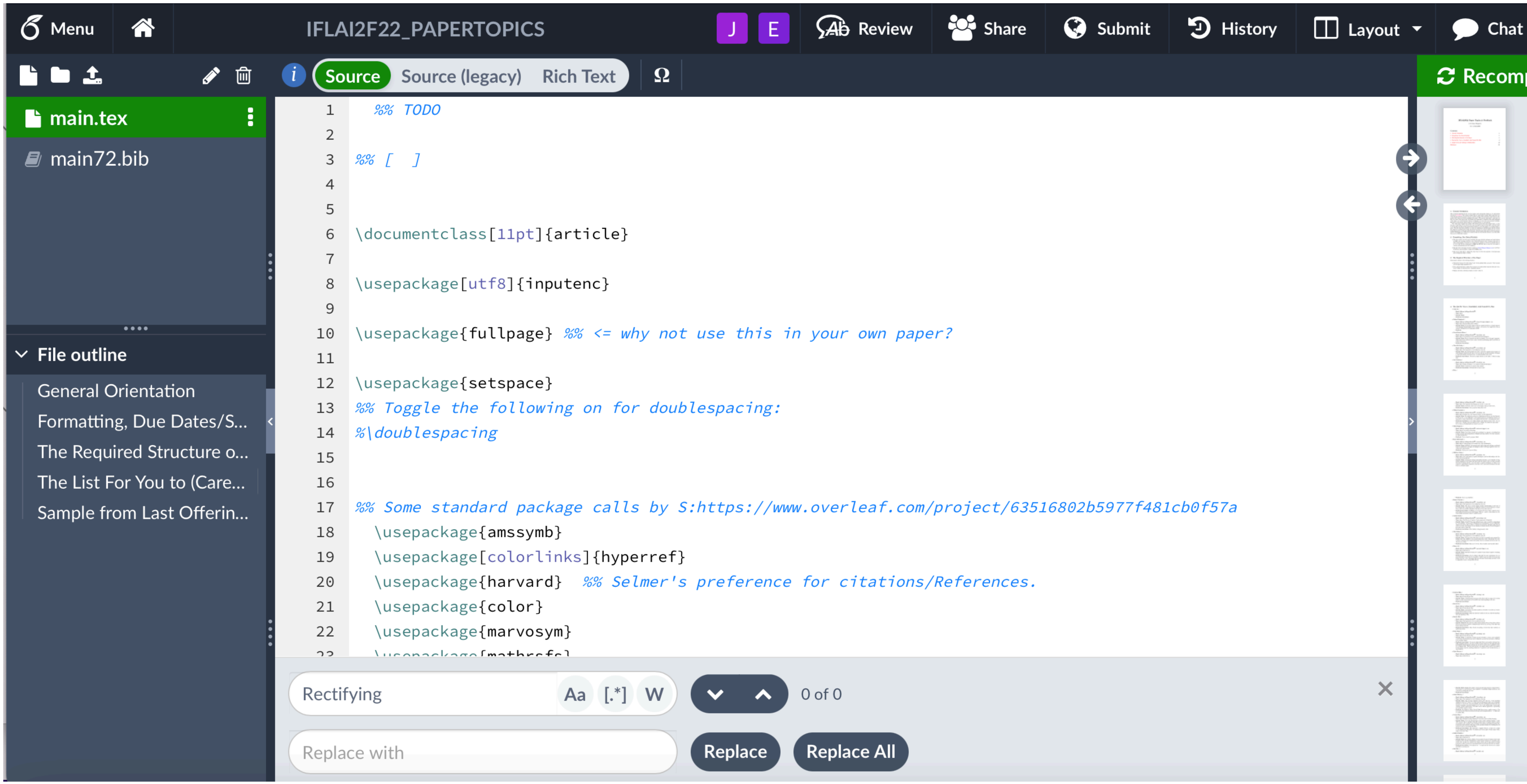
Rensselaer AI & Reasoning (RAIR) Lab
Department of Cognitive Science
Department of Computer Science
Lally School of Management & Technology
Rensselaer Polytechnic Institute (RPI)
Troy, New York 12180 USA

IFLAI2
11/17/2022
ver 1117221237NY

Rensselaer AI and Reasoning Lab

Logistics …

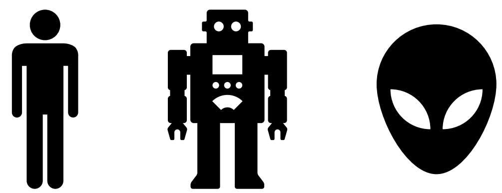
Let's look real-time now, online. Btw, remember, first draft is due *after* Tgiving break.

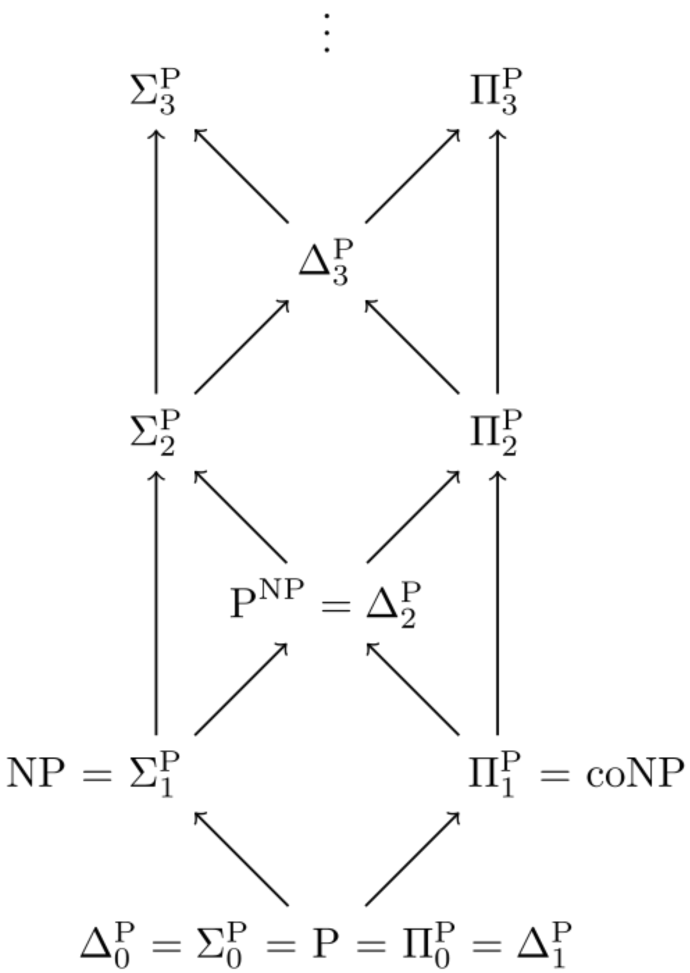


Further delivery on promissory note re building hierarchies via formal logic…

Polynomial Hierarchy, Part II

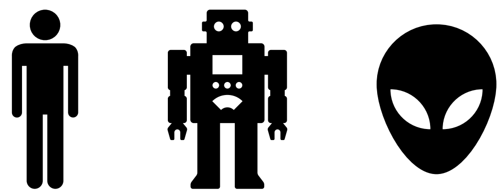
(via formal logic, directly)

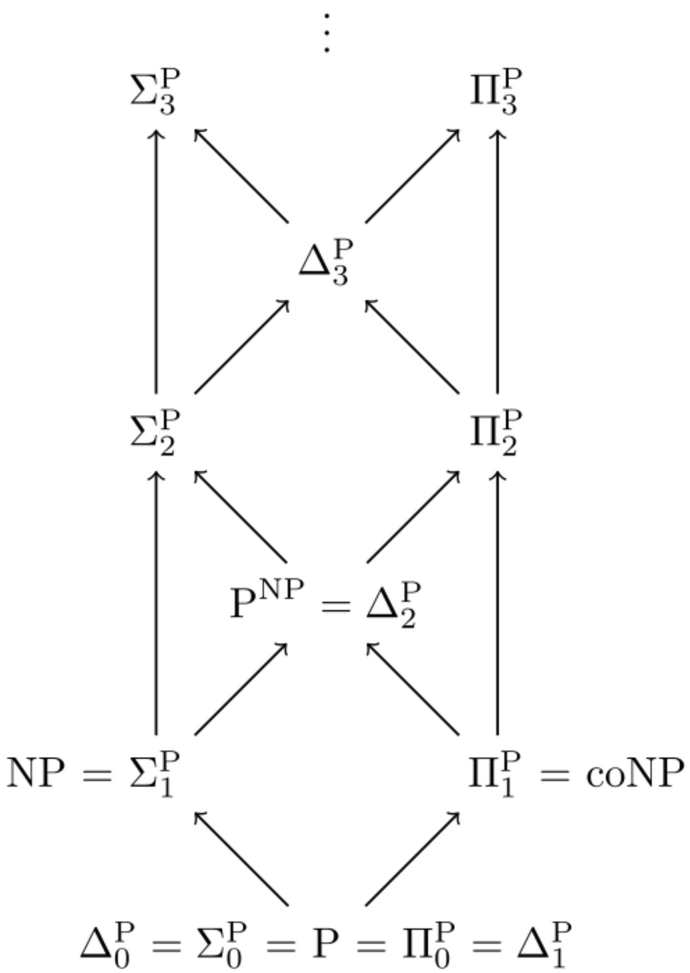




Polynomial Hierarchy, Part II

(via formal logic, directly)

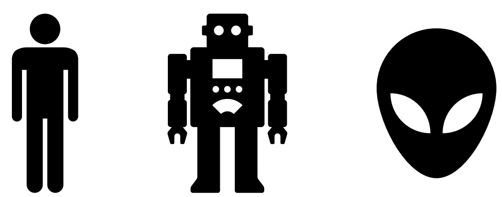


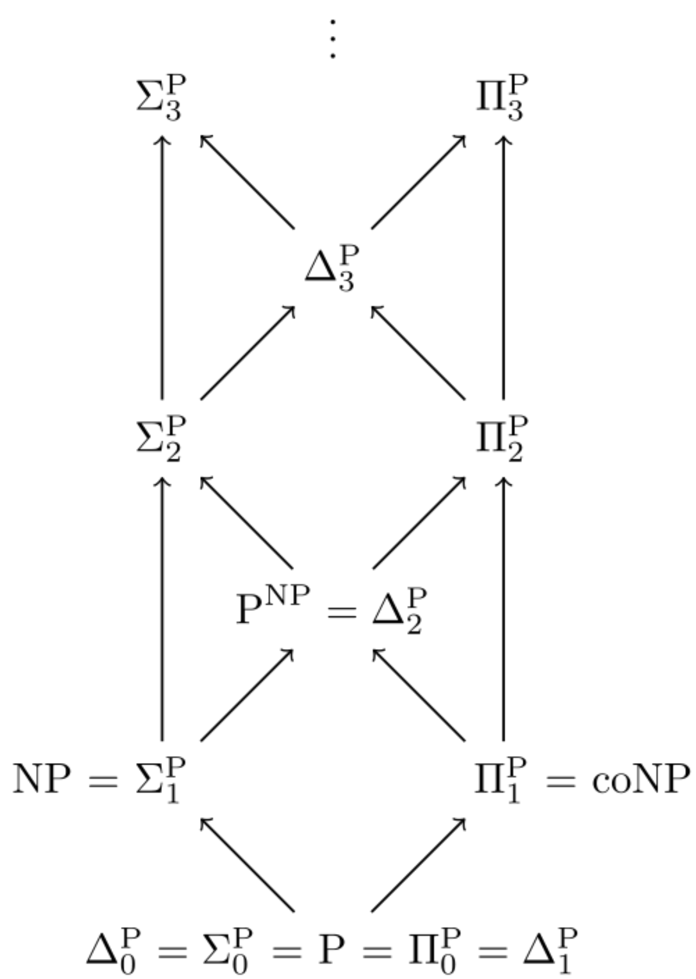


Eg:

Polynomial Hierarchy, Part II

(via formal logic, directly)



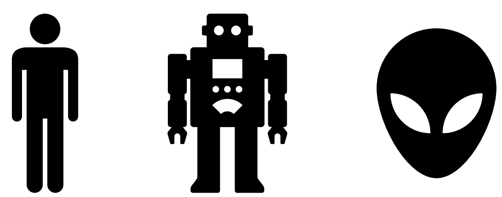


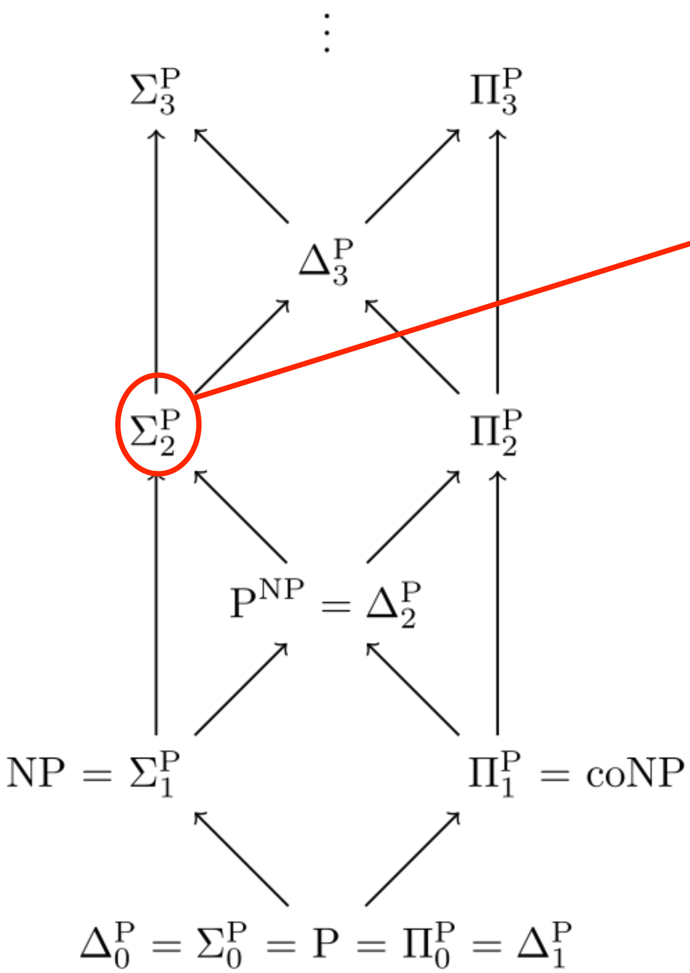
Eg:

$$\langle \phi_1, k \rangle \in L \text{ iff } \exists \phi_2 \forall \alpha KLogEquiv(\phi_1, \phi_2, |\phi_2| \leq k, \alpha(\phi_1) = \alpha(\phi_2))$$

Polynomial Hierarchy, Part II

(via formal logic, directly)



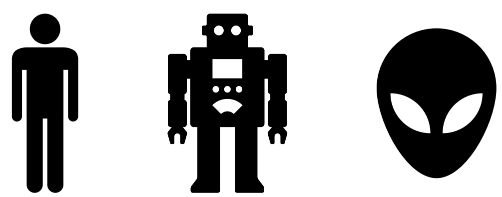


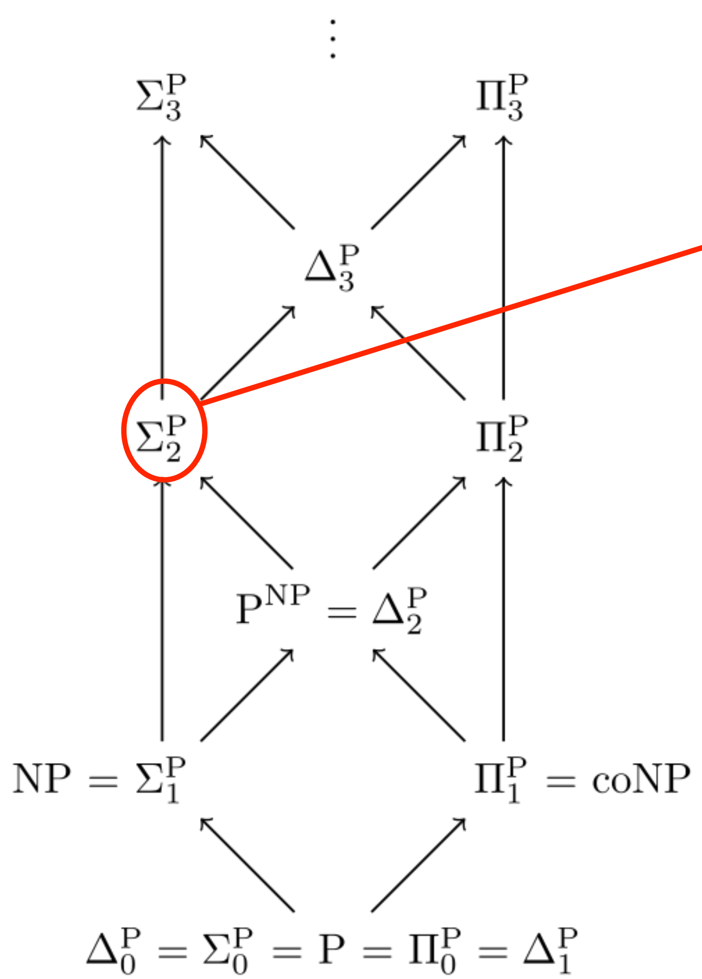
Eg:

$$\langle \phi_1, k \rangle \in L \text{ iff } \exists \phi_2 \forall \alpha KLogEquiv(\phi_1, \phi_2, |\phi_2| \leq k, \alpha(\phi_1) = \alpha(\phi_2))$$

Polynomial Hierarchy, Part II

(via formal logic, directly)





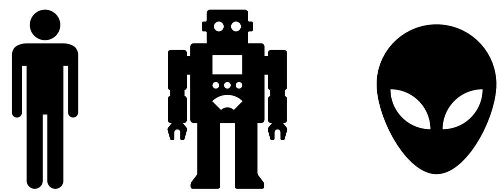
Eg:

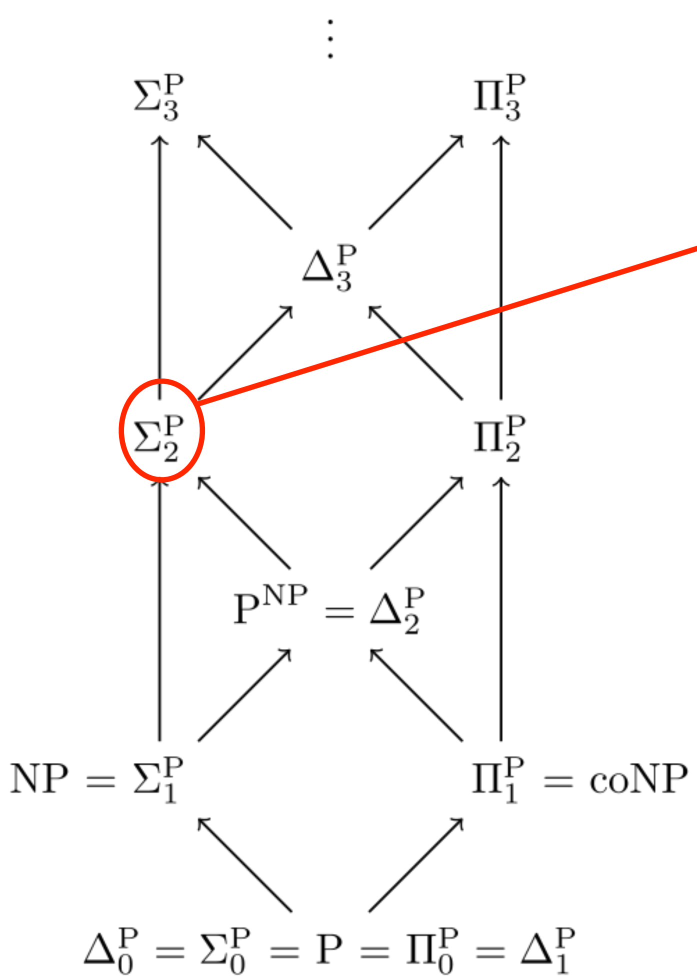
free variables

$$\langle \phi_1, k \rangle \in L \text{ iff } \exists \phi_2 \forall \alpha \, KLogEquiv(\phi_1, \phi_2, |\phi_2| \leq k, \alpha(\phi_1) = \alpha(\phi_2))$$

Polynomial Hierarchy, Part II

(via formal logic, directly)





free variables

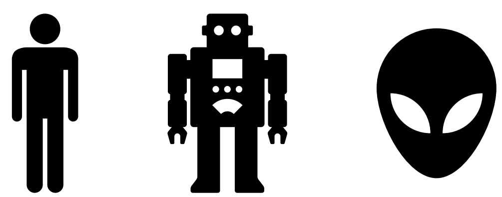
Eg:

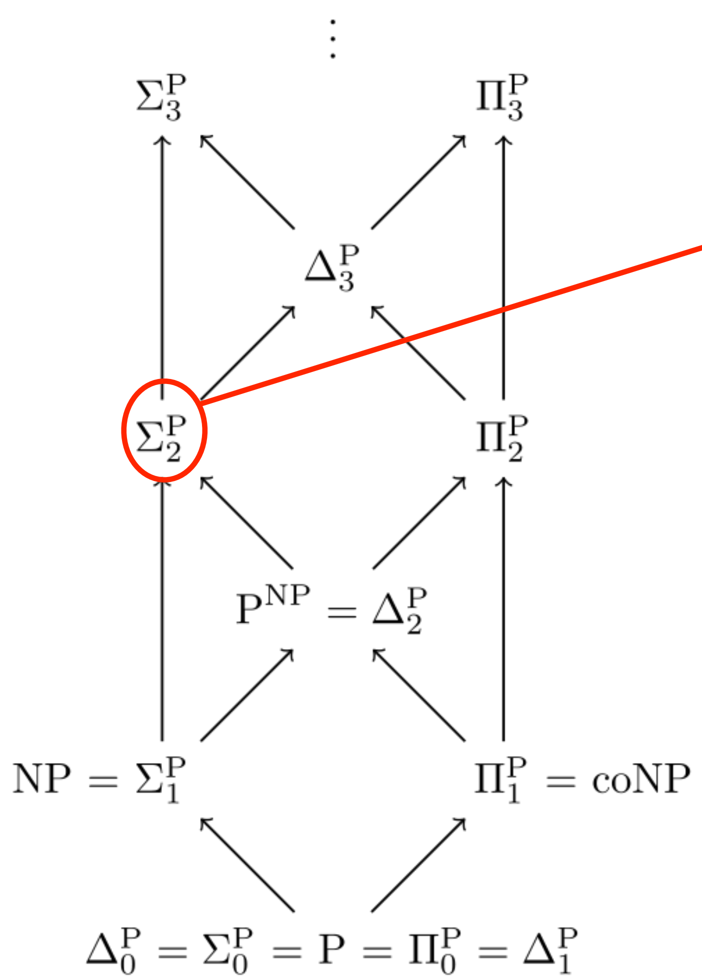
$$\langle \phi_1, k \rangle \in L \text{ iff } \exists \phi_2 \forall \alpha KLogEquiv(\phi_1, \phi_2, |\phi_2| \leq k, \alpha(\phi_1) = \alpha(\phi_2))$$

Now we generalize:

Polynomial Hierarchy, Part II

(via formal logic, directly)





free variables

Eg:

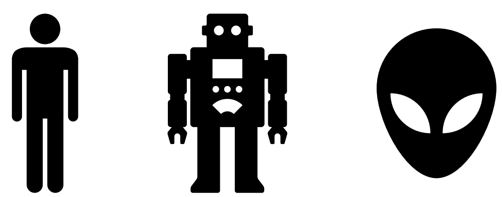
$$\langle \phi_1, k \rangle \in L \text{ iff } \exists \phi_2 \forall \alpha\, KLogEquiv(\phi_1, \phi_2, |\phi_2| \leq k, \alpha(\phi_1) = \alpha(\phi_2))$$

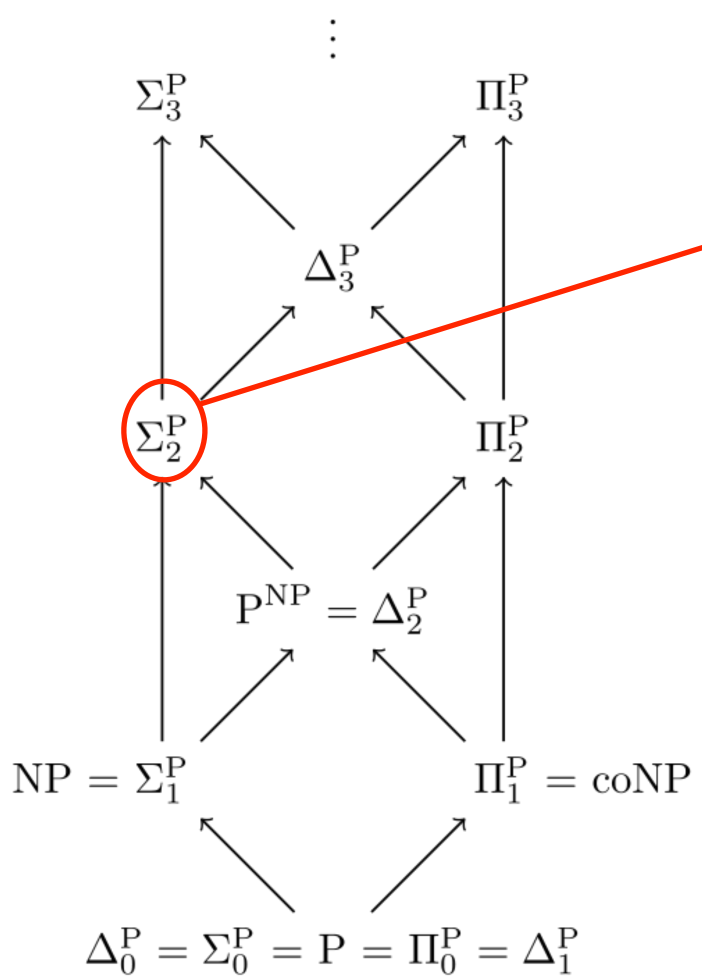
Now we generalize:

$$x \in \Sigma_i \text{ iff } \exists R\ \exists y_1 \forall y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \forall \text{ if } i \text{ even}; Q_i = \exists \text{ if } i \text{ odd})$$

Polynomial Hierarchy, Part II

(via formal logic, directly)





free variables

Eg:

$$\langle \phi_1, k \rangle \in L \text{ iff } \exists \phi_2 \forall \alpha KLogEquiv(\phi_1, \phi_2, |\phi_2| \leq k, \alpha(\phi_1) = \alpha(\phi_2))$$

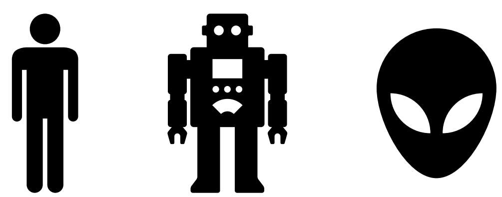
Now we generalize:

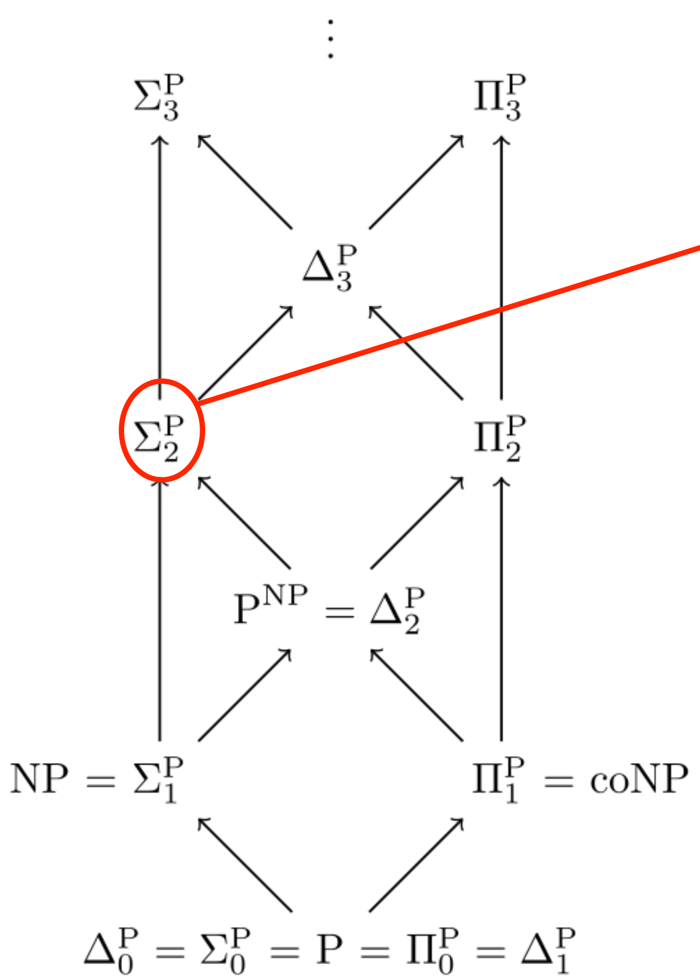
$$x \in \Sigma_i \text{ iff } \exists R \; \exists y_1 \forall y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \forall \text{ if } i \text{ even}; Q_i = \exists \text{ if } i \text{ odd})$$

$$x \in \Pi_i \text{ iff } \exists R \; \forall y_1 \exists y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \exists \text{ if } j \text{ even}; Q_i = \forall \text{ if } j \text{ odd})$$

Polynomial Hierarchy, Part II

(via formal logic, directly)





√ free variables

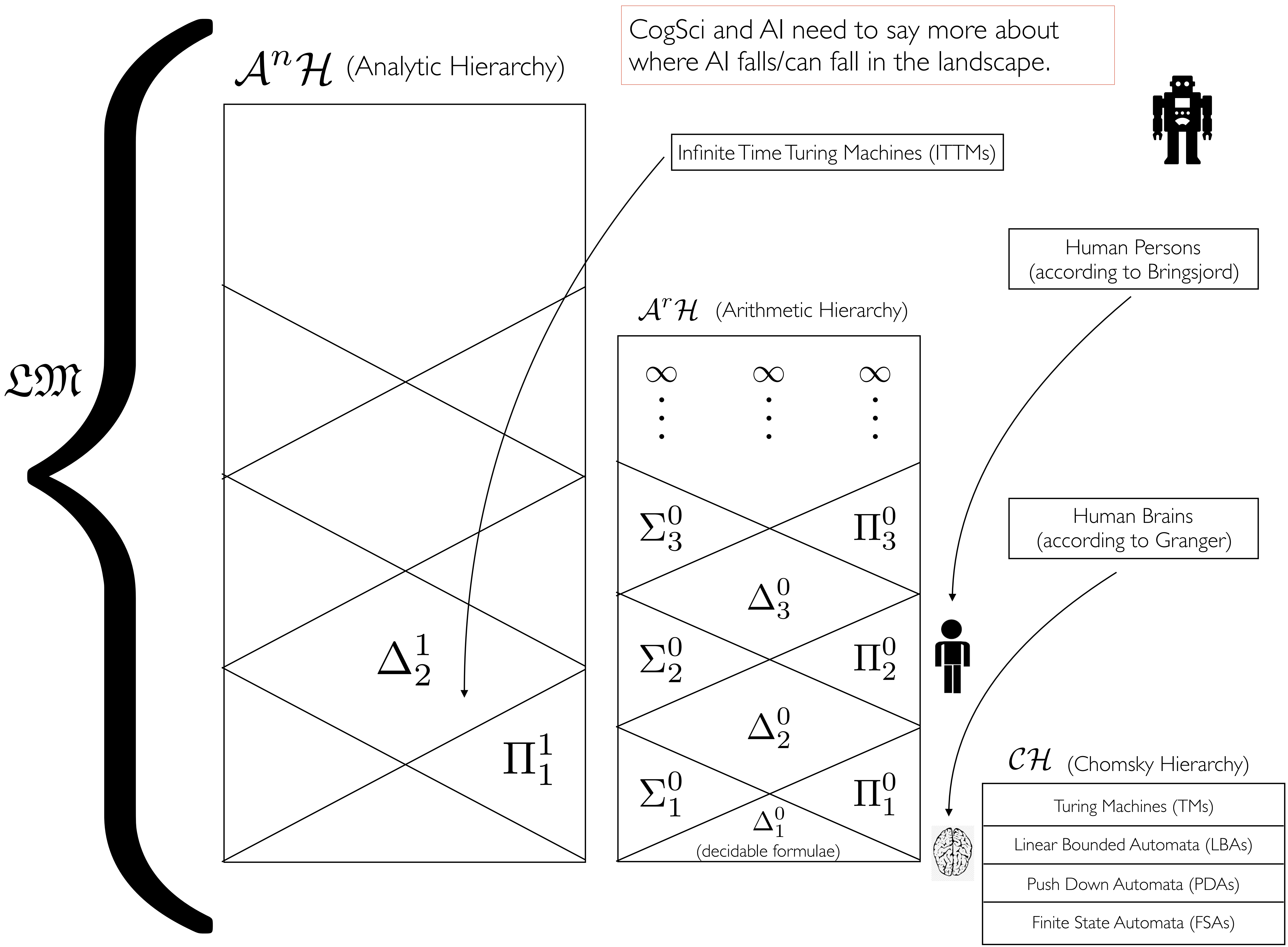
Eg:

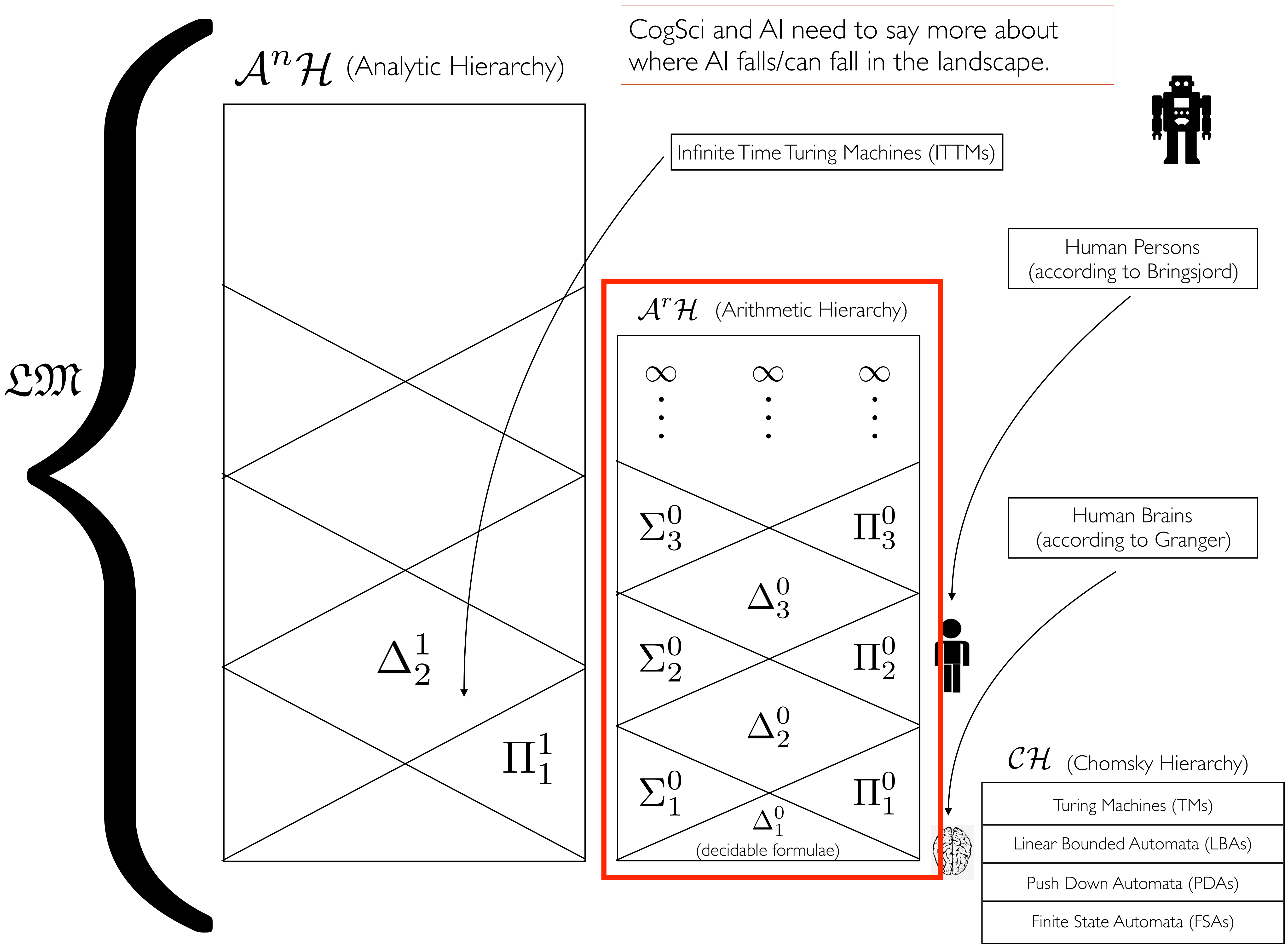
$$\langle \phi_1, k \rangle \in L \text{ iff } \exists \phi_2 \forall \alpha\, KLogEquiv(\phi_1, \phi_2, |\phi_2| \leq k, \alpha(\phi_1) = \alpha(\phi_2))$$

Now we generalize:

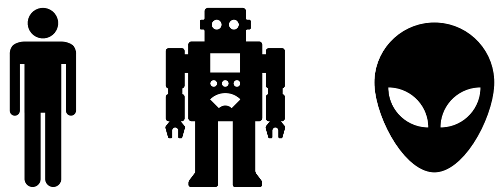
$$x \in \Sigma_i \text{ iff } \exists R\ \exists y_1 \forall y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \forall \text{ if } i \text{ even}; Q_i = \exists \text{ if } i \text{ odd})$$

$$x \in \Pi_i \text{ iff } \exists R\ \forall y_1 \exists y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \exists \text{ if } j \text{ even}; Q_i = \forall \text{ if } j \text{ odd})$$


CogSci and AI need to say more about where AI falls/can fall in the landscape.
$\mathfrak{LM}$
$\mathcal{A}^n\mathcal{H}$ (Analytic Hierarchy)
Infinite Time Turing Machines (ITTMs)
Δ^1_2
Π^1_1
$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)
∞ ∞ ∞
Σ^0_3 Π^0_3
Δ^0_3
Σ^0_2 Π^0_2
Δ^0_2
Σ^0_1 Π^0_1
Δ^0_1
(decidable formulae)
Human Persons
(according to Bringsjord)
Human Brains
(according to Granger)
$\mathcal{CH}$ (Chomsky Hierarchy)
Turing Machines (TMs)
Linear Bounded Automata (LBAs)
Push Down Automata (PDAs)
Finite State Automata (FSAs)

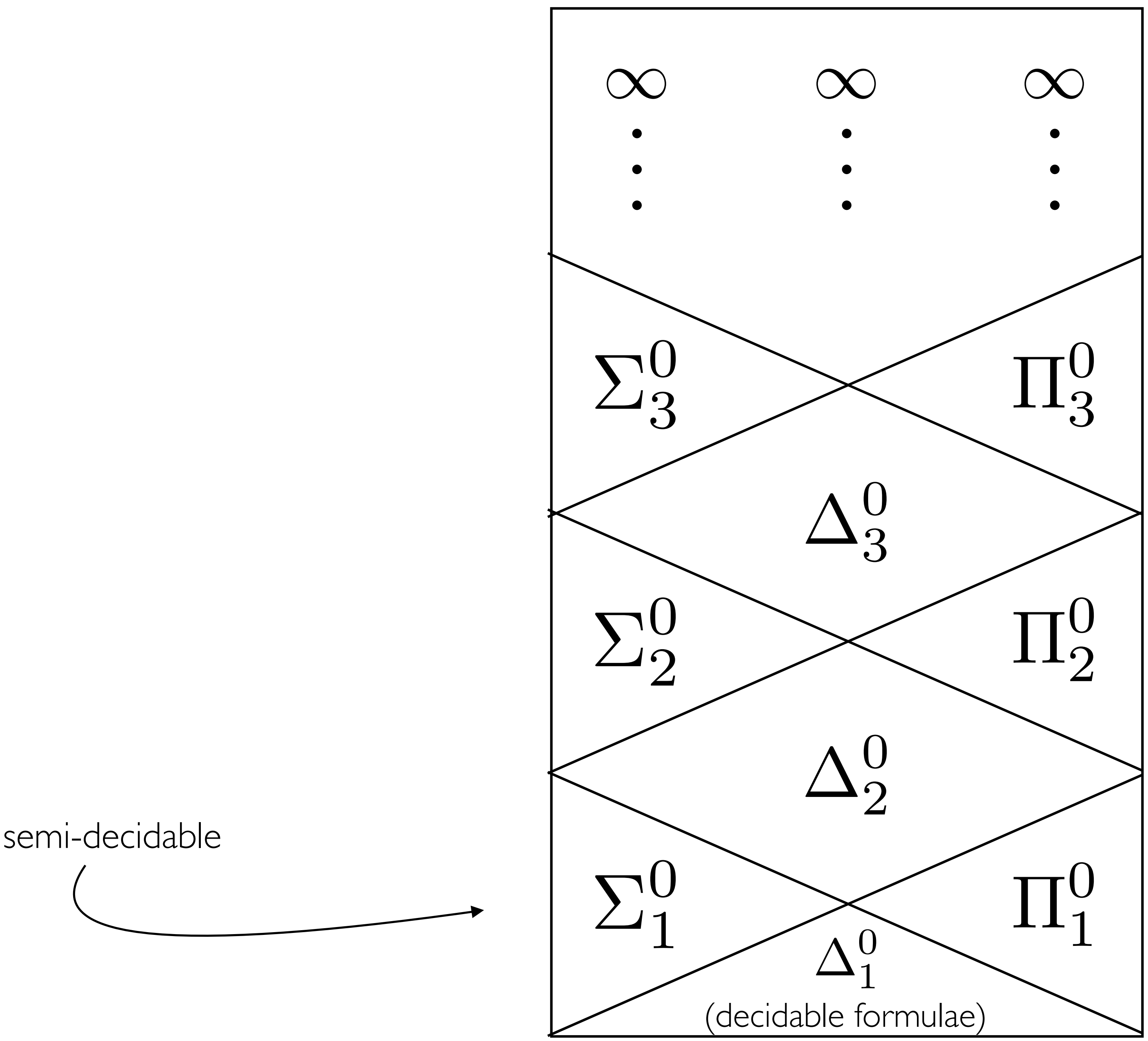


CogSci and AI need to say more about where AI falls/can fall in the landscape.
$\mathfrak{LM}$
$\mathcal{A}^n\mathcal{H}$ (Analytic Hierarchy)
Infinite Time Turing Machines (ITTMs)
Δ^1_2
Π^1_1
$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)
∞
∞
∞
Σ^0_3
Π^0_3
Δ^0_3
Σ^0_2
Π^0_2
Δ^0_2
Σ^0_1
Π^0_1
Δ^0_1
(decidable formulae)
Human Persons (according to Bringsjord)
Human Brains (according to Granger)
$\mathcal{CH}$ (Chomsky Hierarchy)
Turing Machines (TMs)
Linear Bounded Automata (LBAs)
Push Down Automata (PDAs)
Finite State Automata (FSAs)



$$\mathbf{2SAMEFUNC} := \{\mathfrak{m}_1, \mathfrak{m}_2 : \forall u \forall v [\exists k(\langle \mathfrak{m}_1, u\rangle : v, k \leftrightarrow \exists k'(\langle \mathfrak{m}_2, u\rangle : v, k')]$$

$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)



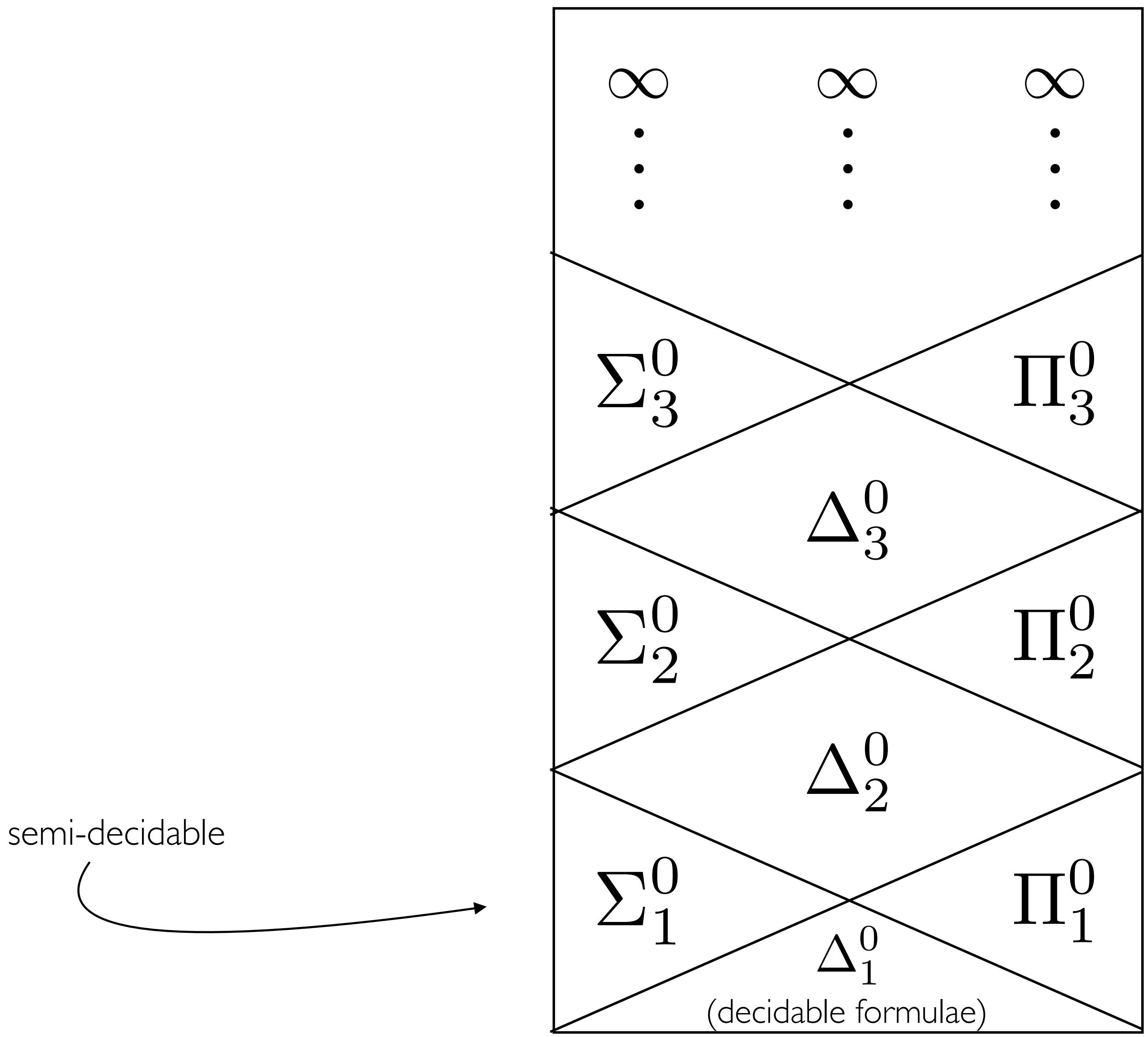
$$\Delta_1^0 = \Sigma_1^0 \cap \Pi_1^0$$

Arithmetic Hierarchy, Part I

Can you see the carryover from PH?

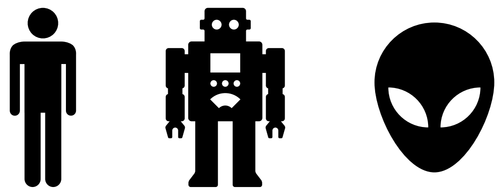
$$\mathbf{2SAMEFUNC} := \{\mathfrak{m}_1, \mathfrak{m}_2 : \forall u \forall v [\exists k(\langle \mathfrak{m}_1, u\rangle : v, k \leftrightarrow \exists k'(\langle \mathfrak{m}_2, u\rangle : v, k')]$$

$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)



$$\Delta_1^0 = \Sigma_1^0 \cap \Pi_1^0$$

Arithmetic Hierarchy, Part I



Can you see the carryover from PH?

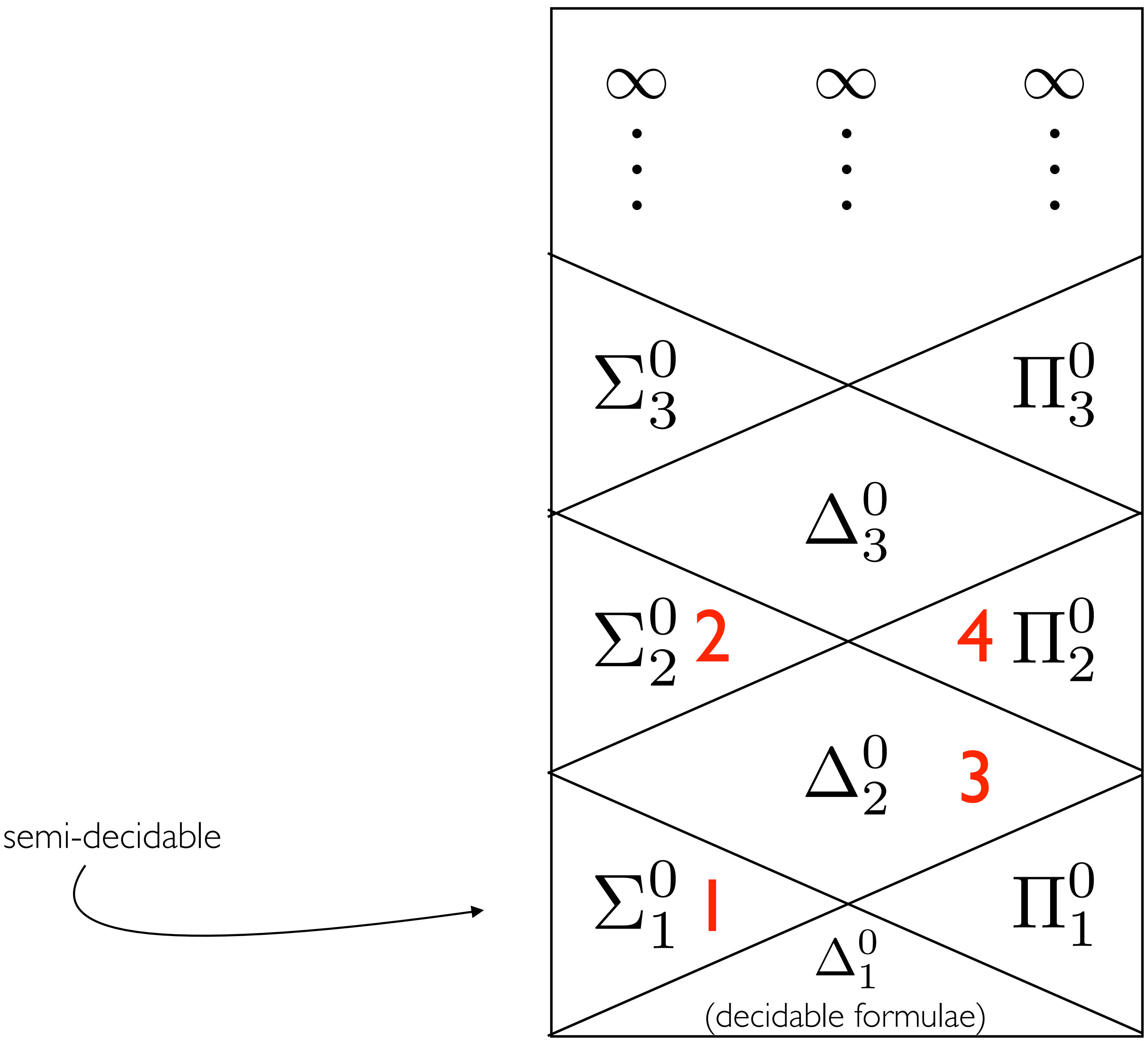
$$\mathbf{2SAMEFUNC} := \{\mathfrak{m}_1, \mathfrak{m}_2 : \forall u \forall v [\exists k (\langle \mathfrak{m}_1, u \rangle : v, k \leftrightarrow \exists k' (\langle \mathfrak{m}_2, u \rangle : v, k')]$$

Let R be a Turing-decidable (= decidable, *simpliciter*) dyadic relation. Where is the set:

$$\{x : \exists y R(x, y)\},$$

1 2 3 or 4?

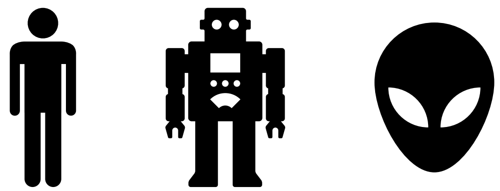
$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)



$$\Delta_1^0 = \Sigma_1^0 \cap \Pi_1^0$$

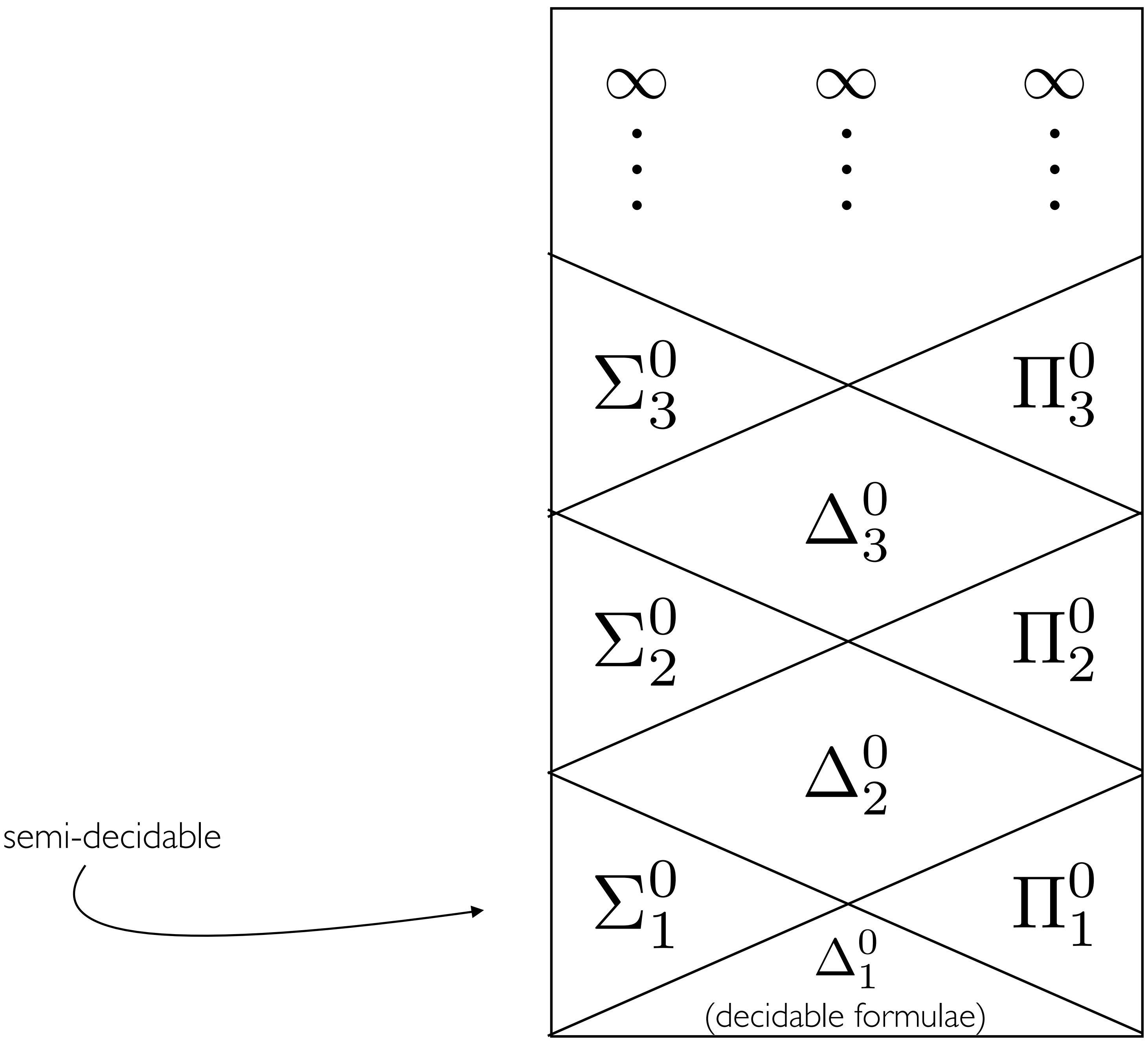
Arithmetic Hierarchy, Part I

Can you see the carryover from PH?



$$\mathbf{2SAMEFUNC} := \{\mathfrak{m}_1, \mathfrak{m}_2 : \forall u \forall v[\exists k(\langle \mathfrak{m}_1, u \rangle : v, k \leftrightarrow \exists k'(\langle \mathfrak{m}_2, u \rangle : v, k')]$$

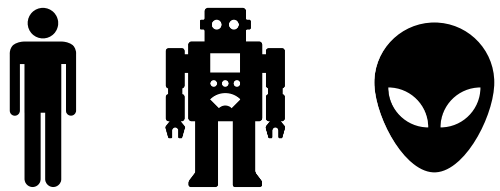
$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)



$$\Delta_1^0 = \Sigma_1^0 \cap \Pi_1^0$$

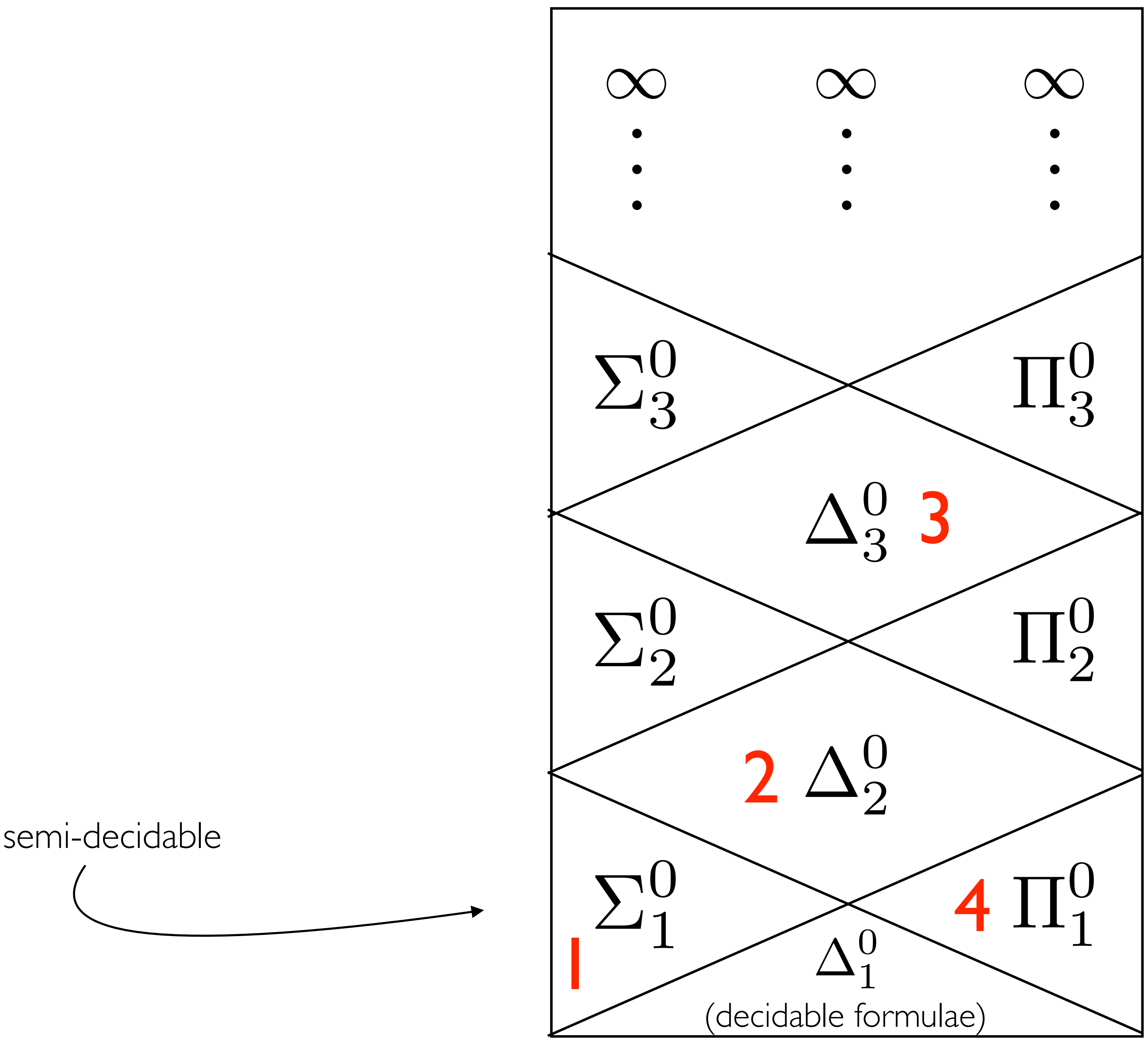
Arithmetic Hierarchy, Part I

Can you see the carryover from PH?



$$\mathbf{2SAMEFUNC} := \{\mathfrak{m}_1, \mathfrak{m}_2 : \forall u \forall v[\exists k(\langle \mathfrak{m}_1, u\rangle : v, k \leftrightarrow \exists k'(\langle \mathfrak{m}_2, u\rangle : v, k')]$$

$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)



Let R be a Turing-decidable (= decidable, *simpliciter*) dyadic relation. Where is the set:

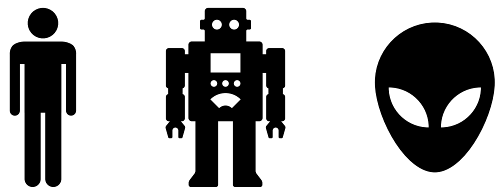
$$\{x : \forall y R(x, y)\},$$

1 2 3 or 4?

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

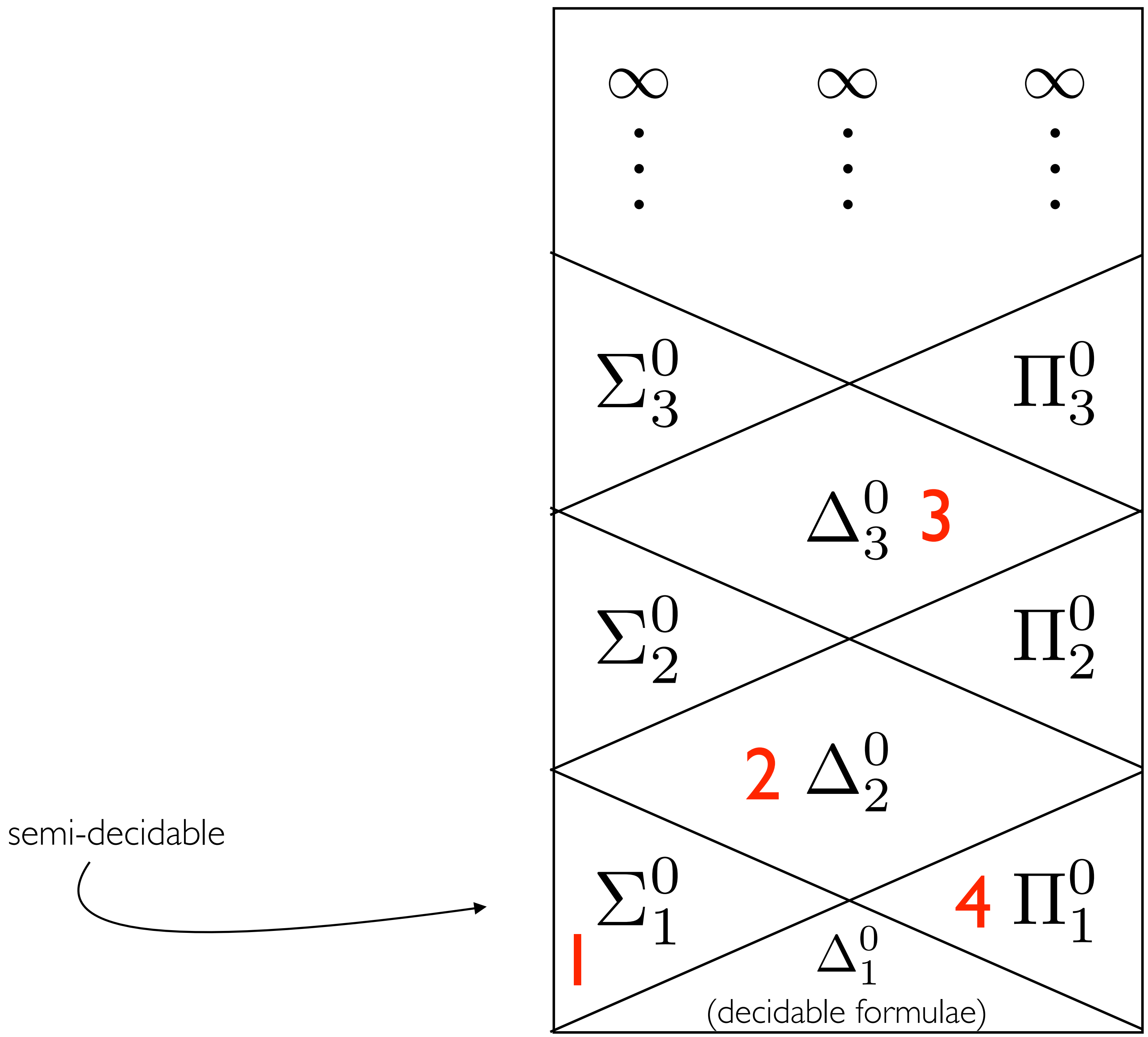
Arithmetic Hierarchy, Part I

Can you see the carryover from PH?



$$\mathbf{2SAMEFUNC} := \{\mathfrak{m}_1, \mathfrak{m}_2 : \forall u \forall v [\exists k(\langle \mathfrak{m}_1, u\rangle : v, k \leftrightarrow \exists k'(\langle \mathfrak{m}_2, u\rangle : v, k')]$$

$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)



Let R be a Turing-decidable (= decidable, *simpliciter*) dyadic relation. Where is the set:

$$\{x : \forall y R(x, y)\},$$

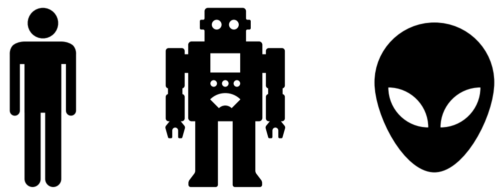
1 2 3 or 4?

$$x \in \Sigma_i \text{ iff } \exists R\ \exists y_1 \forall y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \forall \text{ if } i \text{ even}; Q_i = \exists \text{ if } i \text{ odd})$$

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

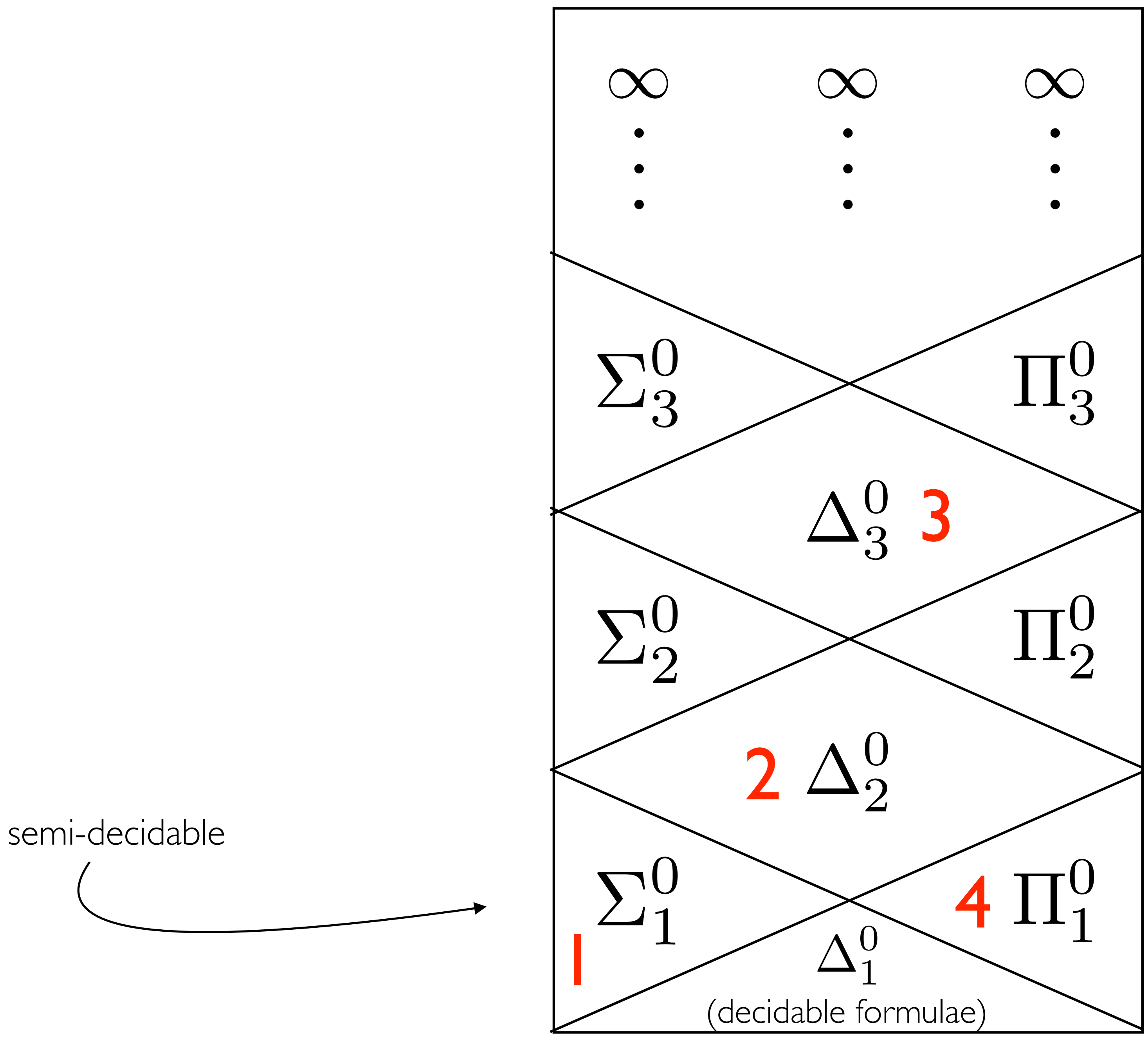
Arithmetic Hierarchy, Part I

Can you see the carryover from PH?



$$\mathbf{2SAMEFUNC} := \{\mathfrak{m}_1, \mathfrak{m}_2 : \forall u \forall v[\exists k(\langle \mathfrak{m}_1, u\rangle : v, k \leftrightarrow \exists k'(\langle \mathfrak{m}_2, u\rangle : v, k')]$$

$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)



Let R be a Turing-decidable (= decidable, *simpliciter*) dyadic relation. Where is the set:

$$\{x : \forall y R(x, y)\},$$

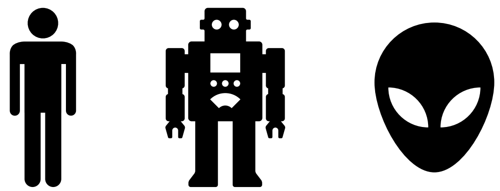
1 2 3 or 4?

$$x \in \Sigma_i \text{ iff } \exists R\ \exists y_1 \forall y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \forall \text{ if } i \text{ even}; Q_i = \exists \text{ if } i \text{ odd})$$

$$x \in \Pi_i \text{ iff } \exists R\ \forall y_1 \exists y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \exists \text{ if } j \text{ even}; Q_i = \forall \text{ if } j \text{ odd})$$

$$\Delta_1^0 = \Sigma_1^0 \cap \Pi_1^0$$

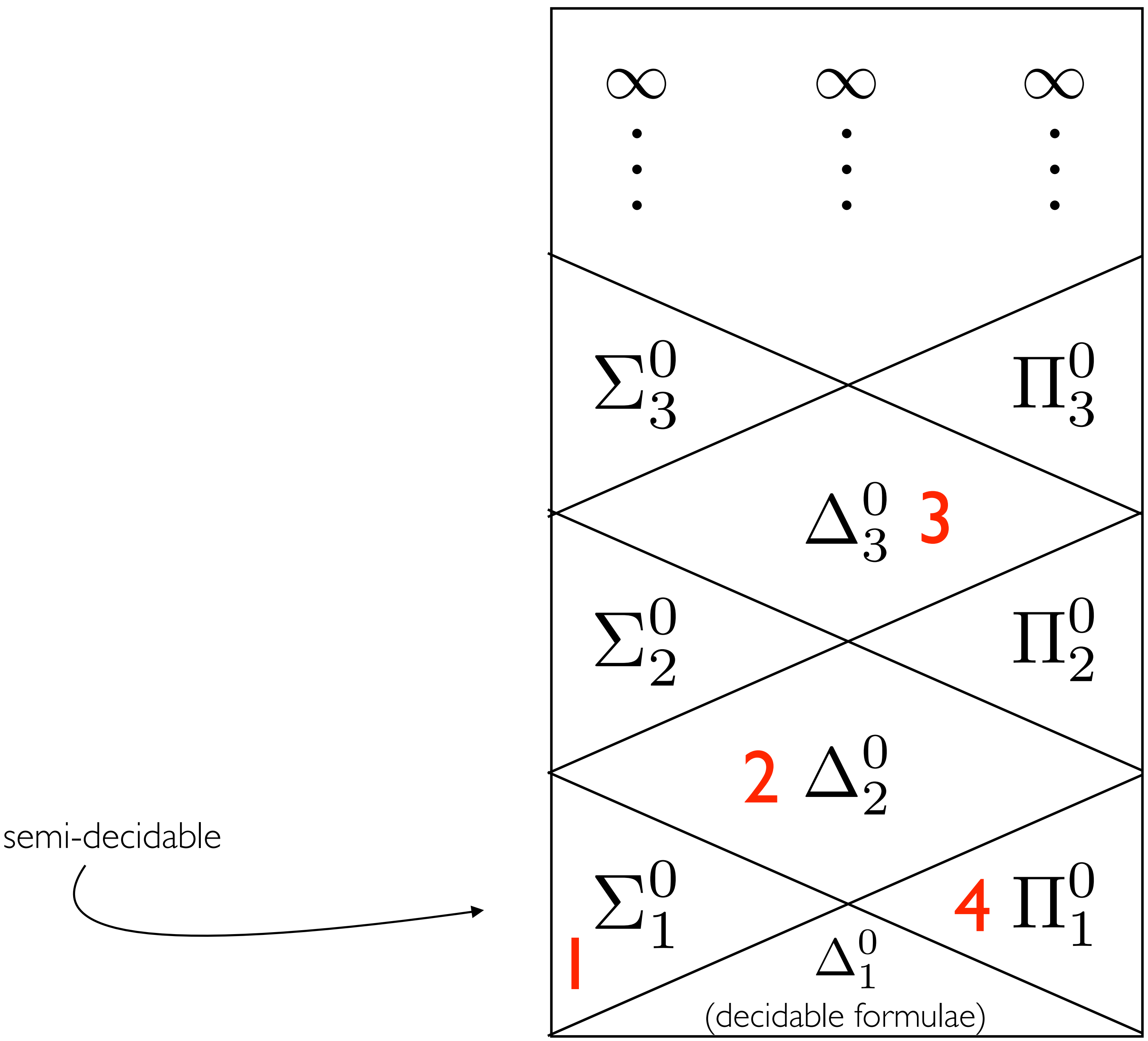
Arithmetic Hierarchy, Part I



Can you see the carryover from PH?

$$\mathbf{2SAMEFUNC} := \{\mathfrak{m}_1, \mathfrak{m}_2 : \forall u \forall v [\exists k (\langle \mathfrak{m}_1, u \rangle : v, k \leftrightarrow \exists k' (\langle \mathfrak{m}_2, u \rangle : v, k')]$$

$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)



Let R be a Turing-decidable (= decidable, *simpliciter*) dyadic relation. Where is the set:

$$\{x : \forall y R(x, y)\},$$

1 2 3 or 4?

$$x \in \Sigma_i \text{ iff } \exists R \; \exists y_1 \forall y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \forall \text{ if } i \text{ even}; Q_i = \exists \text{ if } i \text{ odd})$$

$$x \in \Pi_i \text{ iff } \exists R \; \forall y_1 \exists y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \exists \text{ if } j \text{ even}; Q_i = \forall \text{ if } j \text{ odd})$$

Try your hand at classifying! …

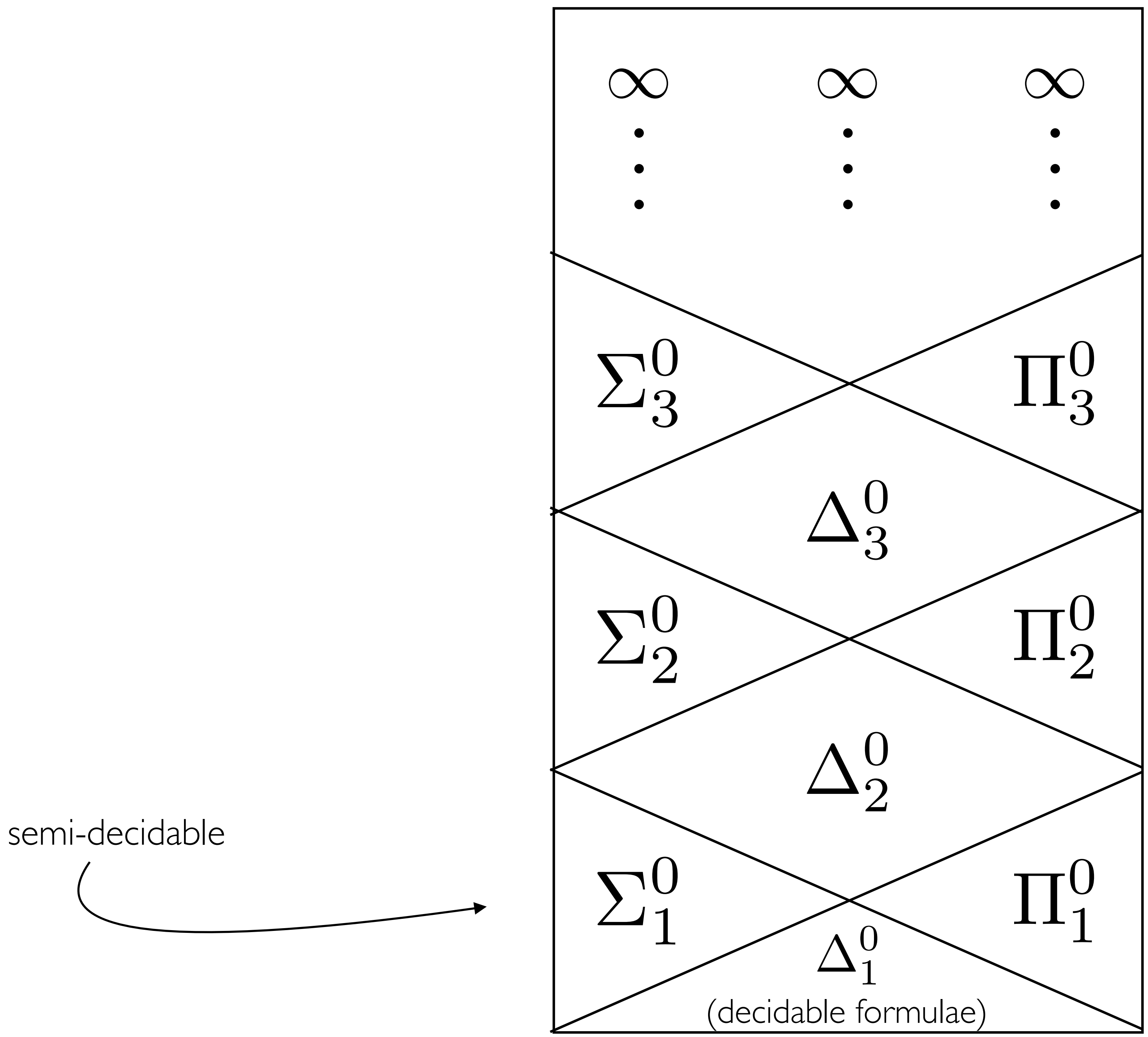
$$\Delta_1^0 = \Sigma_1^0 \cap \Pi_1^0$$

Arithmetic Hierarchy, Part I

Can you see the carryover from PH?

$$\mathbf{2SAMEFUNC} := \{\mathfrak{m}_1, \mathfrak{m}_2 : \forall u \forall v [\exists k (\langle \mathfrak{m}_1, u\rangle : v, k \leftrightarrow \exists k' (\langle \mathfrak{m}_2, u\rangle : v, k')]$$

$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)



$$x \in \Sigma_i \text{ iff } \exists R\ \exists y_1 \forall y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \forall \text{ if } i \text{ even}; Q_i = \exists \text{ if } i \text{ odd})$$

$$x \in \Pi_i \text{ iff } \exists R\ \forall y_1 \exists y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \exists \text{ if } j \text{ even}; Q_i = \forall \text{ if } j \text{ odd})$$

Try your hand at classifying! …

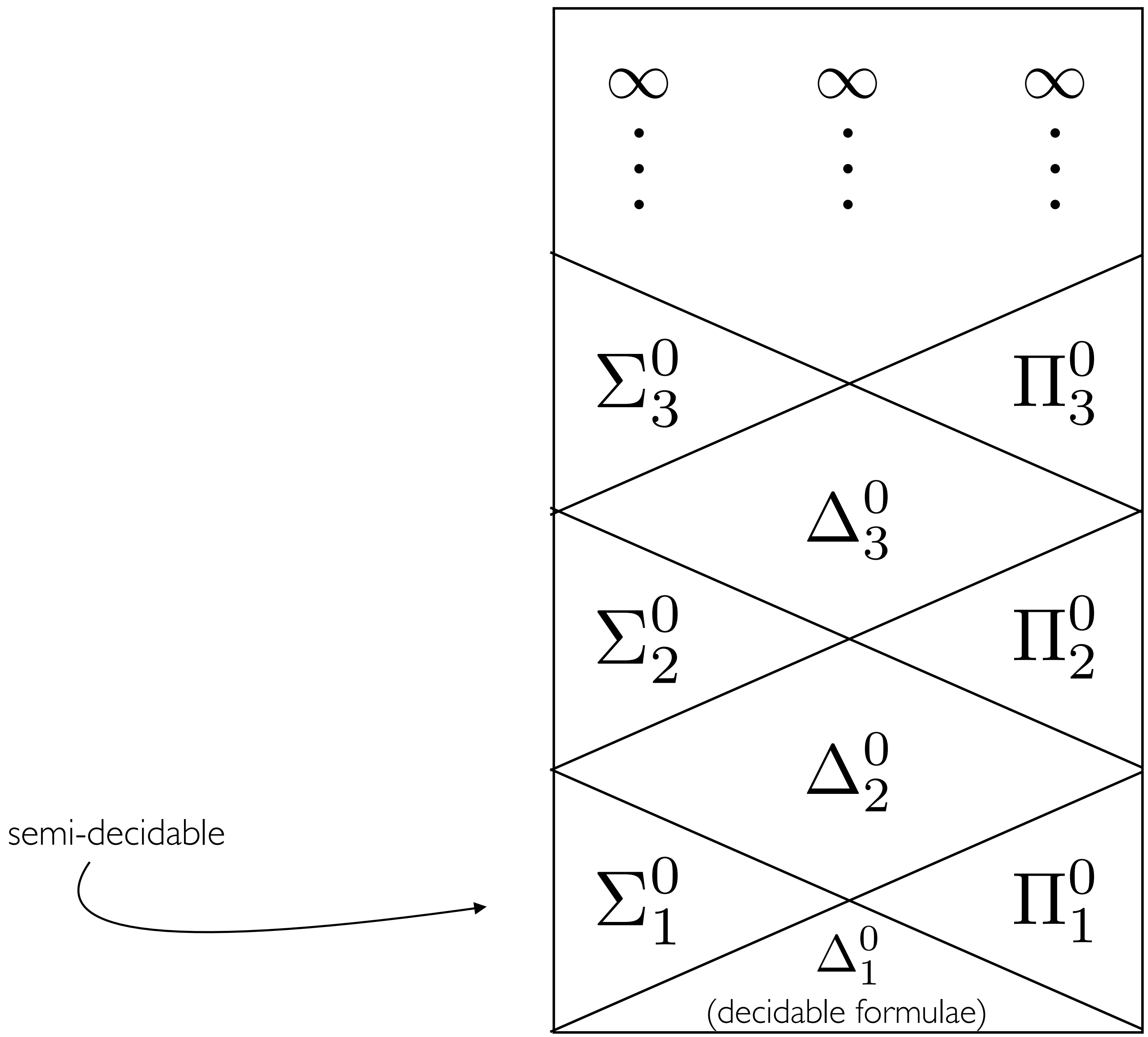
$$\Delta^0_1 = \Sigma^0_1 \cap \Pi^0_1$$

Arithmetic Hierarchy, Part I

Can you see the carryover from PH?

$$\mathbf{2SAMEFUNC} := \{\mathfrak{m}_1, \mathfrak{m}_2 : \forall u \forall v[\exists k(\langle \mathfrak{m}_1, u\rangle : v, k \leftrightarrow \exists k'(\langle \mathfrak{m}_2, u\rangle : v, k')]$$

$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)



$$\Delta_1^0 = \Sigma_1^0 \cap \Pi_1^0$$

$$x \in \Sigma_i \text{ iff } \exists R\ \exists y_1 \forall y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \forall \text{ if } i \text{ even}; Q_i = \exists \text{ if } i \text{ odd})$$

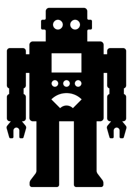
$$x \in \Pi_i \text{ iff } \exists R\ \forall y_1 \exists y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \exists \text{ if } j \text{ even}; Q_i = \forall \text{ if } j \text{ odd})$$

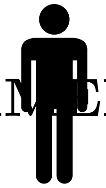
Try your hand at classifying! …

From Kleene: The set to be classified, $\mathscr{K}$, consists of all those inputs to a given Turing machine $\mathfrak{m}$ that results in this machine halting after some number of steps.

Arithmetic Hierarchy, Part I

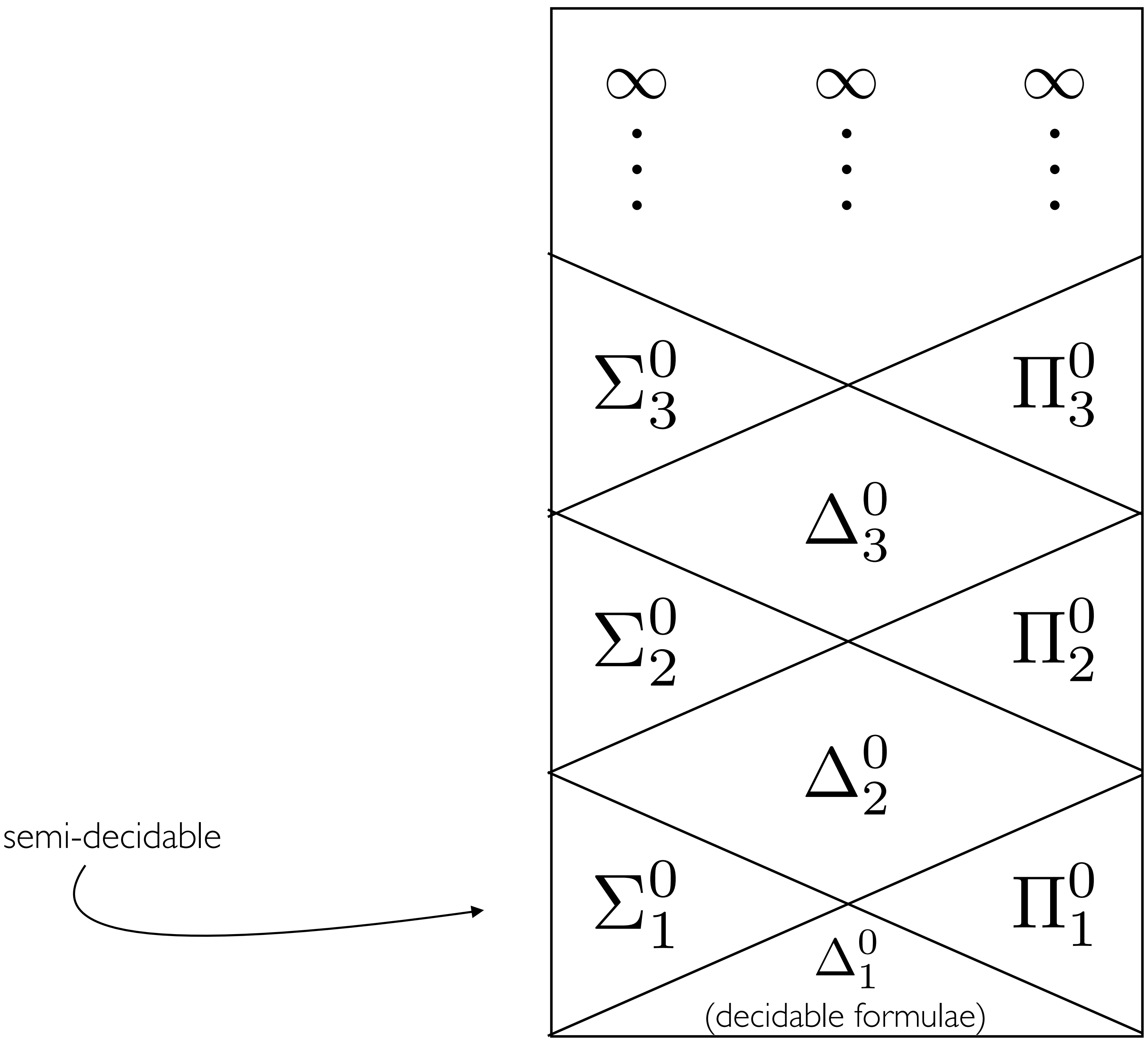
[Can you see the carryover from PH?]





$$\mathbf{2SAMEFUNC} := \{\mathfrak{m}_1, \mathfrak{m}_2 : \forall u \forall v [\exists k(\langle \mathfrak{m}_1, u\rangle : v, k \leftrightarrow \exists k'(\langle \mathfrak{m}_2, u\rangle : v, k')]$$

$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)



$$x \in \Sigma_i \text{ iff } \exists R\ \exists y_1 \forall y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \forall \text{ if } i \text{ even}; Q_i = \exists \text{ if } i \text{ odd})$$

$$x \in \Pi_i \text{ iff } \exists R\ \forall y_1 \exists y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \exists \text{ if } j \text{ even}; Q_i = \forall \text{ if } j \text{ odd})$$

[Try your hand at classifying! …]

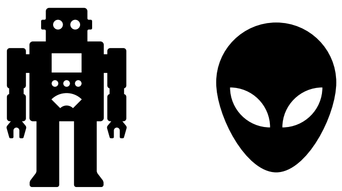
[From Kleene: The set to be classified, $\mathscr{K}$, consists of all those inputs to a given Turing machine $\mathfrak{m}$ that results in this machine halting after some number of steps.]

$$\Delta_1^0 = \Sigma_1^0 \cap \Pi_1^0$$

Arithmetic Hierarchy, Part I

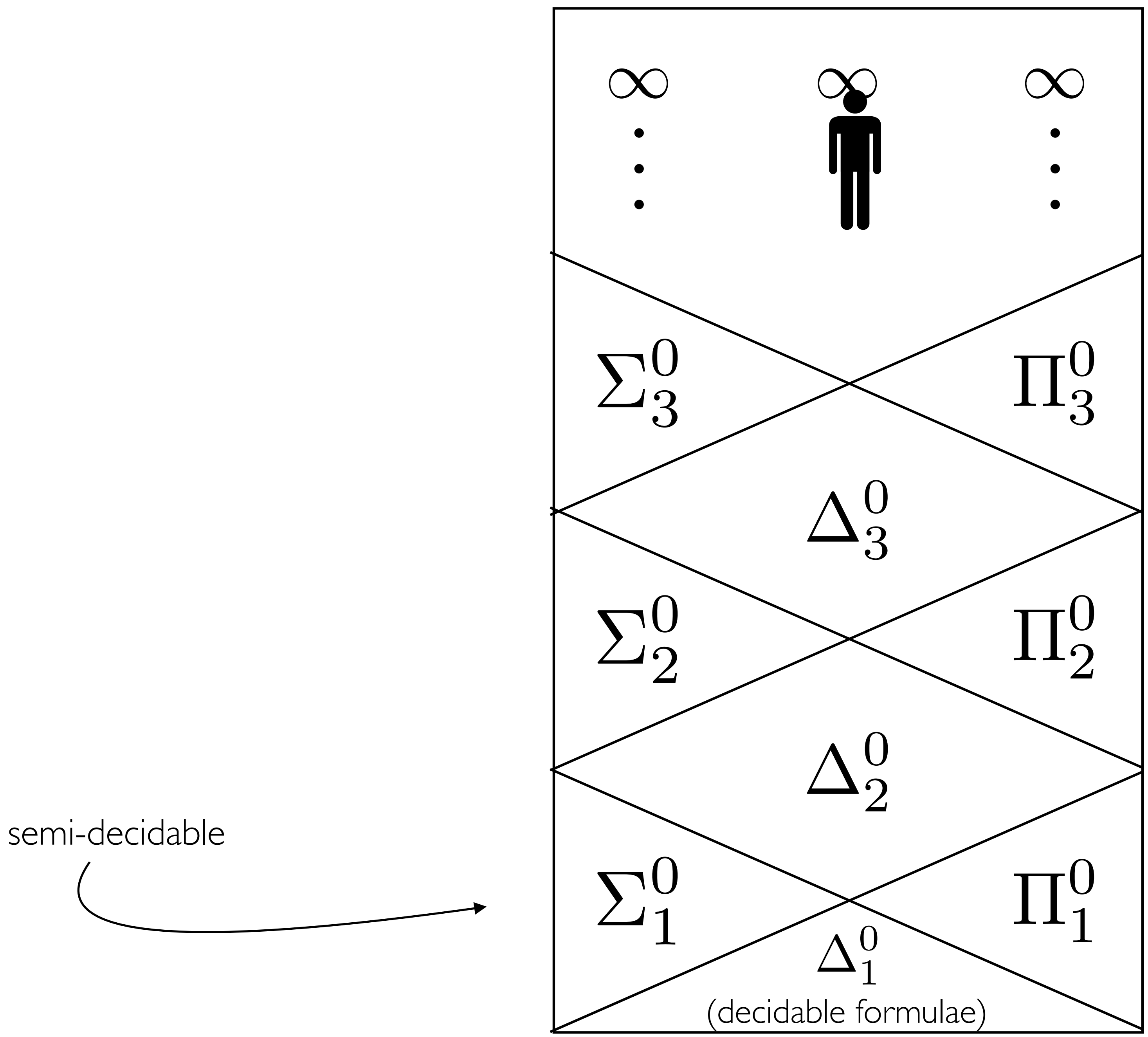
Arithmetic Hierarchy, Part I

Can you see the carryover from PH?



$$\mathbf{2SAMEFUNC} := \{\mathfrak{m}_1, \mathfrak{m}_2 : \forall u \forall v [\exists k(\langle \mathfrak{m}_1, u \rangle : v, k \leftrightarrow \exists k'(\langle \mathfrak{m}_2, u \rangle : v, k')]$$

$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)



$$\Delta_1^0 = \Sigma_1^0 \cap \Pi_1^0$$

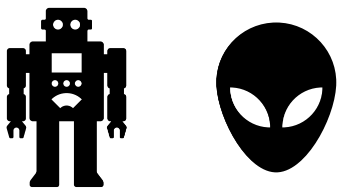
$$x \in \Sigma_i \text{ iff } \exists R\ \exists y_1 \forall y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \forall \text{ if } i \text{ even}; Q_i = \exists \text{ if } i \text{ odd})$$

$$x \in \Pi_i \text{ iff } \exists R\ \forall y_1 \exists y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \exists \text{ if } j \text{ even}; Q_i = \forall \text{ if } j \text{ odd})$$

Try your hand at classifying! …

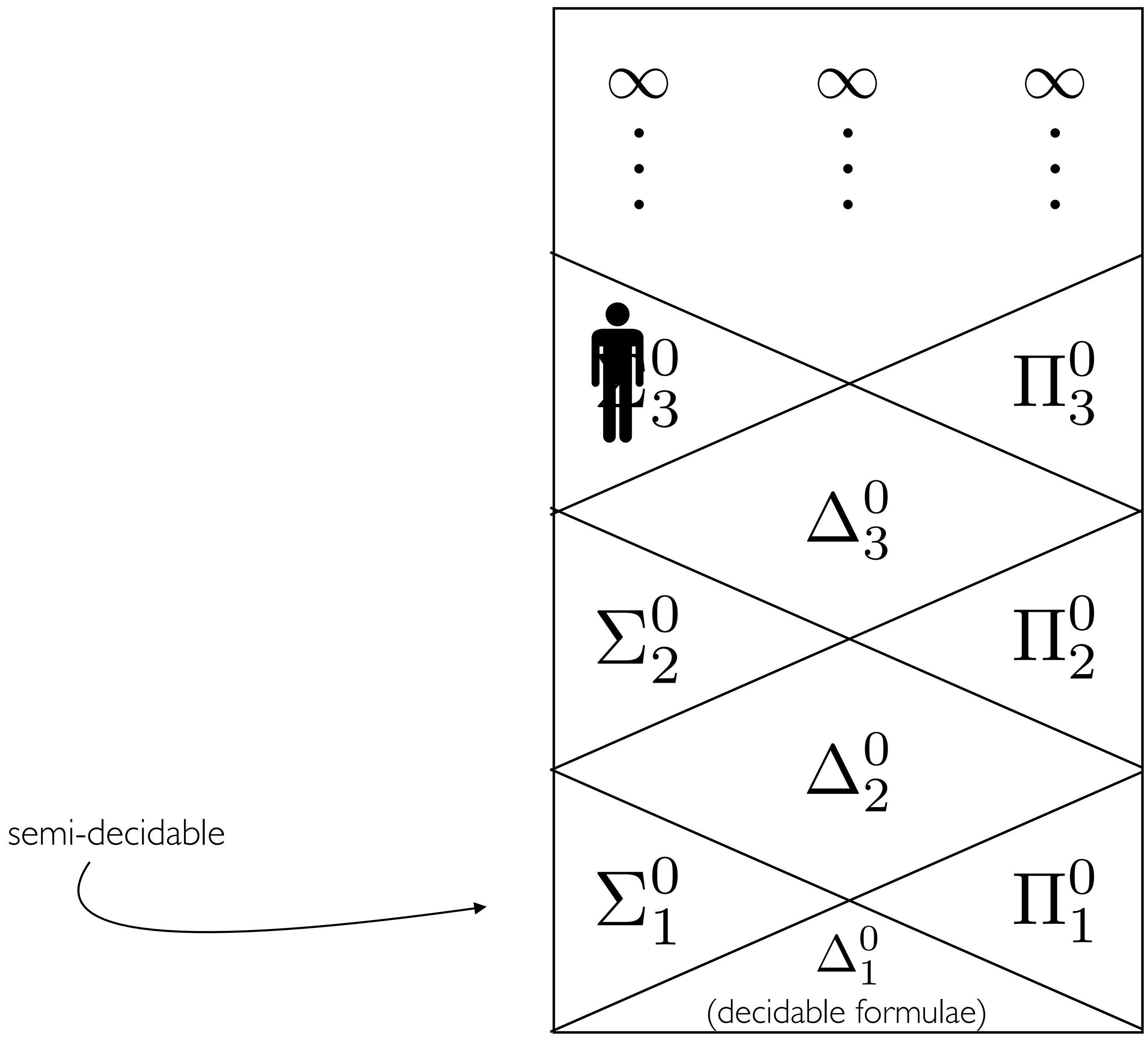
From Kleene: The set to be classified, $\mathscr{K}$, consists of all those inputs to a given Turing machine $\mathfrak{m}$ that results in this machine halting after some number of steps.

Can you see the carryover from PH?



$$\mathbf{2SAMEFUNC} := \{\mathfrak{m}_1, \mathfrak{m}_2 : \forall u \forall v [\exists k (\langle \mathfrak{m}_1, u \rangle : v, k \leftrightarrow \exists k' (\langle \mathfrak{m}_2, u \rangle : v, k')]$$

$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)



$$x \in \Sigma_i \text{ iff } \exists R\ \exists y_1 \forall y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \forall \text{ if } i \text{ even}; Q_i = \exists \text{ if } i \text{ odd})$$

$$x \in \Pi_i \text{ iff } \exists R\ \forall y_1 \exists y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \exists \text{ if } j \text{ even}; Q_i = \forall \text{ if } j \text{ odd})$$

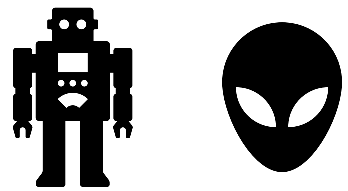
Try your hand at classifying! ...

From Kleene: The set to be classified, $\mathscr{K}$, consists of all those inputs to a given Turing machine $\mathfrak{m}$ that results in this machine halting after some number of steps.

$$\Delta_1^0 = \Sigma_1^0 \cap \Pi_1^0$$

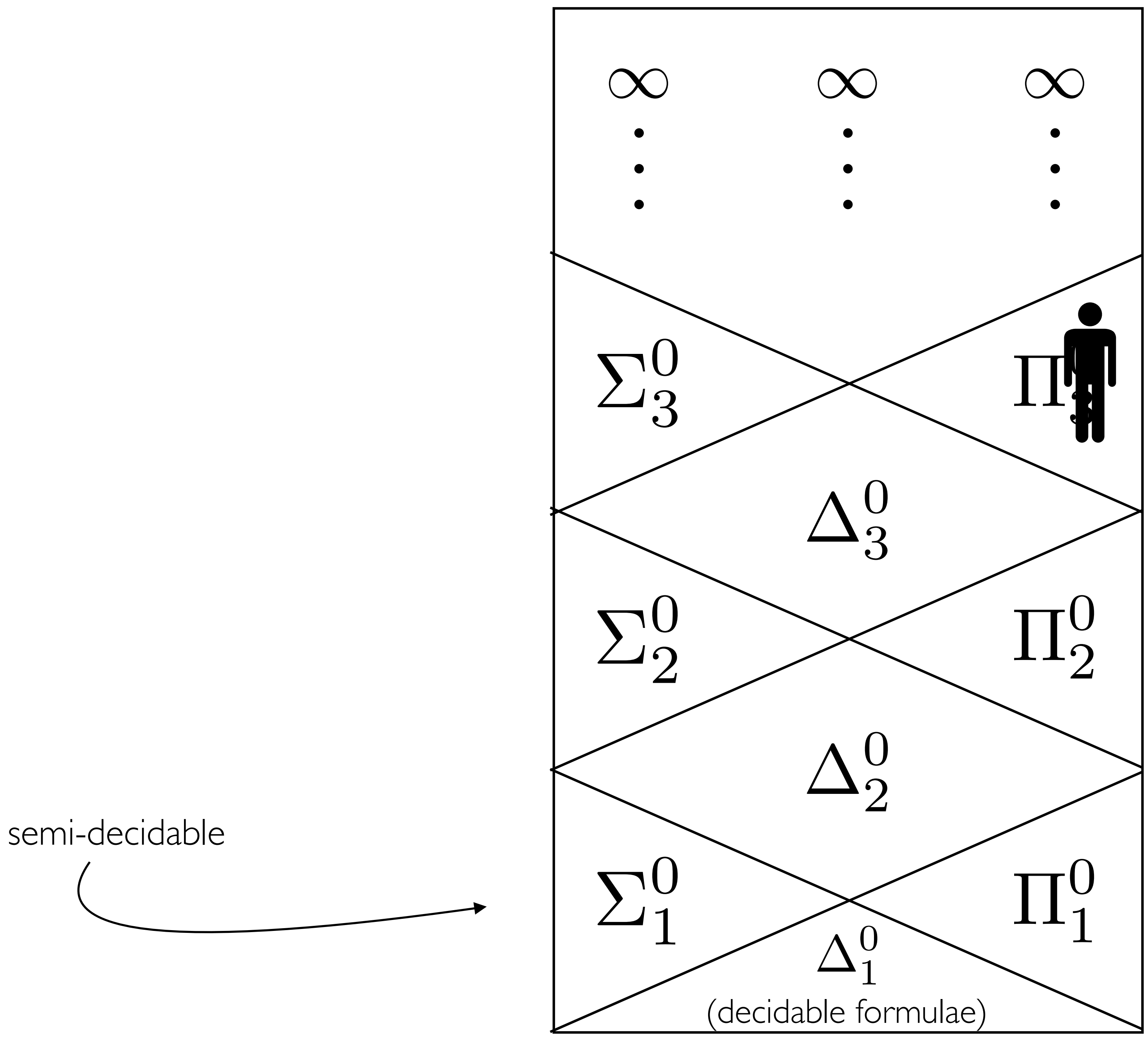
Arithmetic Hierarchy, Part I

Can you see the carryover from PH?



$$\mathbf{2SAMEFUNC} := \{\mathfrak{m}_1, \mathfrak{m}_2 : \forall u \forall v [\exists k (\langle \mathfrak{m}_1, u\rangle : v, k \leftrightarrow \exists k' (\langle \mathfrak{m}_2, u\rangle : v, k')]$$

$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)



$$\Delta_1^0 = \Sigma_1^0 \cap \Pi_1^0$$

$$x \in \Sigma_i \text{ iff } \exists R\ \exists y_1 \forall y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \forall \text{ if } i \text{ even}; Q_i = \exists \text{ if } i \text{ odd})$$

$$x \in \Pi_i \text{ iff } \exists R\ \forall y_1 \exists y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \exists \text{ if } j \text{ even}; Q_i = \forall \text{ if } j \text{ odd})$$

Try your hand at classifying! …

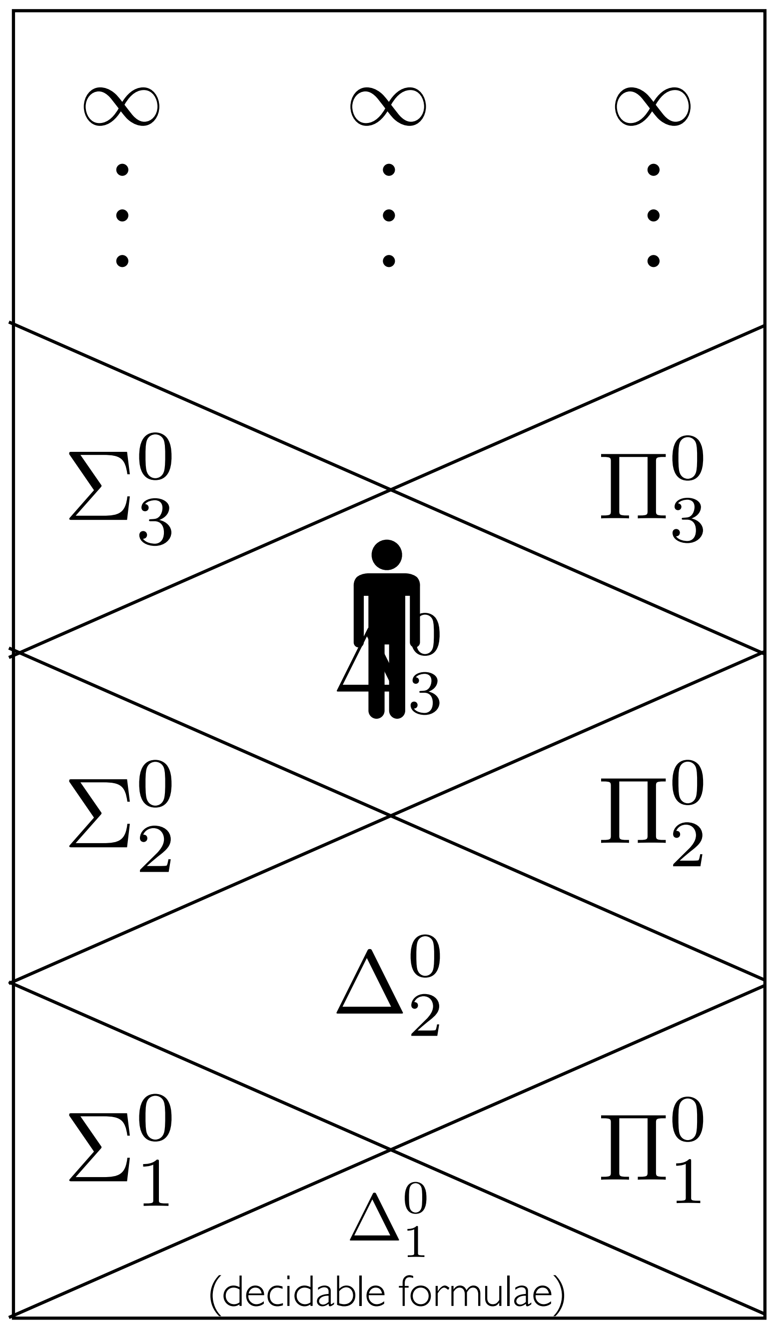
From Kleene: The set to be classified, $\mathscr{K}$, consists of all those inputs to a given Turing machine $\mathfrak{m}$ that results in this machine halting after some number of steps.

Arithmetic Hierarchy, Part I

Can you see the carryover from PH?

$$\mathbf{2SAMEFUNC} := \{\mathfrak{m}_1, \mathfrak{m}_2 : \forall u \forall v[\exists k(\langle \mathfrak{m}_1, u\rangle : v, k \leftrightarrow \exists k'(\langle \mathfrak{m}_2, u\rangle : v, k')]$$

$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)



semi-decidable

$$x \in \Sigma_i \text{ iff } \exists R\ \exists y_1 \forall y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \forall \text{ if } i \text{ even}; Q_i = \exists \text{ if } i \text{ odd})$$

$$x \in \Pi_i \text{ iff } \exists R\ \forall y_1 \exists y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \exists \text{ if } j \text{ even}; Q_i = \forall \text{ if } j \text{ odd})$$

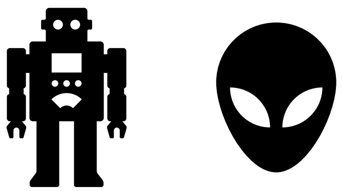
Try your hand at classifying! …

From Kleene: The set to be classified, $\mathscr{K}$, consists of all those inputs to a given Turing machine $\mathfrak{m}$ that results in this machine halting after some number of steps.

$$\Delta_1^0 = \Sigma_1^0 \cap \Pi_1^0$$

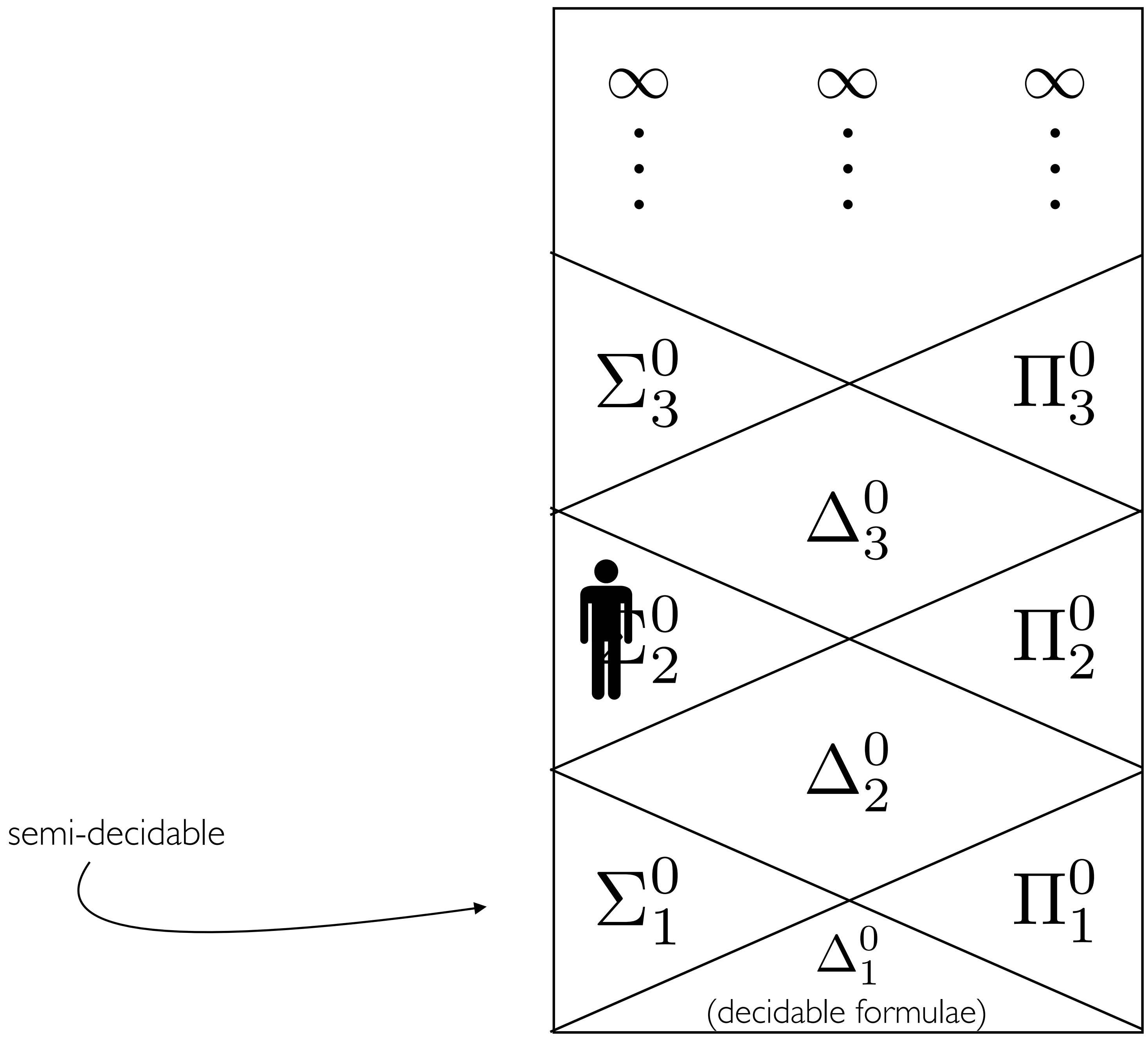
Arithmetic Hierarchy, Part I

Can you see the carryover from PH?



$$\mathbf{2SAMEFUNC} := \{\mathfrak{m}_1, \mathfrak{m}_2 : \forall u \forall v [\exists k (\langle \mathfrak{m}_1, u \rangle : v, k \leftrightarrow \exists k' (\langle \mathfrak{m}_2, u \rangle : v, k')]$$

$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)



$$x \in \Sigma_i \text{ iff } \exists R\ \exists y_1 \forall y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \forall \text{ if } i \text{ even}; Q_i = \exists \text{ if } i \text{ odd})$$

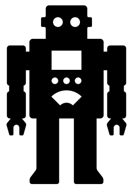
$$x \in \Pi_i \text{ iff } \exists R\ \forall y_1 \exists y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \exists \text{ if } j \text{ even}; Q_i = \forall \text{ if } j \text{ odd})$$

Try your hand at classifying! …

From Kleene: The set to be classified, $\mathscr{K}$, consists of all those inputs to a given Turing machine $\mathfrak{m}$ that results in this machine halting after some number of steps.

$$\Delta_1^0 = \Sigma_1^0 \cap \Pi_1^0$$

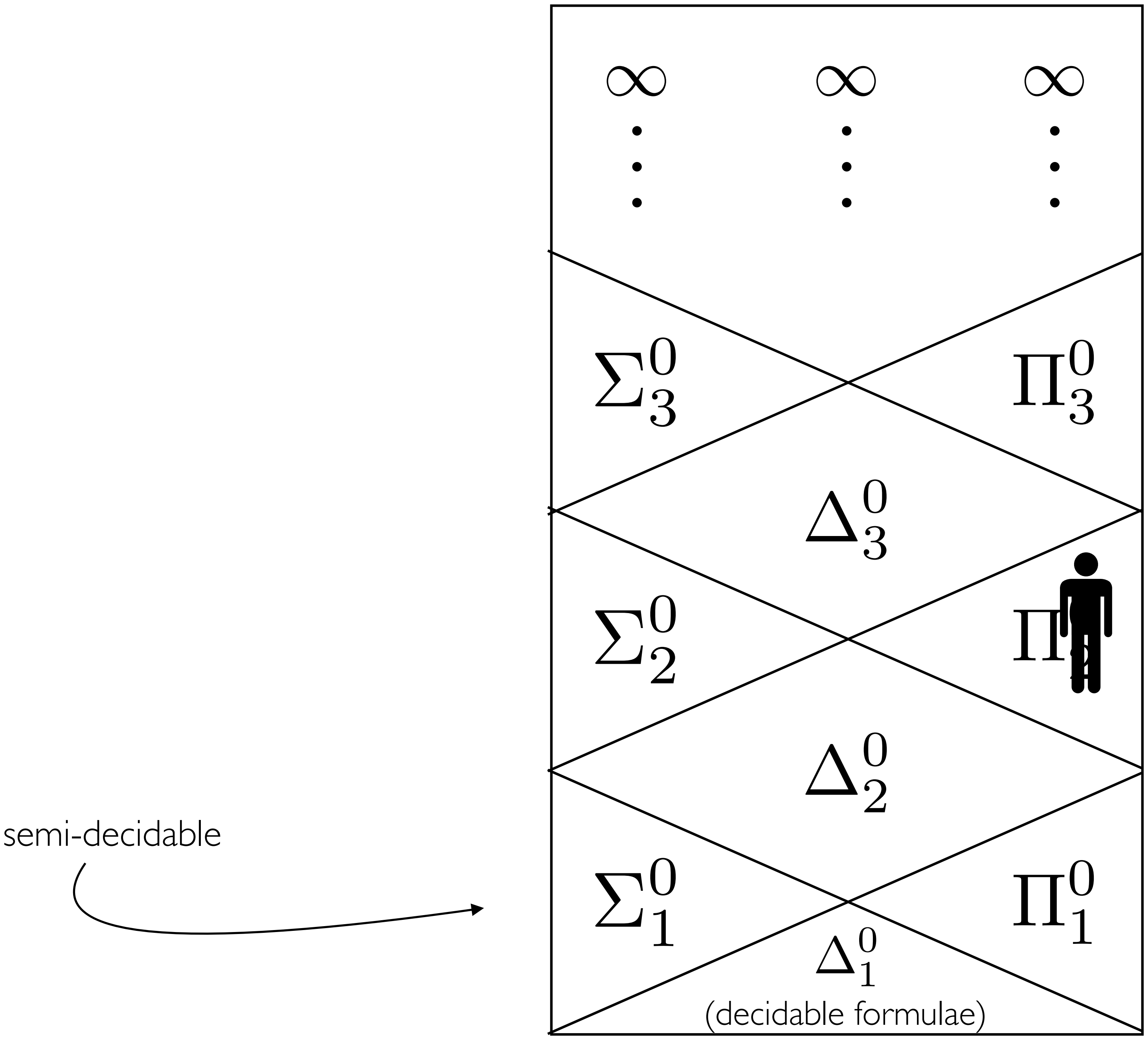
Arithmetic Hierarchy, Part I




Can you see the carryover from PH?

$$\mathbf{2SAMEFUNC} := \{\mathfrak{m}_1, \mathfrak{m}_2 : \forall u \forall v [\exists k(\langle \mathfrak{m}_1, u\rangle : v, k \leftrightarrow \exists k'(\langle \mathfrak{m}_2, u\rangle : v, k')]$$

$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)



$$\Delta_1^0 = \Sigma_1^0 \cap \Pi_1^0$$

$$x \in \Sigma_i \text{ iff } \exists R\ \exists y_1 \forall y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \forall \text{ if } i \text{ even}; Q_i = \exists \text{ if } i \text{ odd})$$

$$x \in \Pi_i \text{ iff } \exists R\ \forall y_1 \exists y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \exists \text{ if } j \text{ even}; Q_i = \forall \text{ if } j \text{ odd})$$

Try your hand at classifying! …

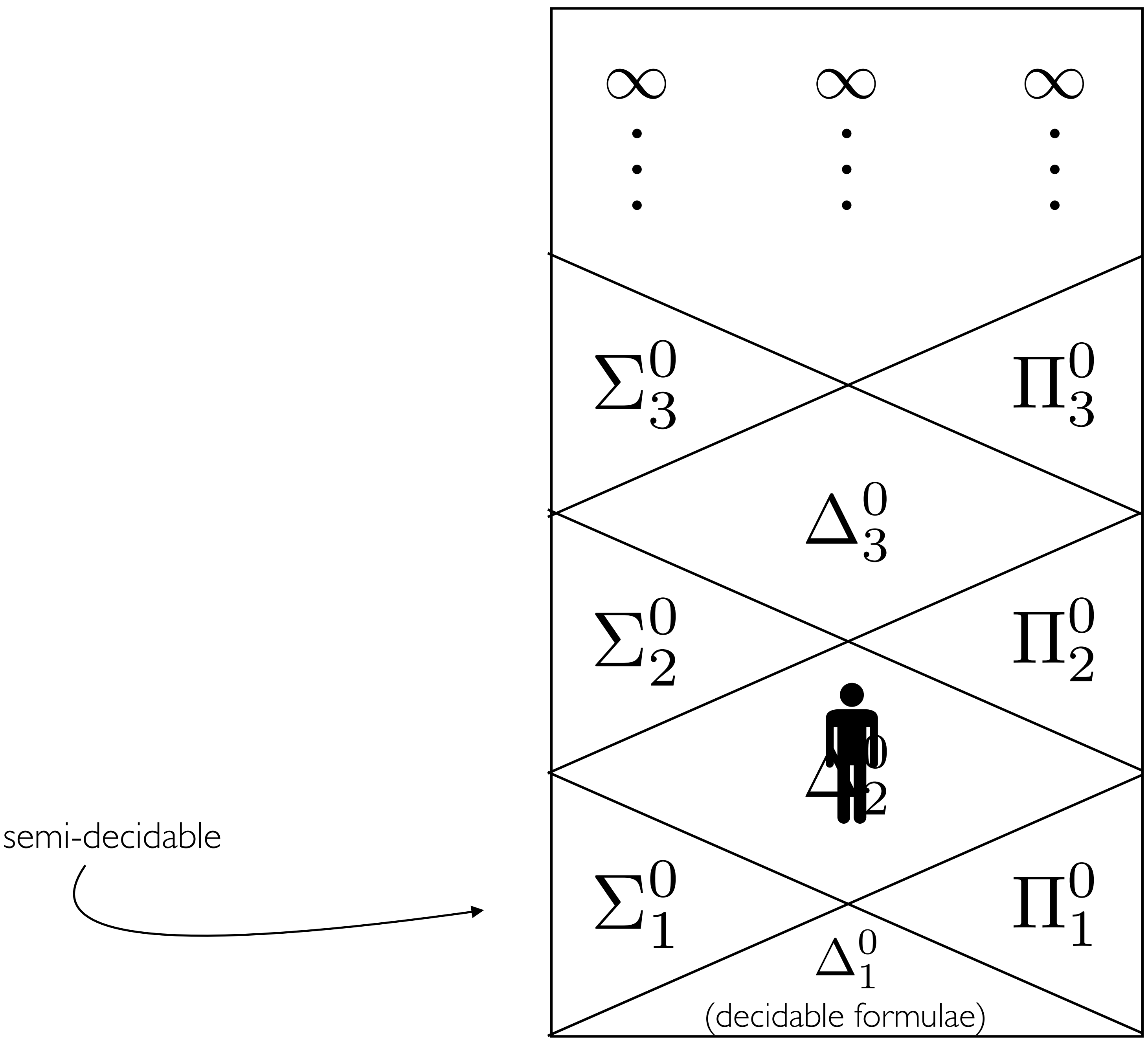
From Kleene: The set to be classified, $\mathscr{K}$, consists of all those inputs to a given Turing machine $\mathfrak{m}$ that results in this machine halting after some number of steps.

Arithmetic Hierarchy, Part I

Can you see the carryover from PH?

$$\mathbf{2SAMEFUNC} := \{\mathfrak{m}_1, \mathfrak{m}_2 : \forall u \forall v [\exists k (\langle \mathfrak{m}_1, u \rangle : v, k \leftrightarrow \exists k' (\langle \mathfrak{m}_2, u \rangle : v, k')]$$

$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)



$$\Delta_1^0 = \Sigma_1^0 \cap \Pi_1^0$$

$$x \in \Sigma_i \text{ iff } \exists R\ \exists y_1 \forall y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \forall \text{ if } i \text{ even}; Q_i = \exists \text{ if } i \text{ odd})$$

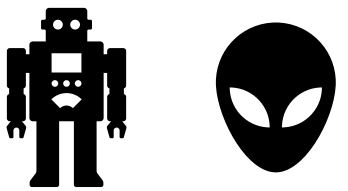
$$x \in \Pi_i \text{ iff } \exists R\ \forall y_1 \exists y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \exists \text{ if } j \text{ even}; Q_i = \forall \text{ if } j \text{ odd})$$

Try your hand at classifying! …

From Kleene: The set to be classified, $\mathscr{K}$, consists of all those inputs to a given Turing machine $\mathfrak{m}$ that results in this machine halting after some number of steps.

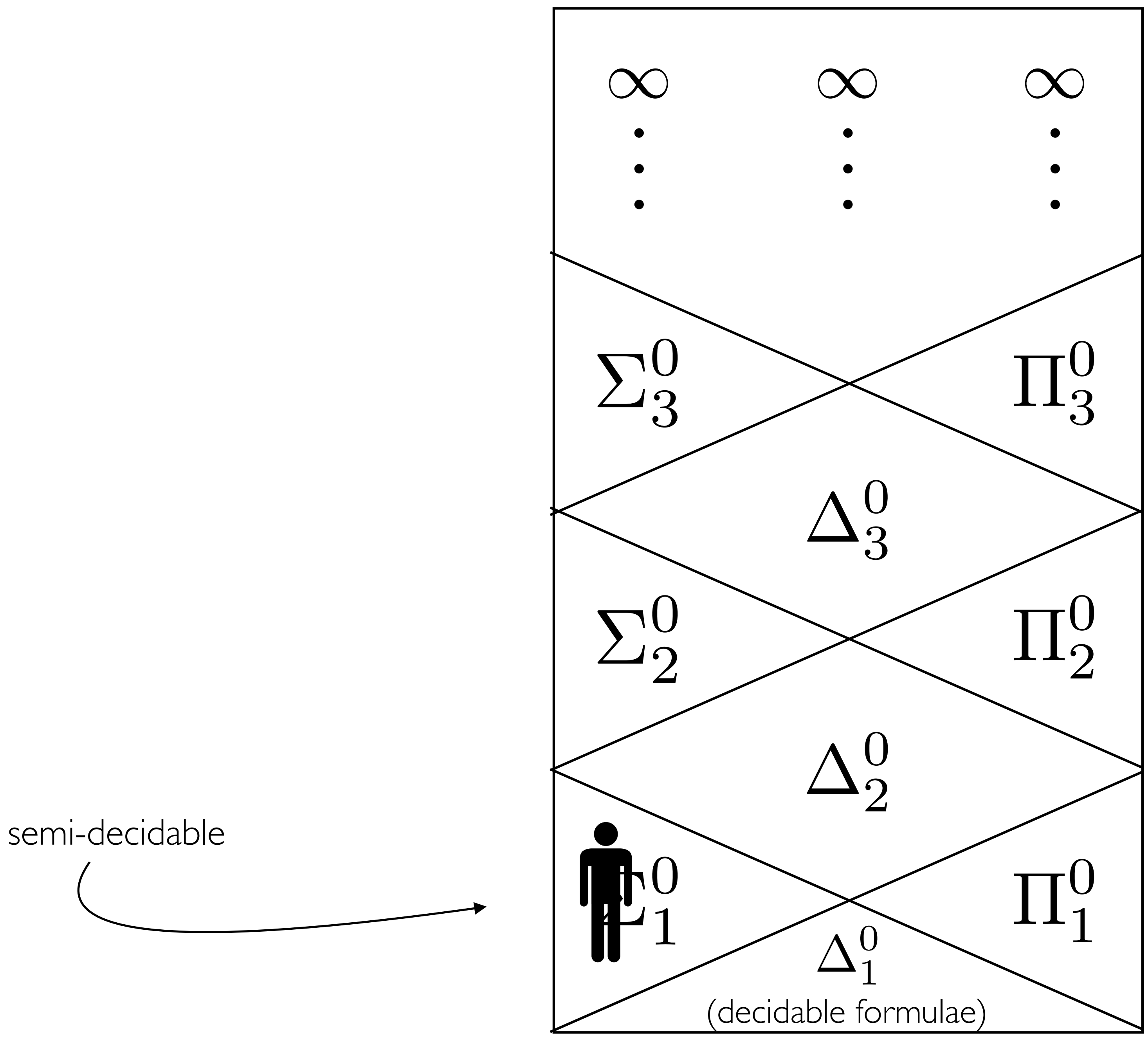
Arithmetic Hierarchy, Part I

Can you see the carryover from PH?



$$\mathbf{2SAMEFUNC} := \{\mathfrak{m}_1, \mathfrak{m}_2 : \forall u \forall v [\exists k(\langle \mathfrak{m}_1, u\rangle : v, k \leftrightarrow \exists k'(\langle \mathfrak{m}_2, u\rangle : v, k')]$$

$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)



$$x \in \Sigma_i \text{ iff } \exists R\ \exists y_1 \forall y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \forall \text{ if } i \text{ even}; Q_i = \exists \text{ if } i \text{ odd})$$

$$x \in \Pi_i \text{ iff } \exists R\ \forall y_1 \exists y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \exists \text{ if } j \text{ even}; Q_i = \forall \text{ if } j \text{ odd})$$

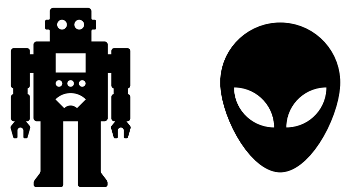
Try your hand at classifying! …

From Kleene: The set to be classified, $\mathscr{K}$, consists of all those inputs to a given Turing machine $\mathfrak{m}$ that results in this machine halting after some number of steps.

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

Arithmetic Hierarchy, Part I

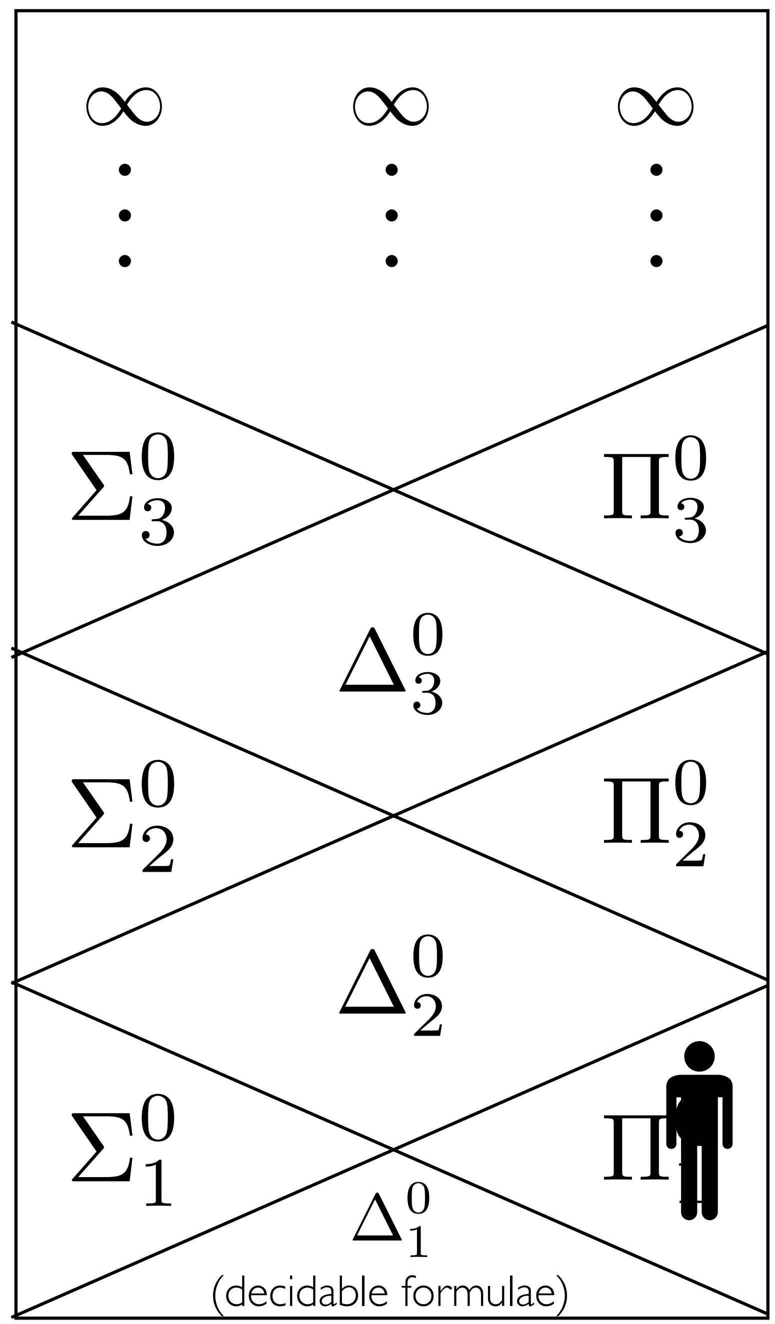
Can you see the carryover from PH?



$$\mathbf{2SAMEFUNC} := \{\mathfrak{m}_1, \mathfrak{m}_2 : \forall u \forall v [\exists k(\langle \mathfrak{m}_1, u\rangle : v, k \leftrightarrow \exists k'(\langle \mathfrak{m}_2, u\rangle : v, k')]$$

$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)

semi-decidable



$$x \in \Sigma_i \text{ iff } \exists R\ \exists y_1 \forall y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \forall \text{ if } i \text{ even}; Q_i = \exists \text{ if } i \text{ odd})$$

$$x \in \Pi_i \text{ iff } \exists R\ \forall y_1 \exists y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \exists \text{ if } j \text{ even}; Q_i = \forall \text{ if } j \text{ odd})$$

Try your hand at classifying! …

From Kleene: The set to be classified, $\mathscr{K}$, consists of all those inputs to a given Turing machine $\mathfrak{m}$ that results in this machine halting after some number of steps.

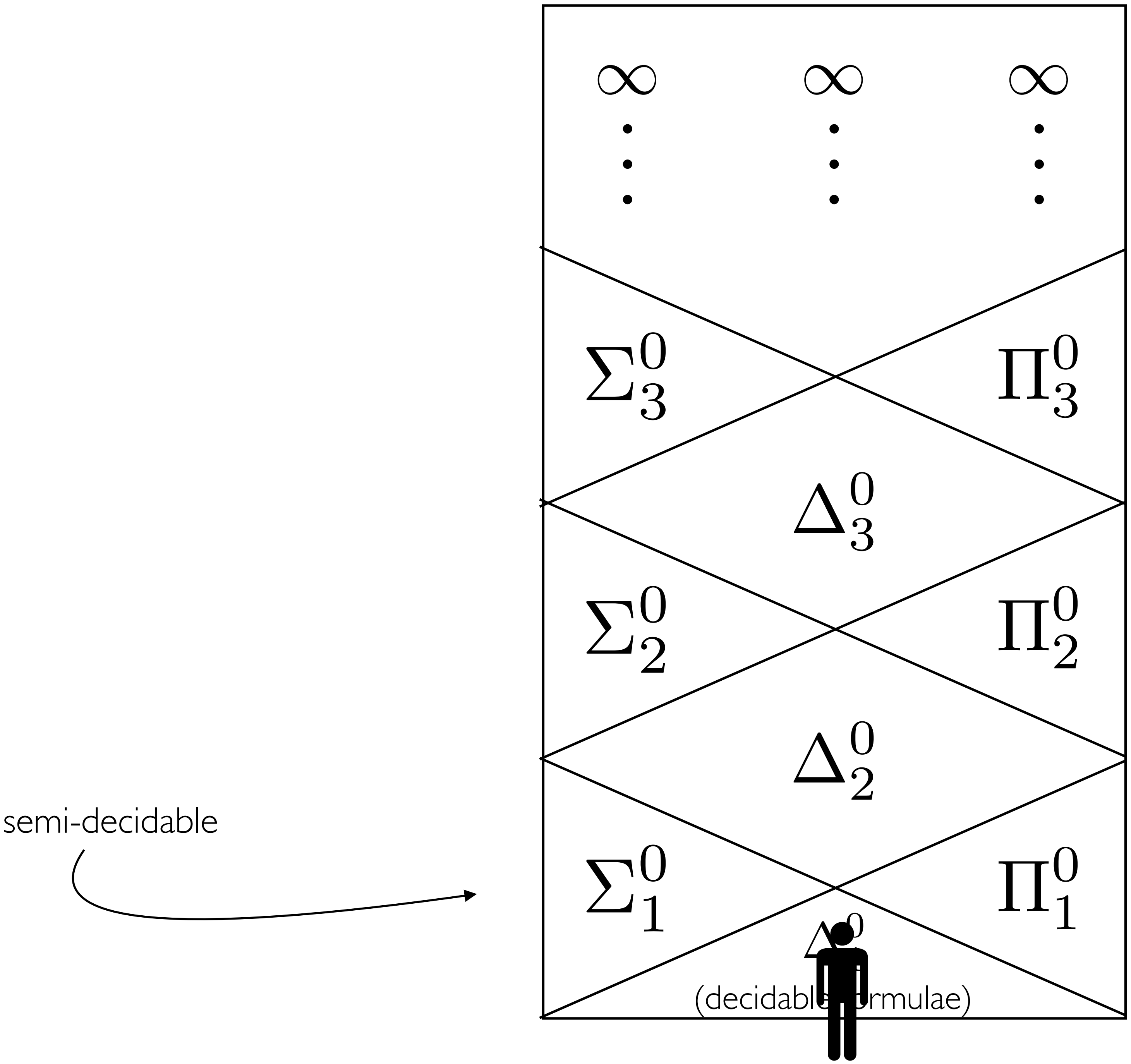
$$\Delta_1^0 = \Sigma_1^0 \cap \Pi_1^0$$

Arithmetic Hierarchy, Part I

Can you see the carryover from PH?

$$\mathbf{2SAMEFUNC} := \{\mathfrak{m}_1, \mathfrak{m}_2 : \forall u \forall v [\exists k (\langle \mathfrak{m}_1, u\rangle : v, k \leftrightarrow \exists k' (\langle \mathfrak{m}_2, u\rangle : v, k')]$$

$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)



$$\Delta_1^0 = \Sigma_1^0 \cap \Pi_1^0$$

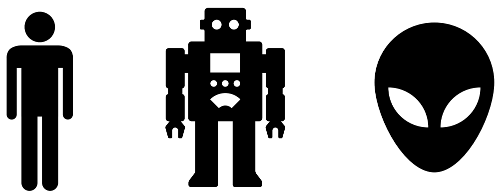
$$x \in \Sigma_i \text{ iff } \exists R\ \exists y_1 \forall y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \forall \text{ if } i \text{ even}; Q_i = \exists \text{ if } i \text{ odd})$$

$$x \in \Pi_i \text{ iff } \exists R\ \forall y_1 \exists y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \exists \text{ if } j \text{ even}; Q_i = \forall \text{ if } j \text{ odd})$$

Try your hand at classifying! …

From Kleene: The set to be classified, $\mathscr{K}$, consists of all those inputs to a given Turing machine $\mathfrak{m}$ that results in this machine halting after some number of steps.

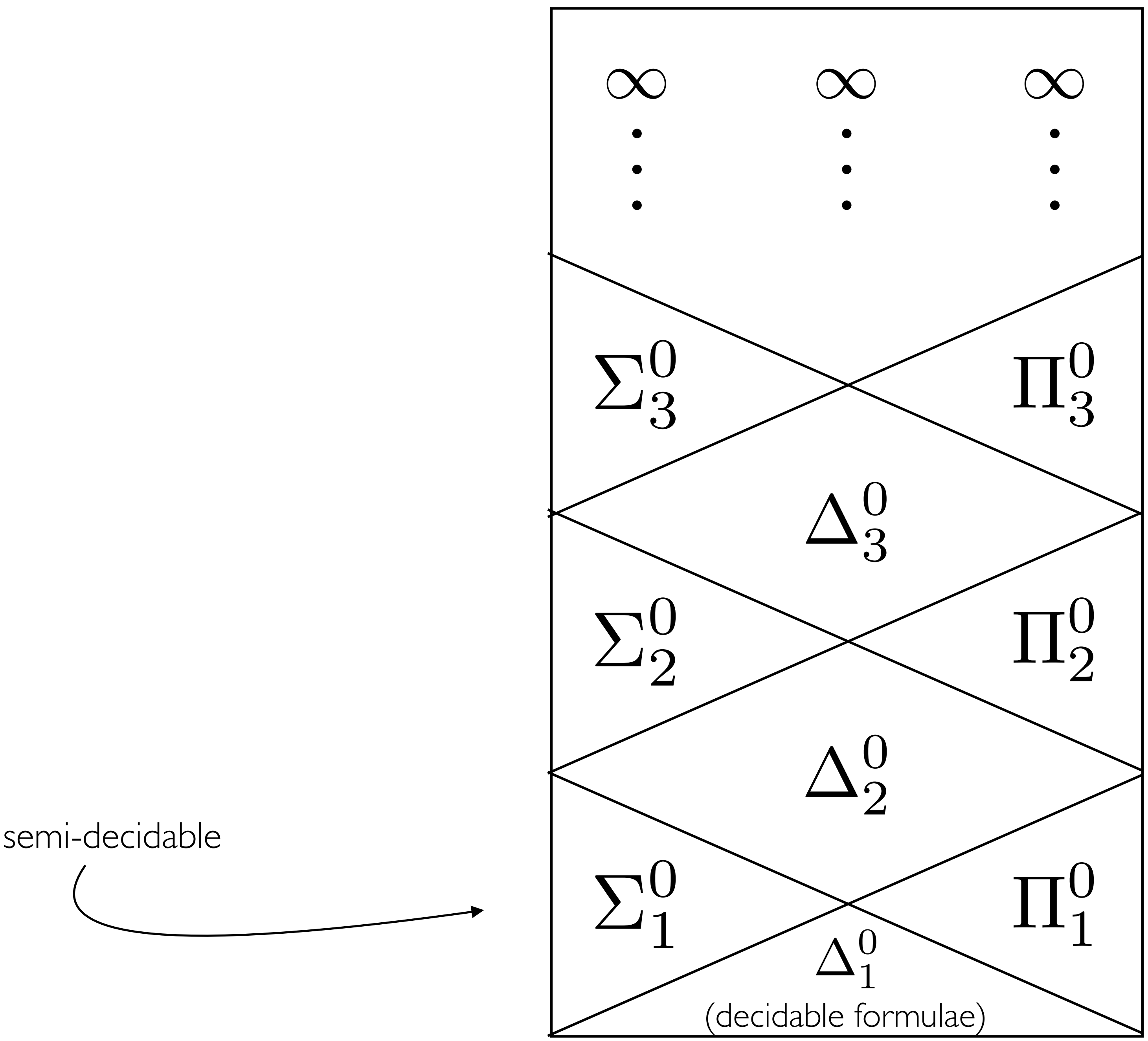
Arithmetic Hierarchy, Part I



Can you see the carryover from PH?

$$\textbf{2SAMEFUNC} := \{\mathfrak{m}_1, \mathfrak{m}_2 : \forall u \forall v [\exists k(\langle \mathfrak{m}_1, u\rangle : v, k \leftrightarrow \exists k'(\langle \mathfrak{m}_2, u\rangle : v, k')]$$

$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)



$$\Delta_1^0 = \Sigma_1^0 \cap \Pi_1^0$$

$$x \in \Sigma_i \text{ iff } \exists R\ \exists y_1 \forall y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \forall \text{ if } i \text{ even}; Q_i = \exists \text{ if } i \text{ odd})$$

$$x \in \Pi_i \text{ iff } \exists R\ \forall y_1 \exists y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \exists \text{ if } j \text{ even}; Q_i = \forall \text{ if } j \text{ odd})$$

Try your hand at classifying! …

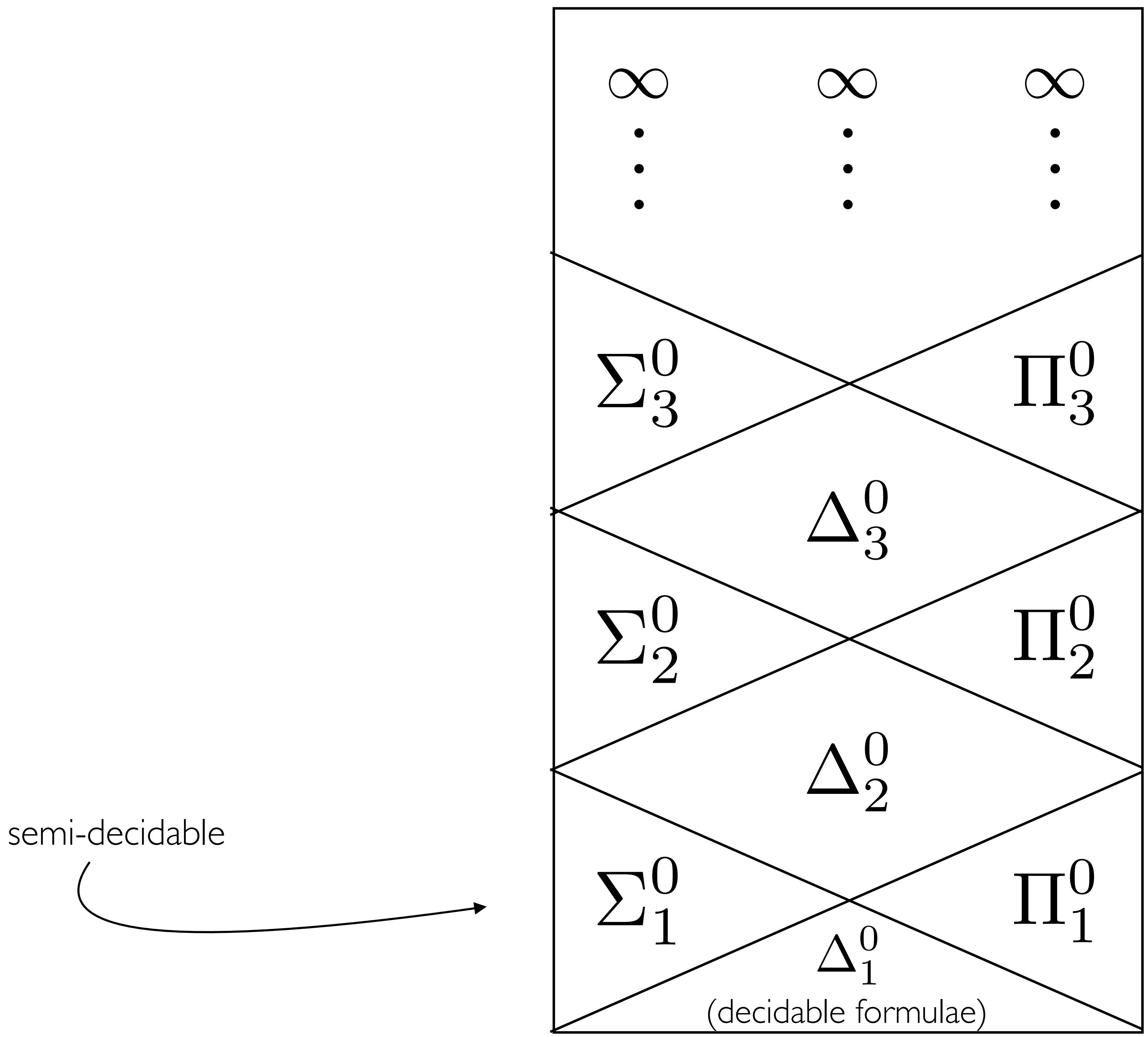
From Kleene: The set to be classified, $\mathscr{K}$, consists of all those inputs to a given Turing machine $\mathfrak{m}$ that results in this machine halting after some number of steps.

Arithmetic Hierarchy, Part I

Can you see the carryover from PH?

$$\mathbf{2SAMEFUNC} := \{\mathfrak{m}_1, \mathfrak{m}_2 : \forall u \forall v [\exists k (\langle \mathfrak{m}_1, u \rangle : v, k \leftrightarrow \exists k' (\langle \mathfrak{m}_2, u \rangle : v, k')]$$

$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)



$$x \in \Sigma_i \text{ iff } \exists R\ \exists y_1 \forall y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \forall \text{ if } i \text{ even}; Q_i = \exists \text{ if } i \text{ odd})$$

$$x \in \Pi_i \text{ iff } \exists R\ \forall y_1 \exists y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \exists \text{ if } j \text{ even}; Q_i = \forall \text{ if } j \text{ odd})$$

Try your hand at classifying! …

From Kleene: The set to be classified, $\mathscr{K}$, consists of all those inputs to a given Turing machine $\mathfrak{m}$ that results in this machine halting after some number of steps.

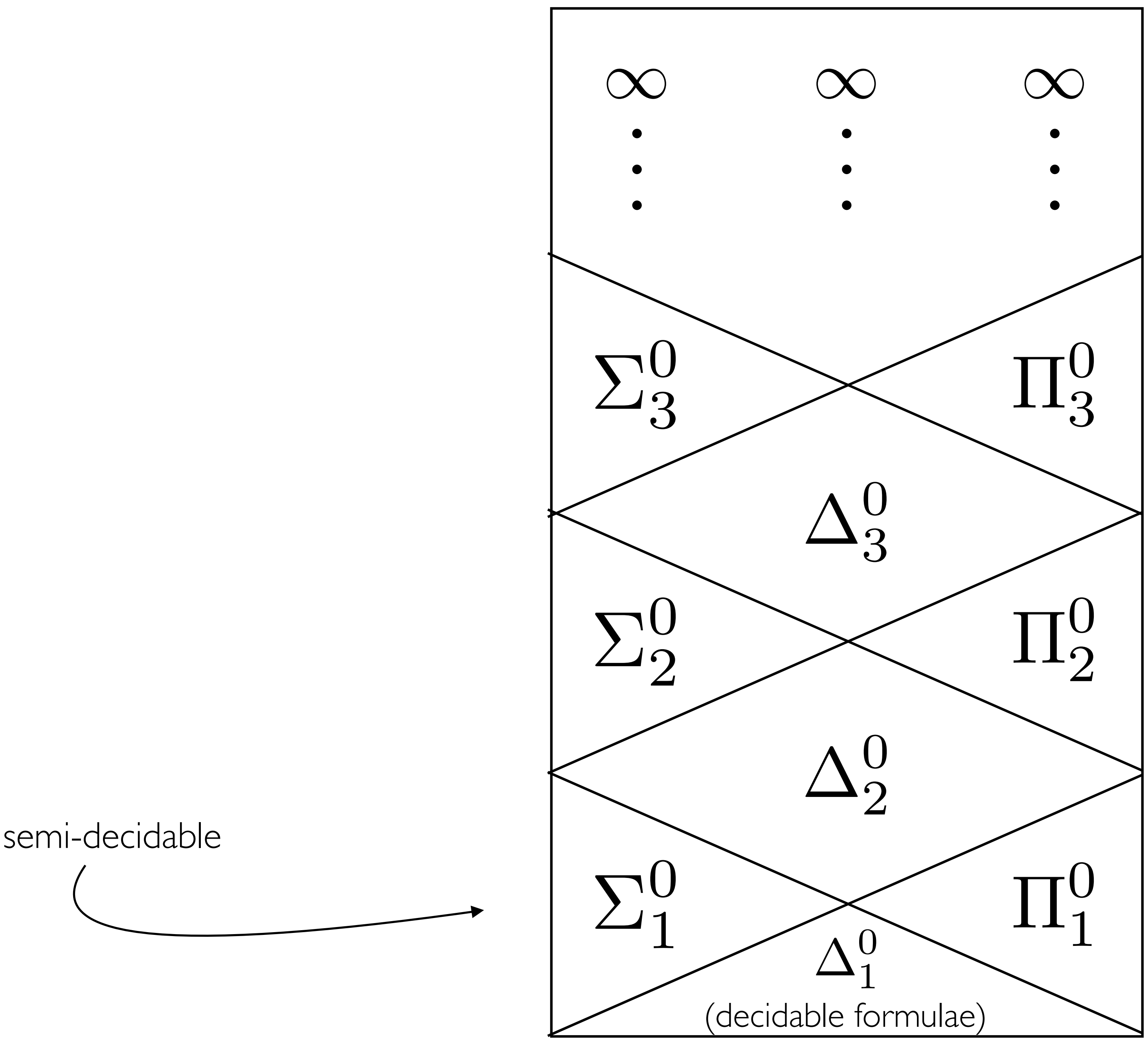
$$\Delta_1^0 = \Sigma_1^0 \cap \Pi_1^0$$

Arithmetic Hierarchy, Part I

Can you see the carryover from PH?

$$\mathbf{2SAMEFUNC} := \{\mathfrak{m}_1, \mathfrak{m}_2 : \forall u \forall v[\exists k(\langle \mathfrak{m}_1, u\rangle : v, k \leftrightarrow \exists k'(\langle \mathfrak{m}_2, u\rangle : v, k')]$$

$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)



$$\Delta_1^0 = \Sigma_1^0 \cap \Pi_1^0$$

$$x \in \Sigma_i \text{ iff } \exists R\ \exists y_1 \forall y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \forall \text{ if } i \text{ even}; Q_i = \exists \text{ if } i \text{ odd})$$

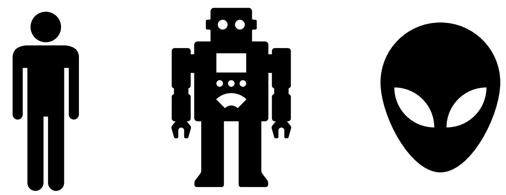
$$x \in \Pi_i \text{ iff } \exists R\ \forall y_1 \exists y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \exists \text{ if } j \text{ even}; Q_i = \forall \text{ if } j \text{ odd})$$

Try your hand at classifying! …

From Kleene: The set to be classified, $\mathscr{K}$, consists of all those inputs to a given Turing machine $\mathfrak{m}$ that results in this machine halting after some number of steps.

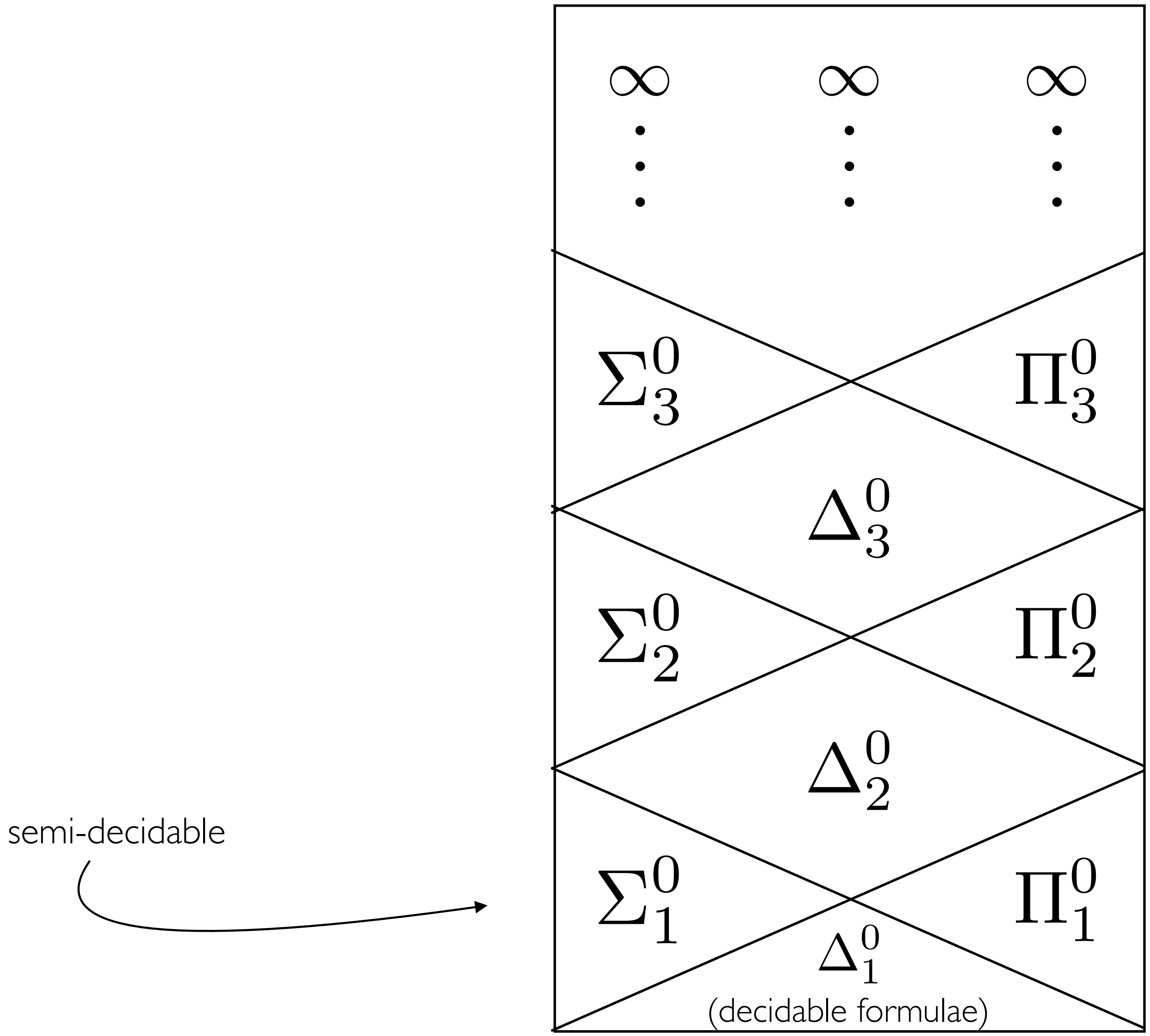
Arithmetic Hierarchy, Part I

Can you see the carryover from PH?



$$\mathbf{2SAMEFUNC} := \{\mathfrak{m}_1, \mathfrak{m}_2 : \forall u \forall v [\exists k (\langle \mathfrak{m}_1, u \rangle : v, k \leftrightarrow \exists k' (\langle \mathfrak{m}_2, u \rangle : v, k')]$$

$\mathcal{A}^r\mathcal{H}$ (Arithmetic Hierarchy)



$$\Delta_1^0 = \Sigma_1^0 \cap \Pi_1^0$$

$$x \in \Sigma_i \text{ iff } \exists R\ \exists y_1 \forall y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \forall \text{ if } i \text{ even}; Q_i = \exists \text{ if } i \text{ odd})$$

$$x \in \Pi_i \text{ iff } \exists R\ \forall y_1 \exists y_2 \cdots Q_i y_i R(x, y_1, y_2, \ldots, y_i)$$
$$(Q_i = \exists \text{ if } j \text{ even}; Q_i = \forall \text{ if } j \text{ odd})$$

Try your hand at classifying! …

From Kleene: The set to be classified, $\mathscr{K}$, consists of all those inputs to a given Turing machine $\mathfrak{m}$ that results in this machine halting after some number of steps.

The set to be classified is the set of all pairs of programs P_1 and P_2 s.t. both compute exactly the same functions.

Arithmetic Hierarchy, Part I

The Four Steps …

The PAID Problem (Level 1):

The PAID Problem (Level 1):

$\forall \mathtt{x} : \mathtt{Agents}$

The PAID Problem (Level 1):

$\forall \mathtt{x} : \mathtt{Agents}$

Powerful(x) + Autonomous(x) + Intelligent(x) => Dangerous(x)

The PAID Problem (Level 1):

$\forall \mathtt{x} : \mathtt{Agents}$

Powerful(x) + Autonomous(x) + Intelligent(x) => Dangerous(x)

↓

The PAID Problem (Level 1):

$\forall \texttt{x} : \texttt{Agents}$

Powerful(x) + Autonomous(x) + Intelligent(x) => Dangerous(x)

↓

$$u(\mathrm{AIA}_i(\pi_j)) > \tau^+ \in \mathbb{Z} \text{ or } \tau^- \in \mathbb{Z}$$

The PAID Problem (Level 1):

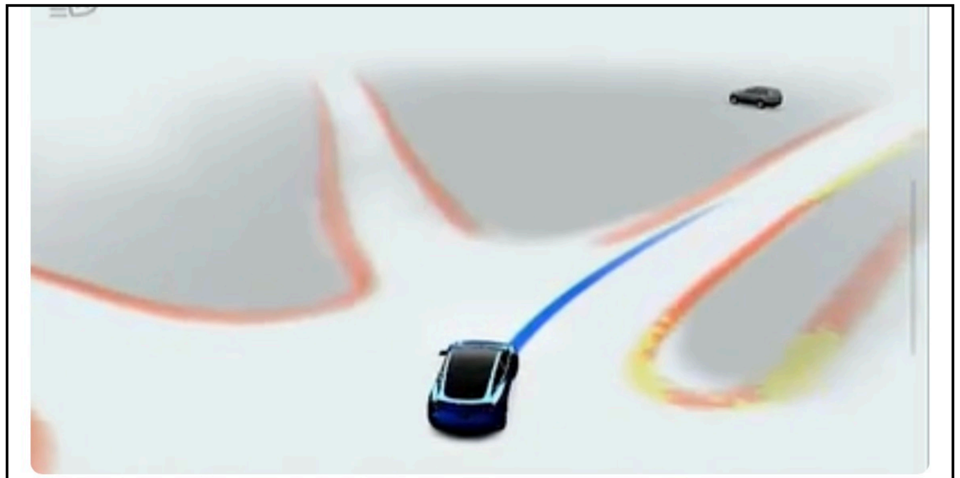
$\forall \mathtt{x} : \mathtt{Agents}$

Powerful(x) + Autonomous(x) + Intelligent(x) => Dangerous(x)

$\downarrow$ ... ...

$$u(\mathrm{AIA}_i(\pi_j)) > \tau^+ \in \mathbb{Z} \text{ or } \tau^- \in \mathbb{Z}$$

PAID Reflected in The News



Tesla dashboard in "Full Self Driving" mode

What Riding in a Self-Driving Tesla Tells Us About the Future of Autonomy

By Cade Metz, Ben Laffin, Hang Do Thi Duc and Ian Clontz. Cade and Ian spent six hours riding in a self-driving car in Jacksonville, Fla., to report this story.

Nov. 14, 2022

After releasing the new beta, Mr. Musk softened his claims about the immediate future of the technology. He now says that the technology will not be widely available until next year — and that regulators are unlikely to approve it for use without hands on the wheel. Autopilot still requires this oversight.

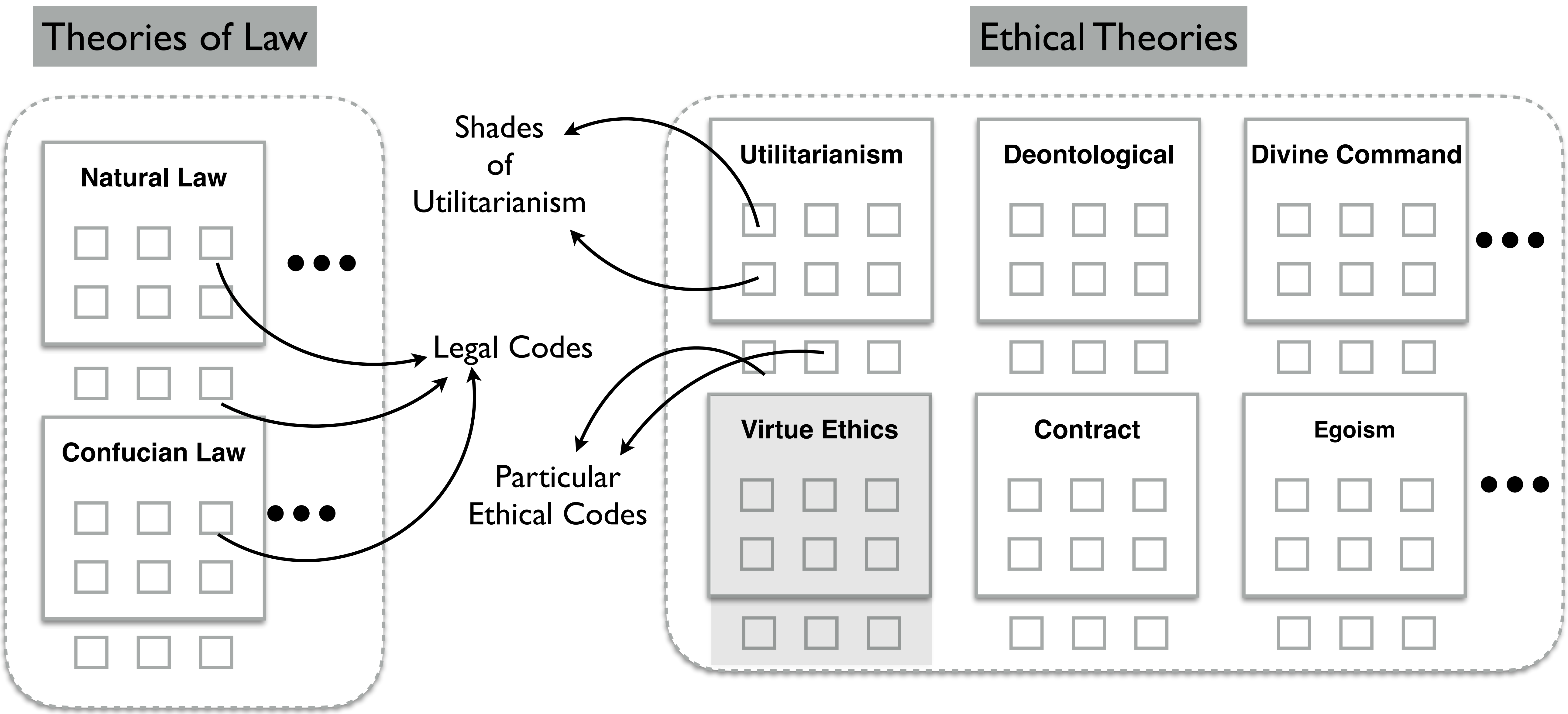
Federal regulators have spent the past several months investigating a series of crashes involving Autopilot, and they have not yet revealed the results. Safety experts worry that the arrival of Full Self-Driving will lead to more accidents.

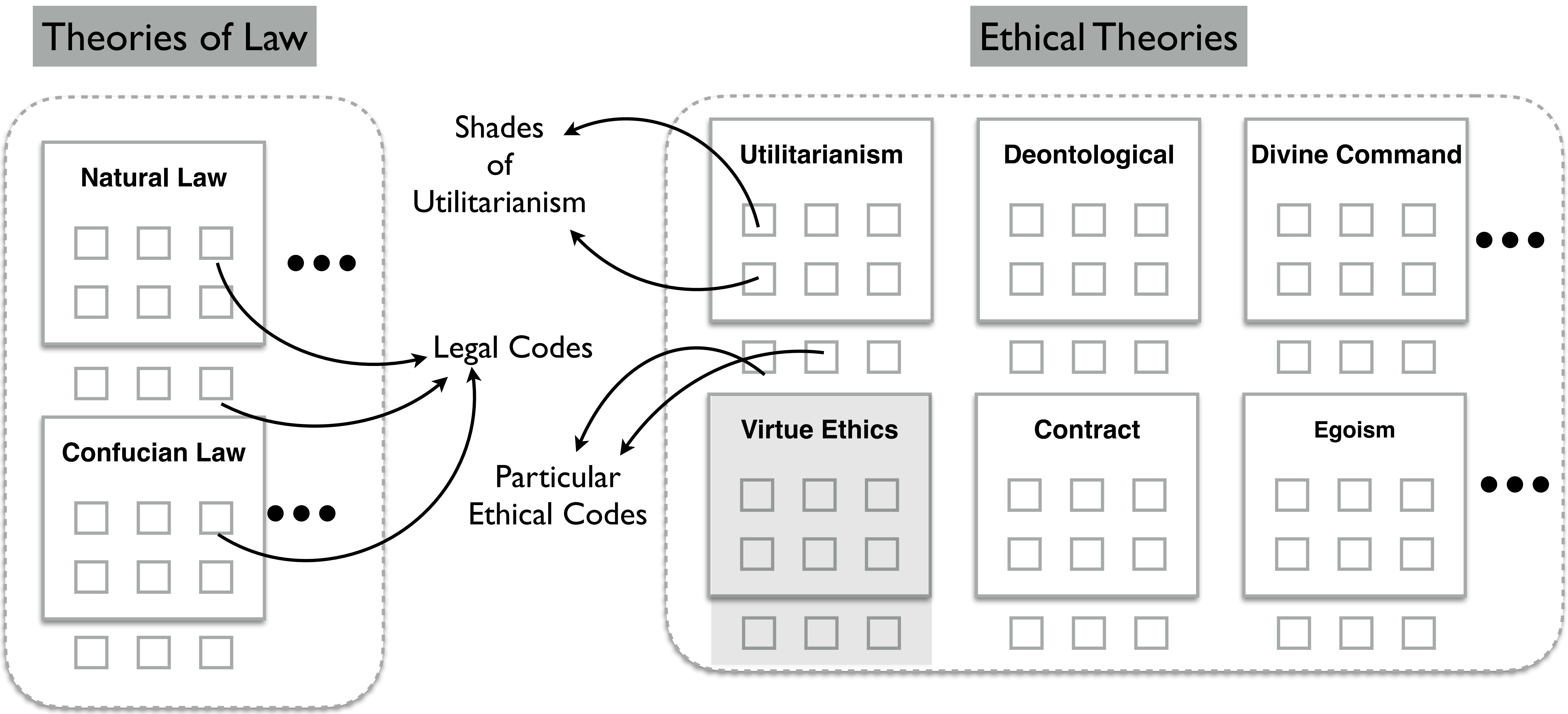
"It is inevitable," said Jake Fisher, senior director of Consumer Reports' Auto Test Center, who has used the technology. "The problem comes as this system gets better and people get complacent. It will still do the unexpected."

Cade Metz reported from Jacksonville, Fla. Video and photographs by Ian Clontz. Reporting and video production by Ben Laffin. Design and development by Hang Do Thi Duc.

Theories of Law
MAKING MORAL MACHINES
MAKING META-MORAL MACHINES
Natural Law
Confucian
Ethical Theories
"Shades" of Util.
Particular Ethical Codes
Legal Codes
Stripped Down Leibniz Hierarchy
Util
Deont.
D.C.
Virtue Eth.
Contract.
Egoism
Sub2
Sub1
F
M.N.
O
Sup1
Sup2
Pick the theory; pick the code; use the Leibnizian operators; engineer the robot
The machine picks the theory; picks the code . . . .
2016

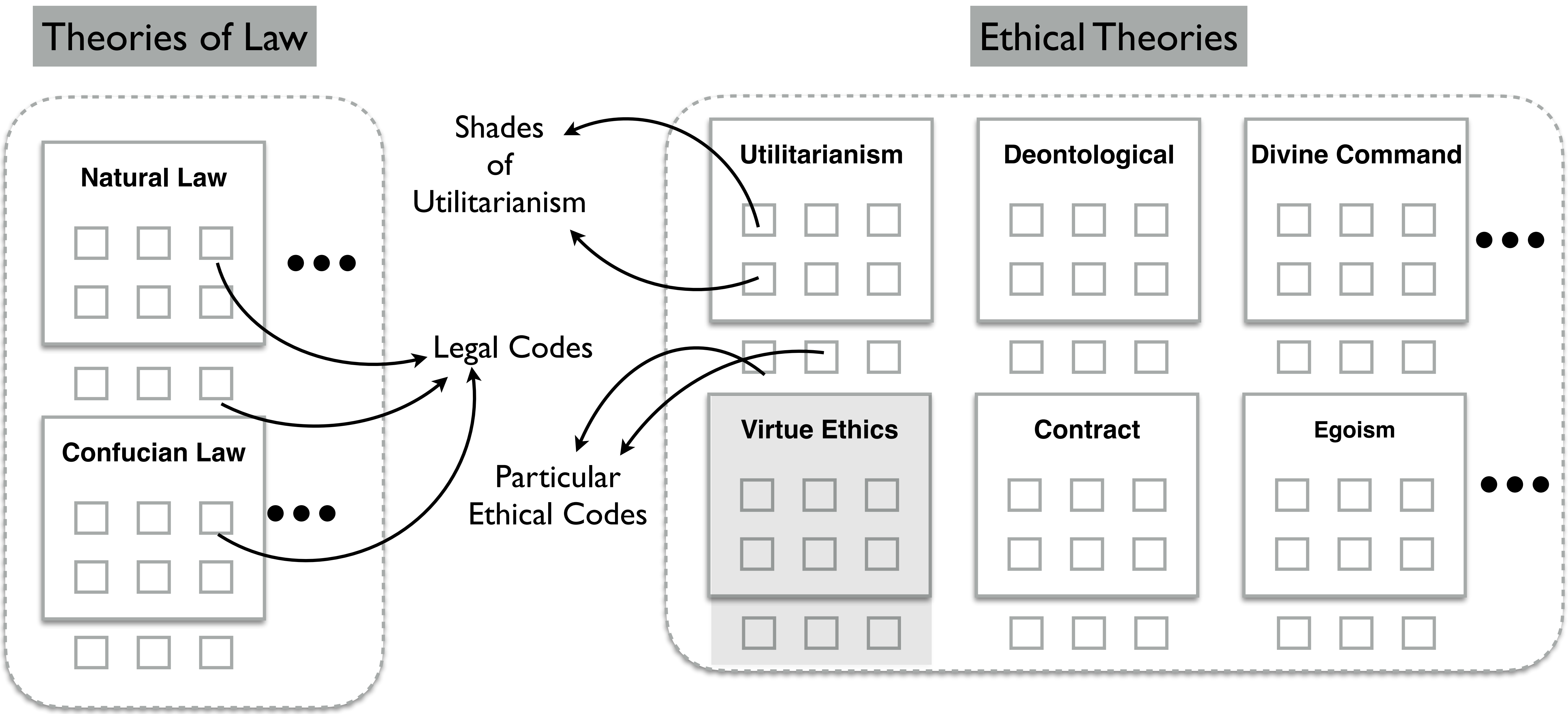
Making Morally X Machines; Only Logic Can Save Us





Theories of Law
Ethical Theories
Natural Law
Confucian Law
Shades of Utilitarianism
Legal Codes
Particular Ethical Codes
Utilitarianism
Deontological
Divine Command
Virtue Ethics
Contract
Egoism

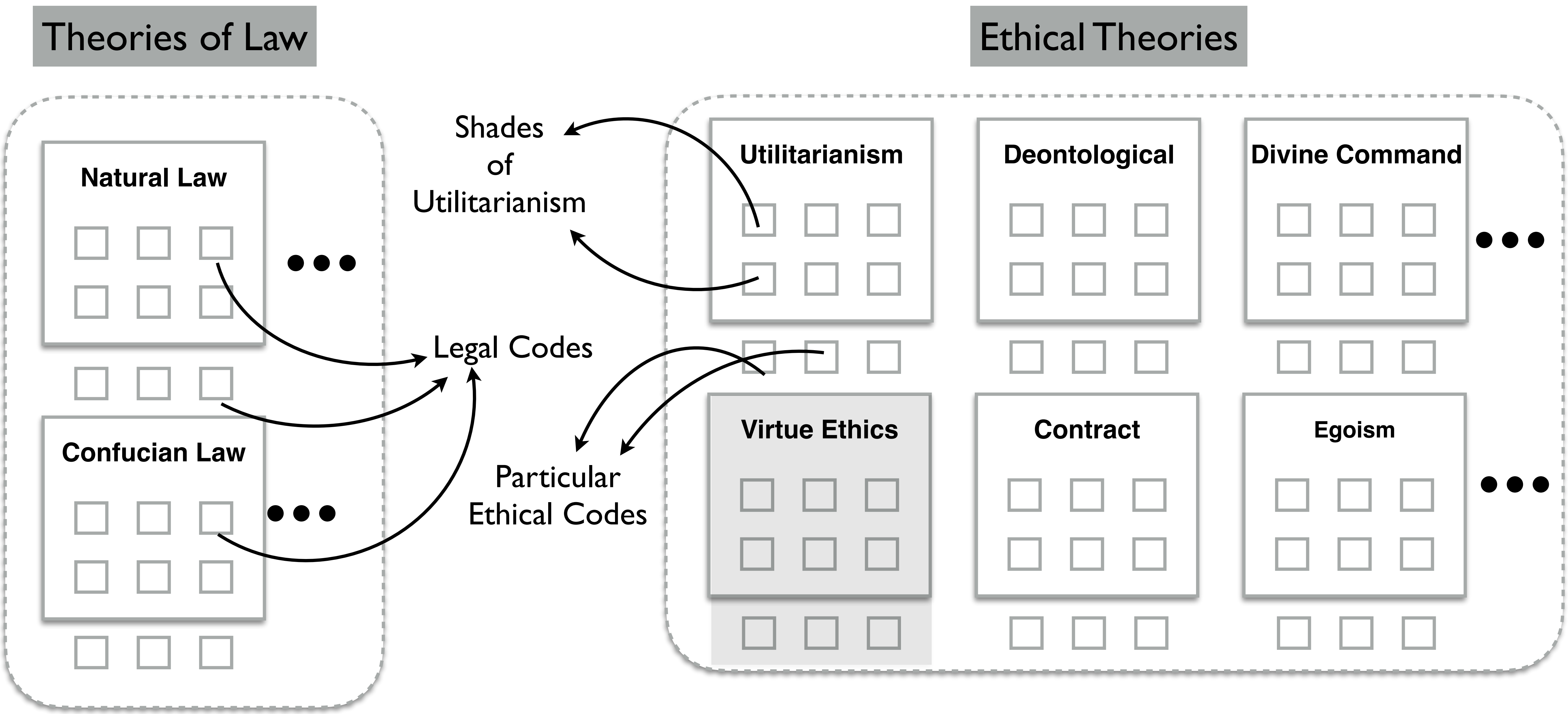
Making Morally X Machines; Only Logic Can Save Us



Step 1

1. Pick a theory
2. Pick a code
3. Run through EH.
4. Formalize in a Cognitive Calculus.

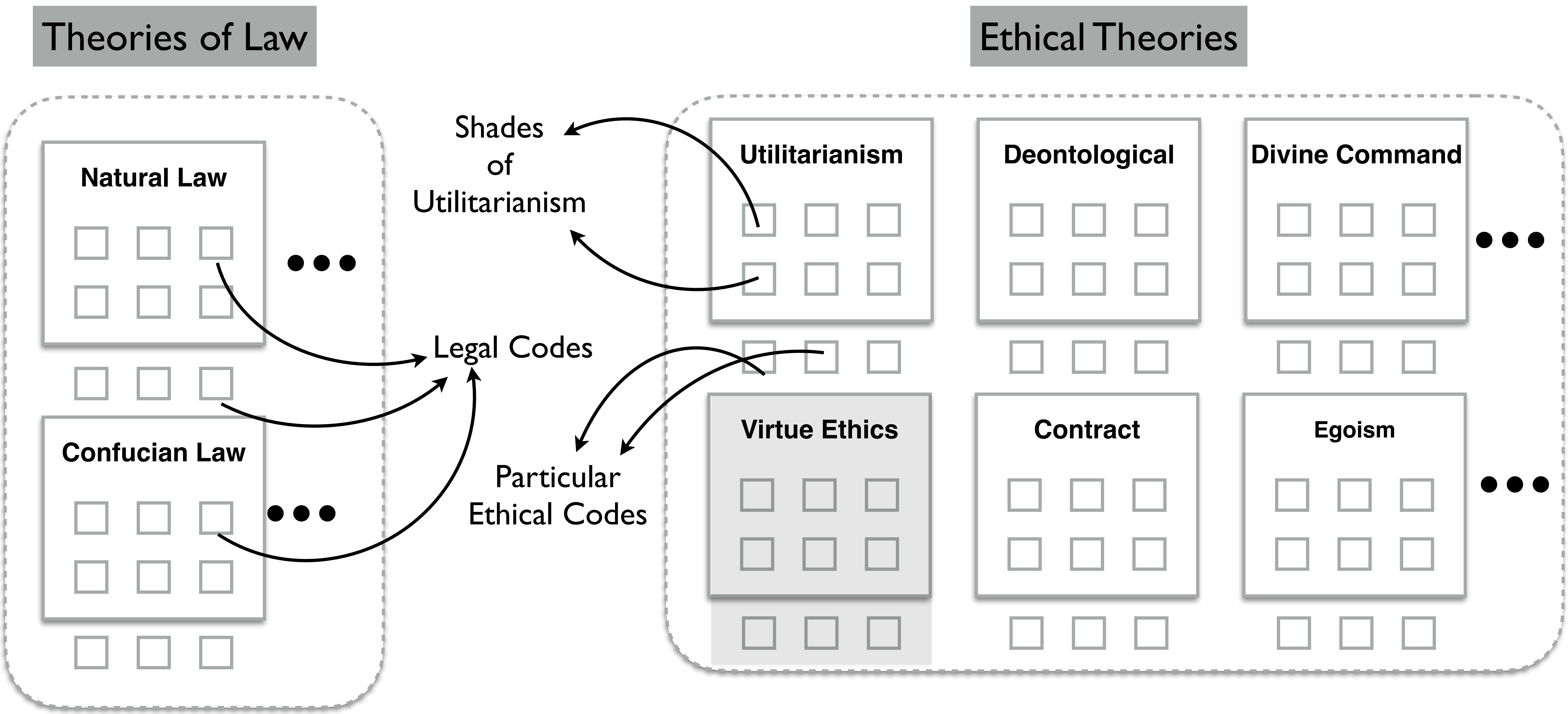
Making Morally X Machines; Only Logic Can Save Us



Step 1

1. Pick a theory
2. Pick a code
3. Run through EH.
4. Formalize in a Cognitive Calculus.

Making Morally X Machines; Only Logic Can Save Us



Step 1

1. Pick a theory
2. Pick a code
3. Run through EH.
4. Formalize in a Cognitive Calculus.

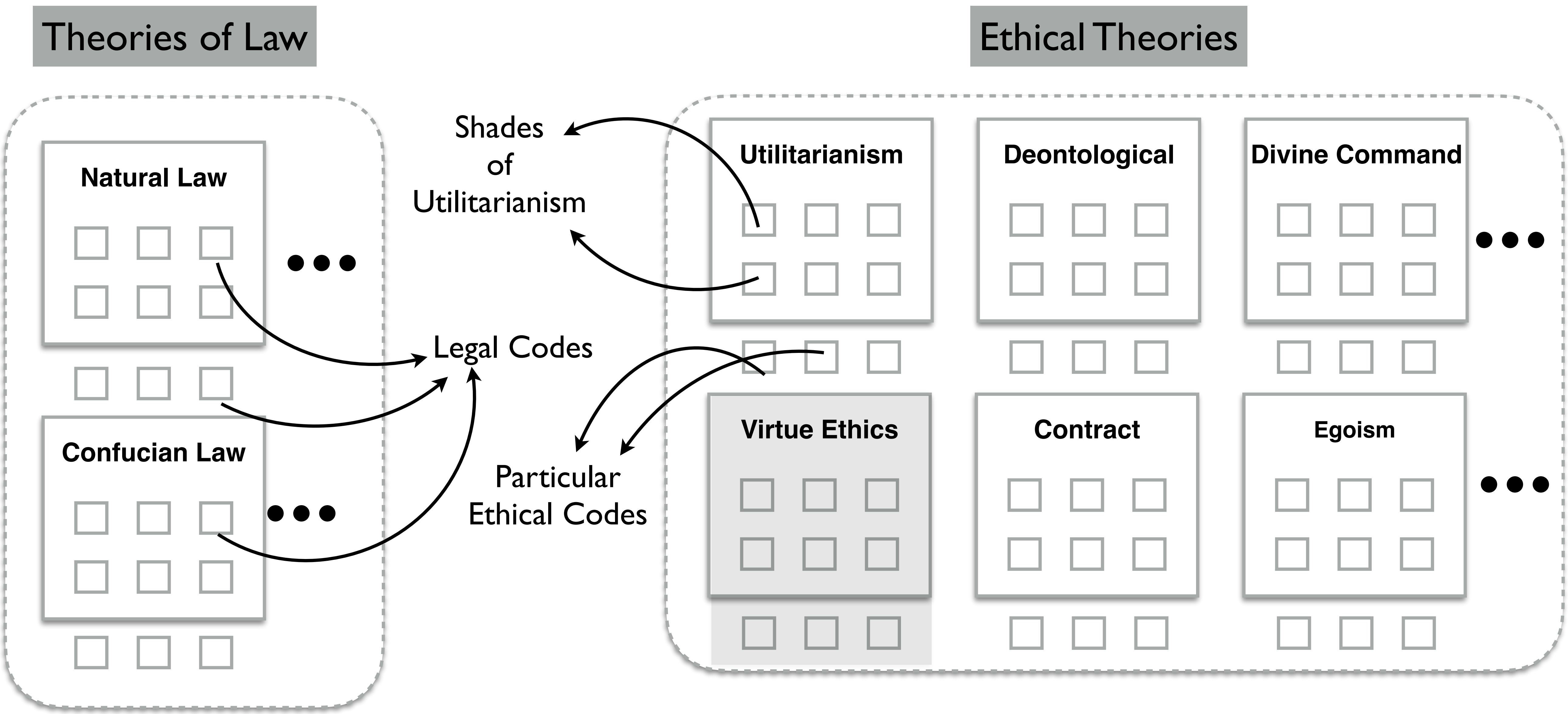
Step 2

Automate

Reasoners

Spectra

Making Morally X Machines; Only Logic Can Save Us



Step 1

1. Pick a theory
2. Pick a code
3. Run through EH.
4. Formalize in a Cognitive Calculus.

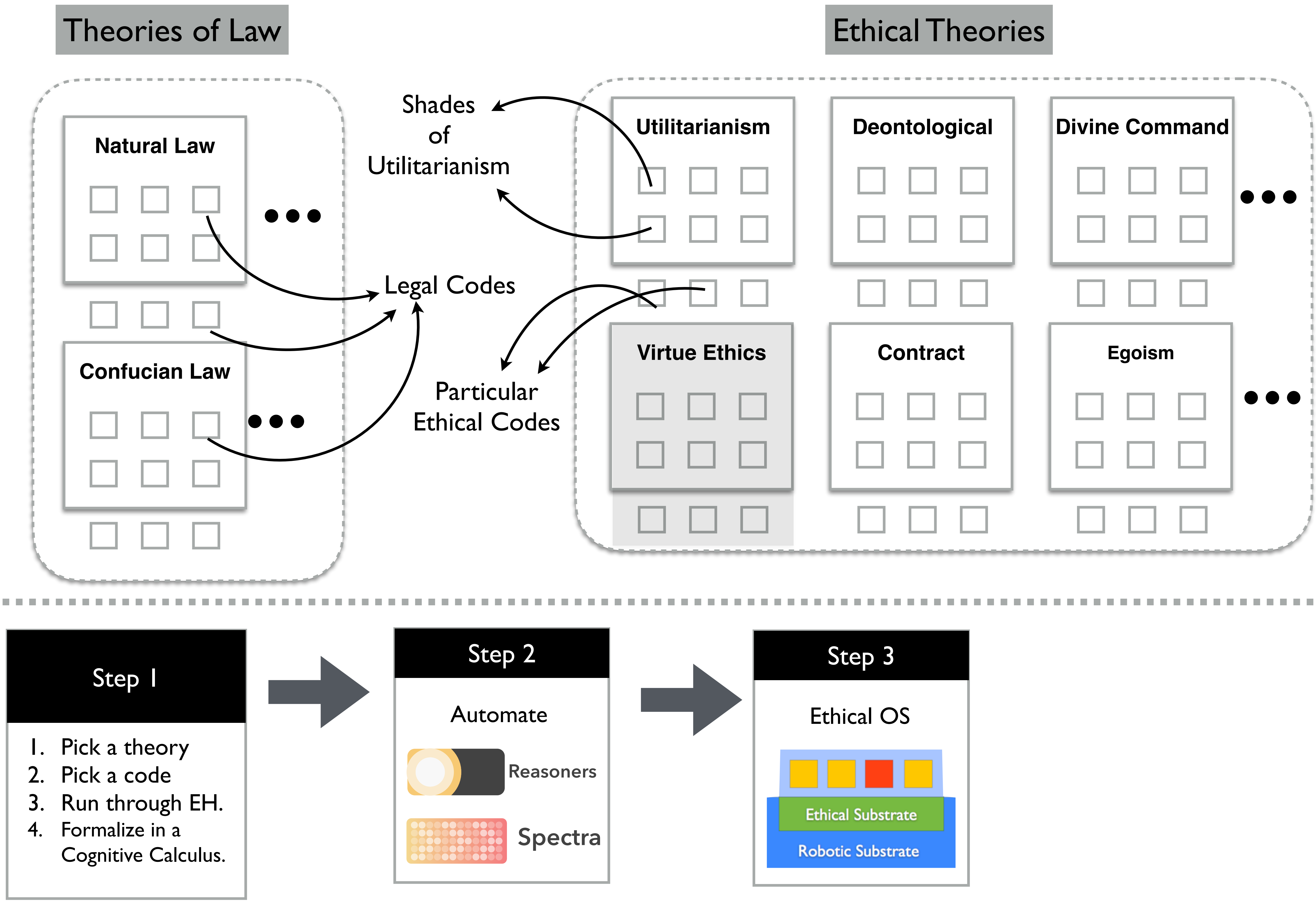
Step 2

Automate

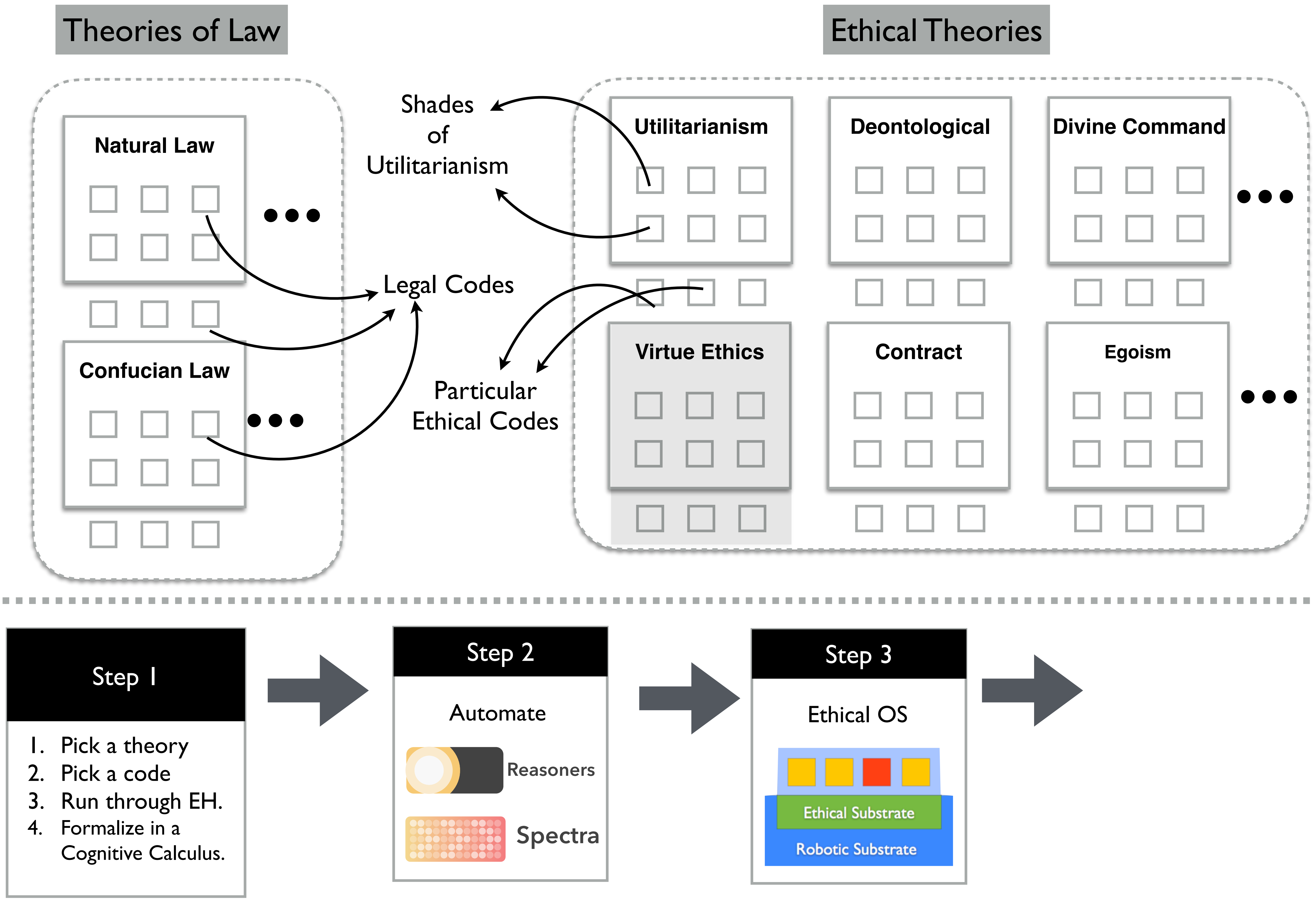
Reasoners

Spectra

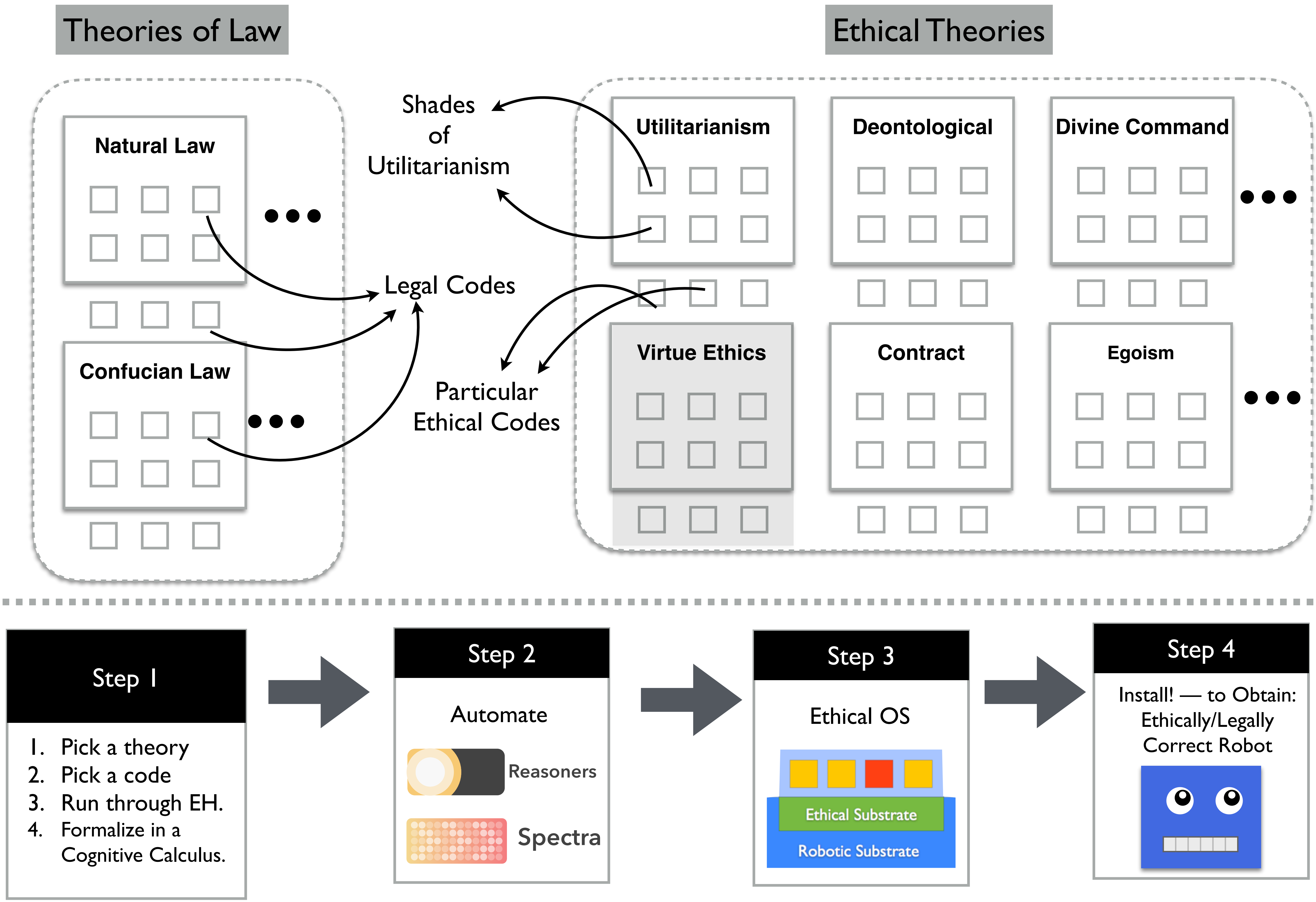
Making Morally X Machines; Only Logic Can Save Us



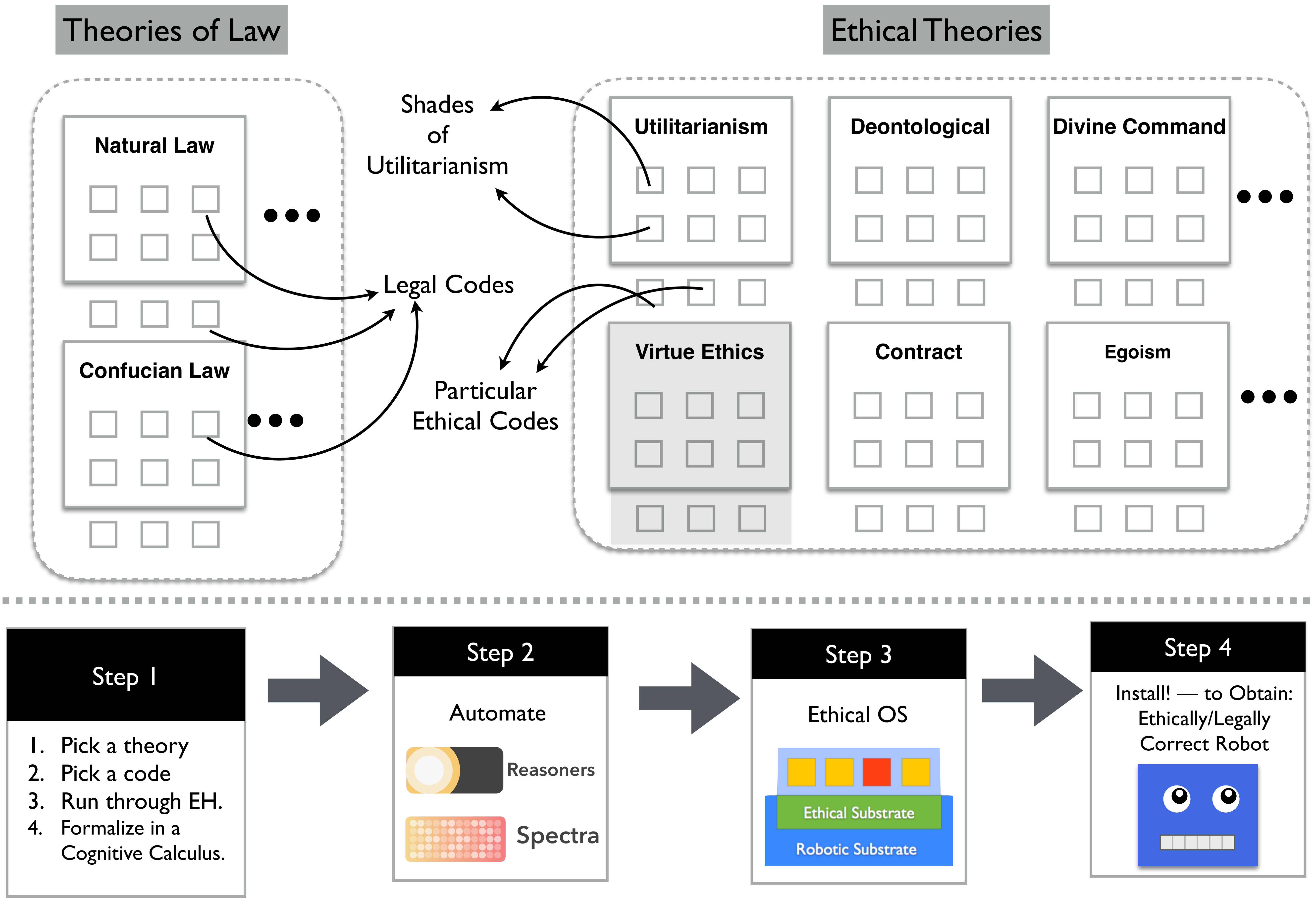
Making Morally X Machines; Only Logic Can Save Us



Making Morally X Machines; Only Logic Can Save Us

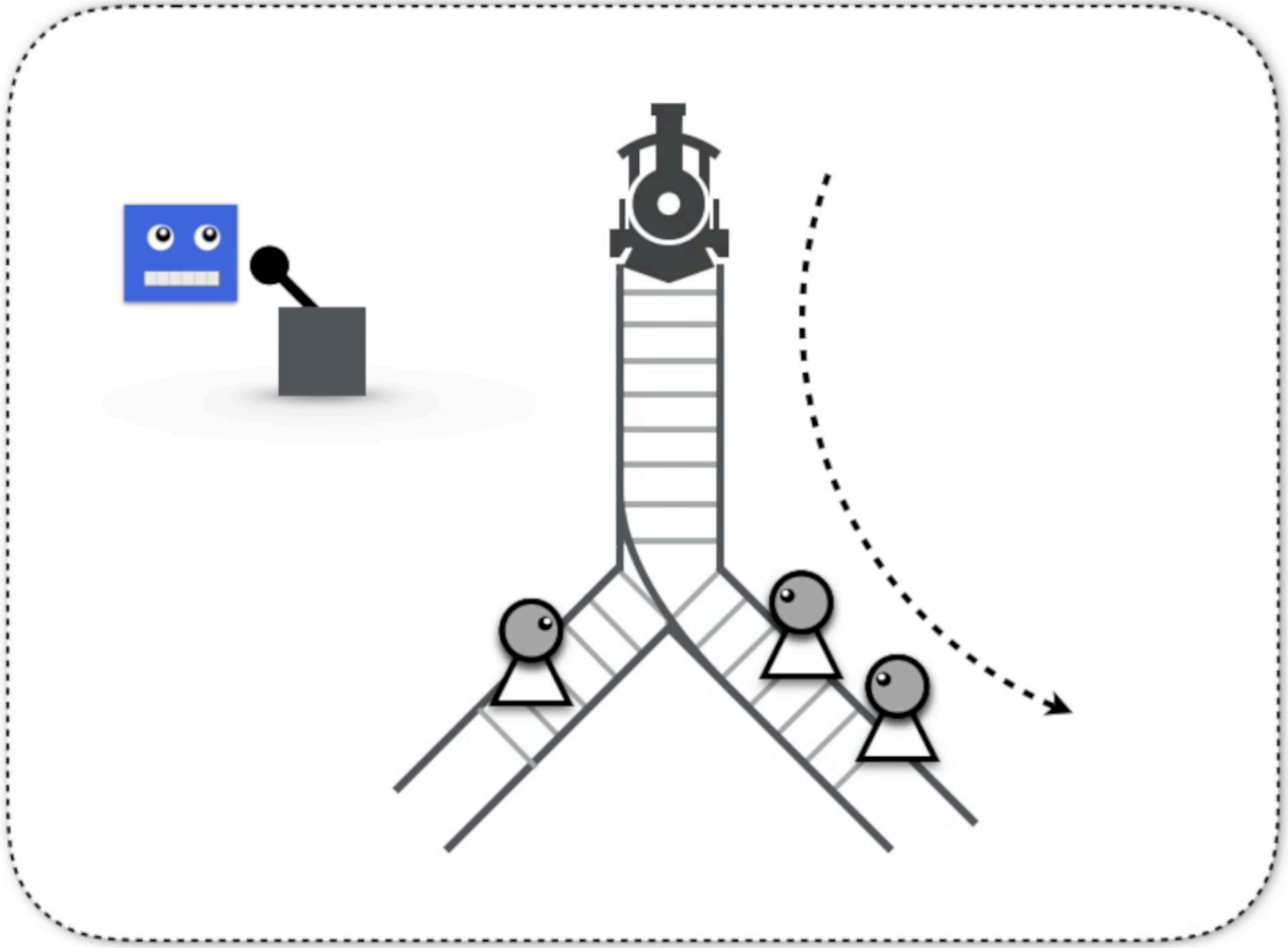


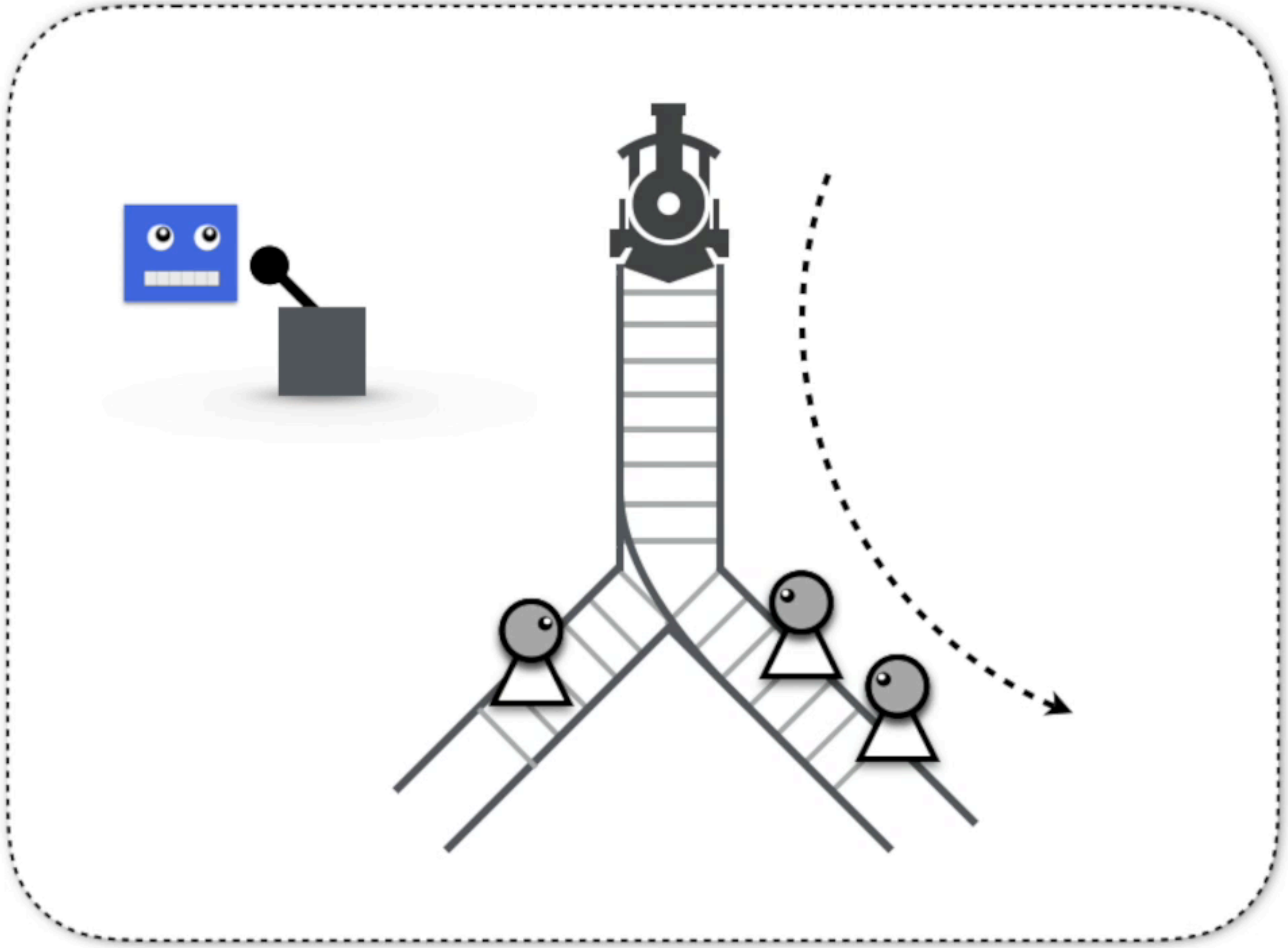
Making Morally X Machines; Only Logic Can Save Us



e.g. "Toward the Engineering of Virtuous Robots" Naveen, Selmer et al.

DDE* & DTE





https://www.ijcai.org/Proceedings/2017/0658.pdf

Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)

On Automating the Doctrine of Double Effect

Naveen Sundar Govindarajulu and **Selmer Bringsjord**
Rensselaer Polytechnic Institute, Troy, NY
{naveensundarg,selmer.bringsjord}@gmail.com

Abstract

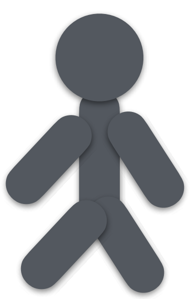
The **doctrine of double effect** ($\mathcal{DDE}$) is a long-studied ethical principle that governs when actions that have both positive and negative effects are to be allowed. The goal in this paper is to automate $\mathcal{DDE}$. We briefly present $\mathcal{DDE}$, and use a first-order modal logic, the **deontic cognitive event calculus**, as our framework to formalize the doctrine.

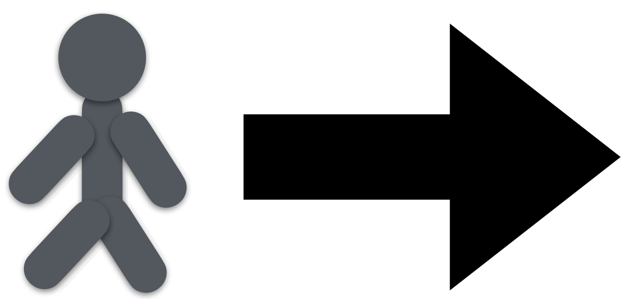
— provided that 1) the harmful effects are not intended; 2) the harmful effects are not used to achieve the beneficial effects (harm is merely a *side*-effect); and 3) benefits outweigh the harm by a significant amount. What distinguishes $\mathcal{DDE}$ from, say, naïve forms of consequentialism in ethics (e.g. act utilitarianism, which holds that an action is obligatory for an autonomous agent if and only if it produces the most utility among all competing actions) is that purely mental intentions in and of themselves, independent of conse-

4 Informal $\mathcal{DDE}$

We now informally but rigorously present $\mathcal{DDE}$. We assume we have at hand an ethical hierarchy of actions as in the deontological case (e.g. forbidden, neutral, obligatory); see [Bringsjord, 2017]. We also assume that we have a utility or goodness function for states of the world or effects as in the consequentialist case. For an autonomous agent a, an action α in a situation σ at time t is said to be $\mathcal{DDE}$-compliant *iff*:

- $\mathbf{C}_1$ the action is not forbidden (where we assume an ethical hierarchy such as the one given by Bringsjord [2017], and require that the action be neutral or above neutral in such a hierarchy);
- $\mathbf{C}_2$ The net utility or goodness of the action is greater than some positive amount γ;
- $\mathbf{C}_{3a}$ the agent performing the action intends only the good effects;
- $\mathbf{C}_{3b}$ the agent does not intend any of the bad effects;
- $\mathbf{C}_4$ the bad effects are not used as a means to obtain the good effects; and
- $\mathbf{C}_5$ if there are bad effects, the agent would rather the situation be different and the agent not have to perform the action. That is, the action is unavoidable.





Formal Conditions for $\mathcal{DDE}$

$\mathbf{F_1}$ α carried out at t is not forbidden. That is:

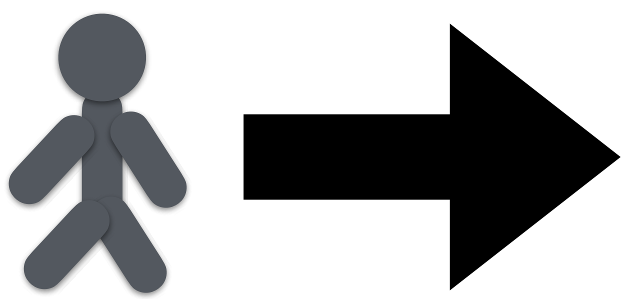
$$\Gamma \nvdash \neg \mathbf{O}\Big(a, t, \sigma, \neg happens\big(action(a, \alpha), t\big)\Big)$$

$\mathbf{F_2}$ The net utility is greater than a given positive real γ:

$$\Gamma \vdash \sum_{y=t+1}^{H} \left(\sum_{f \in \alpha_I^{a,t}} \mu(f, y) - \sum_{f \in \alpha_T^{a,t}} \mu(f, y) \right) > \gamma$$

$\mathbf{F_{3a}}$ The agent a intends at least one good effect. ($\mathbf{F_2}$ should still hold after removing all other good effects.) There is at least one fluent f_g in $\alpha_I^{a,t}$ with $\mu(f_g, y) > 0$, or f_b in $\alpha_T^{a,t}$ with $\mu(f_b, y) < 0$, and some y with $t < y \leq H$ such that the following holds:

$$\Gamma \vdash \begin{pmatrix} \exists f_g \in \alpha_I^{a,t}\ \mathbf{I}\Big(a, t, Holds\big(f_g, y\big)\Big) \\ \vee \\ \exists f_b \in \alpha_T^{a,t}\ \mathbf{I}\Big(a, t, \neg Holds\big(f_b, y\big)\Big) \end{pmatrix}$$



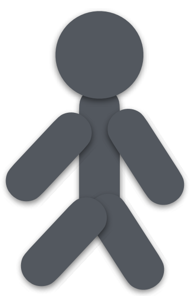
$\mathbf{F_{3b}}$ The agent a does not intend any bad effect. For all fluents f_b in $\alpha_I^{a,t}$ with $\mu(f_b, y) < 0$, or f_g in $\alpha_T^{a,t}$ with $\mu(f_g, y) > 0$, and for all y such that $t < y \leq H$ the following holds:

$$\Gamma \nvdash \mathbf{I}\Big(a, t, Holds\big(f_b, y\big)\Big) \text{ and}$$

$$\Gamma \nvdash \mathbf{I}\Big(a, t, \neg Holds\big(f_g, y\big)\Big)$$

$\mathbf{F_4}$ The harmful effects don't cause the good effects. Four permutations, paralleling the definition of $\rhd$ above, hold here. One such permutation is shown below. For any bad fluent f_b holding at t_1, and any good fluent f_g holding at some t_2, such that $t < t_1, t_2 \leq H$, the following holds:

$$\Gamma \vdash \neg \rhd \Big(Holds\big(f_b, t_1\big), Holds\big(f_g, t_2\big)\Big)$$



Formal Conditions for $\mathcal{DDE}$

F$_1$ α carried out at t is not forbidden. That is:

$$\Gamma \nvdash \neg \mathbf{O}\Big(a, t, \sigma, \neg happens\big(action(a, \alpha), t\big)\Big)$$

F$_2$ The net utility is greater than a given positive real γ:

$$\Gamma \vdash \sum_{y=t+1}^{H} \left(\sum_{f \in \alpha_I^{a,t}} \mu(f, y) - \sum_{f \in \alpha_T^{a,t}} \mu(f, y) \right) > \gamma$$

F$_{3a}$ The agent a intends at least one good effect. (**F$_2$** should still hold after removing all other good effects.) There is at least one fluent f_g in $\alpha_I^{a,t}$ with $\mu(f_g, y) > 0$, or f_b in $\alpha_T^{a,t}$ with $\mu(f_b, y) < 0$, and some y with $t < y \leq H$ such that the following holds:

$$\Gamma \vdash \left(\begin{array}{c} \exists f_g \in \alpha_I^{a,t}\ \mathbf{I}\Big(a, t, Holds\big(f_g, y\big)\Big) \\ \vee \\ \exists f_b \in \alpha_T^{a,t}\ \mathbf{I}\Big(a, t, \neg Holds\big(f_b, y\big)\Big) \end{array} \right)$$

F$_{3b}$ The agent a does not intend any bad effect. For all fluents f_b in $\alpha_I^{a,t}$ with $\mu(f_b, y) < 0$, or f_g in $\alpha_T^{a,t}$ with $\mu(f_g, y) > 0$, and for all y such that $t < y \leq H$ the following holds:

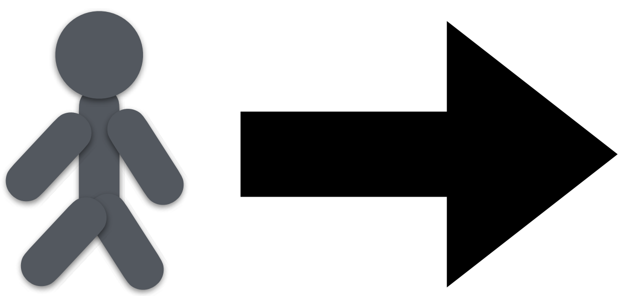
$$\Gamma \nvdash \mathbf{I}\Big(a, t, Holds\big(f_b, y\big)\Big) \text{ and}$$

$$\Gamma \nvdash \mathbf{I}\Big(a, t, \neg Holds\big(f_g, y\big)\Big)$$

F$_4$ The harmful effects don't cause the good effects. Four permutations, paralleling the definition of $\rhd$ above, hold here. One such permutation is shown below. For any bad fluent f_b holding at t_1, and any good fluent f_g holding at some t_2, such that $t < t_1, t_2 \leq H$, the following holds:

$$\Gamma \vdash \neg \rhd \Big(Holds\big(f_b, t_1\big), Holds\big(f_g, t_2\big)\Big)$$

$\mathbb{P}_{\text{DDE}_1}$ + ShadowProver



Formal Conditions for $\mathcal{DDE}$

F_1 α carried out at t is not forbidden. That is:

$$\Gamma \nvdash \neg \mathbf{O}\Big(a, t, \sigma, \neg happens\big(action(a,\alpha), t\big)\Big)$$

F_2 The net utility is greater than a given positive real γ:

$$\Gamma \vdash \sum_{y=t+1}^{H} \left(\sum_{f \in \alpha_I^{a,t}} \mu(f,y) - \sum_{f \in \alpha_T^{a,t}} \mu(f,y) \right) > \gamma$$

F_{3a} The agent a intends at least one good effect. ($\mathbf{F}_2$ should still hold after removing all other good effects.) There is at least one fluent f_g in $\alpha_I^{a,t}$ with $\mu(f_g, y) > 0$, or f_b in $\alpha_T^{a,t}$ with $\mu(f_b, y) < 0$, and some y with $t < y \leq H$ such that the following holds:

$$\Gamma \vdash \begin{pmatrix} \exists f_g \in \alpha_I^{a,t}\ \mathbf{I}\Big(a, t, Holds\big(f_g, y\big)\Big) \\ \vee \\ \exists f_b \in \alpha_T^{a,t}\ \mathbf{I}\Big(a, t, \neg Holds\big(f_b, y\big)\Big) \end{pmatrix}$$

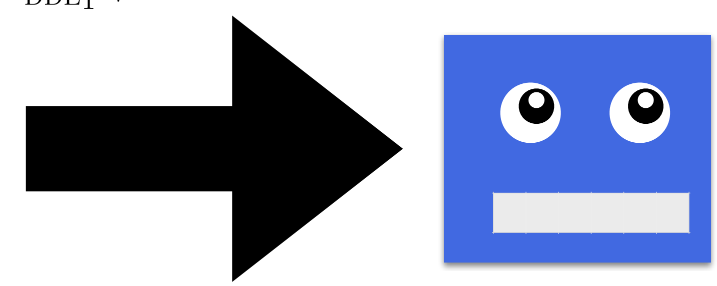
F_{3b} The agent a does not intend any bad effect. For all fluents f_b in $\alpha_I^{a,t}$ with $\mu(f_b, y) < 0$, or f_g in $\alpha_T^{a,t}$ with $\mu(f_g, y) > 0$, and for all y such that $t < y \leq H$ the following holds:

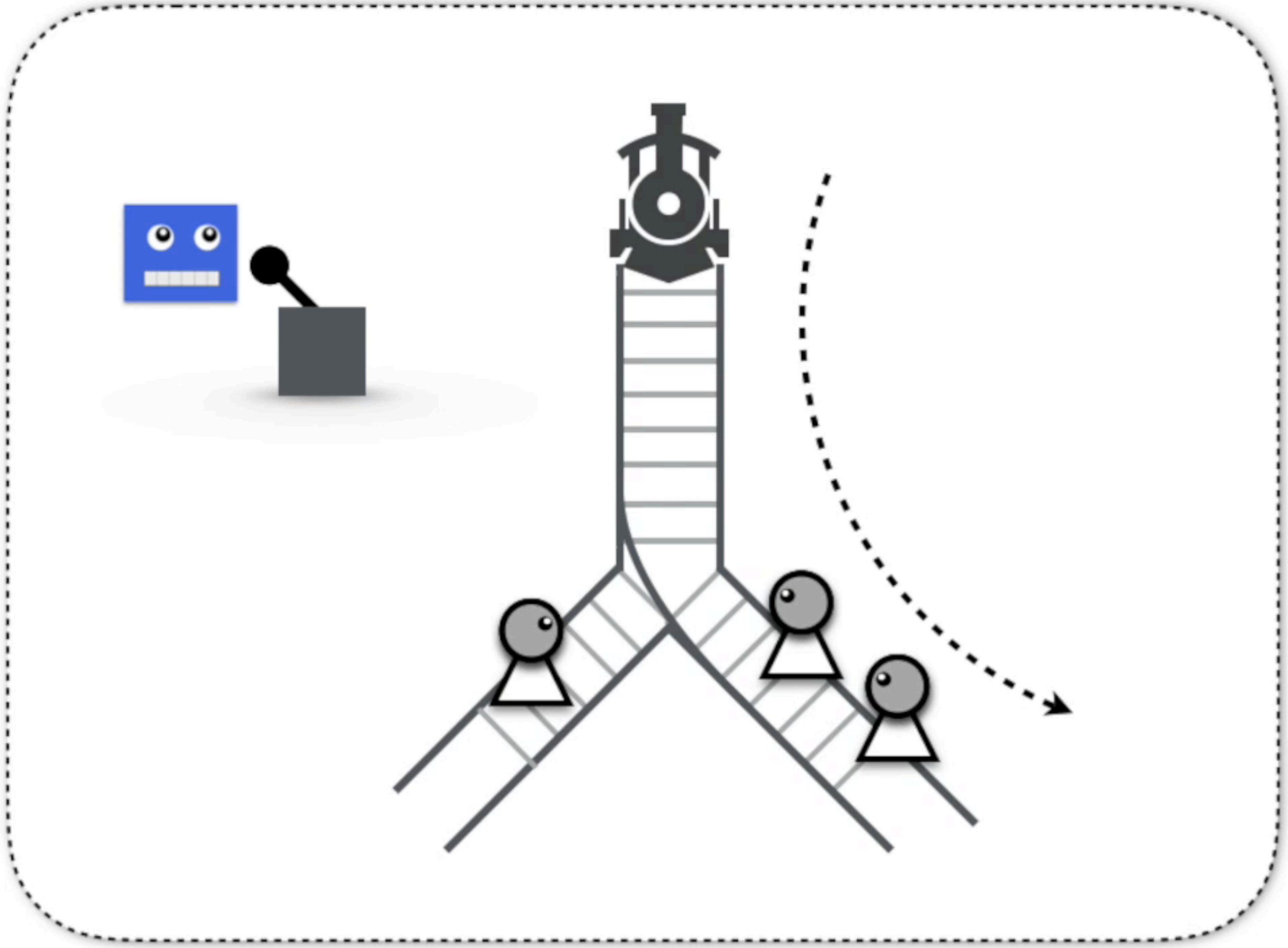
$$\Gamma \nvdash \mathbf{I}\Big(a, t, Holds\big(f_b, y\big)\Big) \text{ and}$$
$$\Gamma \nvdash \mathbf{I}\Big(a, t, \neg Holds\big(f_g, y\big)\Big)$$

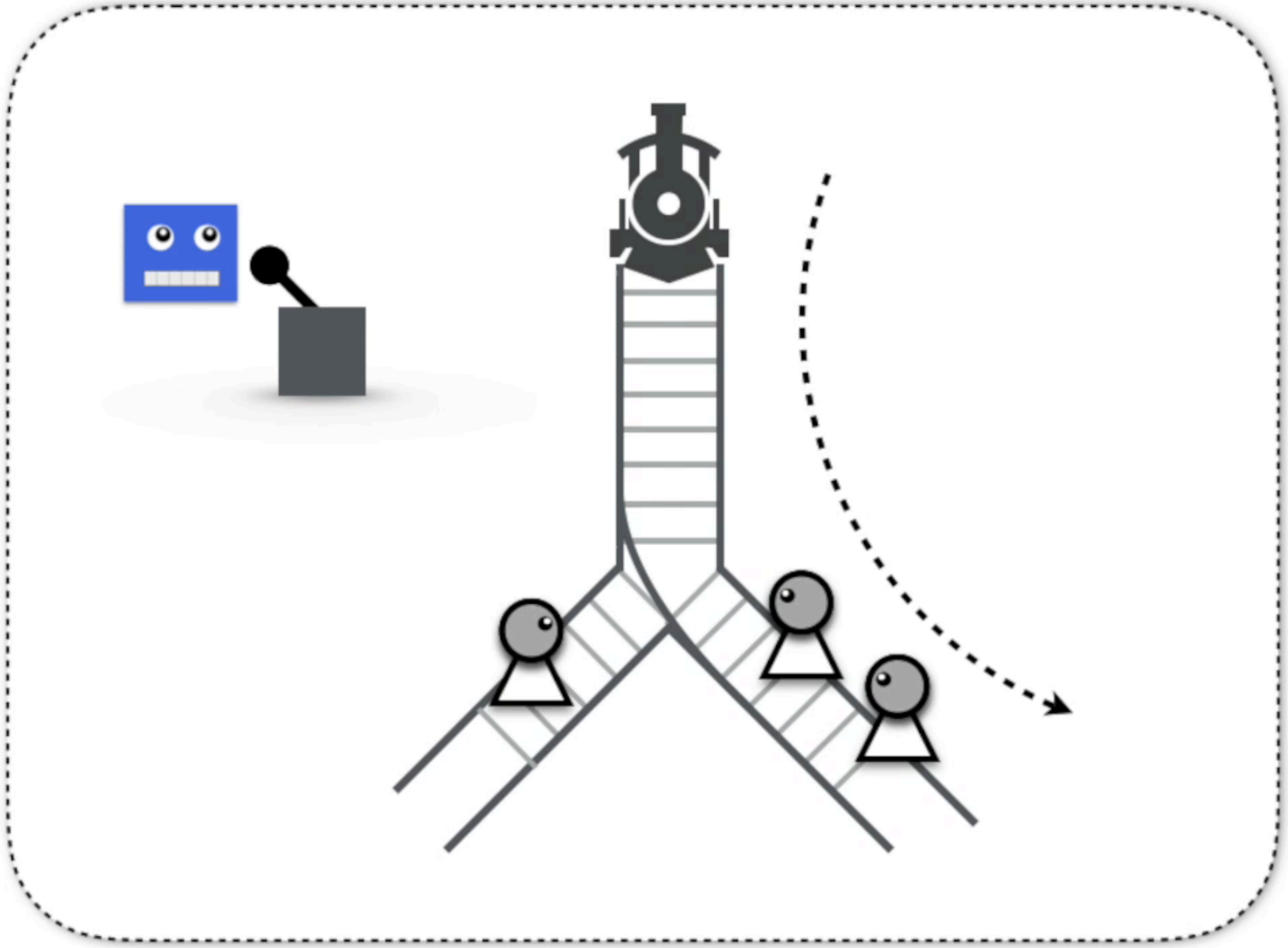
F_4 The harmful effects don't cause the good effects. Four permutations, paralleling the definition of $\rhd$ above, hold here. One such permutation is shown below. For any bad fluent f_b holding at t_1, and any good fluent f_g holding at some t_2, such that $t < t_1, t_2 \leq H$, the following holds:

$$\Gamma \vdash \neg \rhd \Big(Holds\big(f_b, t_1\big), Holds\big(f_g, t_2\big)\Big)$$

$\mathbb{P}_{\mathrm{DDE}_1}$ + ShadowProver







Inference Schemata

$$\frac{\mathbf{K}(a,t_1,\Gamma),\ \ \Gamma\vdash\phi,\ \ t_1\leq t_2}{\mathbf{K}(a,t_2,\phi)}\ [R_{\mathbf{K}}] \qquad \frac{\mathbf{B}(a,t_1,\Gamma),\ \ \Gamma\vdash\phi,\ \ t_1\leq t_2}{\mathbf{B}(a,t_2,\phi)}\ [R_{\mathbf{B}}]$$

$$\frac{}{\mathbf{C}(t,\mathbf{P}(a,t,\phi)\rightarrow\mathbf{K}(a,t,\phi))}\ [R_1] \qquad \frac{}{\mathbf{C}(t,\mathbf{K}(a,t,\phi)\rightarrow\mathbf{B}(a,t,\phi))}\ [R_2]$$

$$\frac{\mathbf{C}(t,\phi)\ t\leq t_1\ldots t\leq t_n}{\mathbf{K}(a_1,t_1,\ldots\mathbf{K}(a_n,t_n,\phi)\ldots)}\ [R_3] \qquad \frac{\mathbf{K}(a,t,\phi)}{\phi}\ [R_4]$$

$$\frac{}{\mathbf{C}(t,\mathbf{K}(a,t_1,\phi_1\rightarrow\phi_2))\rightarrow\mathbf{K}(a,t_2,\phi_1)\rightarrow\mathbf{K}(a,t_3,\phi_2)}\ [R_5]$$

$$\frac{}{\mathbf{C}(t,\mathbf{B}(a,t_1,\phi_1\rightarrow\phi_2))\rightarrow\mathbf{B}(a,t_2,\phi_1)\rightarrow\mathbf{B}(a,t_3,\phi_2)}\ [R_6]$$

$$\frac{}{\mathbf{C}(t,\mathbf{C}(t_1,\phi_1\rightarrow\phi_2))\rightarrow\mathbf{C}(t_2,\phi_1)\rightarrow\mathbf{C}(t_3,\phi_2)}\ [R_7]$$

$$\frac{}{\mathbf{C}(t,\forall x.\ \phi\rightarrow\phi[x\mapsto t])}\ [R_8] \qquad \frac{}{\mathbf{C}(t,\phi_1\leftrightarrow\phi_2\rightarrow\neg\phi_2\rightarrow\neg\phi_1)}\ [R_9]$$

$$\frac{}{\mathbf{C}(t,[\phi_1\wedge\ldots\wedge\phi_n\rightarrow\phi]\rightarrow[\phi_1\rightarrow\ldots\rightarrow\phi_n\rightarrow\psi])}\ [R_{10}]$$

$$\frac{\mathbf{S}(s,h,t,\phi)}{\mathbf{B}(h,t,\mathbf{B}(s,t,\phi))}\ [R_{12}] \qquad \frac{\mathbf{I}(a,t,happens(action(a^*,\alpha),t'))}{\mathbf{P}(a,t,happens(action(a^*,\alpha),t))}\ [R_{13}]$$

$$\frac{\mathbf{B}(a,t,\phi)\quad \mathbf{B}(a,t,\mathbf{O}(a,t,\phi,\chi))\quad \mathbf{O}(a,t,\phi,\chi)}{\mathbf{K}(a,t,\mathbf{I}(a,t,\chi))}\ [R_{14}]$$

Inference Schemata

$$\frac{\mathbf{K}(a,t_1,\Gamma),\ \ \Gamma\vdash\phi,\ \ t_1\leq t_2}{\mathbf{K}(a,t_2,\phi)}\ [R_{\mathbf{K}}] \qquad \frac{\mathbf{B}(a,t_1,\Gamma),\ \ \Gamma\vdash\phi,\ \ t_1\leq t_2}{\mathbf{B}(a,t_2,\phi)}\ [R_{\mathbf{B}}]$$

$$\frac{}{\mathbf{C}(t,\mathbf{P}(a,t,\phi)\rightarrow\mathbf{K}(a,t,\phi))}\ [R_1] \qquad \frac{}{\mathbf{C}(t,\mathbf{K}(a,t,\phi)\rightarrow\mathbf{B}(a,t,\phi))}\ [R_2]$$

$$\frac{\mathbf{C}(t,\phi)\ t\leq t_1\ldots t\leq t_n}{\mathbf{K}(a_1,t_1,\ldots\mathbf{K}(a_n,t_n,\phi)\ldots)}\ [R_3] \qquad \frac{\mathbf{K}(a,t,\phi)}{\phi}\ [R_4]$$

$$\frac{}{\mathbf{C}(t,\mathbf{K}(a,t_1,\phi_1\rightarrow\phi_2))\rightarrow\mathbf{K}(a,t_2,\phi_1)\rightarrow\mathbf{K}(a,t_3,\phi_2)}\ [R_5]$$

$$\frac{}{\mathbf{C}(t,\mathbf{B}(a,t_1,\phi_1\rightarrow\phi_2))\rightarrow\mathbf{B}(a,t_2,\phi_1)\rightarrow\mathbf{B}(a,t_3,\phi_2)}\ [R_6]$$

$$\frac{}{\mathbf{C}(t,\mathbf{C}(t_1,\phi_1\rightarrow\phi_2))\rightarrow\mathbf{C}(t_2,\phi_1)\rightarrow\mathbf{C}(t_3,\phi_2)}\ [R_7]$$

$$\frac{}{\mathbf{C}(t,\forall x.\ \phi\rightarrow\phi[x\mapsto t])}\ [R_8] \qquad \frac{}{\mathbf{C}(t,\phi_1\leftrightarrow\phi_2\rightarrow\neg\phi_2\rightarrow\neg\phi_1)}\ [R_9]$$

$$\frac{}{\mathbf{C}(t,[\phi_1\wedge\ldots\wedge\phi_n\rightarrow\phi]\rightarrow[\phi_1\rightarrow\ldots\rightarrow\phi_n\rightarrow\psi])}\ [R_{10}]$$

$$\frac{\mathbf{S}(s,h,t,\phi)}{\mathbf{B}(h,t,\mathbf{B}(s,t,\phi))}\ [R_{12}] \qquad \frac{\mathbf{I}(a,t,happens(action(a^*,\alpha),t'))}{\mathbf{P}(a,t,happens(action(a^*,\alpha),t))}\ [R_{13}]$$

$$\frac{\mathbf{B}(a,t,\phi)\quad\mathbf{B}(a,t,\mathbf{O}(a,t,\phi,\chi))\quad\mathbf{O}(a,t,\phi,\chi)}{\mathbf{K}(a,t,\mathbf{I}(a,t,\chi))}\ [R_{14}]$$

But what about self-sacrifice?!

3. Goal

In this section we render precise what is needed from a formal model of self-sacrifice. If one is building a self-driving car or a similar robotic system that functions in limited domains, it might be "trivial" to program in self-sacrifice, but we are seeking to understand and formalize what a model of self-sacrifice might look like in *general-purpose* autonomous robotic systems. Consider a sample scenario: A team of n, $(n >= 2)$, soliders from the *blue* team is captured by the *red* team.[b] The leader of the blue team is offered the choice of selecting one member from the team who will be sacrificed to free the rest of the team. Now consider the following actions:

- $\mathbf{a}_1$ The leader picks himself/herself.
- $\mathbf{a}_2$ The leader picks another soldier against their will.
- $\mathbf{a}_3$ The leader chooses a name randomly and it happens to be the leader's name.
- $\mathbf{a}_4$ The leader chooses a name randomly and it happens to be somebody else's name.
- $\mathbf{a}_5$ A soldier volunteers to die; the leader picks their name.

In addition to robotic systems with the capability for self-sacrifice in the right situations, we need systems that can understand human decisions in ethically-charged scenarios. We need a framework that can discern that: only $\mathbf{a}_1$ and $\mathbf{a}_5$ involve *true* self-sacrifice; $\mathbf{a}_3$ is *accidental* self-sacrifice; and $\mathbf{a}_2$ might be immoral.

Govindarajulu, N.S., Bringsjord, S., Ghosh, R.\ \& Peveler, M.\ (2019) ``Beyond the Doctrine of Double Effect: A Formal Model of True Self-Sacrifice'' in Ferreira, M.I.A., Sequeira, J.S., Virk, G.S., Tokhi, M.O., Kadar, E.E., eds., \textit{Robots and Well-Being}, in the series \textit{Intelligent Systems, Control and Automation: Science and Engineering} (Basel, Switzerland: Springer), pp.\ 39--54. A (rather rough) preprint of the paper is available at the link immediately below.
http://kryten.mm.rpi.edu/NSG_SB_RG_MP_DDE_SelfSac_110617.pdf

Logicization of Self-Reference

Three Levels of Self-Representation

de dicto Agent r with the name or description ν has come to believe on the basis of prior information Γ that the statement ϕ holds for the agent with the name or description ν.

$$\Gamma \vdash_r \mathbf{B}\Big(\mathsf{I}_r, \mathsf{now}, \exists a{:}\mathsf{Agent}\ \big[named(a,\nu) \wedge \phi(a)\big]\Big)$$

de re Agent r with the name or description ν has come to believe on the basis of prior information Γ that the statement ϕ holds of the agent with the name or description ν.

$$\exists a{:}\mathsf{Agent}\ named\,(a,\nu)\ \Big[\Gamma \vdash_r \mathbf{B}\big(\mathsf{I}_r, \mathsf{now}, \phi(a)\big)\Big]$$

de se Agent r believes on the basis of Γ that the statement ϕ holds of itself ν.

$$\Gamma \vdash_r \mathbf{B}\big(\mathsf{I}_r, \mathsf{now}, \phi(\mathsf{I_r} *)\big)$$

Govindarajulu, N.S., Bringsjord, S., Ghosh, R.\ \& Peveler, M.\ (2019) ``Beyond the Doctrine of Double Effect: A Formal Model of True Self-Sacrifice'' in Ferreira, M.I.A., Sequeira, J.S., Virk, G.S., Tokhi, M.O., Kadar, E.E., eds., \textit{Robots and Well-Being}, in the series \textit{Intelligent Systems, Control and Automation: Science and Engineering} (Basel, Switzerland: Springer), pp.\ 39--54. A (rather rough) preprint of the paper is available at the link immediately below.
http://kryten.mm.rpi.edu/NSG_SB_RG_MP_DDE_SelfSac_110617.pdf

DDE Abstracted

7. Formal $\mathcal{DDE}^*$

Assume we have an autonomous agent or robot r with a knowledge-base Γ. In Ref. 7, the predicate $\mathcal{DDE}(\Gamma, \sigma, a, \alpha, t, H)$ is formalized — and is read as "**from a set of premises Γ, and in situation σ, we can say that action α by agent a at time t operating with horizon H is $\mathcal{DDE}$-compliant.**" The formalization is broken up into four clauses corresponding to the informal clauses $\mathbf{C}_1$–$\mathbf{C}_4$ given above in Section 5:

$$\mathcal{DDE}(\Gamma, \sigma, a, \alpha, t, H) \leftrightarrow \Big(\mathbf{F}_1(\Gamma, \sigma, a, \alpha, t, H) \wedge \mathbf{F}_2(\ldots) \wedge \mathbf{F}_3(\ldots) \wedge \mathbf{F}_4(\ldots) \quad \Big)$$

Govindarajulu, N.S., Bringsjord, S., Ghosh, R.\ \& Peveler, M.\ (2019) ``Beyond the Doctrine of Double Effect: A Formal Model of True Self-Sacrifice'' in Ferreira, M.I.A., Sequeira, J.S., Virk, G.S., Tokhi, M.O., Kadar, E.E., eds., \textit{Robots and Well-Being}, in the series \textit{Intelligent Systems, Control and Automation: Science and Engineering} (Basel, Switzerland: Springer), pp.\ 39--54. A (rather rough) preprint of the paper is available at the link immediately below.
http://kryten.mm.rpi.edu/NSG_SB_RG_MP_DDE_SelfSac_110617.pdf

higher-order 6-place relation

DDE Abstracted

7. Formal $\mathcal{DDE}$*

Assume we have an autonomous agent or robot r with a knowledge-base Γ. In Ref. 7, the predicate $\mathcal{DDE}(\Gamma, \sigma, a, \alpha, t, H)$ is formalized — and is read as "**from a set of premises Γ, and in situation σ, we can say that action α by agent a at time t operating with horizon H is $\mathcal{DDE}$-compliant.**" The formalization is broken up into four clauses corresponding to the informal clauses $\mathbf{C}_1$–$\mathbf{C}_4$ given above in Section 5:

$$\mathcal{DDE}(\Gamma, \sigma, a, \alpha, t, H) \leftrightarrow \Big(\mathbf{F}_1(\Gamma, \sigma, a, \alpha, t, H) \wedge \mathbf{F}_2(\ldots) \wedge \mathbf{F}_3(\ldots) \wedge \mathbf{F}_4(\ldots) \quad \Big)$$

Govindarajulu, N.S., Bringsjord, S., Ghosh, R.\ \& Peveler, M.\ (2019) ``Beyond the Doctrine of Double Effect: A Formal Model of True Self-Sacrifice'' in Ferreira, M.I.A., Sequeira, J.S., Virk, G.S., Tokhi, M.O., Kadar, E.E., eds., \textit{Robots and Well-Being}, in the series \textit{Intelligent Systems, Control and Automation: Science and Engineering} (Basel, Switzerland: Springer), pp.\ 39--54. A (rather rough) preprint of the paper is available at the link immediately below.
http://kryten.mm.rpi.edu/NSG_SB_RG_MP_DDE_SelfSac_110617.pdf

DDE*

With the formal machinery at hand, enhancing $\mathcal{DDE}$ to $\mathcal{DDE}^*$ is straighforward. Now, corresponding to the augmented informal definition in Section 5, we take the $\mathcal{DDE}$ predicate defined in Ref. 7 and added disjunction.

$$\mathcal{DDE}^*(\ldots) \Leftrightarrow \begin{cases} \mathcal{DDE}\,(\Gamma, \sigma, a, \alpha, t, H) \bigvee \\ \mathbf{F}_1 \wedge \mathbf{F}_2 \wedge \mathbf{F}_3 \wedge \mathbf{K}\left(a, t, \begin{pmatrix} \left[\forall b. \left(b \neq a^*\right) \rightarrow \nu(\alpha, a, b, t) \gg 0\right] \wedge \\ \nu(\alpha, a, a^*, t) \ll 0 \end{pmatrix}\right) \end{cases}$$

The disjunction simply states that the new principle $\mathcal{DDE}^*$ applies when — (1) $\mathcal{DDE}$ applies; or (2) when conditions $\mathbf{F}_1$, $\mathbf{F}_2$, and $\mathbf{F}_3$ apply along with the condition that the agent performing the action **knows** that all of the bad effects are directed toward itself, and the good effects are great in magnitude and apply only to other agents.

Simulation: We take a formalization of the standard trolley scenario[7] and

Govindarajulu, N.S., Bringsjord, S., Ghosh, R.\ \& Peveler, M.\ (2019) ``Beyond the Doctrine of Double Effect: A Formal Model of True Self-Sacrifice'' in Ferreira, M.I.A., Sequeira, J.S., Virk, G.S., Tokhi, M.O., Kadar, E.E., eds., \textit{Robots and Well-Being}, in the series \textit{Intelligent Systems, Control and Automation: Science and Engineering} (Basel, Switzerland: Springer), pp.\ 39--54. A (rather rough) preprint of the paper is available at the link immediately below.
http://kryten.mm.rpi.edu/NSG_SB_RG_MP_DDE_SelfSac_110617.pdf

What about DTripleE??

Peveler, M.,~Govindarajulu, N.S.\ \& Bringsjord, S.\ (2018)
``Toward Automating the Doctrine of Triple Effect'' in
S.\ Bringsjord, M.\ Osman Tokhi, M.\ Isabel Aldinhas Ferreira,
N.S.\ Govindarajulu, eds., \textit{Hybrid Worlds: Societal and
Ethical Challenges; Proceedings of the International Conference
on Robot Ethics and Standards (ICRES) 2018}, ISBN
978-1-9164490-1-5, London, UK: CLAWAR, pp.\ 82--88. The link
below goes to the entire ebook in which the official version of
this paper appears.
http://kryten.mm.rpi.edu/HybridWorlds.pdf

What about D<u>Triple</u>E??

ICRES 2018: International Conference on Robot
Ethics and Standards, Troy, NY, 20-21 August 2018.
https://doi.org/10.13180/icres.2018.20-21.08.020

TOWARD AUTOMATING THE DOCTRINE OF TRIPLE EFFECT

M. PEVELER*, N. S. GOVINDARAJULU, and S. BRINGSJORD
Rensselaer AI & Reasoning (RAIR) Lab
Rensselaer Polytechnic Institue (RPI)
Troy NY 12180 USA
**E-mail: pevelm@rpi.com, naveensundarg@gmail.com, selmer.bringsjord@gmail.com*

The **Doctrine of Double Effect** ($\mathcal{DDE}$) is a long-studied ethical principle governing whether taking an action that has both significant positive and negative effects is ethically permissible. Unfortunately, despite its storied history, $\mathcal{DDE}$ does not fully account for the permissibility of actions taken in certain particularly challenging moral dilemmas that have recently arrived on the scene. The **Doctrine of *Triple* Effect** ($\mathcal{DTE}$) can be employed in these dilemmas, to separate the intention to perform an action *because* an effect will occur, versus *in order* for that effect to occur. This distinction allows an agent to permissibly pursue actions that may have foreseeable negative effects resulting from those actions — as long as the negative effect is not the agent's primary intention. By $\mathcal{DDE}$ such actions are not classified as ethically permissible. We briefly present $\mathcal{DTE}$ and, using a first-order multi-operator modal logic (the **deontic cognitive event calculus**), formalize this doctrine. We then give a proof-sketch of a situation for which $\mathcal{DTE}$ but not $\mathcal{DDE}$ can be used to classify a relevant action as permissible. We end with a look forward to future work.

Keywords: doctrine of double effect, doctrine of triple effect, machine ethics, AI

Peveler, M.,~Govindarajulu, N.S.\ \& Bringsjord, S.\ (2018)
``Toward Automating the Doctrine of Triple Effect'' in
S.\ Bringsjord, M.\ Osman Tokhi, M.\ Isabel Aldinhas Ferreira,
N.S.\ Govindarajulu, eds., \textit{Hybrid Worlds: Societal and
Ethical Challenges; Proceedings of the International Conference
on Robot Ethics and Standards (ICRES) 2018}, ISBN
978-1-9164490-1-5, London, UK: CLAWAR, pp.\ 82--88. The link
below goes to the entire ebook in which the official version of
this paper appears.
http://kryten.mm.rpi.edu/HybridWorlds.pdf

Making Morally  Machines

Selmer Bringsjord ∧ Naveen Sundar Govindarajulu ∧ John Licato

Making Morally  **Machines**

Selmer Bringsjord ∧ Naveen Sundar Govindarajulu ∧ John Licato

er løsningen, med nok penger!