Putting Perspective into OWL [sic]:

Complexity-Neutral Standpoint Reasoning for Ontology Languages via Monodic S5 over Counting Two-Variable First-Order Logic

Lucía Gómez Álvarez, Sebastian Rudolph

Inria
Technische Universität Dresden

Motivation

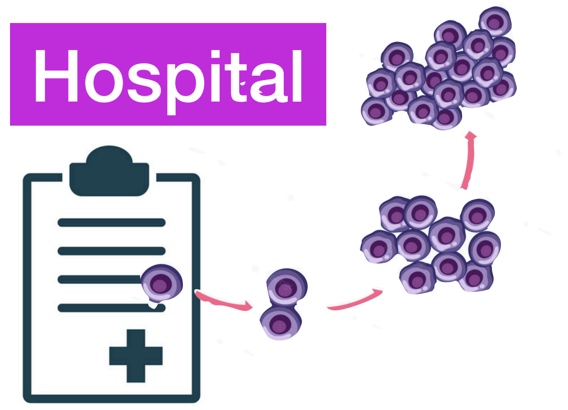
Multiperspective Reasoning

Motivation

Non-trivial combinations of the huge diversity of knowledge sources available
Knowledge sources embed the perspectives of their creators!

SNOMED CT

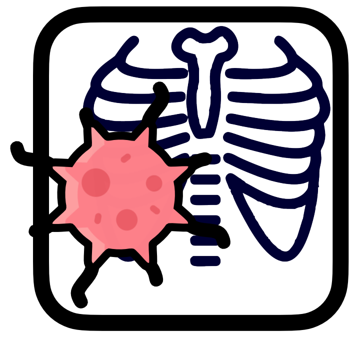
Hospital



WIKIDATA

yago select knowledge

Private Lab



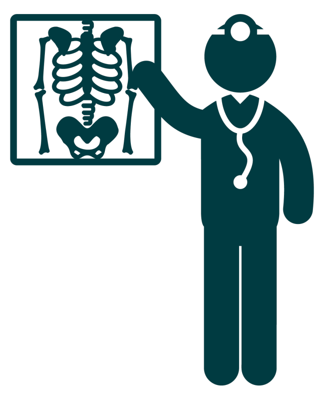
DBpedia

Diverse Knowledge Sources

Motivation

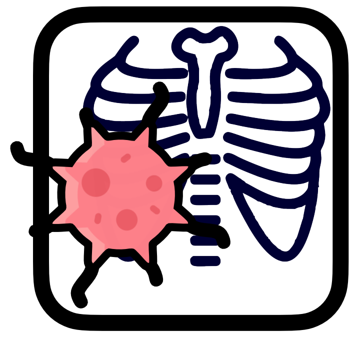
Non-trivial combinations of the huge diversity of knowledge sources available
Knowledge sources embed the perspectives of their creators!

Private Lab



Tumor(t)
hasTissue(p, t)

Private Lab



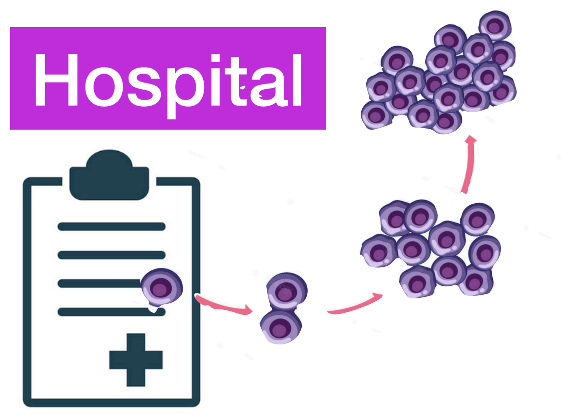
SNOMED CT

WIKIDATA

yago select knowledge

DBpedia

Hospital



Diverse Knowledge Sources

Motivation

Non-trivial combinations of the huge diversity of knowledge sources available

Knowledge sources embed the perspectives of their creators!

Diverse Knowledge Sources

Motivation

Non-trivial combinations of the huge diversity of knowledge sources available

Knowledge sources embed the perspectives of their creators!

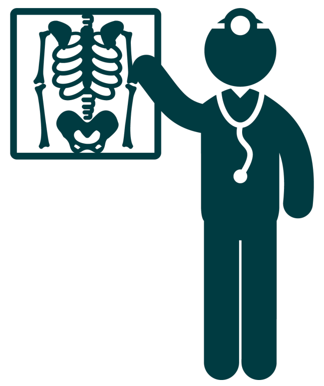
Diverse Knowledge Sources

Motivation

Non-trivial combinations of the huge diversity of knowledge sources available
Knowledge sources embed the perspectives of their creators!

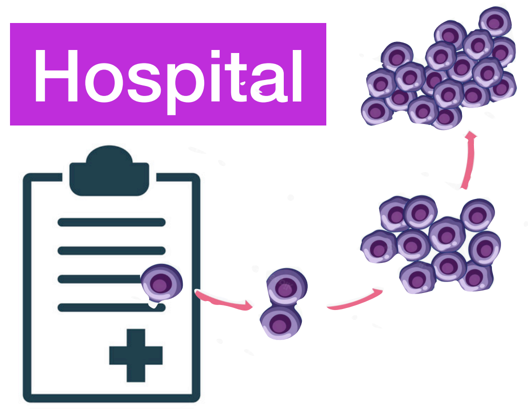
Private Lab

Tumor(t)
hasTissue(p, t)

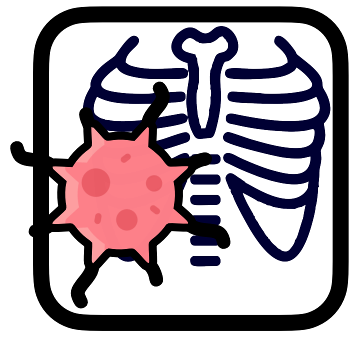


SNOMED CT

Hospital



Private Lab



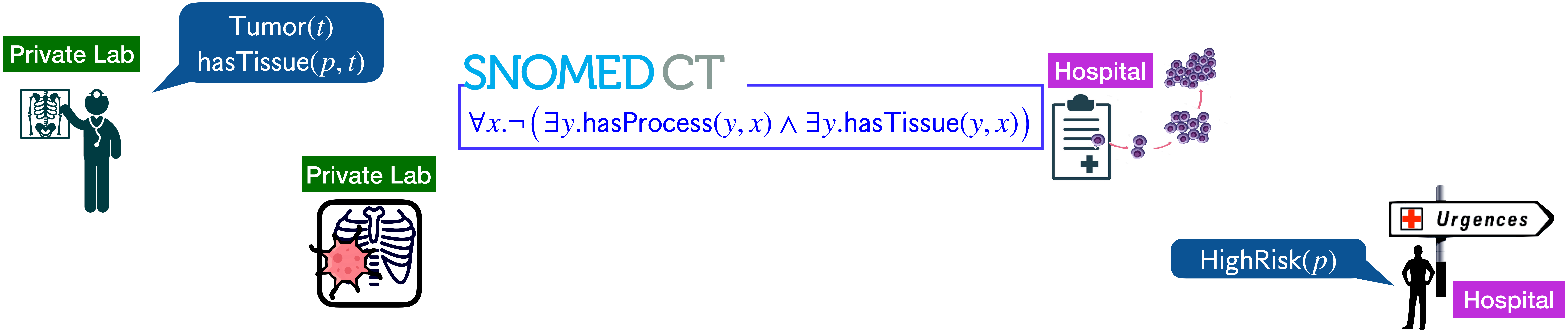
HighRisk(p)

Hospital

Diverse Knowledge Sources

Motivation

Non-trivial combinations of the huge diversity of knowledge sources available
Knowledge sources embed the perspectives of their creators!

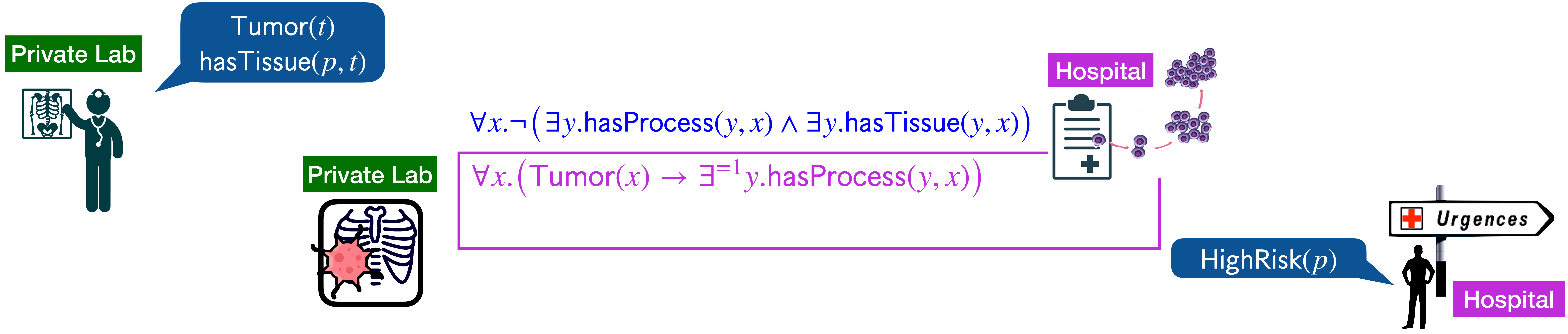


Diverse Knowledge Sources

Motivation

Non-trivial combinations of the huge diversity of knowledge sources available

Knowledge sources embed the perspectives of their creators!

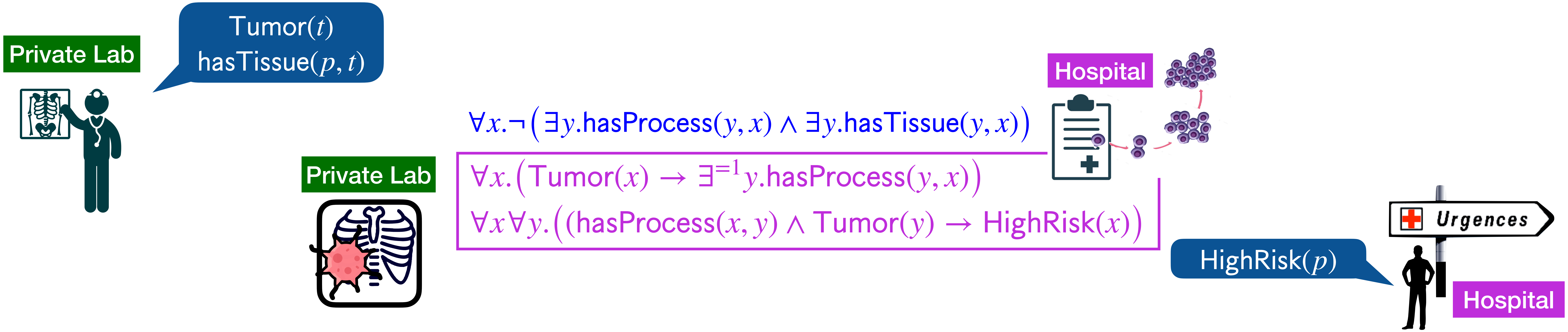


Diverse Knowledge Sources

Motivation

Non-trivial combinations of the huge diversity of knowledge sources available

Knowledge sources embed the perspectives of their creators!



Diverse Knowledge Sources

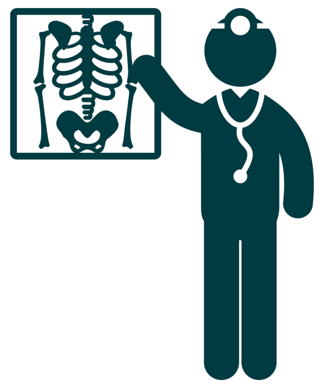
Motivation

Non-trivial combinations of the huge diversity of knowledge sources available

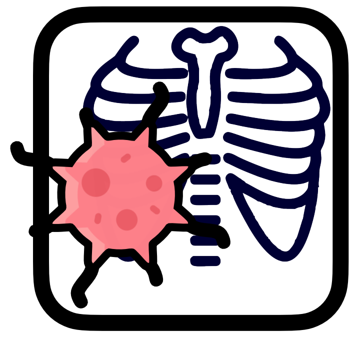
Knowledge sources embed the perspectives of their creators!

Private Lab

Tumor(t)
hasTissue(p, t)



Private Lab



$\forall x.\neg\big(\exists y.\text{hasProcess}(y,x) \wedge \exists y.\text{hasTissue}(y,x)\big)$

$\forall x.\big(\text{Tumor}(x) \rightarrow \exists^{=1} y.\text{hasProcess}(y,x)\big)$

$\forall x \forall y.\big((\text{hasProcess}(x,y) \wedge \text{Tumor}(y) \rightarrow \text{HighRisk}(x)\big)$

$\forall x.\big(\text{Tumor}(x) \rightarrow \exists^{=1} y.\text{hasTissue}(y,x)\big)$

HighRisk(p)

Hospital

Diverse Knowledge Sources

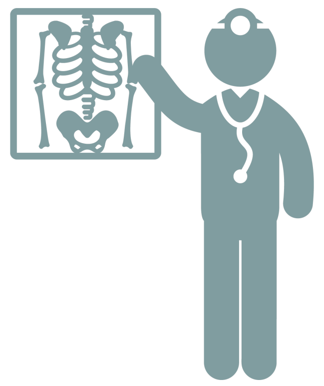
Motivation

Non-trivial combinations of the huge diversity of knowledge sources available
Knowledge sources embed the perspectives of their creators!

Challenge: Integration

Private Lab

Tumor(t)
hasTissue(p, t)



Knowledge Fusion

$$\forall x.\neg\big(\exists y.\mathsf{hasProcess}(y,x) \land \exists y.\mathsf{hasTissue}(y,x)\big)$$

$$\forall x.\big(\mathsf{Tumor}(x) \rightarrow \exists^{=1} y.\mathsf{hasProcess}(y,x)\big)$$

$$\forall x \forall y.\big((\mathsf{hasProcess}(x,y) \land \mathsf{Tumor}(y) \rightarrow \mathsf{HighRisk}(x)\big)$$

$$\forall x.\big(\mathsf{Tumor}(x) \rightarrow \exists^{=1} y.\mathsf{hasTissue}(y,x)\big)$$

HighRisk(p)

Hospital

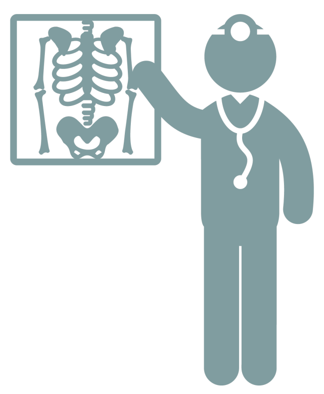
Motivation

Non-trivial combinations of the huge diversity of knowledge sources available
Knowledge sources embed the perspectives of their creators!

Challenge: Integration

$\mathsf{Tumor}(t)$

Private Lab



$\mathsf{Tumor}(t)$
$\mathsf{hasTissue}(p, t)$

Knowledge Fusion

$\forall x.\neg\big(\exists y.\mathsf{hasProcess}(y, x) \land \exists y.\mathsf{hasTissue}(y, x)\big)$

$\forall x.\big(\mathsf{Tumor}(x) \rightarrow \exists^{=1} y.\mathsf{hasProcess}(y, x)\big)$

$\forall x \forall y.\big((\mathsf{hasProcess}(x, y) \land \mathsf{Tumor}(y) \rightarrow \mathsf{HighRisk}(x)\big)$

$\forall x.\big(\mathsf{Tumor}(x) \rightarrow \exists^{=1} y.\mathsf{hasTissue}(y, x)\big)$

$\mathsf{HighRisk}(p)$

Hospital

Motivation

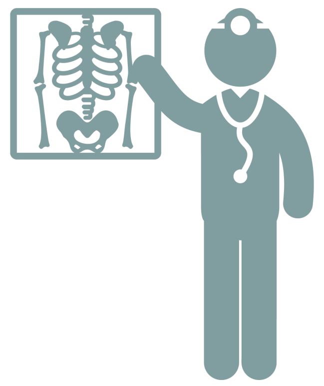
Non-trivial combinations of the huge diversity of knowledge sources available
Knowledge sources embed the perspectives of their creators!

Challenge: Integration

$\mathsf{Tumor}(t)$

$\mathsf{hasProcess}(\,.\,, t)$

Private Lab



$\mathsf{Tumor}(t)$
$\mathsf{hasTissue}(p, t)$

Knowledge Fusion

$$\forall x.\neg\big(\exists y.\mathsf{hasProcess}(y, x) \land \exists y.\mathsf{hasTissue}(y, x)\big)$$

$$\forall x.\big(\mathsf{Tumor}(x) \rightarrow \exists^{=1} y.\mathsf{hasProcess}(y, x)\big)$$

$$\forall x \forall y.\big((\mathsf{hasProcess}(x, y) \land \mathsf{Tumor}(y) \rightarrow \mathsf{HighRisk}(x)\big)$$

$$\forall x.\big(\mathsf{Tumor}(x) \rightarrow \exists^{=1} y.\mathsf{hasTissue}(y, x)\big)$$

$\mathsf{HighRisk}(p)$

Hospital

Motivation

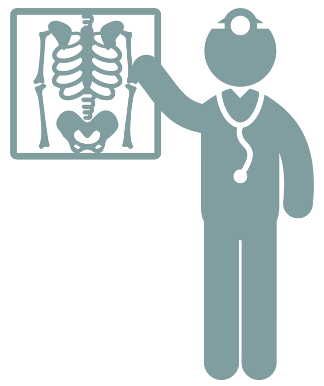
Non-trivial combinations of the huge diversity of knowledge sources available
Knowledge sources embed the perspectives of their creators!

Challenge: Integration

$\text{Tumor}(t)$

$\text{hasProcess}(\,.\,, t)$ $\qquad$ $\text{hasTissue}(\,.\,, t)$

Private Lab



$\text{Tumor}(t)$
$\text{hasTissue}(p, t)$

Knowledge Fusion

$\forall x.\neg\big(\exists y.\text{hasProcess}(y, x) \wedge \exists y.\text{hasTissue}(y, x)\big)$

$\forall x.\big(\text{Tumor}(x) \rightarrow \exists^{=1} y.\text{hasProcess}(y, x)\big)$

$\forall x \forall y.\big((\text{hasProcess}(x, y) \wedge \text{Tumor}(y) \rightarrow \text{HighRisk}(x)\big)$

$\forall x.\big(\text{Tumor}(x) \rightarrow \exists^{=1} y.\text{hasTissue}(y, x)\big)$

$\text{HighRisk}(p)$

Hospital

Motivation

Non-trivial combinations of the huge diversity of knowledge sources available

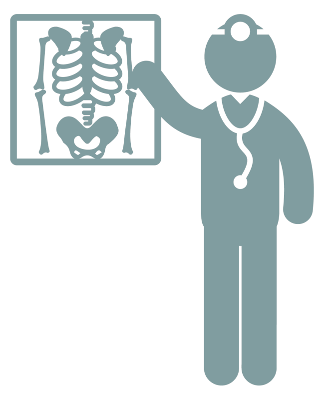
Knowledge sources embed the perspectives of their creators!

Challenge: Integration

$\mathsf{Tumor}(t)$

$\mathsf{hasProcess}(\,.\,, t) \rightarrow \bot \leftarrow \mathsf{hasTissue}(\,.\,, t)$

Private Lab



$\mathsf{Tumor}(t)$
$\mathsf{hasTissue}(p, t)$

Knowledge Fusion

$\forall x. \neg \big(\exists y.\mathsf{hasProcess}(y, x) \wedge \exists y.\mathsf{hasTissue}(y, x) \big)$

$\forall x. \big(\mathsf{Tumor}(x) \rightarrow \exists^{=1} y.\mathsf{hasProcess}(y, x) \big)$

$\forall x \forall y. \big((\mathsf{hasProcess}(x, y) \wedge \mathsf{Tumor}(y) \rightarrow \mathsf{HighRisk}(x) \big)$

$\forall x. \big(\mathsf{Tumor}(x) \rightarrow \exists^{=1} y.\mathsf{hasTissue}(y, x) \big)$

$\mathsf{HighRisk}(p)$

Hospital

Multiperspective Knowledge Management

Challenge: combining diverse (potentially conflicting) sources without weakening them

Standpoint Logic

➡ **Multimodal logic** characterised by simplified Kripke semantics

➡ Knowledge relative to “points of view” (standpoints)

Multiperspective Knowledge Management

Challenge: combining diverse (potentially conflicting) sources without weakening them

Standpoint Logic

- ➡ **Multimodal logic** characterised by simplified Kripke semantics
- ➡ Knowledge relative to “points of view” (standpoints)

$$\forall x. \neg \big(\exists y.\mathsf{hasProcess}(y,x) \land \exists y.\mathsf{hasTissue}(y,x) \big)$$

$$\forall x. \big(\mathsf{Tumor}(x) \rightarrow \exists^{=1} y.\mathsf{hasTissue}(x,y) \big)$$

$$\forall x. \big(\mathsf{Tumor}(x) \rightarrow \exists^{=1} y.\mathsf{hasProcess}(y,x) \big)$$

$$\forall x \forall y. \big(\mathsf{hasProcess}(x,y) \land \mathsf{Tumor}(y) \rightarrow \mathsf{HighRisk}(x) \big)$$

Multiperspective Knowledge Management

Challenge: combining diverse (potentially conflicting) sources without weakening them

Standpoint Logic

➡ **Multimodal logic** characterised by simplified Kripke semantics

➡ Knowledge relative to "points of view" (standpoints)

$\Box_e$ Unequivocal according to e

$\Diamond_e$ Conceivable according to e

CONSISTENT

$\Box_* \forall x.\neg\big(\exists y.\mathrm{hasProcess}(y,x) \wedge \exists y.\mathrm{hasTissue}(y,x)\big)$

$\Box_L \forall x.\big(\mathrm{Tumor}(x) \rightarrow \exists^{=1} y.\mathrm{hasTissue}(x,y)\big)$

$\Box_H \forall x.\big(\mathrm{Tumor}(x) \rightarrow \exists^{=1} y.\mathrm{hasProcess}(y,x)\big)$

$\Box_H \forall x \forall y.\big(\mathrm{hasProcess}(x,y) \wedge \mathrm{Tumor}(y) \rightarrow \mathrm{HighRisk}(x)\big)$

Multiperspective Knowledge Management

Challenge: combining diverse (potentially conflicting) sources without weakening them

Standpoint Logic

➡ **Multimodal logic** characterised by simplified Kripke semantics

➡ Knowledge relative to "points of view" (standpoints)

$\Box_e$ Unequivocal according to e

$\Diamond_e$ Conceivable according to e

CONSISTENT

$$\Box_* \forall x.\neg\big(\exists y.\mathsf{hasProcess}(y,x) \land \exists y.\mathsf{hasTissue}(y,x)\big)$$

$$\Box_L \forall x.\big(\mathsf{Tumor}(x) \rightarrow \exists^{=1} y.\mathsf{hasTissue}(x,y)\big)$$

$$\Box_H \forall x.\big(\mathsf{Tumor}(x) \rightarrow \exists^{=1} y.\mathsf{hasProcess}(y,x)\big)$$

$$\Box_H \forall x \forall y.\big(\mathsf{hasProcess}(x,y) \land \mathsf{Tumor}(y) \rightarrow \mathsf{HighRisk}(x)\big)$$

$$\forall x.\big(\Diamond_L\big(\exists y\, \mathsf{hasTissue}(x,y) \land \mathsf{Tumor}(y)\big) \rightarrow \Box_H \exists y.\big(\mathsf{hasProcess}(x,y) \land \mathsf{Tumor}(y)\big)$$

Multiperspective Knowledge Management

Challenge: combining diverse (potentially conflicting) sources without weakening them

Standpoint Logic

➡ **Multimodal logic** characterised by simplified Kripke semantics

➡ Knowledge relative to "points of view" (standpoints)

$\Box_e$ Unequivocal according to e

$\Diamond_e$ Conceivable according to e

CONSISTENT

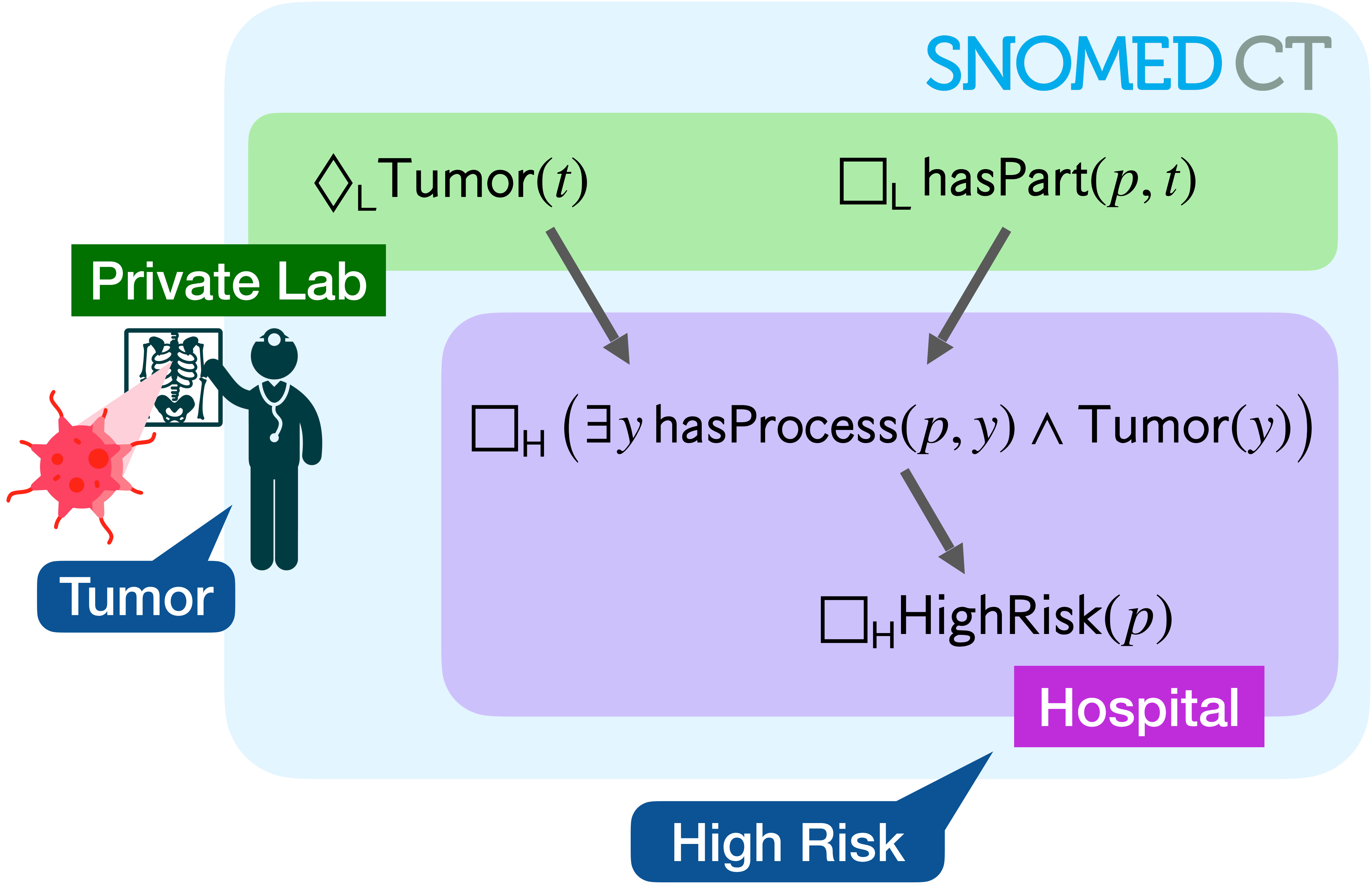
$\Box_* \forall x.\neg\big(\exists y.\mathrm{hasProcess}(y,x) \wedge \exists y.\mathrm{hasTissue}(y,x)\big)$

$\Box_L \forall x.\big(\mathrm{Tumor}(x) \rightarrow \exists^{=1} y.\mathrm{hasTissue}(x,y)\big)$

$\Box_H \forall x.\big(\mathrm{Tumor}(x) \rightarrow \exists^{=1} y.\mathrm{hasProcess}(y,x)\big)$

$\Box_H \forall x \forall y.\big(\mathrm{hasProcess}(x,y) \wedge \mathrm{Tumor}(y) \rightarrow \mathrm{HighRisk}(x)\big)$

$\forall x.\big(\Diamond_L\big(\exists y\, \mathrm{hasTissue}(x,y) \wedge \mathrm{Tumor}(y)\big) \rightarrow \Box_H \exists y.\big(\mathrm{hasProcess}(x,y) \wedge \mathrm{Tumor}(y)\big)\big)$



Contents

- ➡ Motivation for the framework (DONE)
- ➡ First Order Standpoint Logic and Monodic Standpoint C^2 (THE LANGUAGE)
- ➡ Transformations (PREPROCESSING)
- ➡ Satisfiability in Monodic Standpoint C^2 (MAIN TECHNICAL RESULT)
- ➡ Application to Ontology Languages (OWL)
- ➡ Nominals Cause Trouble
- ➡ Final Observations and Conclusion

First-Order Standpoint Logic
(Syntax and Semantics)

First-Order Standpoint Logic: Syntax

Syntax of $\mathbb{S}$

First-Order Standpoint Logic: Syntax

Syntax of $\mathbb{S}$

Signature $\langle \mathbf{P}, \mathbf{C}, \mathbf{S} \rangle$ of predicates, constants and standpoints.

First-Order Standpoint Logic: Syntax

Syntax of $\mathbb{S}$

Signature $\langle \mathbf{P}, \mathbf{C}, \mathbf{S} \rangle$ of predicates, constants and standpoints.

FOSL formulas:

$$\phi, \psi ::= P(t_1, \ldots, t_k) \mid t_1 \doteq t_2 \mid \neg\phi \mid \phi \land \psi \mid \exists^{\triangleleft n} x.\phi \mid \lozenge_{\mathsf{e}}\phi$$

- $\lozenge_{\mathsf{e}}\phi \longrightarrow$ "To *e*, it is **conceivable** that ϕ"

First-Order Standpoint Logic: Syntax

Syntax of $\mathbb{S}$

Signature $\langle \mathbf{P}, \mathbf{C}, \mathbf{S} \rangle$ of predicates, constants and standpoints.

FOSL formulas:

$$\phi, \psi ::= P(t_1, \ldots, t_k) \mid t_1 \doteq t_2 \mid \neg\phi \mid \phi \wedge \psi \mid \exists^{\triangleleft n} x.\phi \mid \Diamond_{\mathsf{e}}\phi$$

counting quantifiers with $\triangleleft \in \{\leq, =, \geq\}$,

- $\Diamond_{\mathsf{e}}\phi \longrightarrow$ "To *e*, it is **conceivable** that ϕ"

First-Order Standpoint Logic: Syntax

Syntax of $\mathbb{S}$

Signature $\langle \mathbf{P}, \mathbf{C}, \mathbf{S} \rangle$ of predicates, constants and standpoints.

FOSL formulas:

$$\phi, \psi ::= P(t_1, \ldots, t_k) \mid t_1 \doteq t_2 \mid \neg\phi \mid \phi \wedge \psi \mid \exists^{\lhd n} x.\phi \mid \Diamond_{\mathrm{e}}\phi$$

counting quantifiers with $\lhd \in \{ \leq, =, \geq \}$,

$\Diamond_{\mathrm{e}}\neg\phi \equiv \neg\Box_{\mathrm{e}}\,\phi$ (dual)

- $\Diamond_{\mathrm{e}}\phi \longrightarrow$ "To *e*, it is **conceivable** that ϕ"
- $\Box_{\mathrm{e}}\phi \longrightarrow$ "To *e*, it is **unequivocal** that ϕ"

First-Order Standpoint Logic: Syntax

Syntax of $\mathbb{S}$

Signature $\langle \mathbf{P}, \mathbf{C}, \mathbf{S} \rangle$ of predicates, constants and standpoints.

FOSL formulas:

$$\phi, \psi ::= P(t_1, \ldots, t_k) \mid t_1 \doteq t_2 \mid \neg\phi \mid \phi \wedge \psi \mid \exists^{\lhd n} x.\phi \mid \Diamond_{\mathrm{e}} \phi$$

counting quantifiers with $\lhd \in \{ \leq, =, \geq \}$,

$\Diamond_{\mathrm{e}} \neg\phi \equiv \neg \Box_{\mathrm{e}}\, \phi$ (dual)

Standpoint expressions:

$$\mathrm{e}_1, \mathrm{e}_2 ::= * \mid s \mid \mathrm{e}_1 \cup \mathrm{e}_2 \mid \mathrm{e}_1 \cap \mathrm{e}_2 \mid \mathrm{e}_1 \backslash \mathrm{e}_2$$

- $\Diamond_{\mathrm{e}} \phi \longrightarrow$ "To *e*, it is **conceivable** that ϕ"
- $\Box_{\mathrm{e}} \phi \longrightarrow$ "To *e*, it is **unequivocal** that ϕ"

First-Order Standpoint Logic: Syntax

Syntax of $\mathbb{S}$

Signature $\langle \mathbf{P}, \mathbf{C}, \mathbf{S} \rangle$ of predicates, constants and standpoints.

FOSL formulas:

$$\phi, \psi ::= P(t_1, \ldots, t_k) \mid t_1 \doteq t_2 \mid \neg\phi \mid \phi \wedge \psi \mid \exists^{\triangleleft n} x.\phi \mid \Diamond_{\mathrm{e}} \phi$$

counting quantifiers with $\triangleleft \in \{ \leq, =, \geq \}$,

$\Diamond_{\mathrm{e}} \neg\phi \equiv \neg \Box_{\mathrm{e}} \phi$ (dual)

Standpoint expressions:

$$\mathrm{e}_1, \mathrm{e}_2 ::= * \mid s \mid \mathrm{e}_1 \cup \mathrm{e}_2 \mid \mathrm{e}_1 \cap \mathrm{e}_2 \mid \mathrm{e}_1 \backslash \mathrm{e}_2$$

- $\Diamond_{\mathrm{e}} \phi \longrightarrow$ “To *e*, it is **conceivable** that ϕ”
- $\Box_{\mathrm{e}} \phi \longrightarrow$ “To *e*, it is **unequivocal** that ϕ”

A formula is:

- **Monodic** : at most one free variable in each subformula of the form $\Diamond_{\mathrm{e}} \phi$ (or $\Box_{\mathrm{e}} \phi$)

$\forall x.\mathrm{Tissue}(x) \rightarrow \Box_{*} \mathrm{Tissue}(x)$ ✅

$\forall x \forall y.\mathrm{hasTissue}(x, y) \rightarrow \Box_{*} \mathrm{hasTissue}(x, y)$ ❌

First-Order Standpoint Logic: Syntax

Syntax of $\mathbb{S}$

Signature $\langle \mathbf{P}, \mathbf{C}, \mathbf{S} \rangle$ of predicates, constants and standpoints.

FOSL formulas:

$$\phi, \psi ::= P(t_1, \ldots, t_k) \mid t_1 \doteq t_2 \mid \neg\phi \mid \phi \wedge \psi \mid \exists^{\triangleleft n} x.\phi \mid \Diamond_{\mathrm{e}} \phi$$

counting quantifiers with $\triangleleft \in \{ \leq, =, \geq \}$,

$$\Diamond_{\mathrm{e}} \neg\phi \equiv \neg \Box_{\mathrm{e}}\, \phi \text{ (dual)}$$

Standpoint expressions:

$$\mathrm{e}_1, \mathrm{e}_2 ::= * \mid s \mid \mathrm{e}_1 \cup \mathrm{e}_2 \mid \mathrm{e}_1 \cap \mathrm{e}_2 \mid \mathrm{e}_1 \backslash \mathrm{e}_2$$

- $\Diamond_{\mathrm{e}} \phi \longrightarrow$ "To *e*, it is **conceivable** that ϕ"
- $\Box_{\mathrm{e}} \phi \longrightarrow$ "To *e*, it is **unequivocal** that ϕ"

A formula is:

- **Monodic** : at most one free variable in each subformula of the form $\Diamond_{\mathrm{e}} \phi$ (or $\Box_{\mathrm{e}}\, \phi$)

  $\forall x.\mathsf{Tissue}(x) \rightarrow \Box_{*}\, \mathsf{Tissue}(x)$ ✅

  $\forall x \forall y.\mathsf{hasTissue}(x, y) \rightarrow \Box_{*}\, \mathsf{hasTissue}(x, y)$ ❌

- C^2 : only 2 variables, all predicates of arity ≤ 2

  $\Box_{*}\, \forall x \forall y.\mathsf{hasTissue}(x, y) \rightarrow \exists x.\mathsf{hasCell}(y, x)$ ✅

  $\Box_{*}\, \forall x \forall y.\mathsf{hasTissue}(x, y) \rightarrow \exists z.\mathsf{hasOrgan}(x, z) \wedge \mathsf{from}(y, z)$ ❌

First-Order Standpoint Logic: Semantics

Semantics of $\mathbb{S}$

Relational semantics:

$$\mathcal{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$$

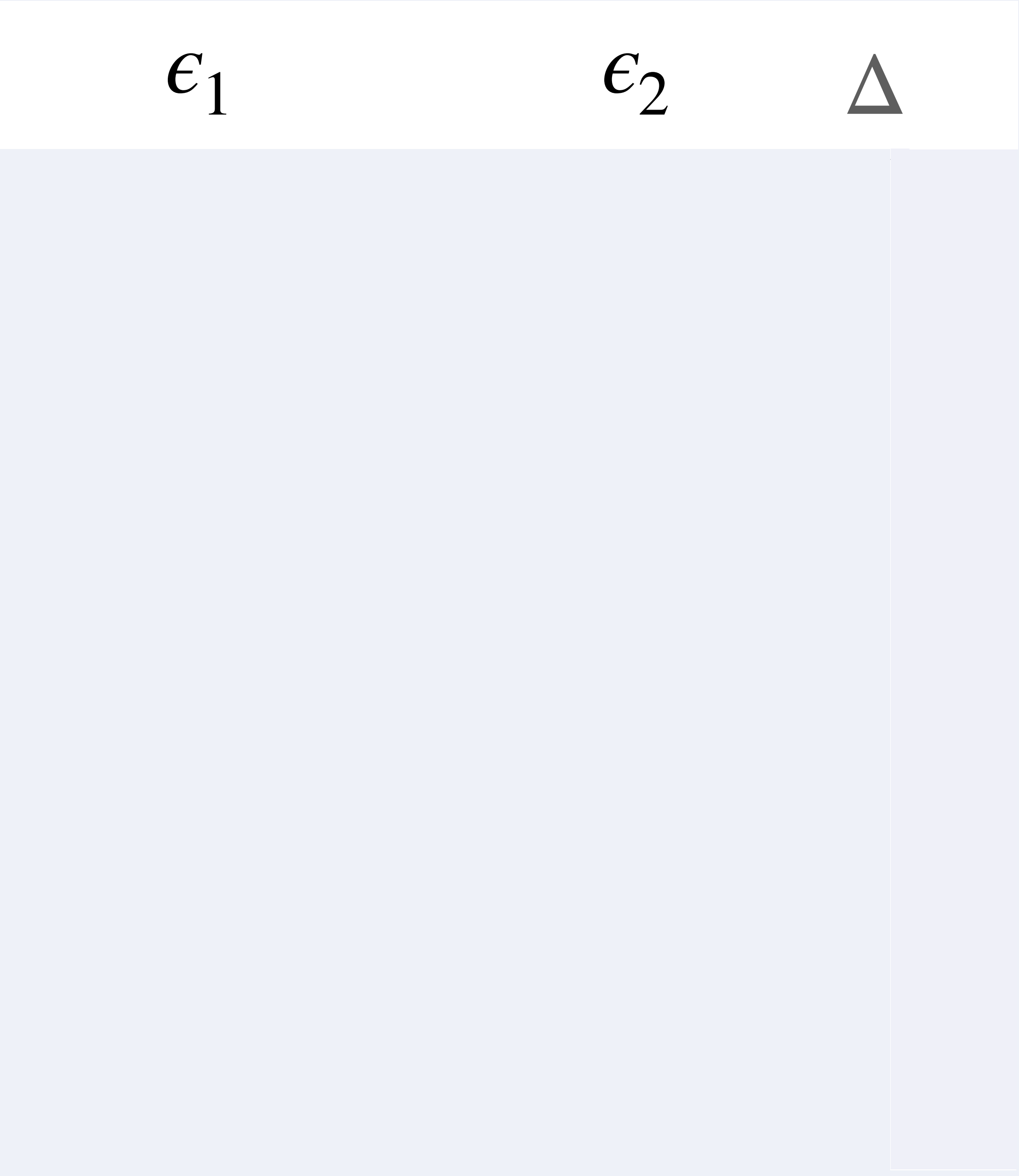
First-Order Standpoint Logic: Semantics

Semantics of $\mathbb{S}$

Relational semantics:

$$\mathcal{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$$

- $\Delta = \{\epsilon_1, \epsilon_2\}$ domain



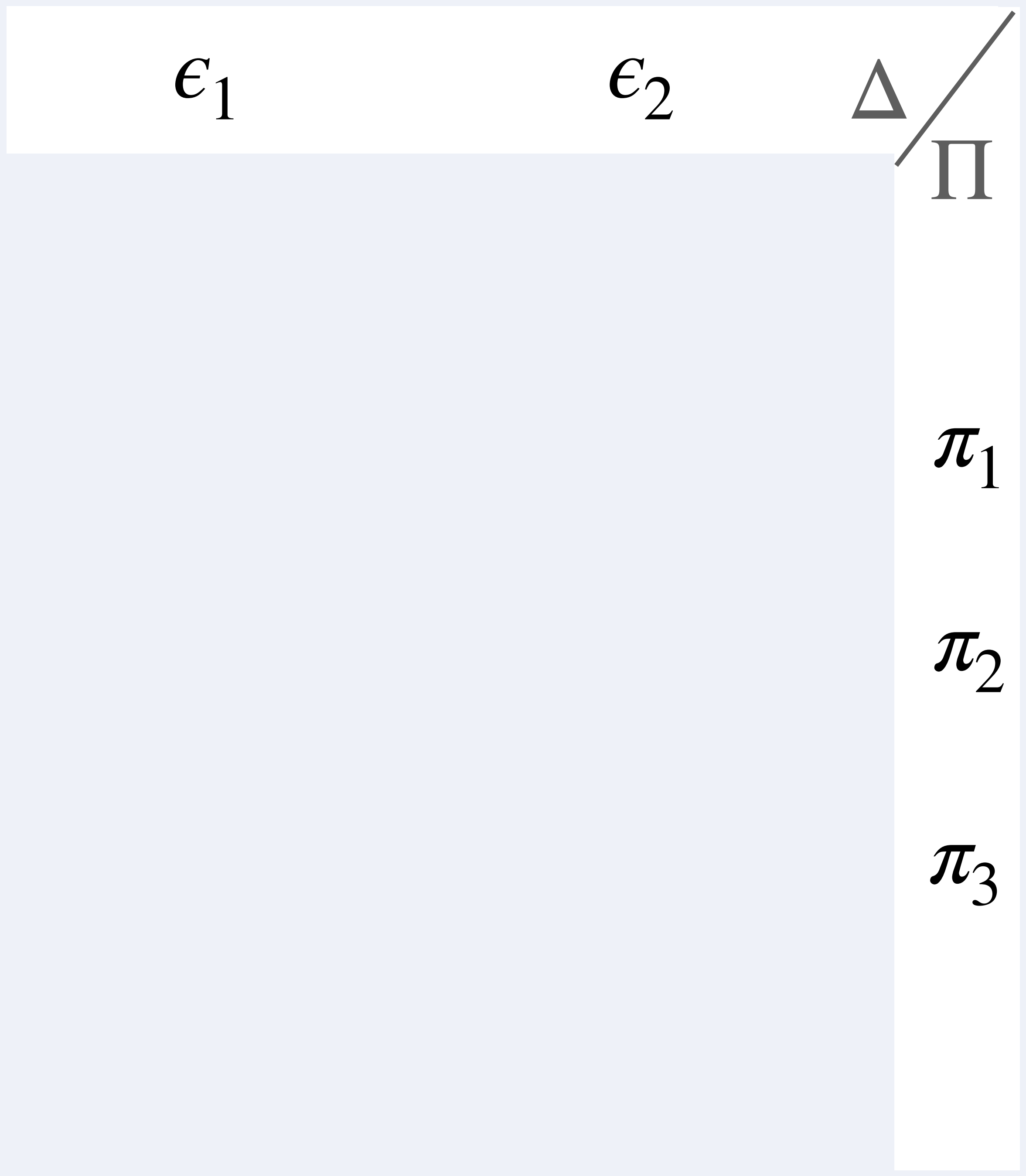
First-Order Standpoint Logic: Semantics

Semantics of $\mathbb{S}$

Relational semantics:

$$\mathcal{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$$

- $\Delta = \{\epsilon_1, \epsilon_2\}$ domain
- $\Pi = \{\pi_1, \pi_2, \pi_3\}$ worlds



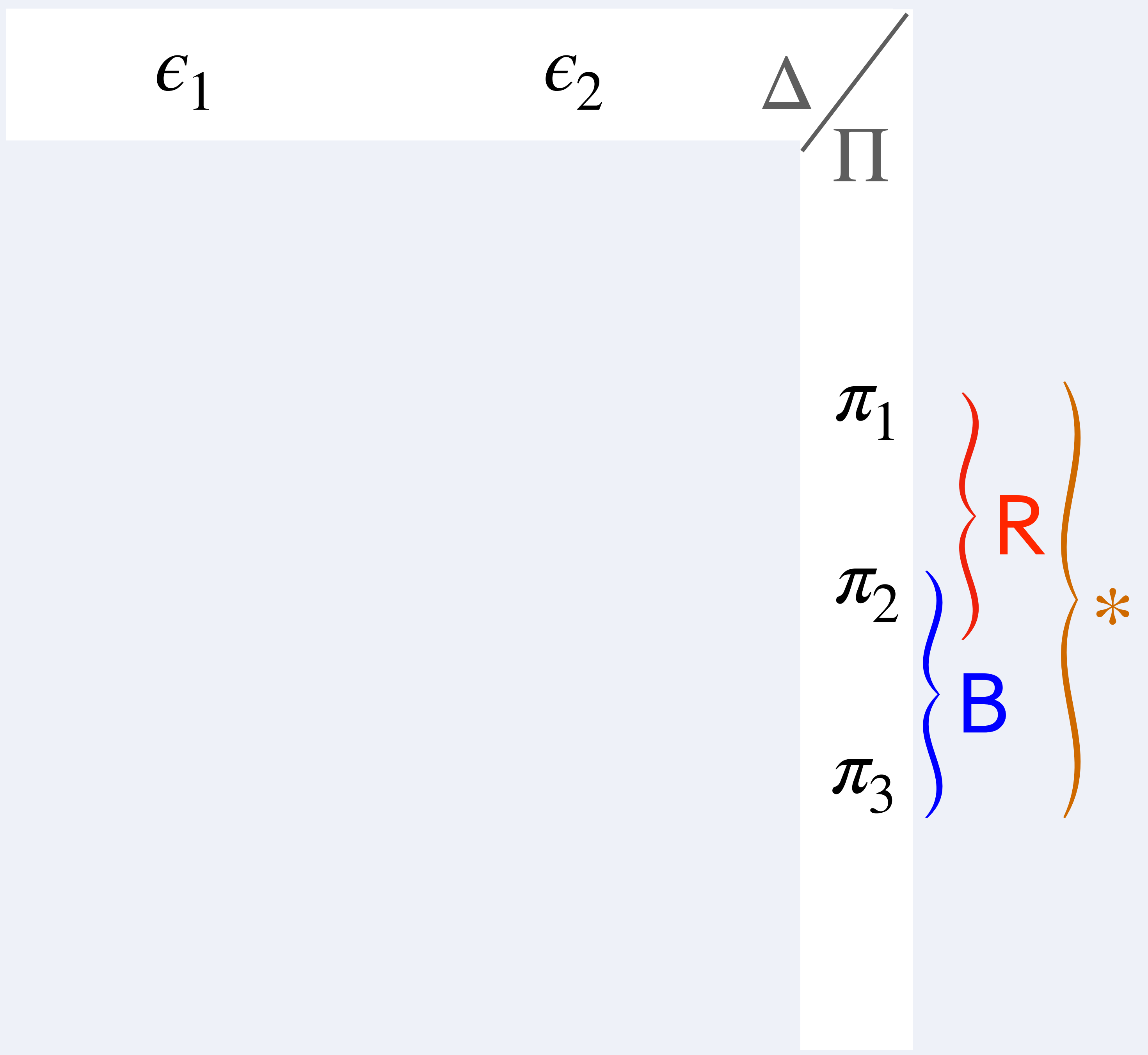
First-Order Standpoint Logic: Semantics

Semantics of $\mathbb{S}$

Relational semantics:

$$\mathcal{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$$

- $\Delta = \{\epsilon_1, \epsilon_2\}$ domain
- $\Pi = \{\pi_1, \pi_2, \pi_3\}$ worlds
- $\sigma(R) = \{\pi_1, \pi_2\}$ standpoint
  $\sigma(B) = \{\pi_2, \pi_3\}$ assignment



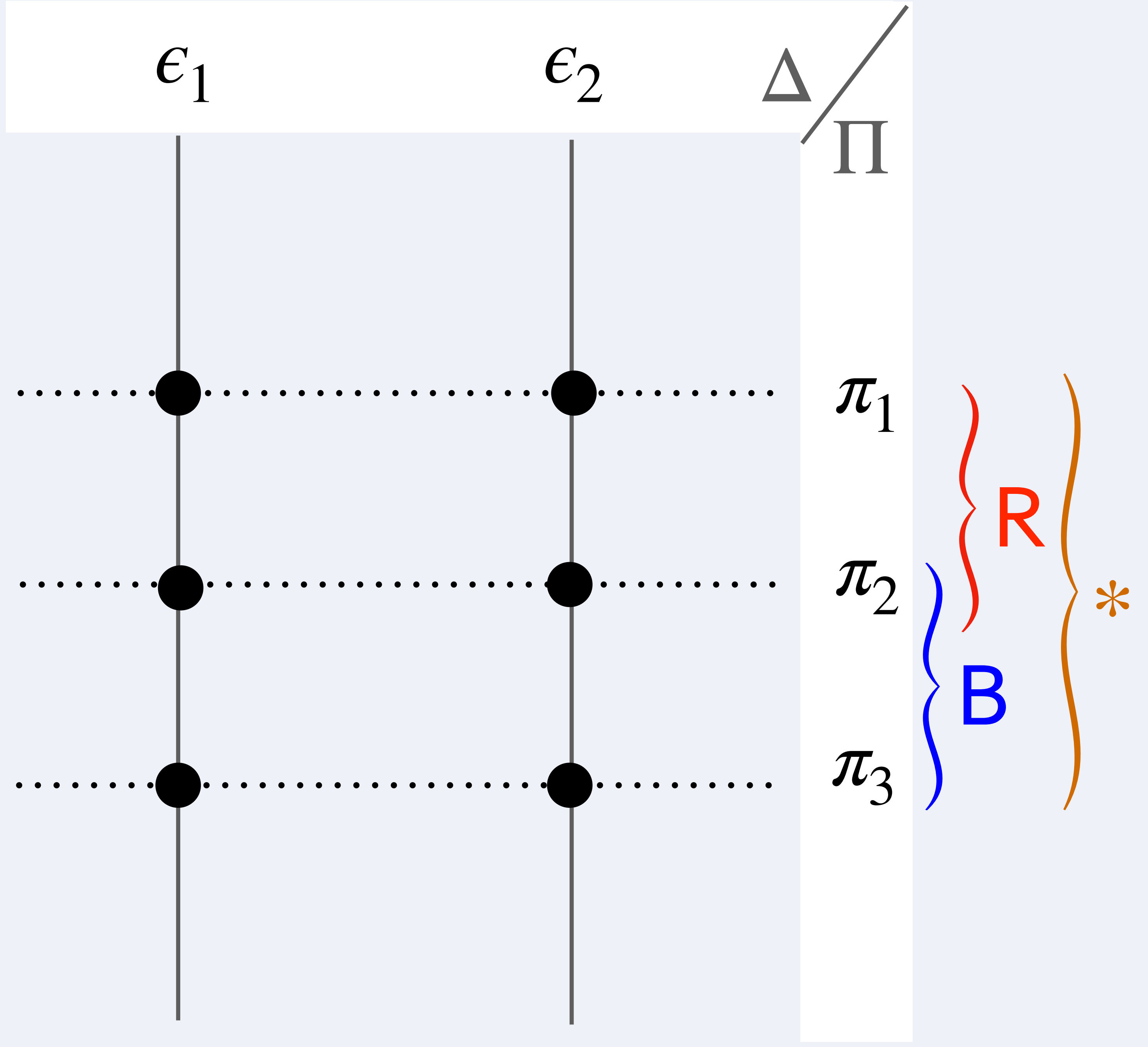
First-Order Standpoint Logic: Semantics

Semantics of $\mathbb{S}$

Relational semantics:

$$\mathscr{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$$

- $\Delta = \{\epsilon_1, \epsilon_2\}$ domain
- $\Pi = \{\pi_1, \pi_2, \pi_3\}$ worlds
- $\sigma(\mathsf{R}) = \{\pi_1, \pi_2\}$ standpoint
  $\sigma(\mathsf{B}) = \{\pi_2, \pi_3\}$ assignment
- $\gamma(\pi_1) = \{p \mapsto \emptyset, \ldots\}$ interpretation assignment to worlds



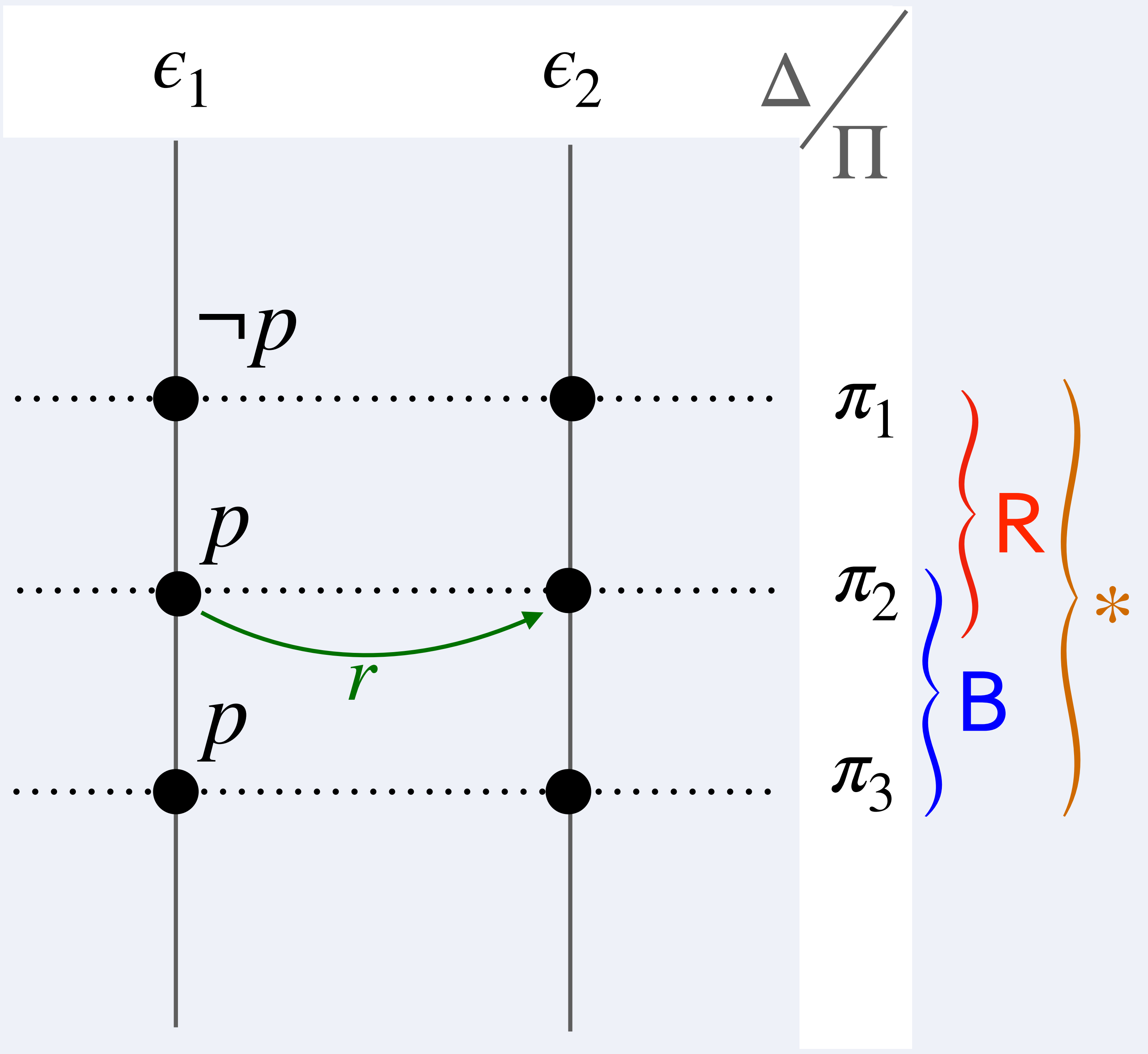
First-Order Standpoint Logic: Semantics

Semantics of $\mathbb{S}$

Relational semantics:

$$\mathscr{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$$

- $\Delta = \{\epsilon_1, \epsilon_2\}$ domain
- $\Pi = \{\pi_1, \pi_2, \pi_3\}$ worlds
- $\sigma(\mathsf{R}) = \{\pi_1, \pi_2\}$ standpoint
  $\sigma(\mathsf{B}) = \{\pi_2, \pi_3\}$ assignment
- $\gamma(\pi_1) = \{p \mapsto \emptyset, \dots\}$ interpretation assignment to worlds



First-Order Standpoint Logic: Semantics

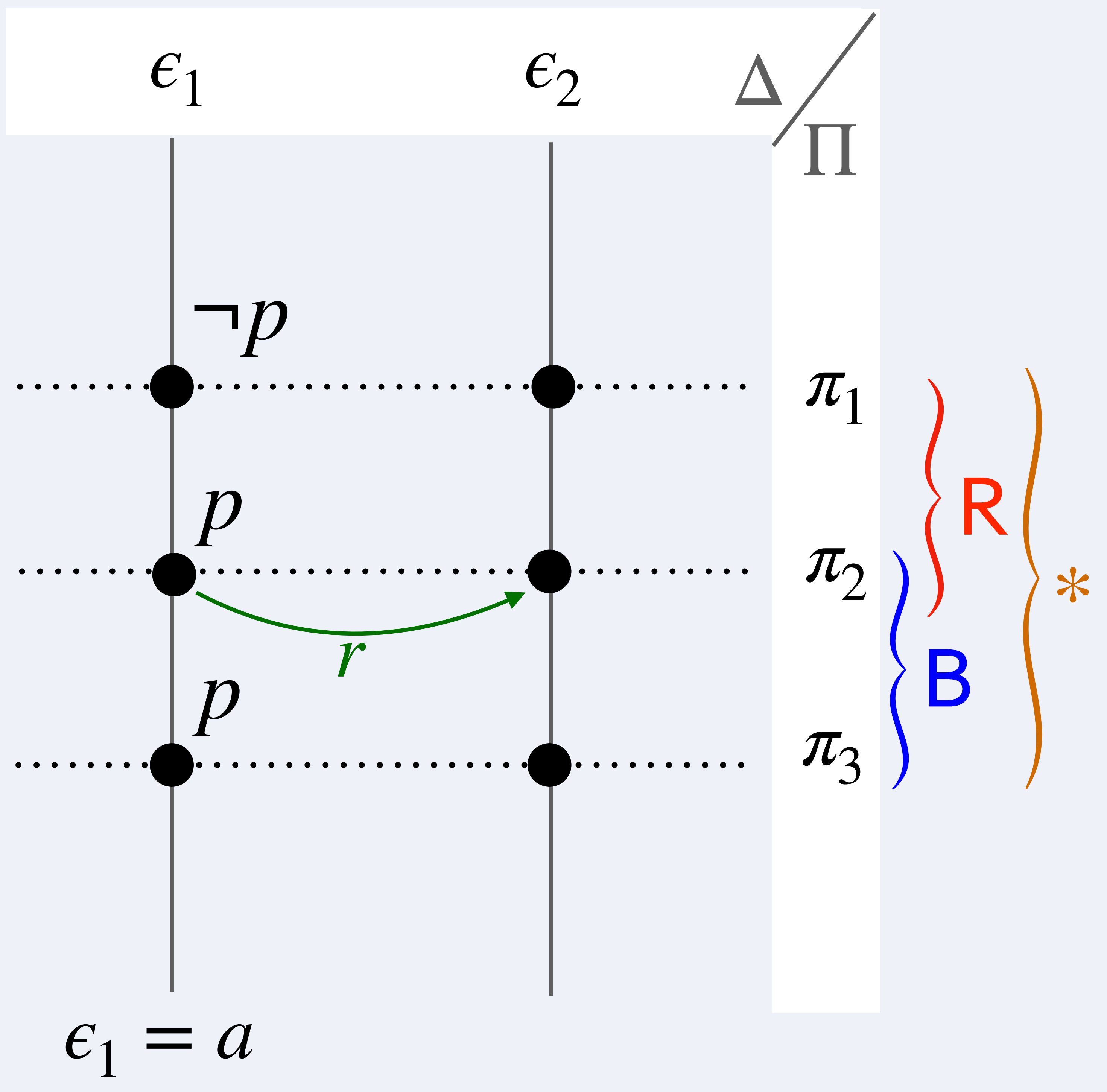
Semantics of $\mathbb{S}$

Relational semantics:

$$\mathcal{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$$

- $\Delta = \{\epsilon_1, \epsilon_2\}$ domain
- $\Pi = \{\pi_1, \pi_2, \pi_3\}$ worlds
- $\sigma(\mathsf{R}) = \{\pi_1, \pi_2\}$ standpoint
  $\sigma(\mathsf{B}) = \{\pi_2, \pi_3\}$ assignment
- $\gamma(\pi_1) = \{p \mapsto \emptyset, \ldots\}$ interpretation assignment to worlds

*Rigid domains and constants



First-Order Standpoint Logic: Semantics

Semantics of $\mathbb{S}$

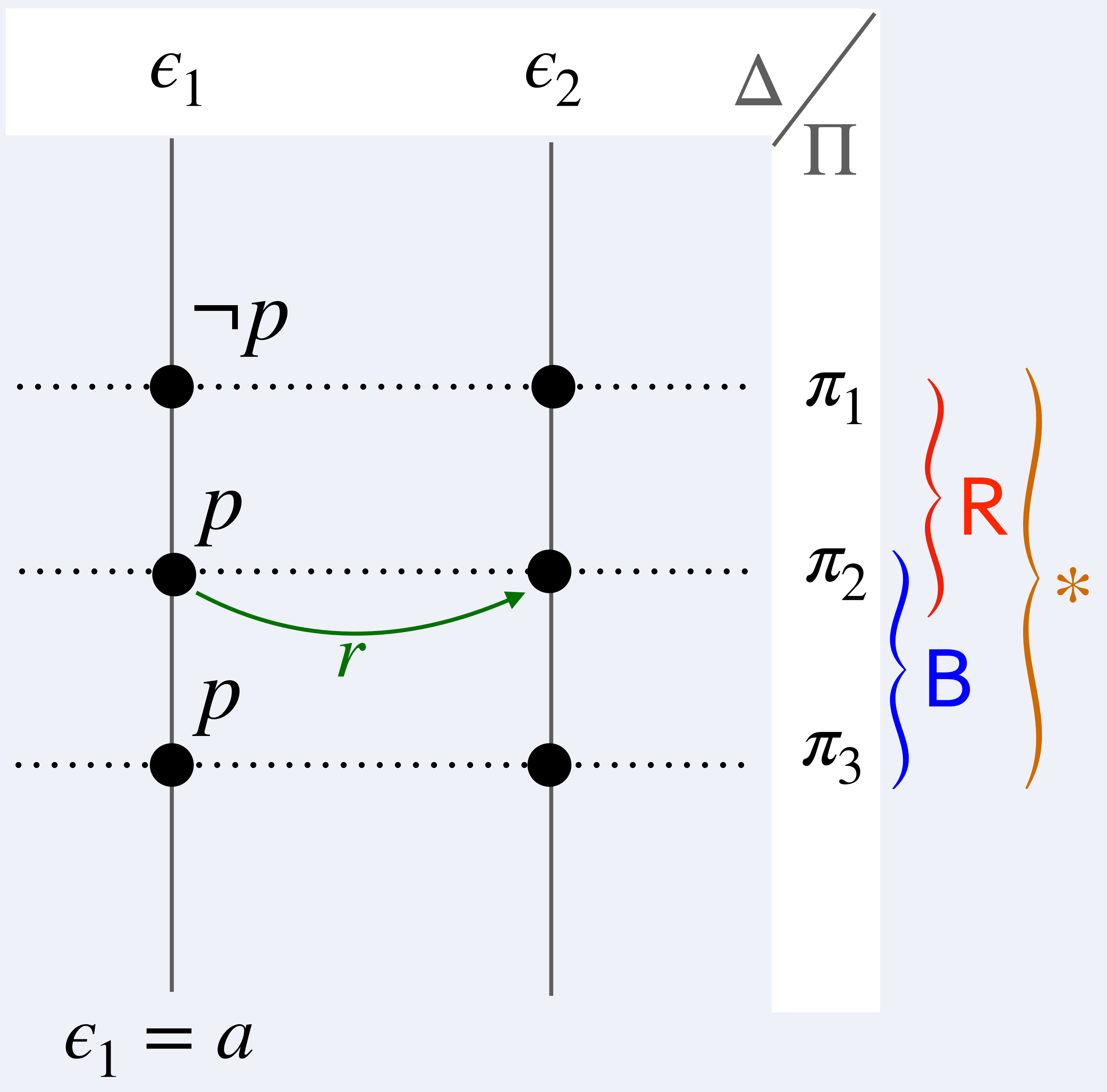
Relational semantics:

$$\mathscr{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$$

- $\Delta = \{\epsilon_1, \epsilon_2\}$ domain
- $\Pi = \{\pi_1, \pi_2, \pi_3\}$ worlds
- $\sigma(\text{R}) = \{\pi_1, \pi_2\}$ standpoint
  $\sigma(\text{B}) = \{\pi_2, \pi_3\}$ assignment
- $\gamma(\pi_1) = \{p \mapsto \emptyset, \dots\}$ interpretation assignment to worlds

- $\mathscr{M} \models \Box_{\text{B}}\, p(a)$

*Rigid domains and constants



First-Order Standpoint Logic: Semantics

Semantics of $\mathbb{S}$

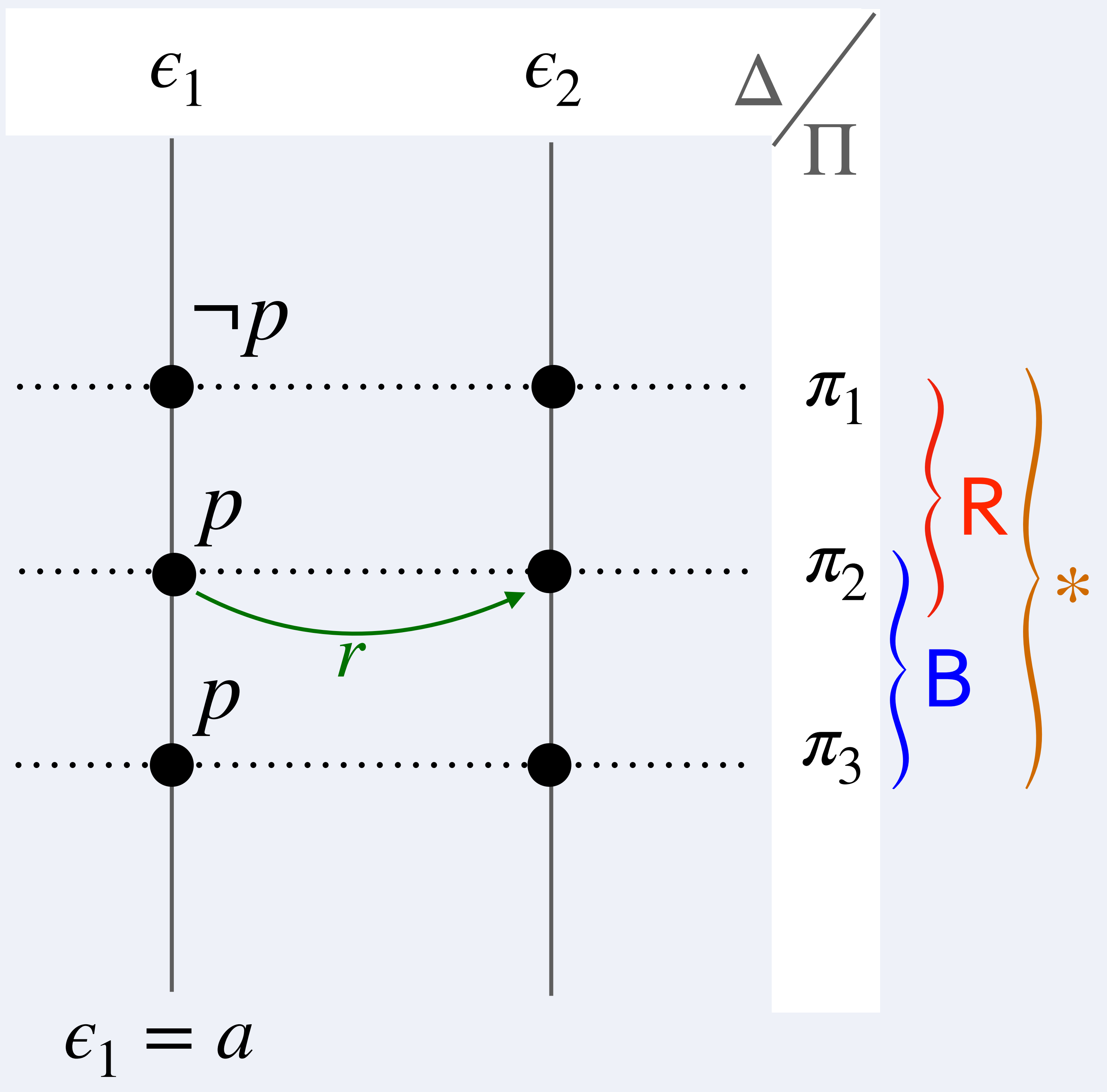
Relational semantics:

$$\mathscr{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$$

- $\Delta = \{\epsilon_1, \epsilon_2\}$ domain
- $\Pi = \{\pi_1, \pi_2, \pi_3\}$ worlds
- $\sigma(\mathsf{R}) = \{\pi_1, \pi_2\}$ standpoint
  $\sigma(\mathsf{B}) = \{\pi_2, \pi_3\}$ assignment
- $\gamma(\pi_1) = \{p \mapsto \emptyset, \ldots\}$ interpretation assignment to worlds

- $\mathscr{M} \models \square_{\mathsf{B}}\, p(a)$
- $\mathscr{M} \models \exists x.\big(\lozenge_{\mathsf{R}} p(x) \land \lozenge_{\mathsf{R}} \neg p(x)\big)$

*Rigid domains and constants



First-Order Standpoint Logic: Semantics

Semantics of $\mathbb{S}$

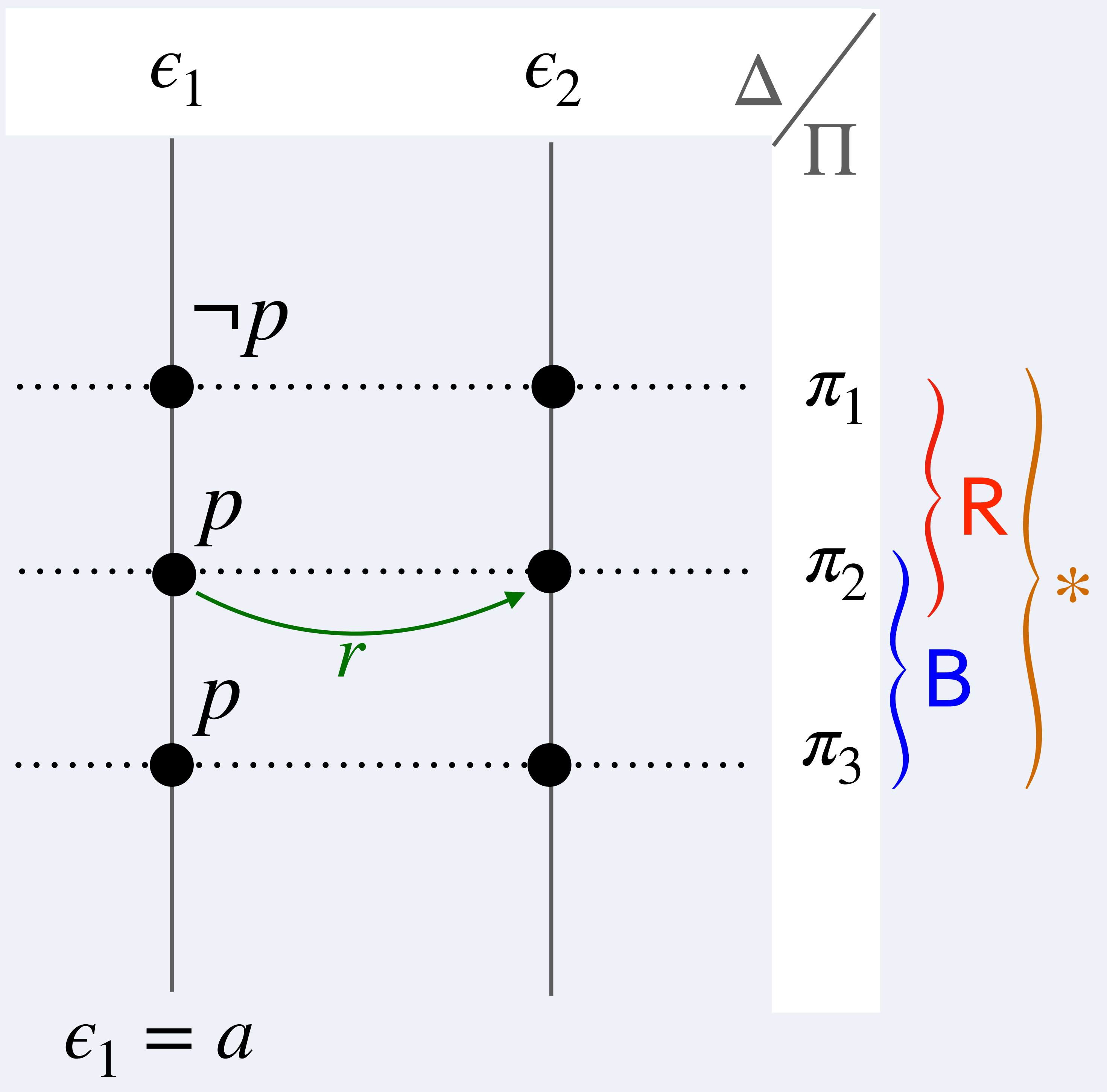
Relational semantics:

$$\mathcal{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$$

- $\Delta = \{\epsilon_1, \epsilon_2\}$ domain
- $\Pi = \{\pi_1, \pi_2, \pi_3\}$ worlds
- $\sigma(\mathsf{R}) = \{\pi_1, \pi_2\}$ standpoint
  $\sigma(\mathsf{B}) = \{\pi_2, \pi_3\}$ assignment
- $\gamma(\pi_1) = \{p \mapsto \emptyset, \ldots\}$ interpretation assignment to worlds

- $\mathcal{M} \models \square_{\mathsf{B}}\, p(a)$
- $\mathcal{M} \models \exists x.\big(\lozenge_{\mathsf{R}} p(x) \land \lozenge_{\mathsf{R}} \neg p(x)\big)$
- $\mathcal{M} \models \square_{\mathsf{R}} \forall x.p(x) \rightarrow (\exists y.r(x,y))$

*Rigid domains and constants



Transformations

Stepwise Translation

monodic C^2 FOSL → <u>nullary-</u> and <u>constant-free</u> <u>S5</u> monodic C^2 FOSL

Stepwise Translation

monodic C^2 FOSL ⟶ <u>nullary-</u> and <u>constant-free</u> <u>S5</u> monodic C^2 FOSL

$$\exists x.\Box_{\mathsf{L}} \left(\mathsf{Tumor}(x) \wedge \mathsf{hasTissue}(John, x)\right)$$

Stepwise Translation

monodic C^2 FOSL ➡ <u>nullary-</u> and <u>constant-free</u> <u>S5</u> monodic C^2 FOSL

$$\exists x.\Box_{\mathsf{L}}\left(\mathsf{Tumor}(x) \wedge \mathsf{hasTissue}(John, x)\right)$$

⬇

$$\exists x.\Box_{*}\left(\mathsf{L} \rightarrow (\mathsf{Tumor}(x) \wedge \mathsf{hasTissue}(John, x))\right)$$

S5: no standpoint expressions except *

Simulate standpoint expressions by marking worlds with nullary predicates.

Stepwise Translation

monodic C^2 FOSL → nullary- and constant-free S5 monodic C^2 FOSL

$$\exists x.\square_{\mathsf{L}} \big(\mathsf{Tumor}(x) \land \mathsf{hasTissue}(John, x)\big)$$

↓

$$\exists x.\square_{*} \big(\mathsf{L} \to (\mathsf{Tumor}(x) \land \mathsf{hasTissue}(John, x))\big)$$

↓

$$\exists x.\square_{*} \big(\forall x.(\mathsf{L}'(x)) \to (\mathsf{Tumor}(x) \land \mathsf{hasTissue}(John, x))\big)$$

S5: no standpoint expressions except *

Simulate standpoint expressions by marking worlds with nullary predicates.

nullary-free: all predicates of arity ≥1

Easy: Simulate nullary predicates by immediately quantified unary ones.

Stepwise Translation

monodic C^2 FOSL → nullary- and constant-free S5 monodic C^2 FOSL

$$\exists x.\Box_{\mathsf{L}} \left(\mathsf{Tumor}(x) \wedge \mathsf{hasTissue}(John, x)\right)$$

↓

$$\exists x.\Box_{*} \left(\mathsf{L} \rightarrow (\mathsf{Tumor}(x) \wedge \mathsf{hasTissue}(John, x))\right)$$

↓

$$\exists x.\Box_{*} \left(\forall x.(\mathsf{L}'(x)) \rightarrow (\mathsf{Tumor}(x) \wedge \mathsf{hasTissue}(John, x))\right)$$

↓

$$\exists x.\Box_{*} \left(\forall x.(\mathsf{L}'(x)) \rightarrow \left(\mathsf{Tumor}(x) \wedge \exists y.(\mathsf{hasTissue}(y, x) \wedge \mathsf{John}(y))\right)\right)$$
$$\wedge\ \exists^{=1} x\,.\,\mathsf{John}(x) \wedge \exists^{=1} x\,.\,\Box_{*}\,\mathsf{John}(x)$$

S5: no standpoint expressions except *

Simulate standpoint expressions by marking worlds with nullary predicates.

nullary-free: all predicates of arity ≥1

Easy: Simulate nullary predicates by immediately quantified unary ones.

constant-free: no constants

Simulate constants by unary predicates (axiomatising uniqueness and rigidity).

Satisfiability in Monodic Standpoint C^2

Context

➡ C^2 is NExpTime-complete

Context

- ➡ C^2 is NExpTime-complete
- ➡ for other logics' standpoint extensions, we often relied on existence of models with few worlds ("small model property")

Context

- ➡ C^2 is NExpTime-complete
- ➡ for other logics' standpoint extensions, we often relied on existence of models with few worlds ("small model property")
- ➡ yet, here we can enforce infinitely many worlds:

$$\Box_* \forall x. \Big(\big(\mathsf{First}(x) \leftrightarrow \neg \exists y\, \mathsf{Next}(y,x)\big) \wedge \exists^{\leq 1} y\, \big(\mathsf{Next}(y,x)\big) \wedge \exists^{=1} y \big(\mathsf{Next}(x,y)\big) \Big)$$

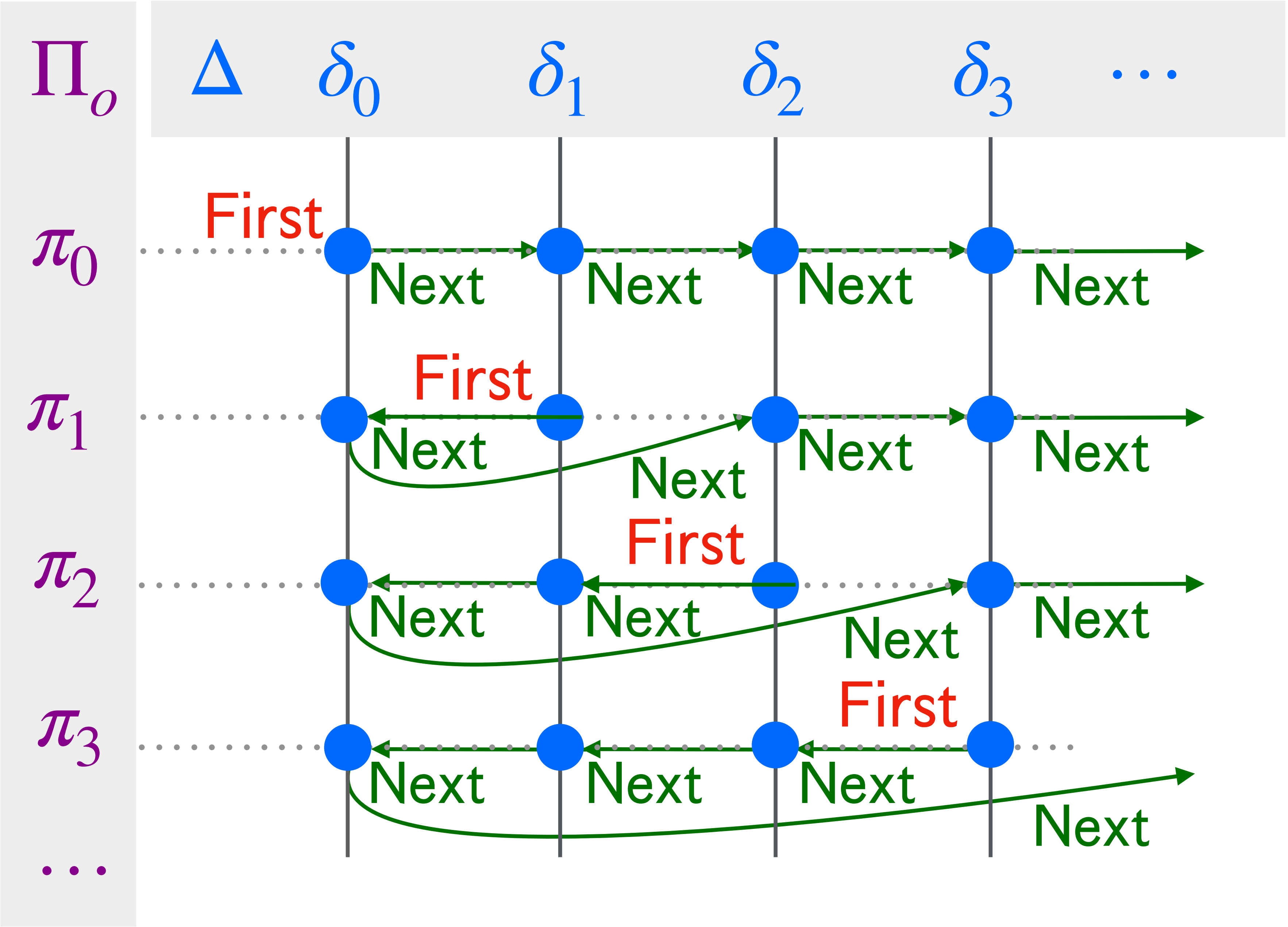
infinite domain

$$\forall x. \Diamond_* \big(\mathsf{First}(x)\big)$$

every entity is first in some world

$$\Box_* \exists^{=1} x. \big(\mathsf{First}(x)\big)$$

only one first entity per world



Argument Overview

(nullary- and constant-free S5)

monodic C^2 FOSL formula

Argument Overview

(nullary- and constant-free S5)

monodic C^2 FOSL formula

satisfied in

FOSL
Structure

Argument Overview

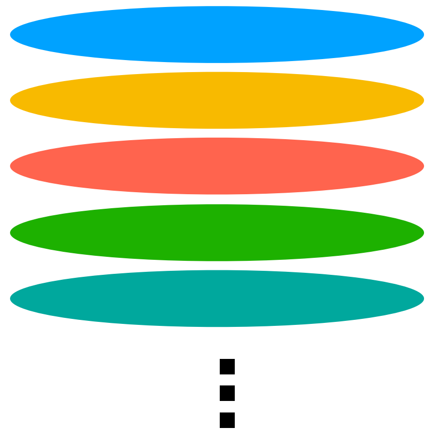
(nullary- and constant-free S5)

monodic C^2 FOSL formula

satisfied in ↓

FOSL Structure

(possibly infinite, irregular)



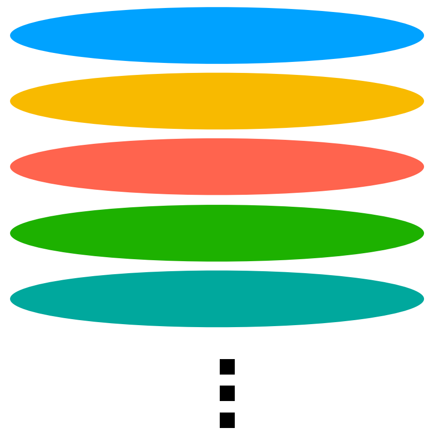
Argument Overview

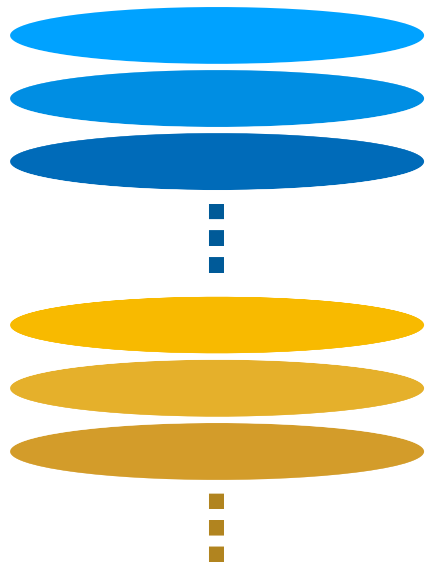
(nullary- and constant-free S5)

monodic C^2 FOSL formula

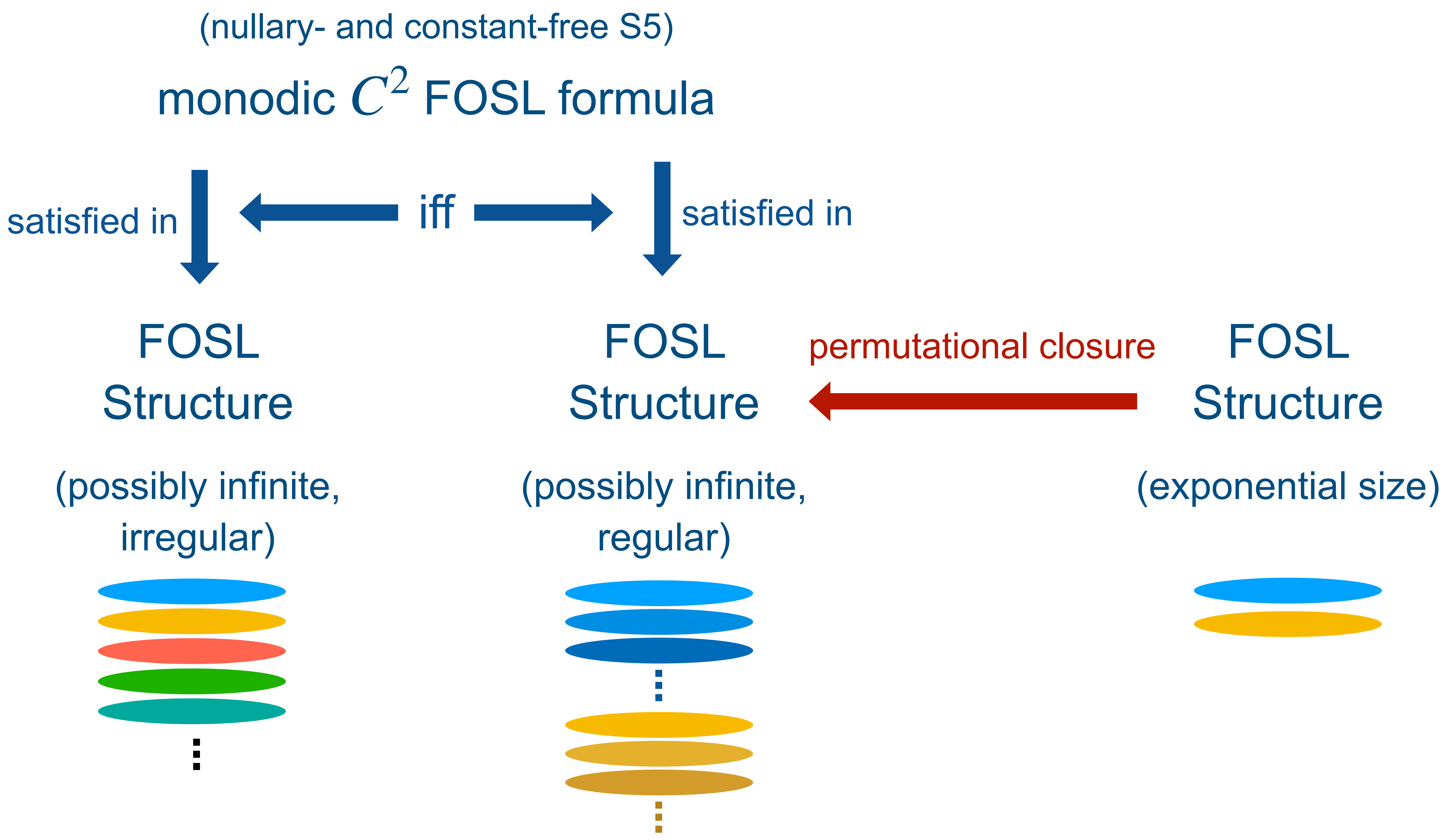
satisfied in ← iff → satisfied in

FOSL Structure	FOSL Structure
(possibly infinite, irregular)	(possibly infinite, regular)

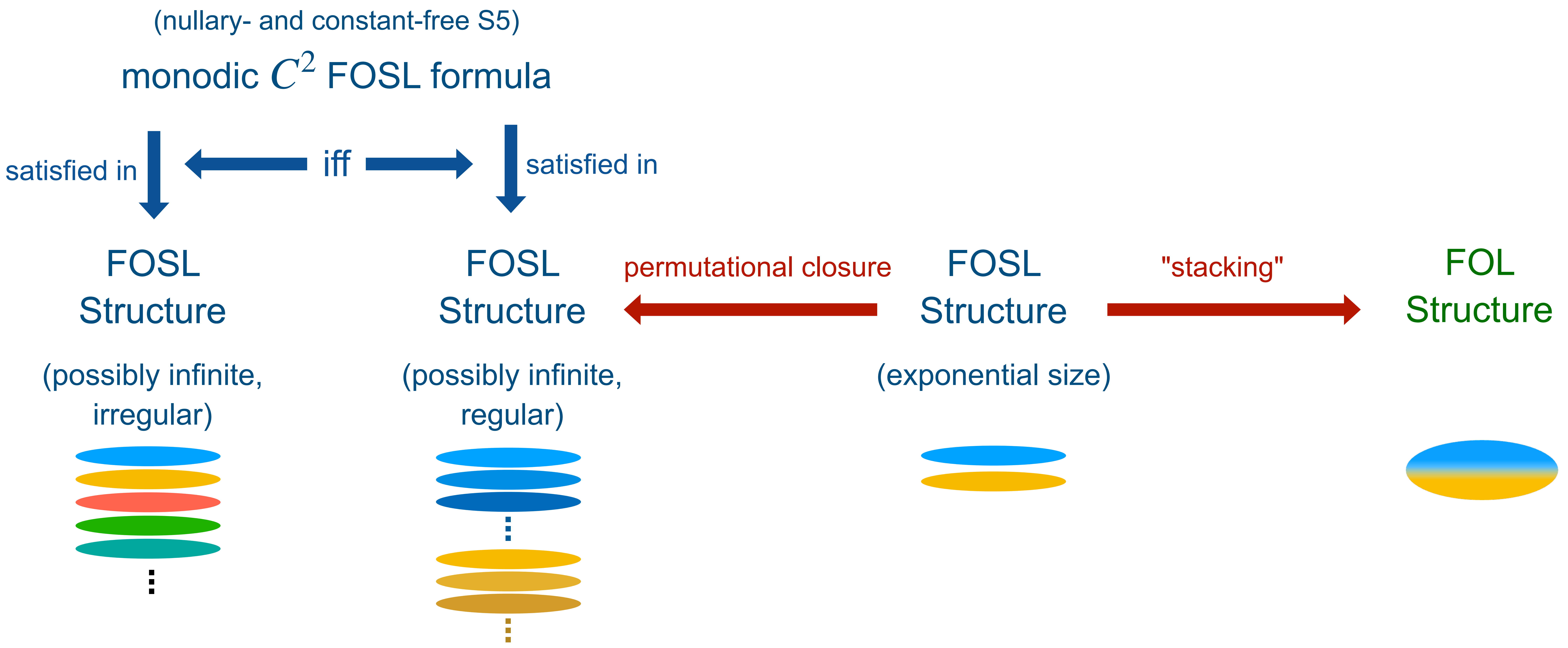




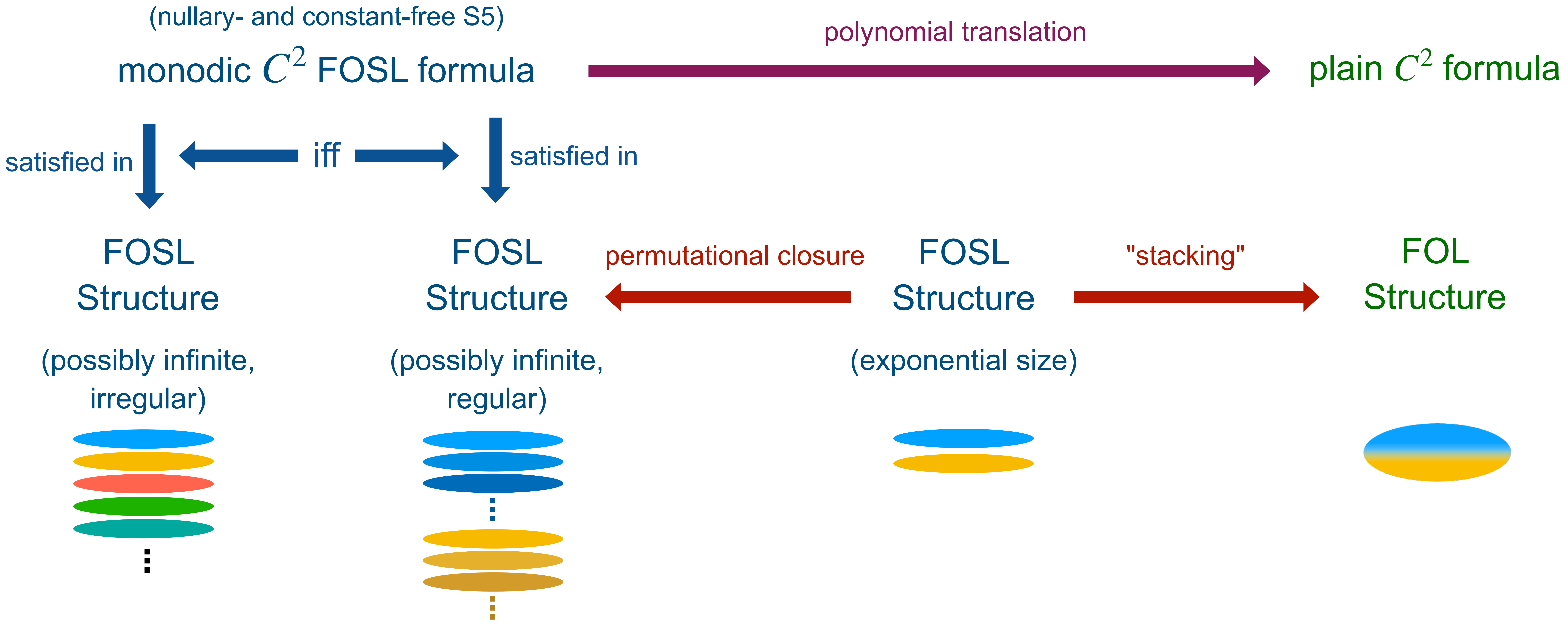
Argument Overview



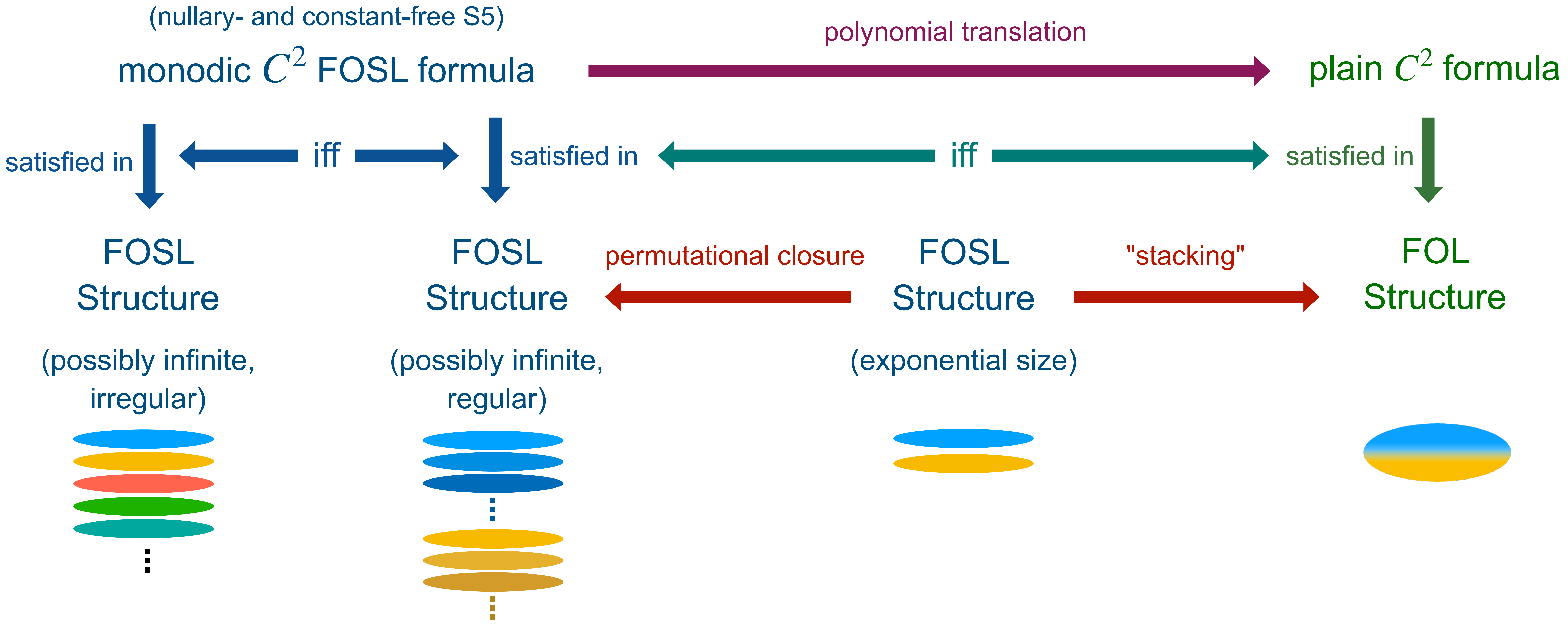
Argument Overview



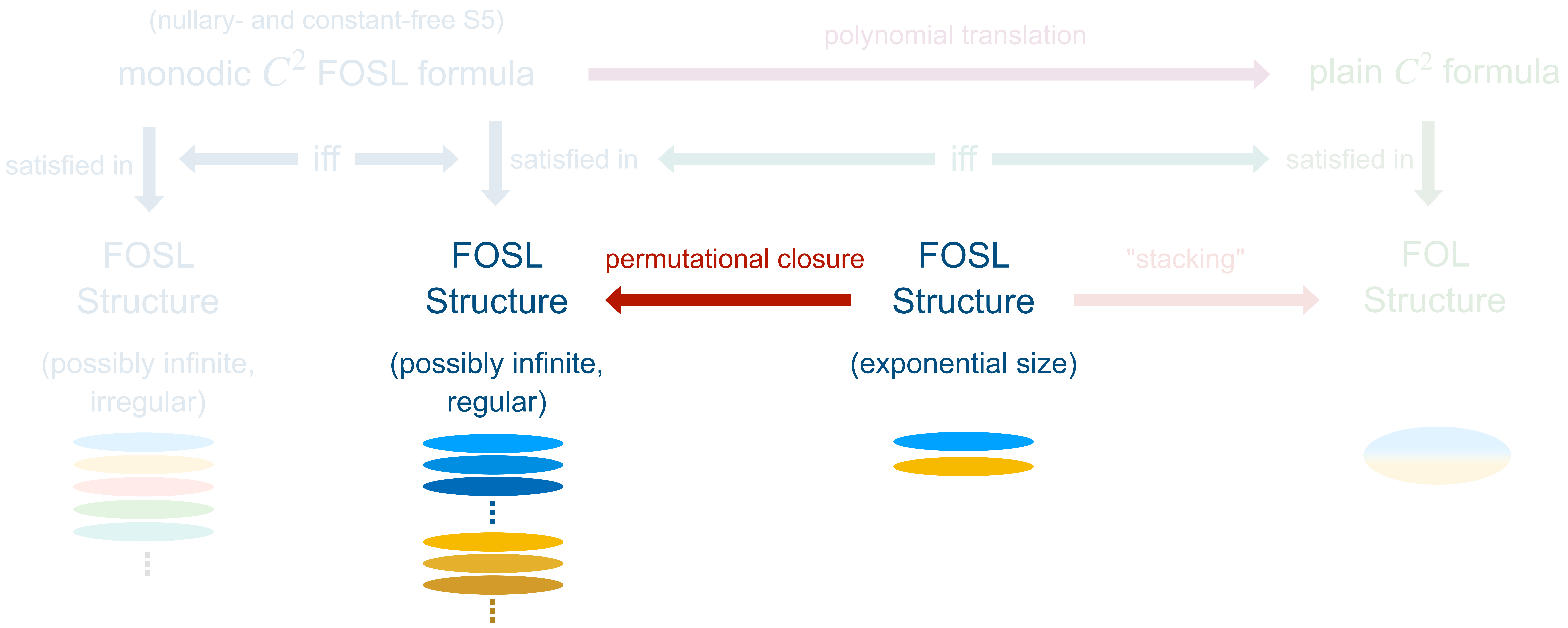
Argument Overview



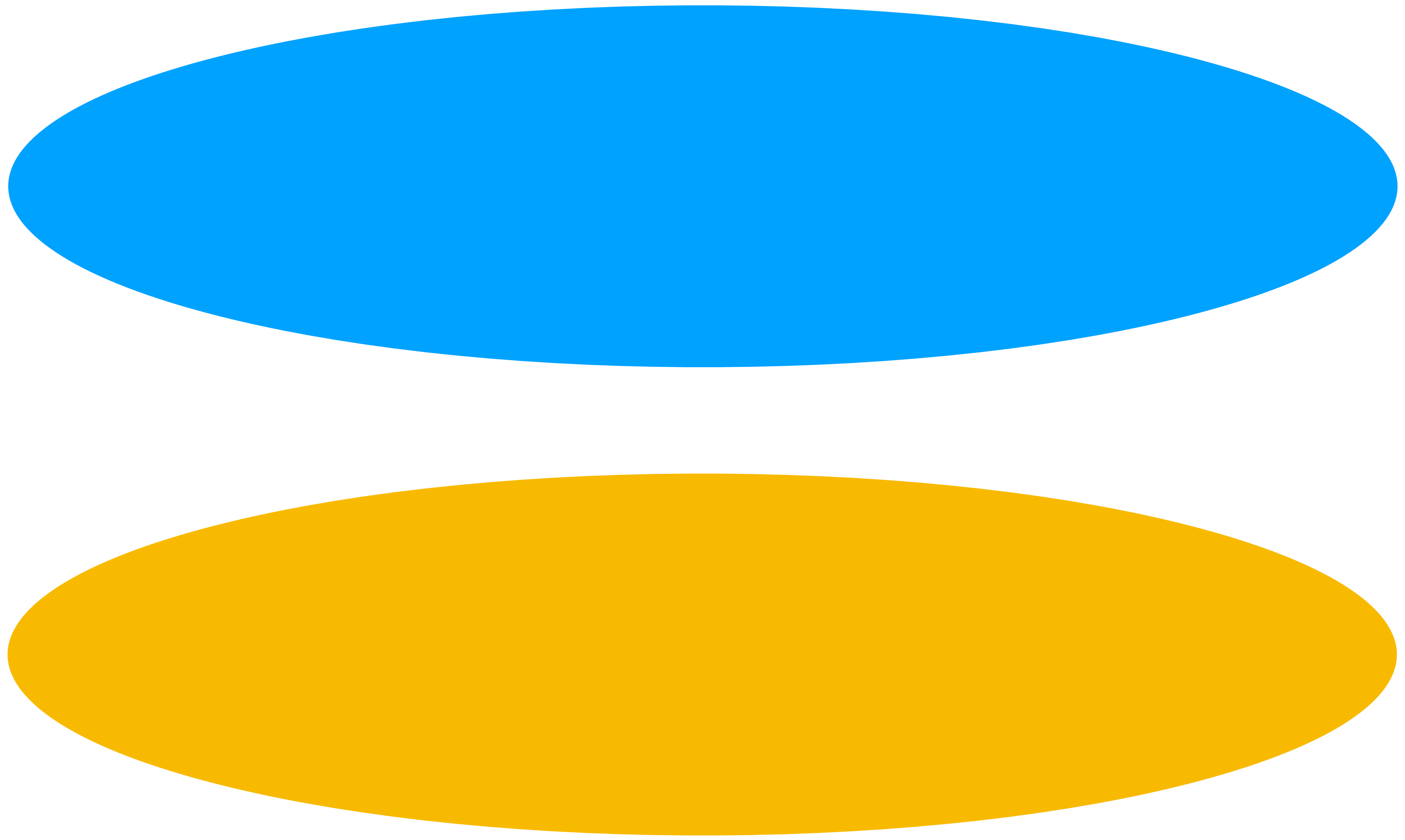
Argument Overview



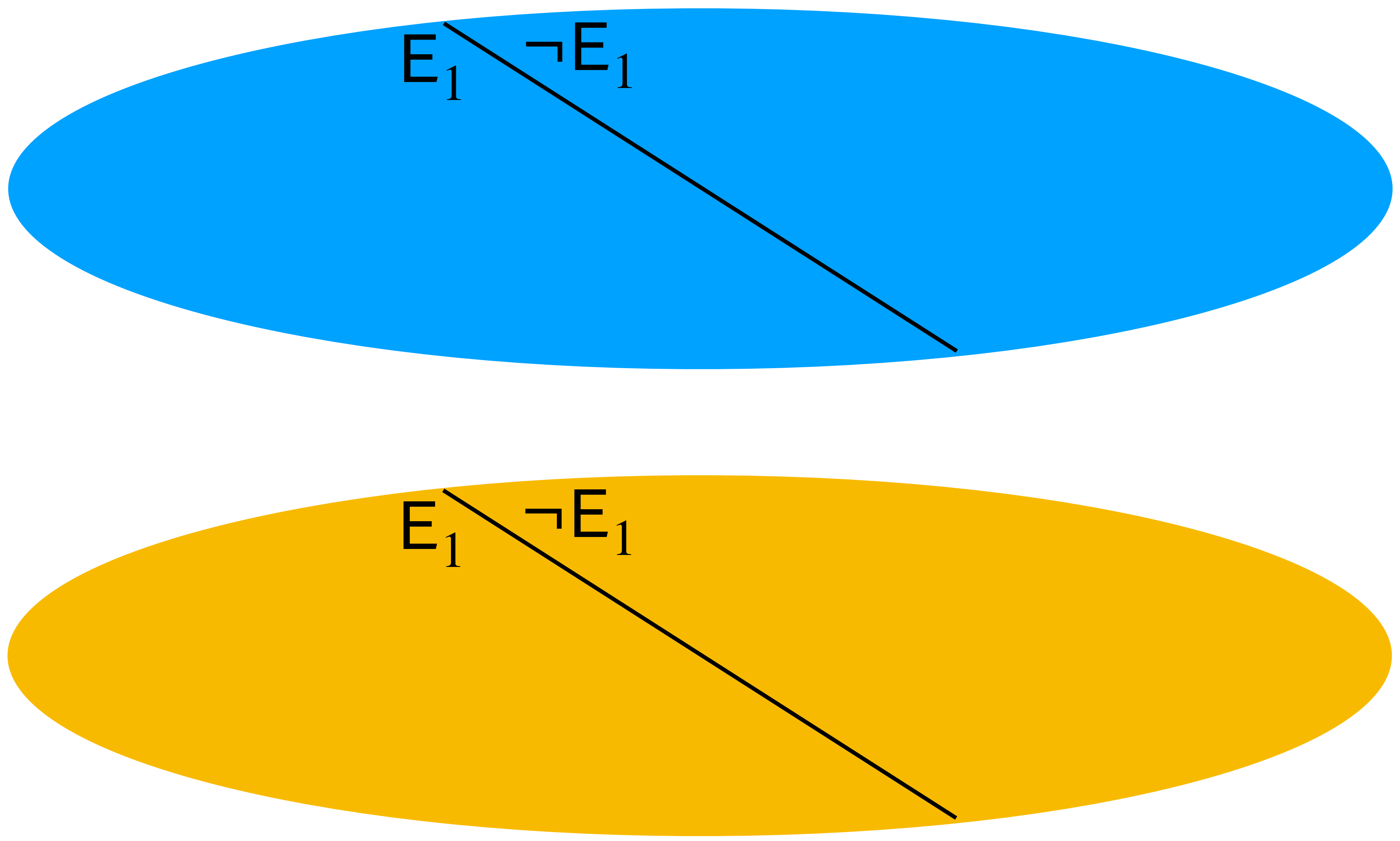
Argument Overview



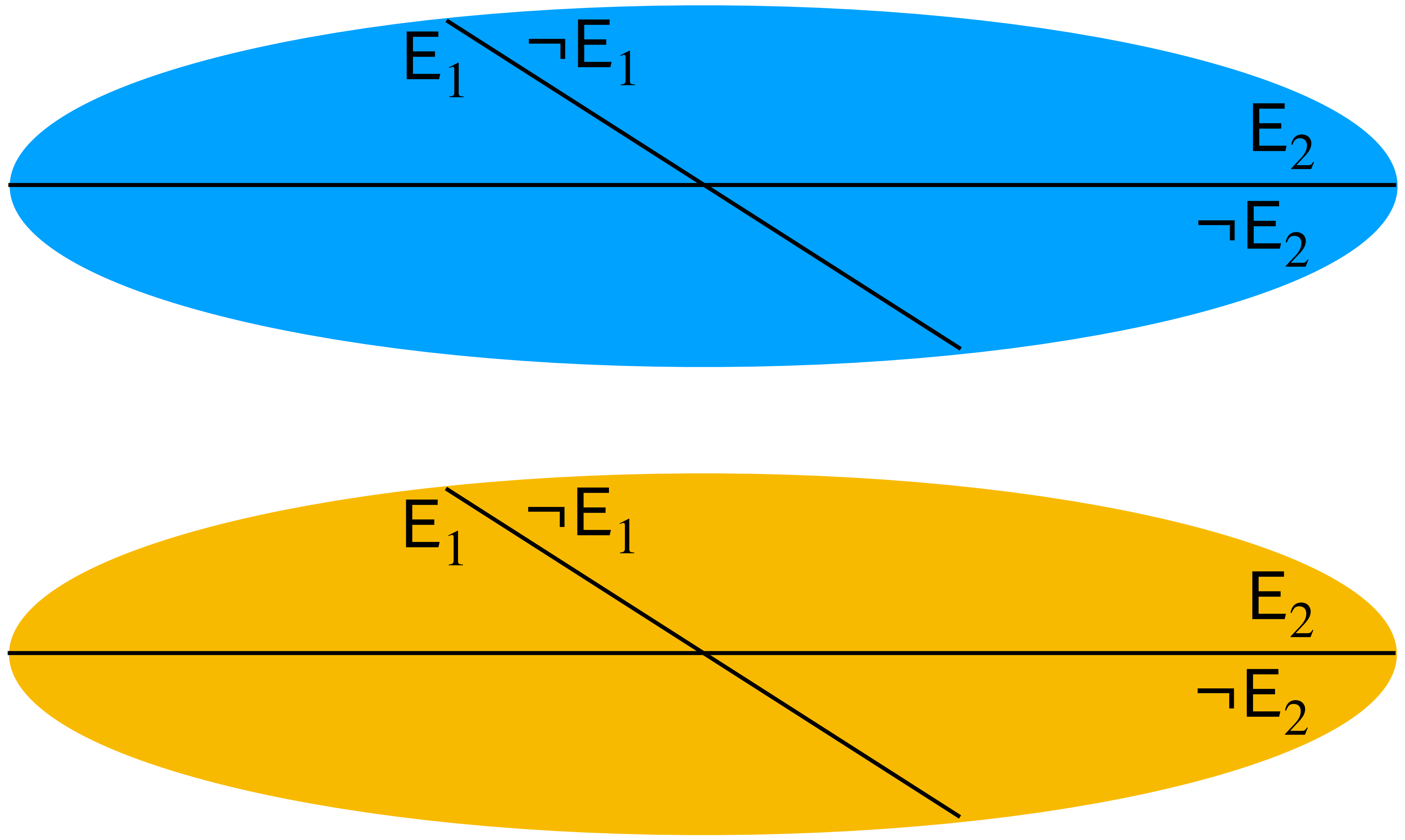
Permutational Closure



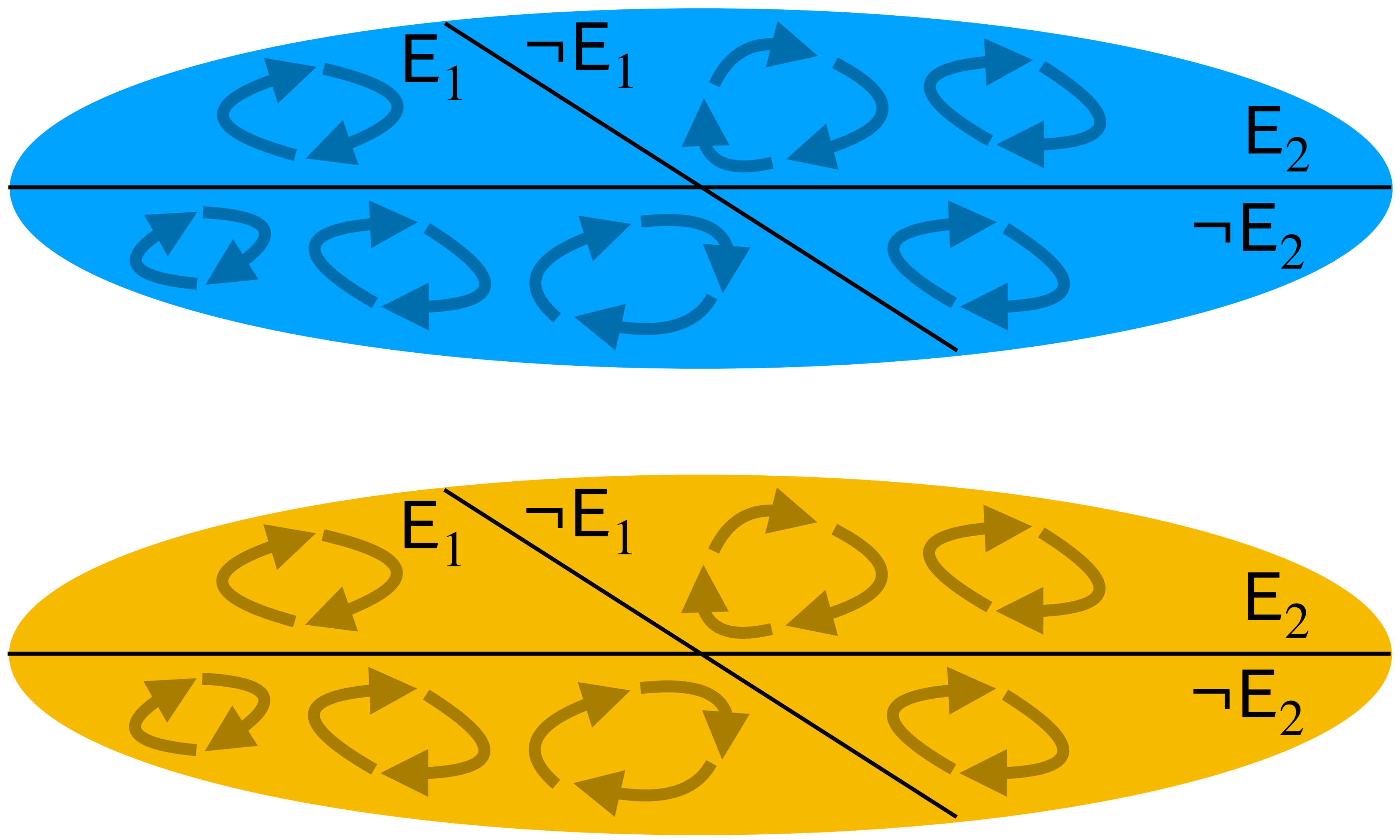
Permutational Closure



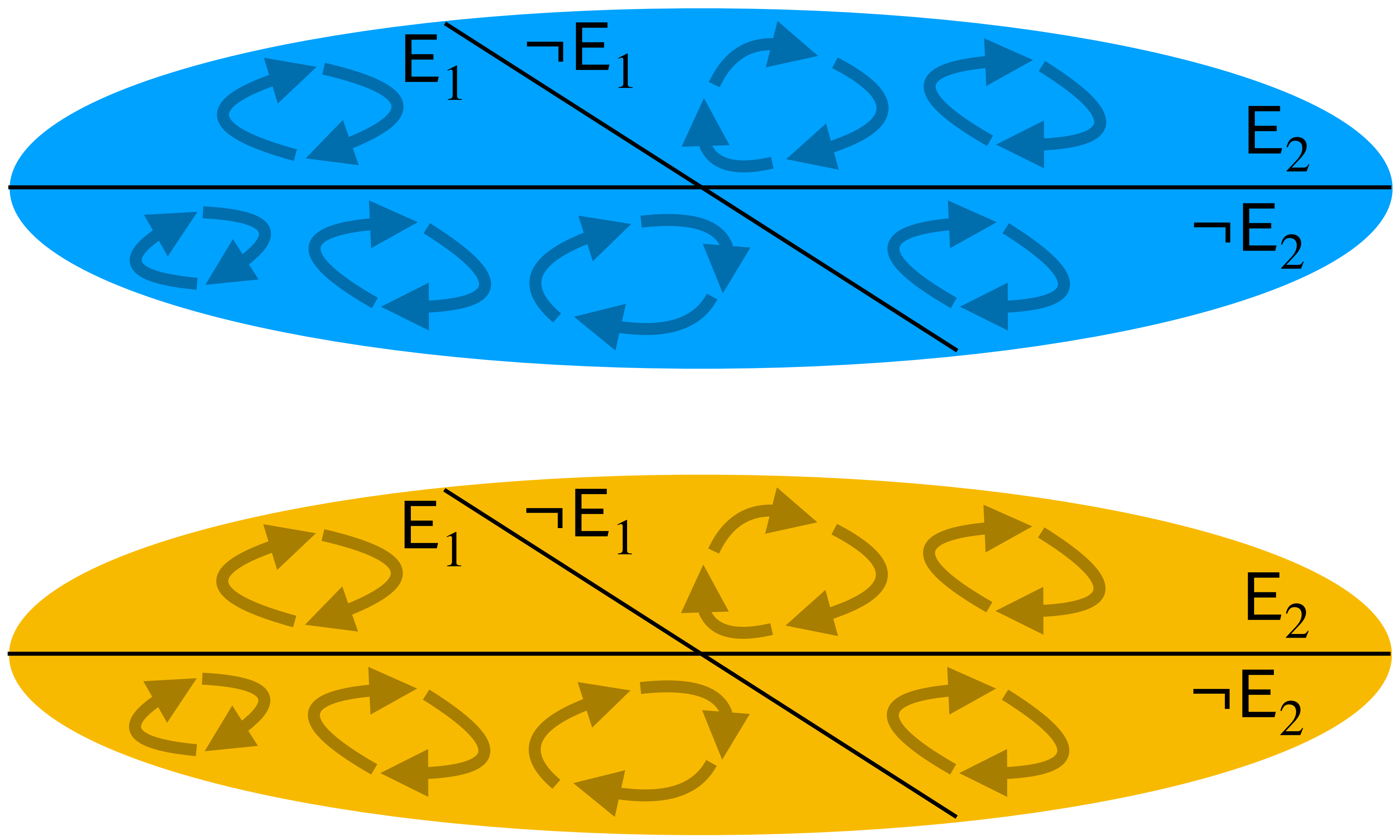
Permutational Closure

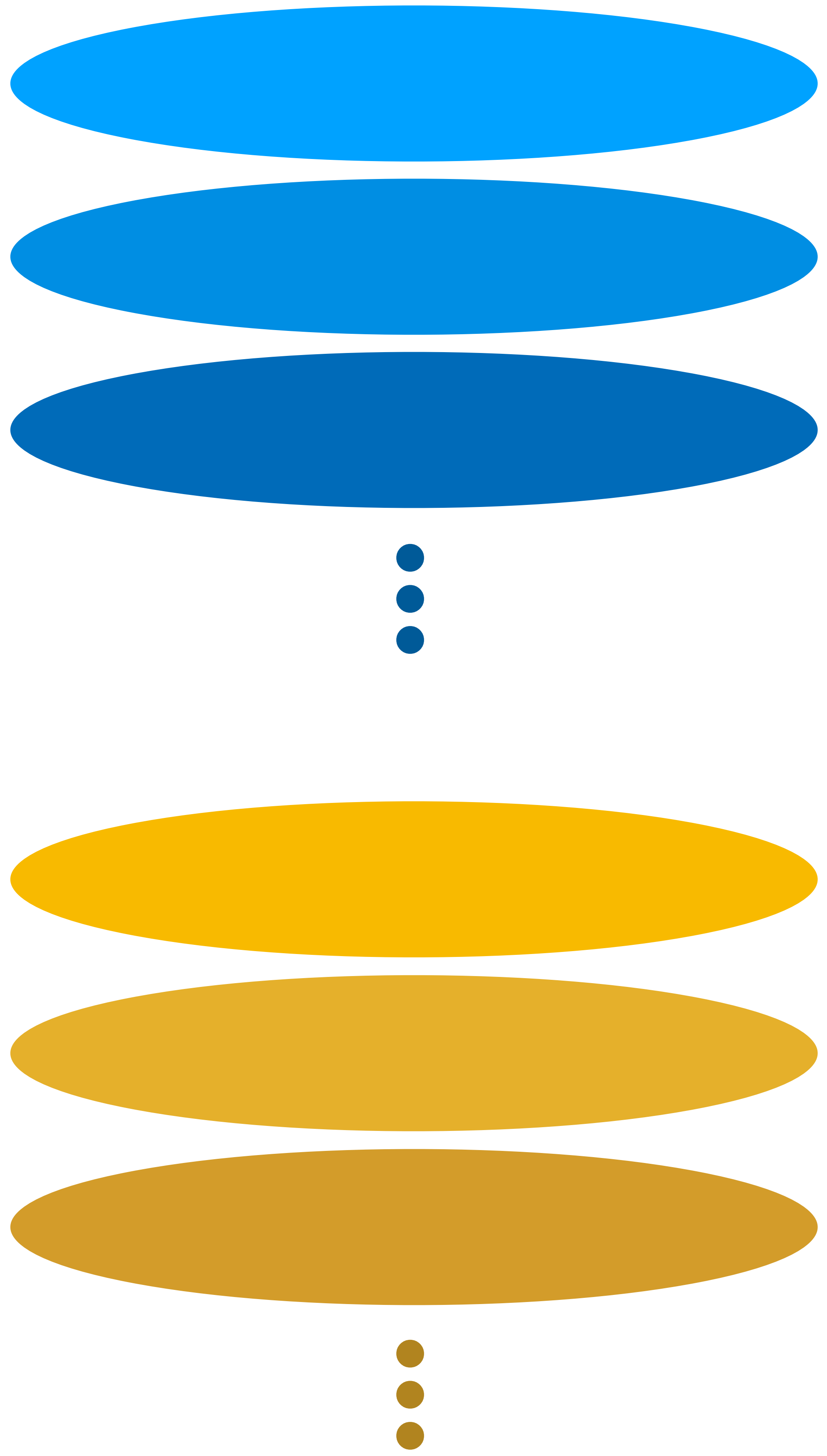


Permutational Closure

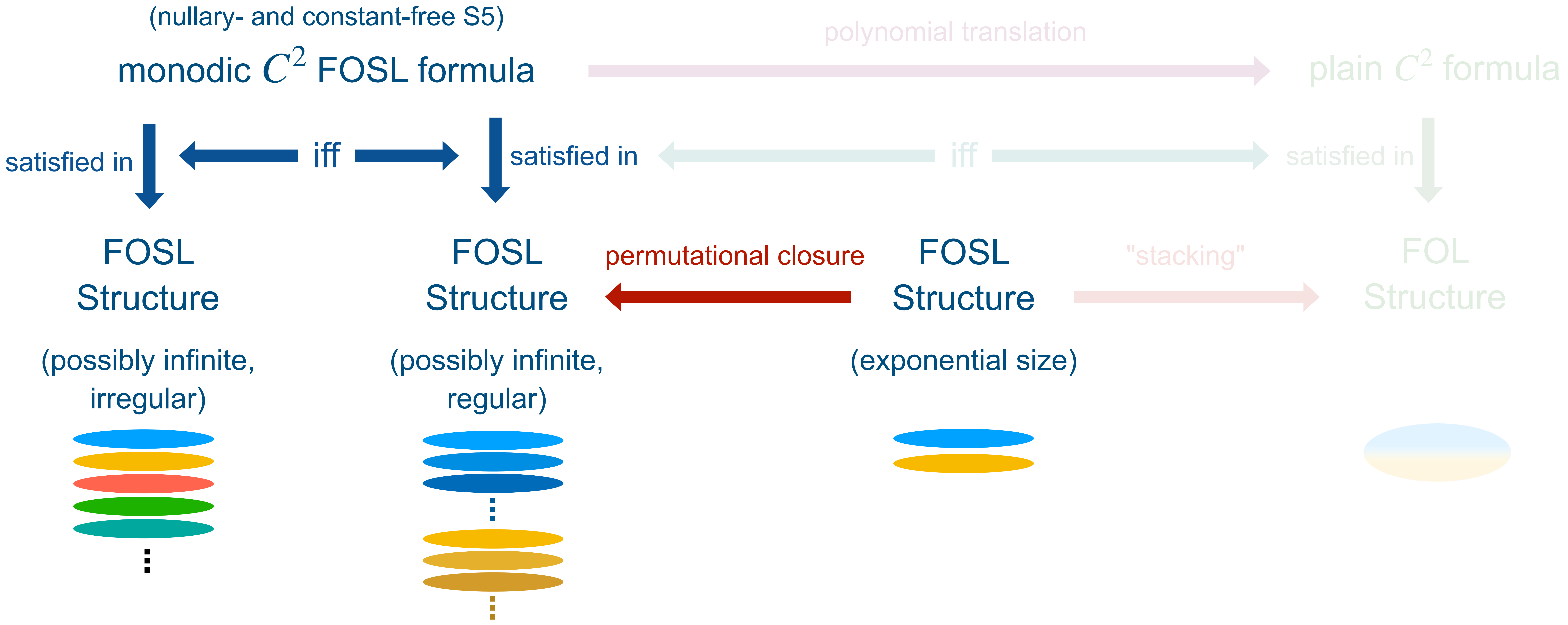


Permutational Closure

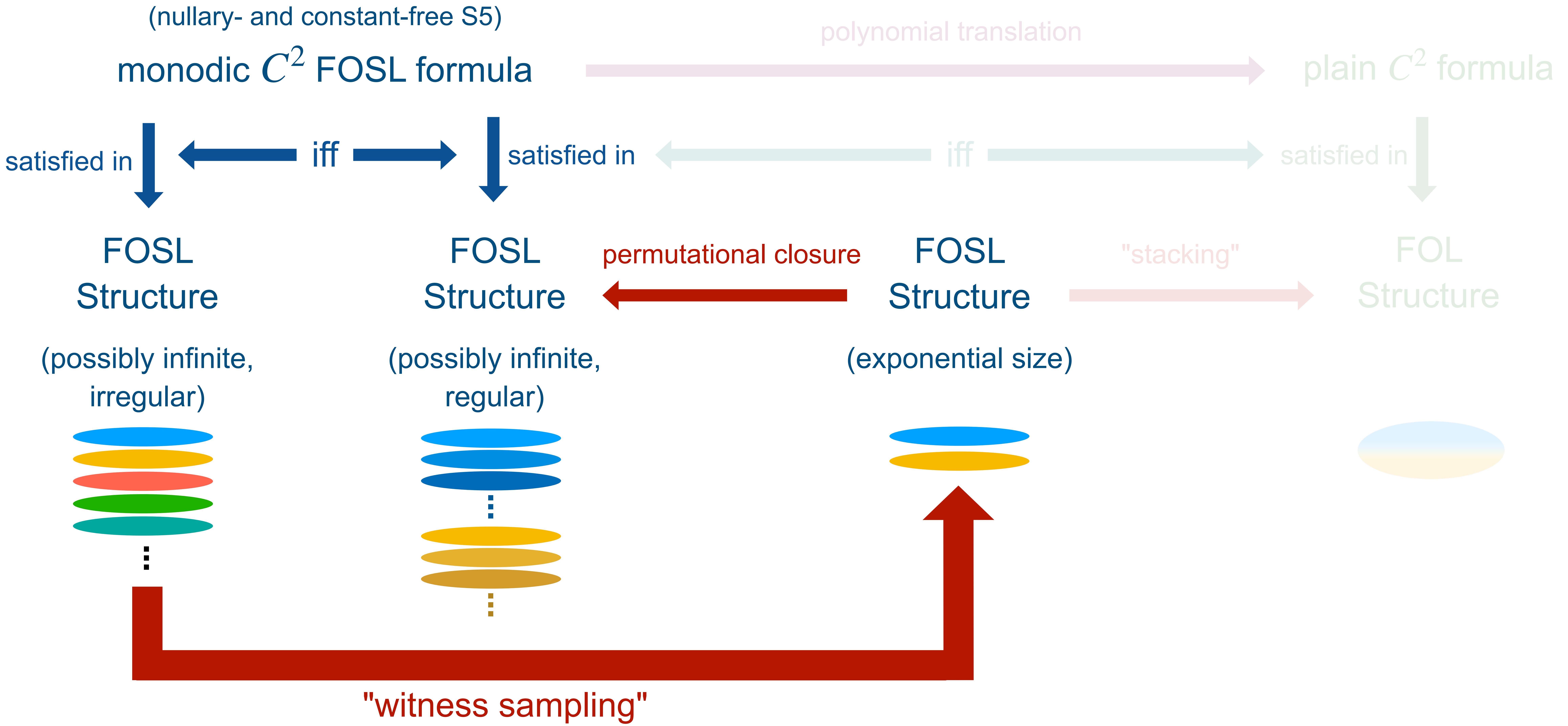




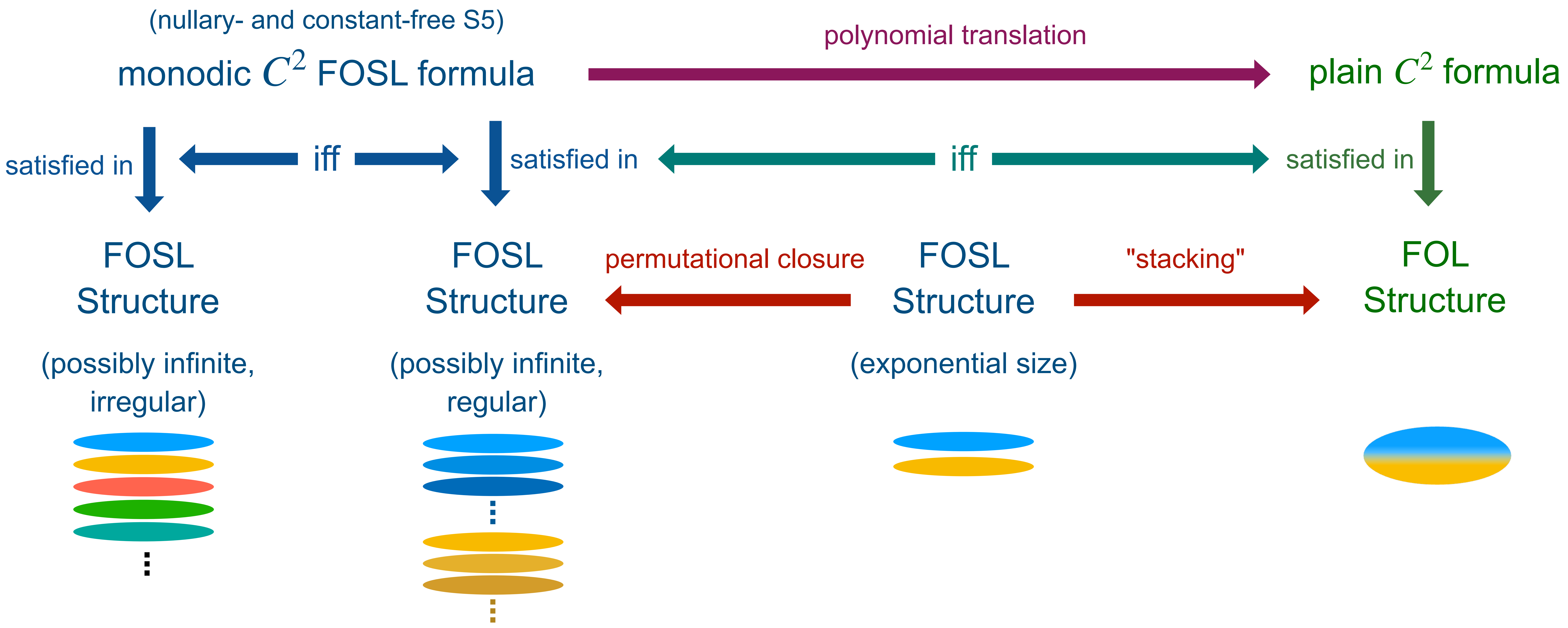
Argument Overview



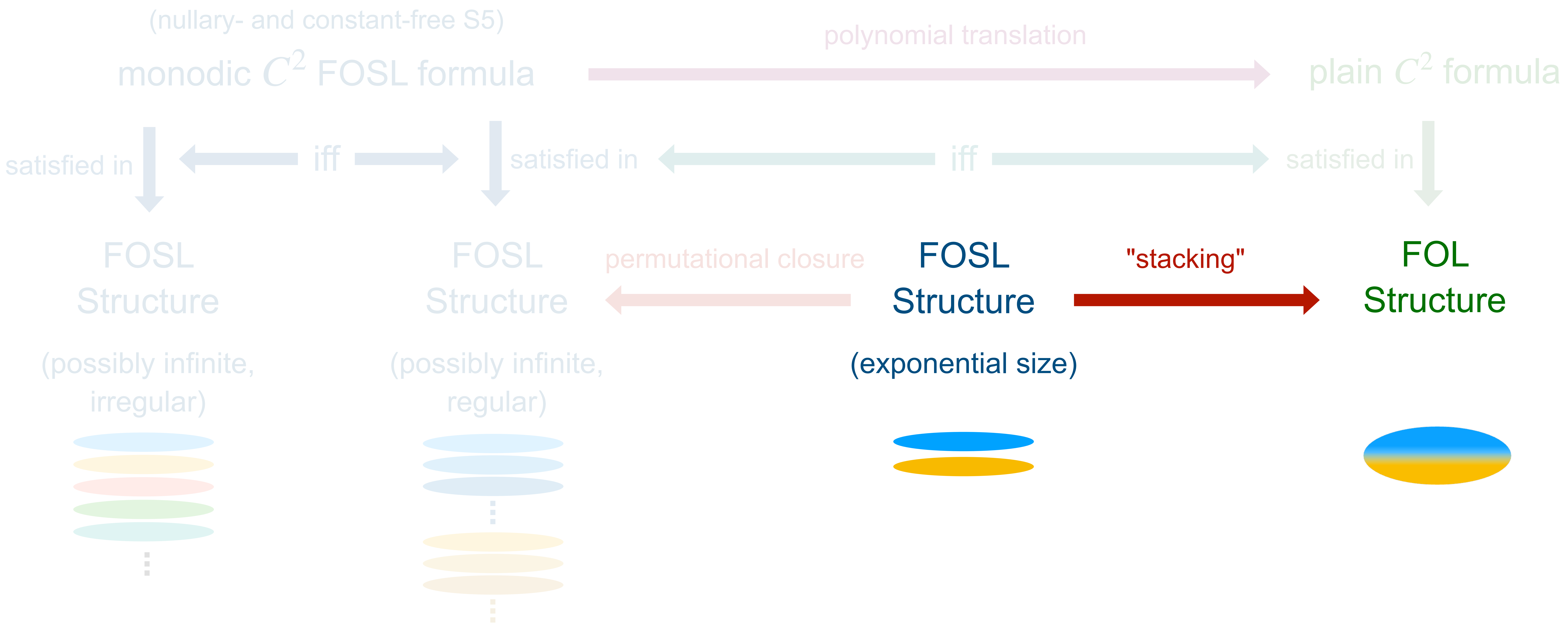
Argument Overview



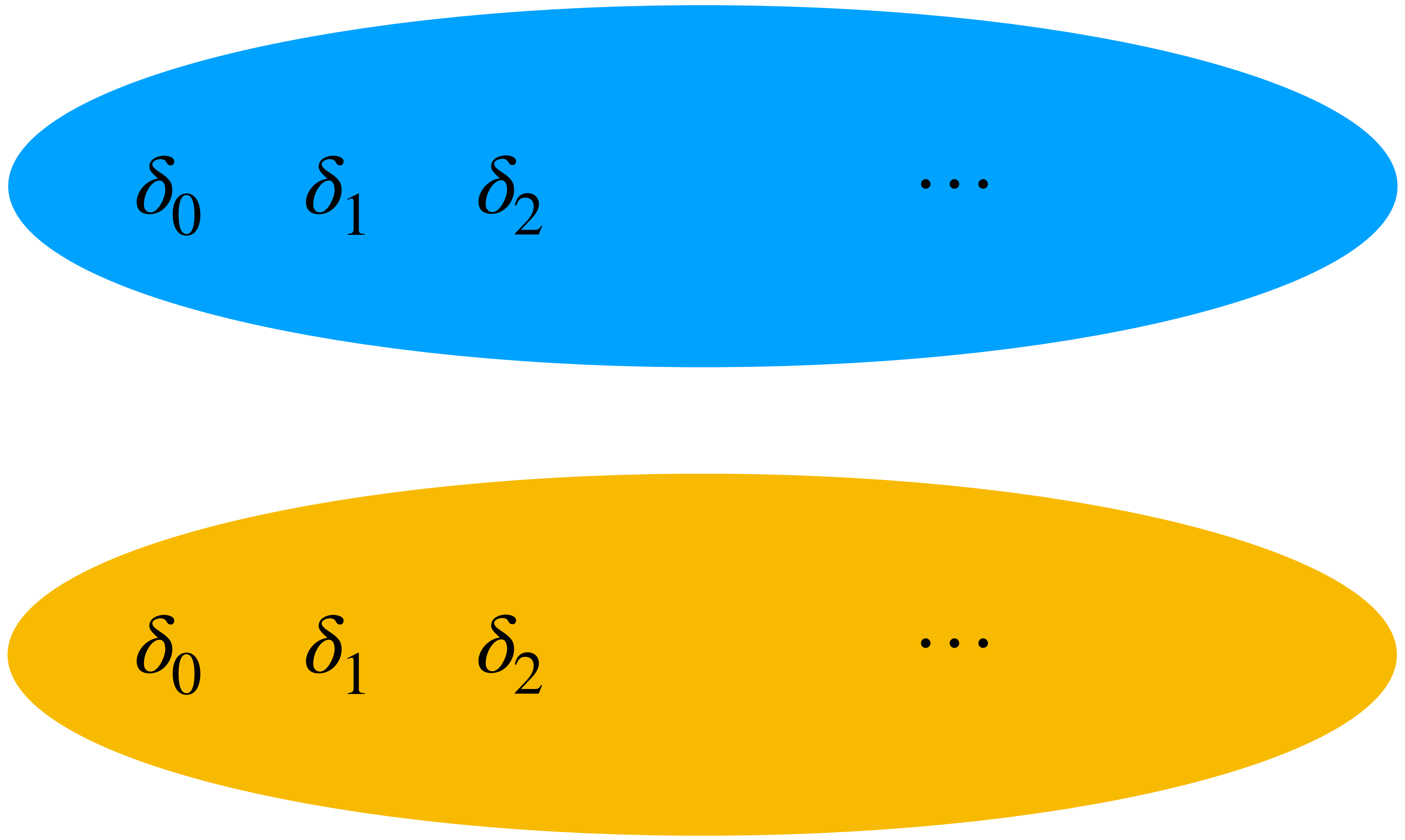
Argument Overview



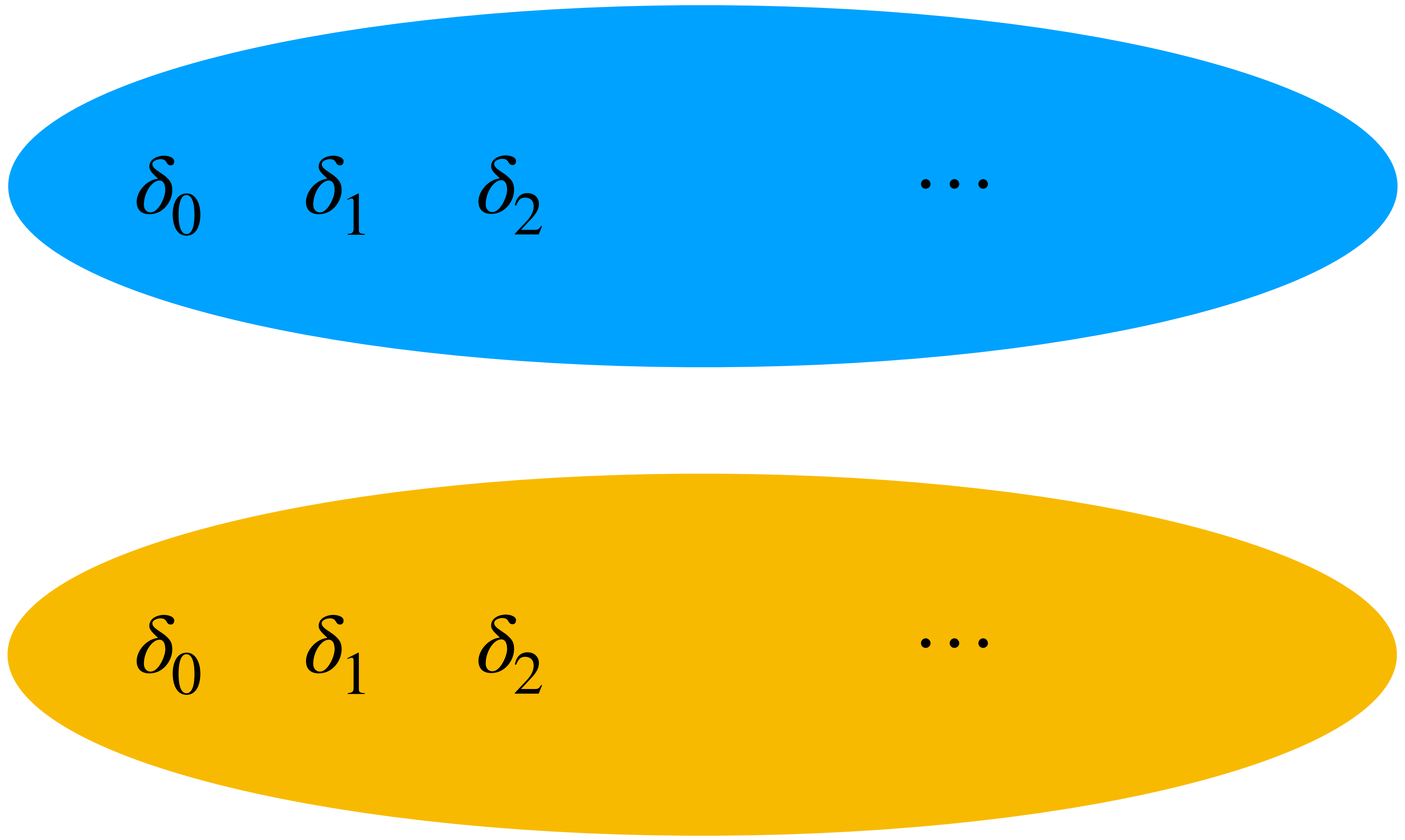
Argument Overview

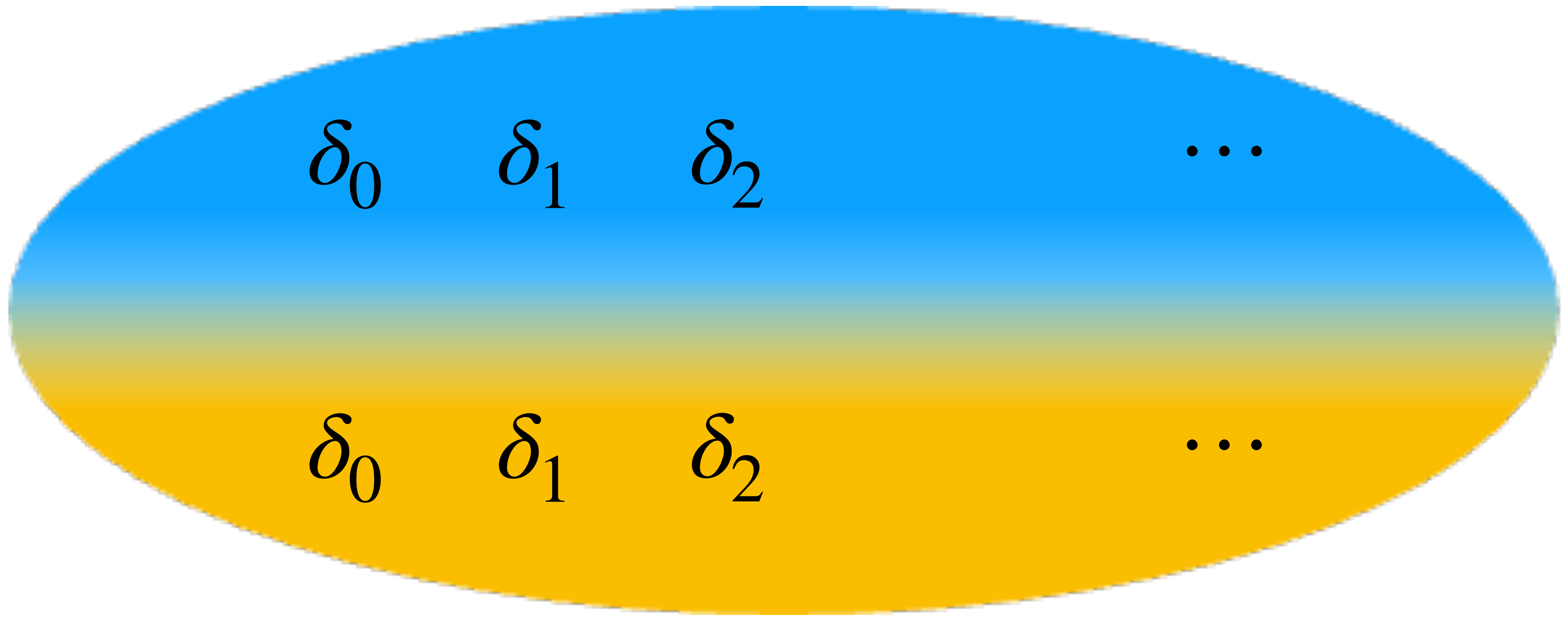


Stacking Worlds



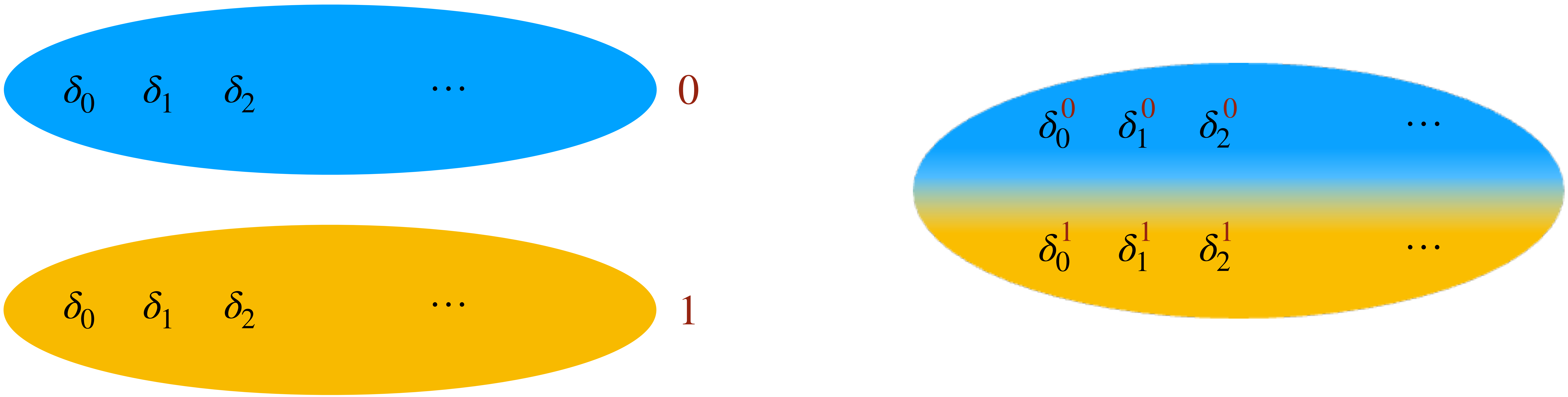
Stacking Worlds





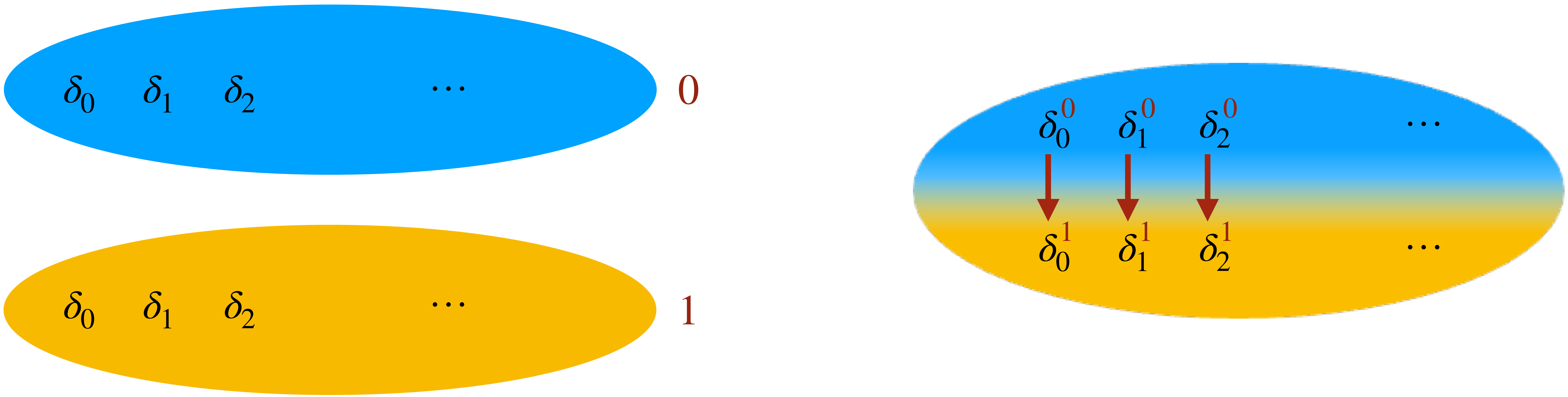
- Disjoint union

Stacking Worlds



- Disjoint union
- Bookkeeping 1: indicate originating worlds by assigning numbers (bit-encoding by unary predicates L_0,L_1,...)

Stacking Worlds



- Disjoint union
- Bookkeeping 1: indicate originating worlds by assigning numbers (bit-encoding by unary predicates L_0,L_1,...)
- Bookkeeping 2: connect corresponding elements with consecutive numbers by binary predicate F
- N.B.: Being "well-stacked" interpretation (i.e. the result of such a stacking) can be characterized in C^2

Translating to plain C^2

A polytime translation maps a (pretransformed) monodic C^2 FOSL formula to a plain C^2 formula.

Translating to plain C^2

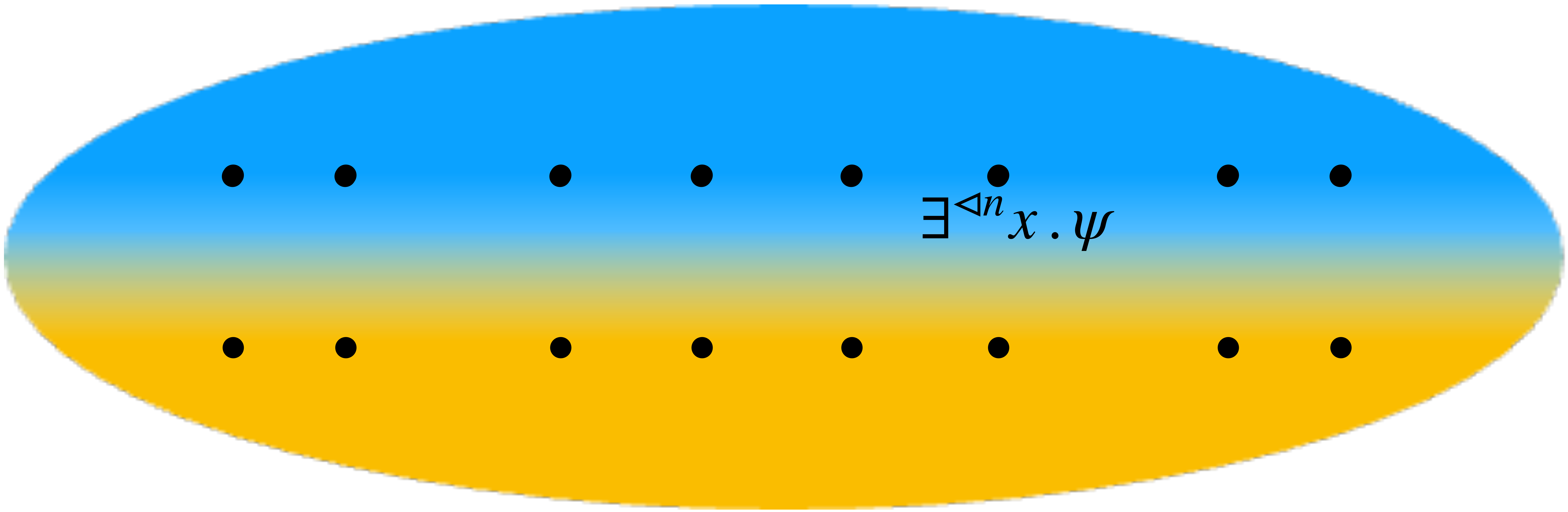
A polytime translation maps a (pretransformed) monodic C^2 FOSL formula to a plain C^2 formula.

Basic idea for the interesting cases: consider formulae with free variable y

Translating to plain C^2

A polytime translation maps a (pretransformed) monodic C^2 FOSL formula to a plain C^2 formula.

Basic idea for the interesting cases: consider formulae with free variable y

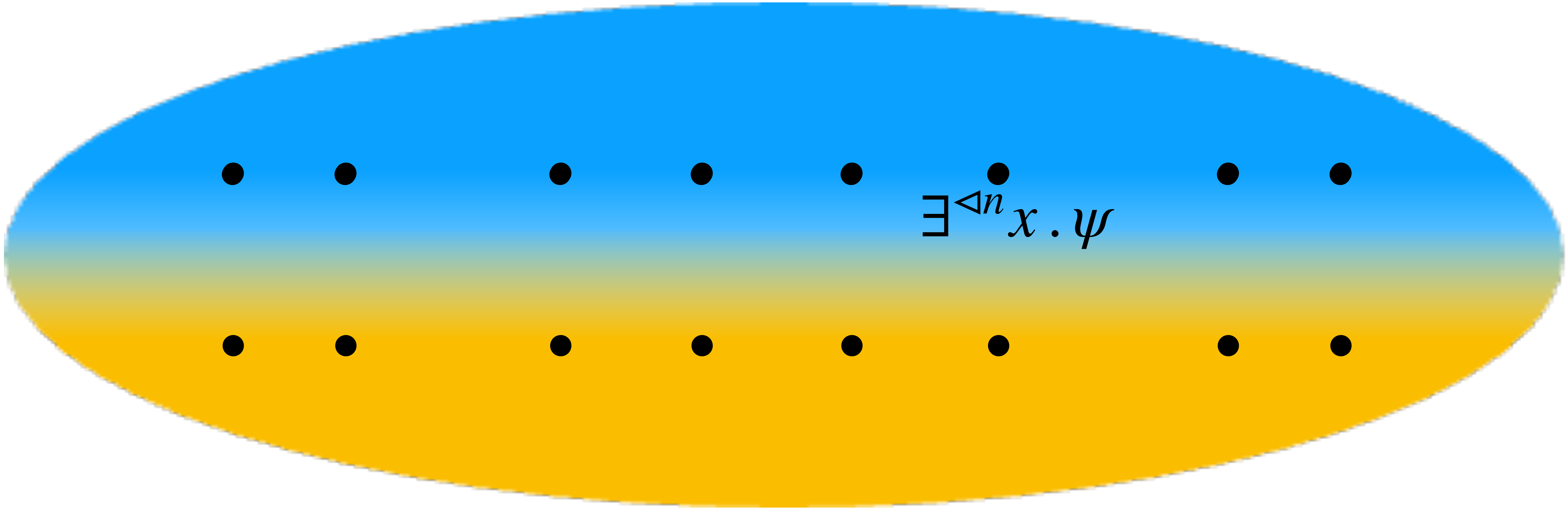


Translating to plain C^2

A polytime translation maps a (pretransformed) monodic C^2 FOSL formula to a plain C^2 formula.

Basic idea for the interesting cases: consider formulae with free variable y

- $\exists^{\lhd n} x \,.\, \psi(x, y)$ implemented as "there are $\lhd n$ elements x *in the same* L-*layer as* y satisfying ψ"

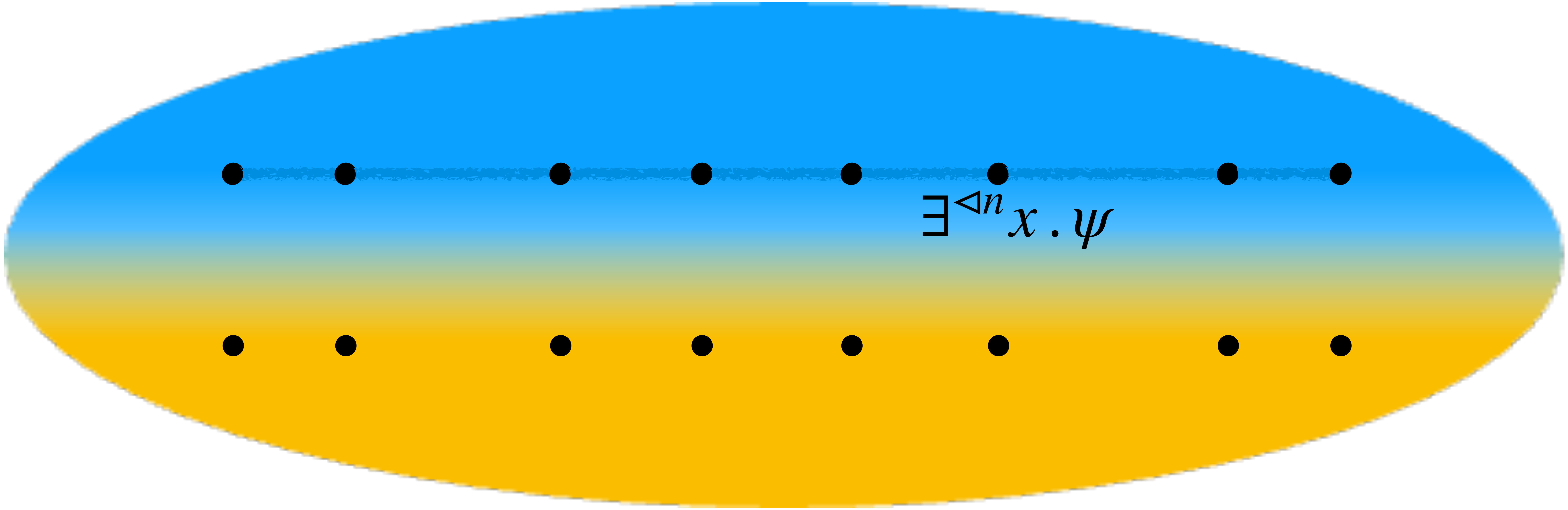


Translating to plain C^2

A polytime translation maps a (pretransformed) monodic C^2 FOSL formula to a plain C^2 formula.

Basic idea for the interesting cases: consider formulae with free variable y

- $\exists^{\lhd n} x \,.\, \psi(x, y)$ implemented as "there are $\lhd n$ elements x *in the same* L*-layer as* y satisfying ψ "

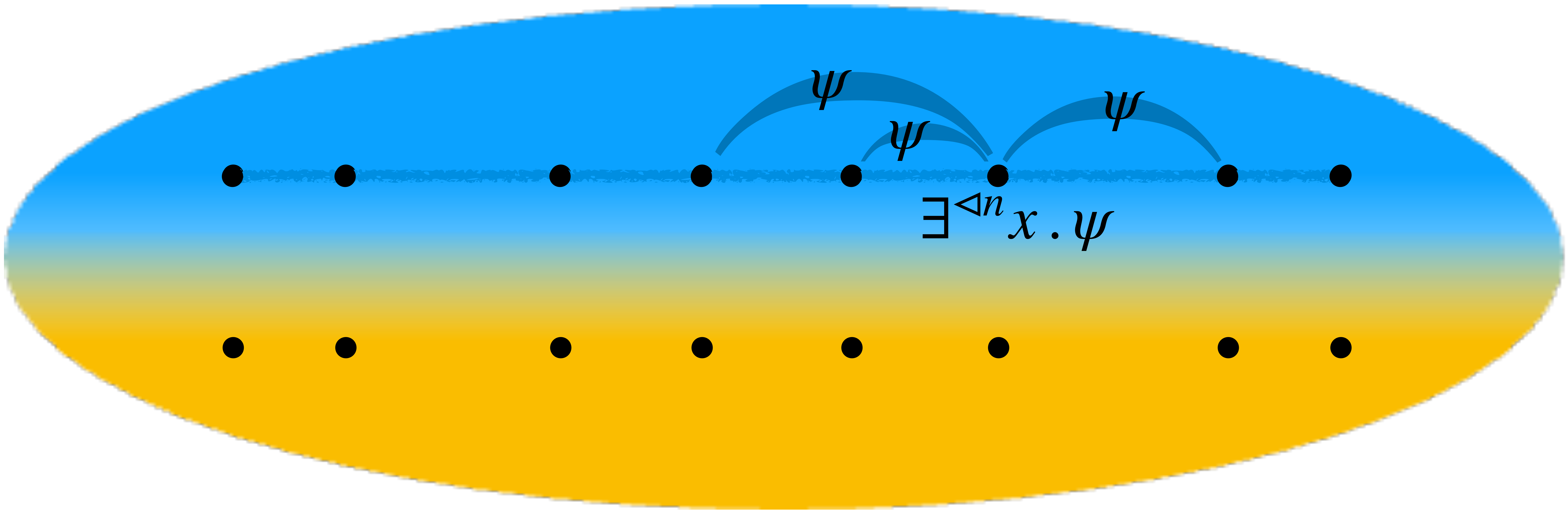


Translating to plain C^2

A polytime translation maps a (pretransformed) monodic C^2 FOSL formula to a plain C^2 formula.

Basic idea for the interesting cases: consider formulae with free variable y

- $\exists^{\lhd n} x \,.\, \psi(x, y)$ implemented as "there are $\lhd n$ elements x *in the same* L*-layer as* y satisfying ψ"

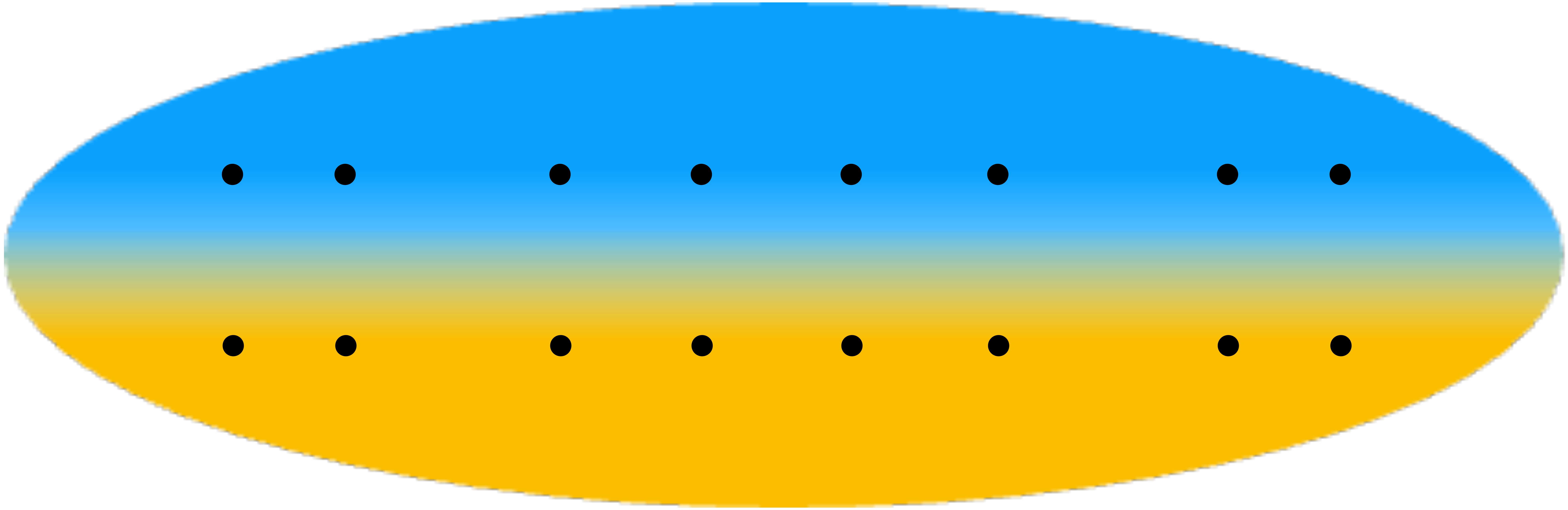


Translating to plain C^2

A polytime translation maps a (pretransformed) monodic C^2 FOSL formula to a plain C^2 formula.

Basic idea for the interesting cases: consider formulae with free variable y

- $\exists^{\lhd n} x \,.\, \psi(x, y)$ implemented as "there are $\lhd n$ elements x *in the same* L*-layer as* y satisfying ψ "
- $\Diamond_* \phi(y)$ implemented as "there is an element *from the same* E*-class as* y satisfying ϕ "

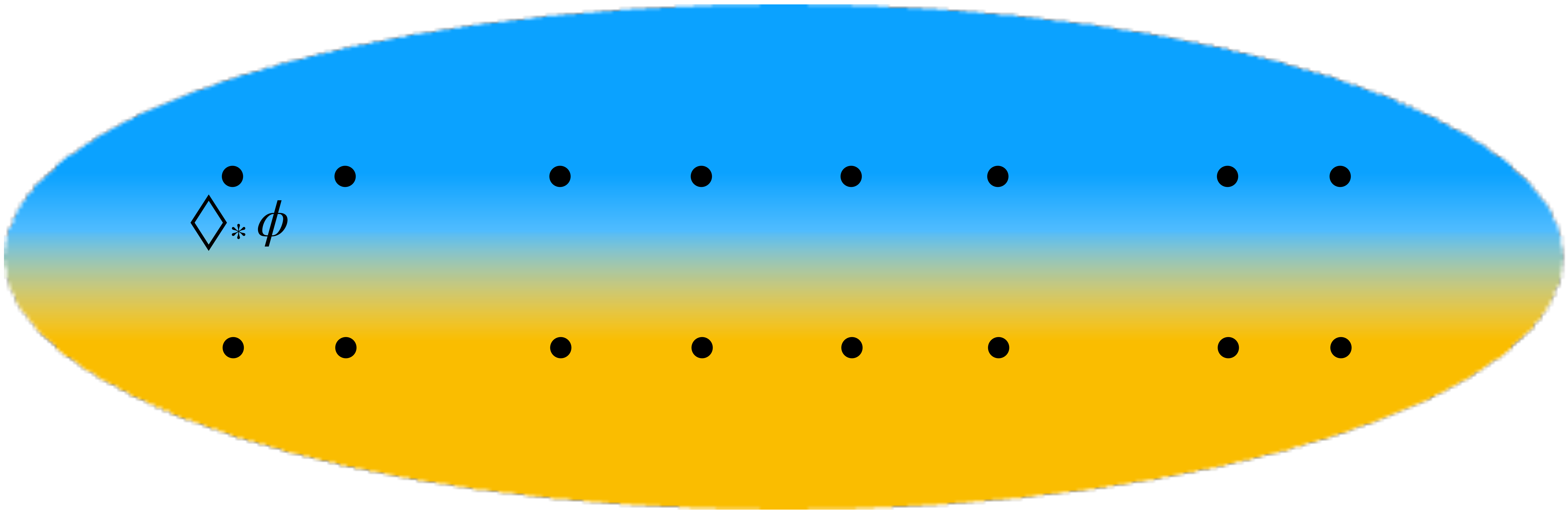


Translating to plain C^2

A polytime translation maps a (pretransformed) monodic C^2 FOSL formula to a plain C^2 formula.

Basic idea for the interesting cases: consider formulae with free variable y

- $\exists^{\lhd n} x \,.\, \psi(x, y)$ implemented as "there are $\lhd n$ elements x *in the same* L*-layer as* y satisfying ψ "
- $\Diamond_* \phi(y)$ implemented as "there is an element *from the same* E*-class as* y satisfying ϕ "

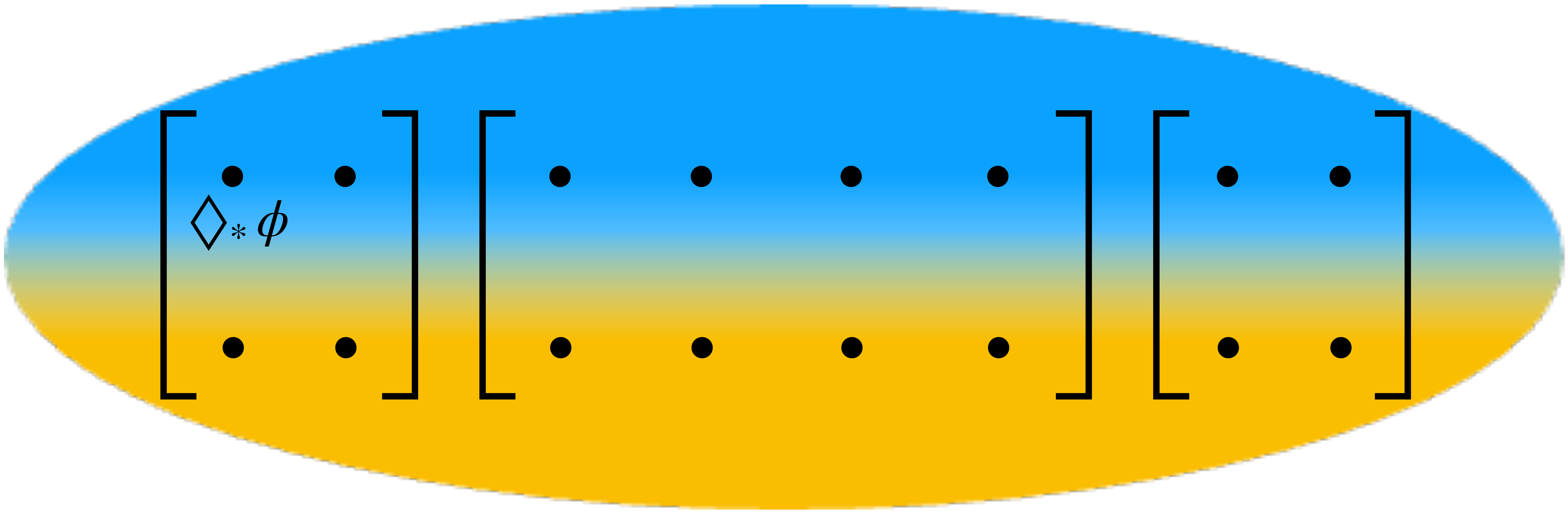


Translating to plain C^2

A polytime translation maps a (pretransformed) monodic C^2 FOSL formula to a plain C^2 formula.

Basic idea for the interesting cases: consider formulae with free variable y

- $\exists^{\triangleleft n} x \,.\, \psi(x, y)$ implemented as "there are $\triangleleft n$ elements x *in the same* L-*layer as* y satisfying ψ "
- $\Diamond_* \phi(y)$ implemented as "there is an element *from the same* E-*class as* y satisfying ϕ "

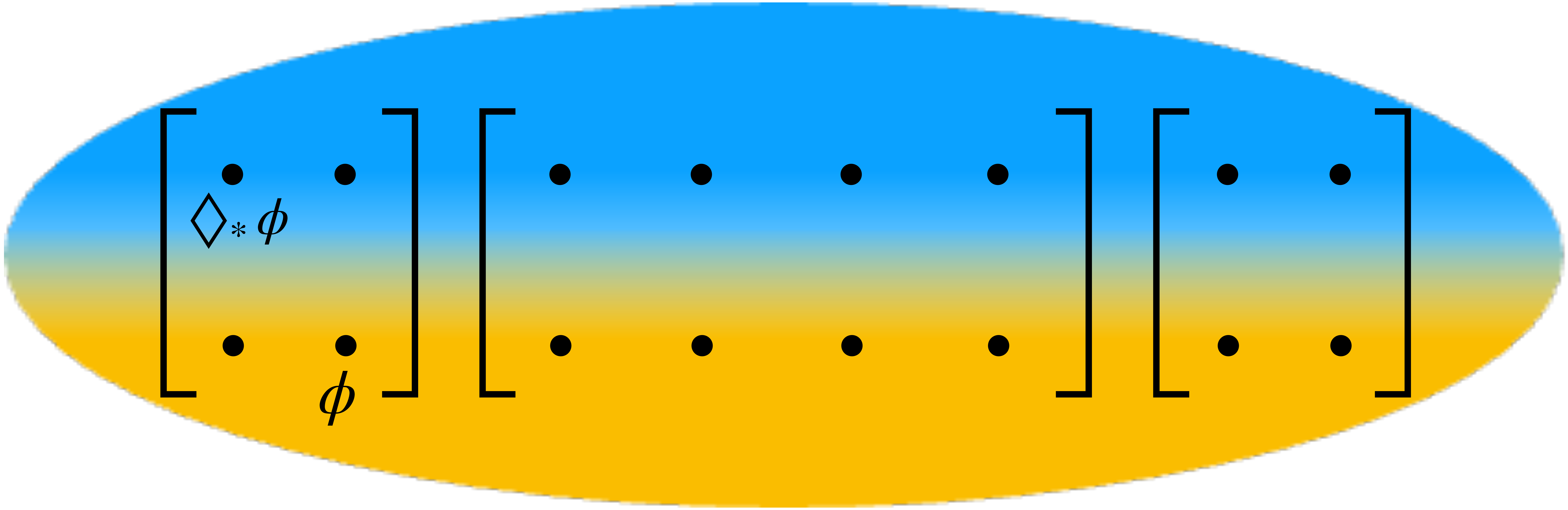


Translating to plain C^2

A polytime translation maps a (pretransformed) monodic C^2 FOSL formula to a plain C^2 formula.

Basic idea for the interesting cases: consider formulae with free variable y

- $\exists^{\lhd n} x \,.\, \psi(x,y)$ implemented as "there are $\lhd n$ elements x *in the same* L-*layer as* y satisfying ψ"
- $\Diamond_* \phi(y)$ implemented as "there is an element *from the same* E-*class as* y satisfying ϕ"



Translating to plain C^2

A polytime translation maps a monodic C^2 FOSL formula to a plain C^2 formula.

Basic idea for the interesting cases: consider formulae with free variable y

- $\exists^{\lhd n} x \,.\, \psi(x, y)$ implemented as "there are $\lhd n$ elements x *in the same* L*-layer as* y satisfying ψ "
- $\Diamond_* \phi(y)$ implemented as "there is an element *from the same* E*-class as* y satisfying ϕ "

The translation preserves satisfiabiliy.

Translating to plain C^2

A polytime translation maps a monodic C^2 FOSL formula to a plain C^2 formula.

Basic idea for the interesting cases: consider formulae with free variable y

- $\exists^{\lhd n} x \,.\, \psi(x, y)$ implemented as "there are $\lhd n$ elements x *in the same* L*-layer as* y satisfying ψ "
- $\Diamond_{*}\, \phi(y)$ implemented as "there is an element *from the same* E*-class as* y satisfying ϕ "

The translation preserves satisfiabiliy.

Corollary: *Satisfiability in monodic standpoint* C^2 *is* $\textsc{NExpTime}$*-complete*

Application to Ontology Languages

Application to Description Logics and Ontology Languages

Application to Description Logics and Ontology Languages

Adding monodic standpoints to popular expressive DLs does not increase their computational complexity:

Application to Description Logics and Ontology Languages

Adding monodic standpoints to popular expressive DLs does not increase their computational complexity:

➡ We can add monodic standpoints to the description logic $\mathcal{ALCOIQB}^{\mathsf{Self}}$

(Easy: Essentially just C^2 in diguise)

Application to Description Logics and Ontology Languages

Adding monodic standpoints to popular expressive DLs does not increase their computational complexity:

➡ We can add monodic standpoints to the description logic $\mathcal{ALCOIQB}^{\text{Self}}$

(Easy: Essentially just C^2 in diguise)

➡ We show how to accommodate (possibly standpoint-annotated) role chain axioms.

We obtain monodic standpoint $\mathcal{SHOIQB}_\mathcal{S}$ and $\mathcal{SROIQB}_\mathcal{S}$

subsuming **OWL 1** and **OWL 2**

Application to Description Logics and Ontology Languages

Adding monodic standpoints to popular expressive DLs does not increase their computational complexity:

➡ We can add monodic standpoints to the description logic $\mathcal{ALCOIQB}^{\mathsf{Self}}$

(Easy: Essentially just C^2 in diguise)

➡ We show how to accommodate (possibly standpoint-annotated) role chain axioms.

We obtain monodic standpoint $\mathcal{SHOIQB}_\mathcal{S}$ and $\mathcal{SROIQB}_\mathcal{S}$

subsuming **OWL 1** and **OWL 2**

Theorem: *Checking satisfiability of monodic standpoint $\mathcal{SHOIQB}_\mathcal{S}$ sentences is $\textsc{NExpTime}$-complete.*

Checking satisfiability of monodic standpoint $\mathcal{SHOIQB}_\mathcal{S}$ sentences is $\textsc{N2ExpTime}$-complete.

Nominals Cause Trouble

Nominals Cause Trouble

Nominals Cause Trouble

As observed in earlier works, the interplay of nominals and standpoint modalities is quite troublesome for reasoning:

Nominals Cause Trouble

As observed in earlier works, the interplay of nominals and standpoint modalities is quite troublesome for reasoning:

➡ The satisfiability even of monodic standpoint $\mathcal{ALCO}$ is already $\textsc{NExpTime}$-hard (tiling of exponential size grid), as opposed to monodic standpoint $\mathcal{SHIQ}$, which is known to be $\textsc{ExpTime}$-complete.

Nominals Cause Trouble

As observed in earlier works, the interplay of nominals and standpoint modalities is quite troublesome for reasoning:

➡ The satisfiability even of monodic standpoint $\mathcal{ALCO}$ is already $\textsc{NExpTime}$-hard

(tiling of exponential size grid),

as opposed to monodic standpoint $\mathcal{SHIQ}$, which is known to be $\textsc{ExpTime}$-complete.

➡ Lifting monodicity gentliest (1 binary rigid predicate) causes undecidability even for $\mathcal{ALCOIF}$

(tiling of infinite grid),

while monodic standpoint $\mathcal{ALCIF}$ with rigid predicates (and more) is known to be decidable.

Conclusions

Recap:

- ➡ Managing perspectives is interesting in knowledge integration scenarios.
- ➡ Reasoning with monodic Standpoint C^2 has the same complexity as with plain C^2.
- ➡ This warrants complexity-neutral extensions of Ontology Languages by monodic standpoints.
- ➡ The interplay of standpoint modalities and nominals causes trouble.

Conclusions

Recap:

➡ Managing perspectives is interesting in knowledge integration scenarios.

➡ Reasoning with monodic Standpoint C^2 has the same complexity as with plain C^2.

➡ This warrants complexity-neutral extensions of Ontology Languages by monodic standpoints.

➡ The interplay of standpoint modalities and nominals causes trouble.

Future Work:

➡ Implementation of translations and integration with existing reasoners

➡ Lifting the monodicity restriction

➡ Towards conceptual modelling with standpoints for knowledge integration challenges

Bonus Material

Permutational Representatives

Permutational Representatives

For any satisfiable ϕ, we construct an (exp.) structure that yields a model

Permutational Representatives

For any satisfiable ϕ, we construct an (exp.) structure that yields a model

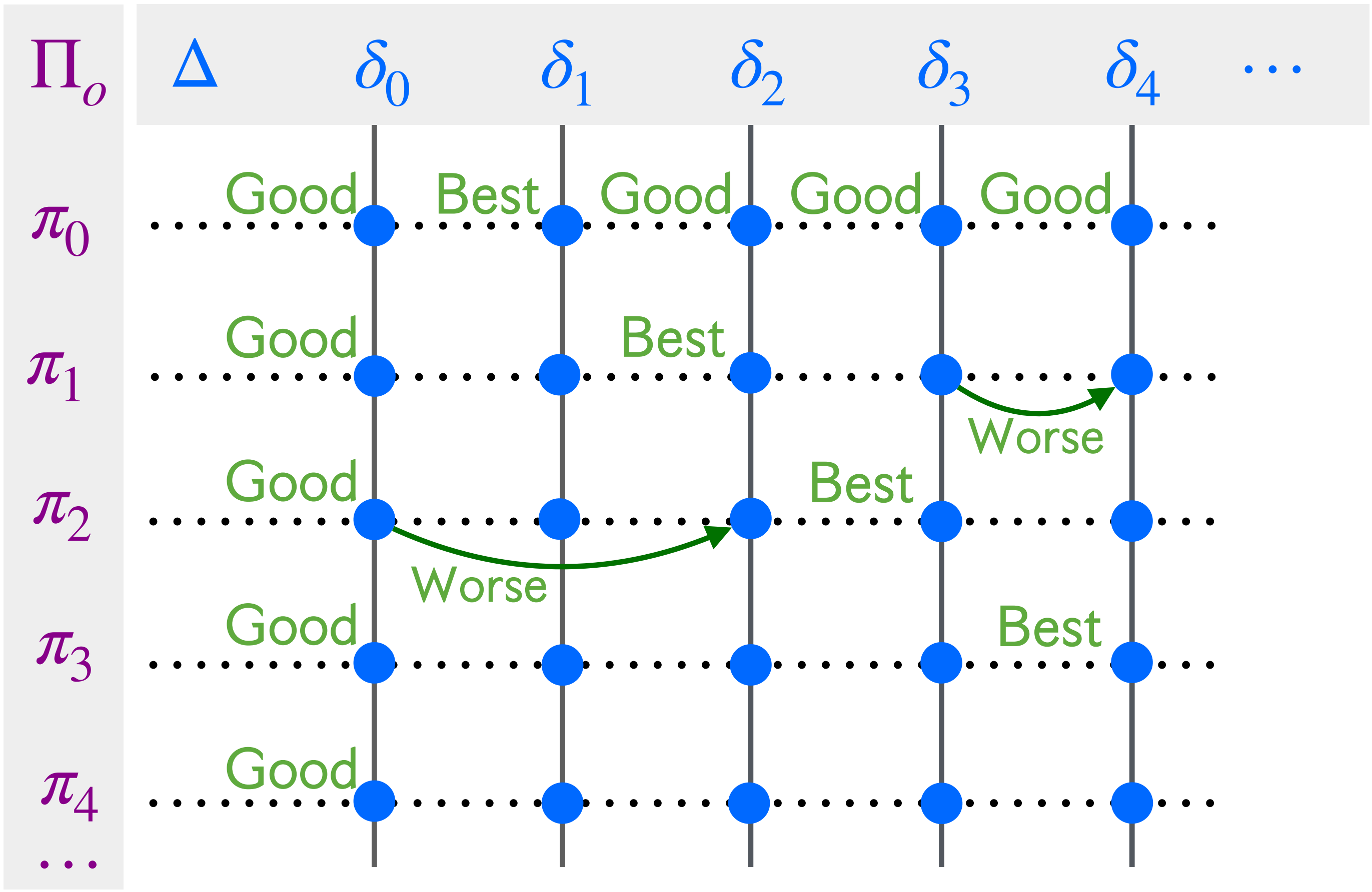
$$\phi = \exists^{=1} x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$

$$E_0 = \Box_*(\texttt{Good}(x))$$

$$E_1 = E_0 \vee \Diamond_*(\texttt{Best}(x) \wedge \forall y.(\texttt{Best}(y) \leftrightarrow x \doteq y))$$

$$E_2 = \Diamond_*(\forall x.\texttt{Good}(x) \vee \texttt{Best}(x))$$

$$\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$$



Permutational Representatives

For any satisfiable ϕ, we construct an (exp.) structure that yields a model

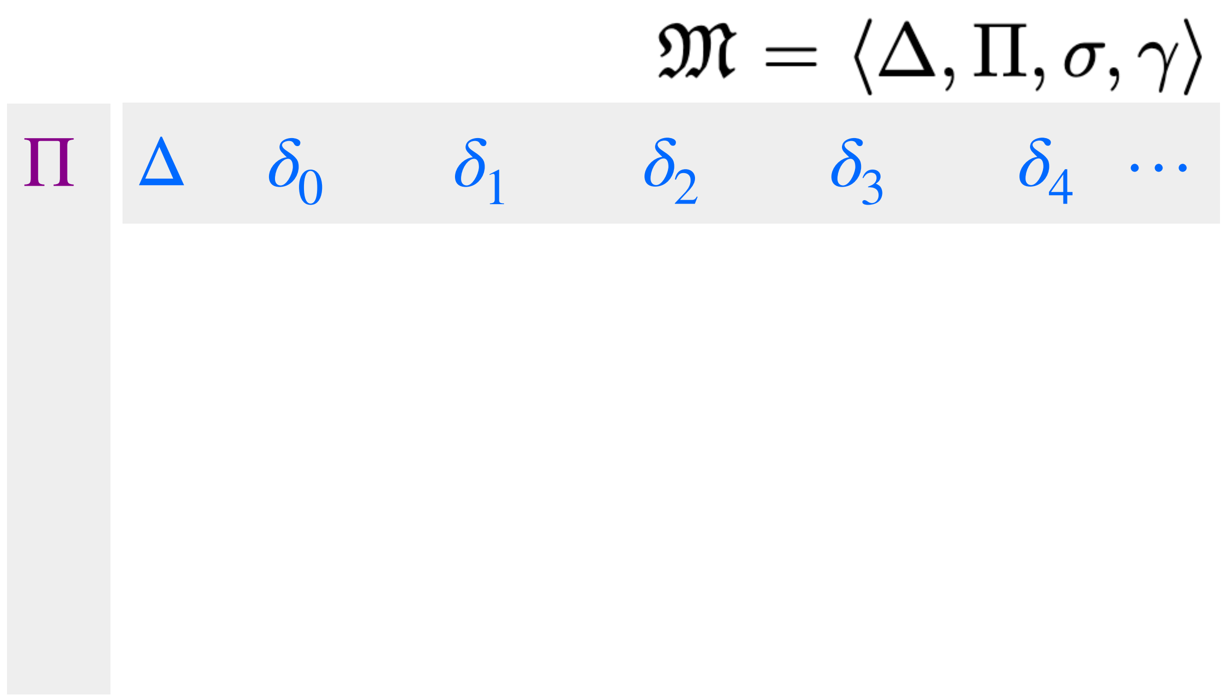
$$\phi = \exists^{=1}x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$

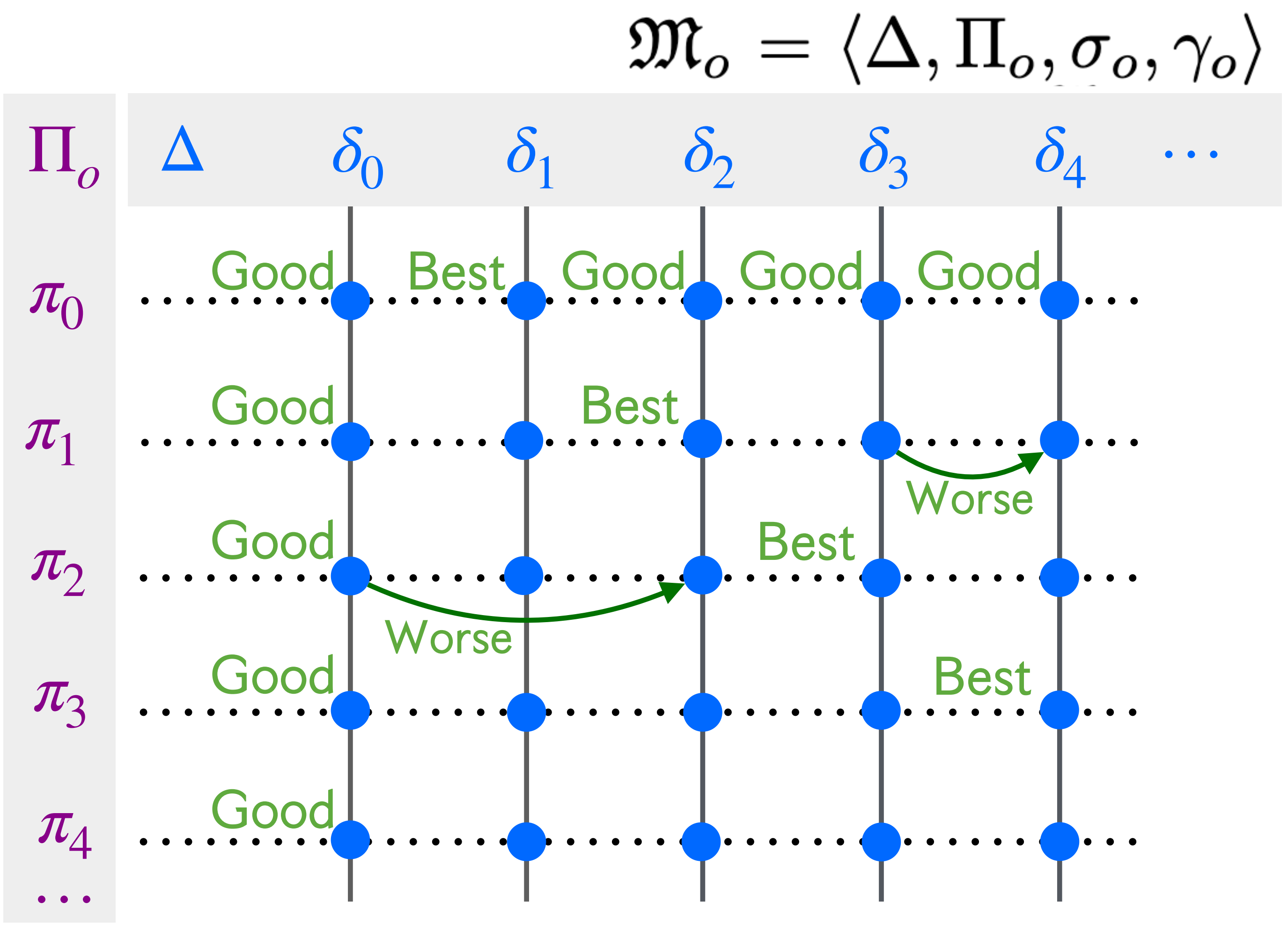
$$E_0 = \Box_*(\mathtt{Good}(x))$$

$$E_1 = E_0 \vee \Diamond_*(\mathtt{Best}(x) \wedge \forall y.(\mathtt{Best}(y) \leftrightarrow x \doteq y))$$

$$E_2 = \Diamond_*(\forall x.\mathtt{Good}(x) \vee \mathtt{Best}(x))$$

Building the exponential structure from a model $\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$:





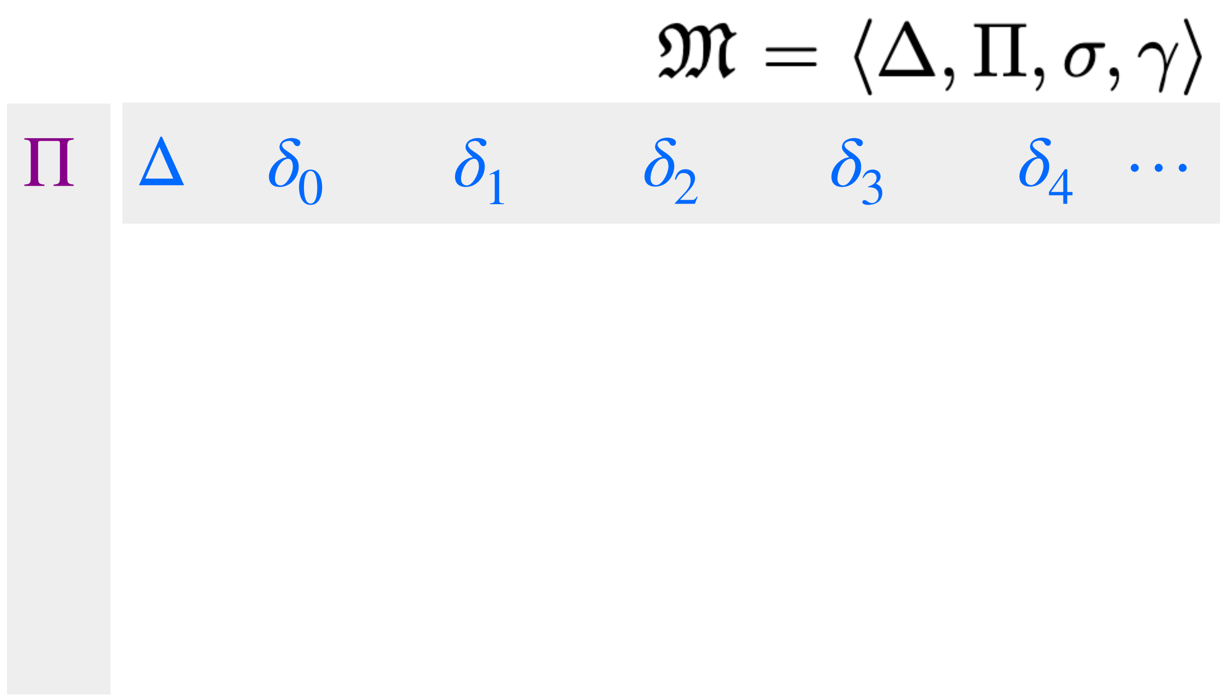
Permutational Representatives

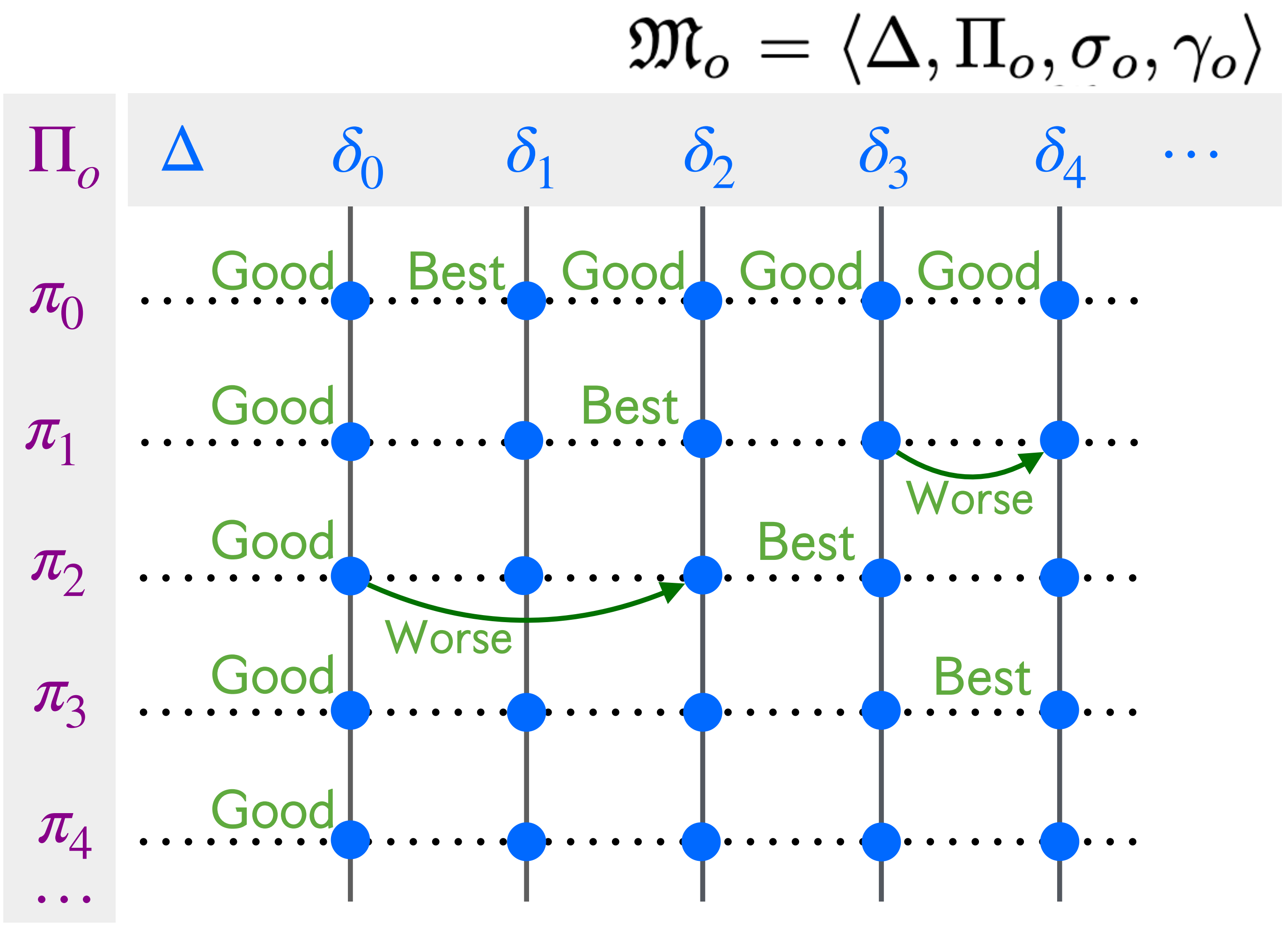
For any satisfiable ϕ, we construct an (exp.) structure that yields a model

$$\phi = \exists^{=1}x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$
$$E_0 = \Box_*(\mathtt{Good}(x))$$
$$E_1 = E_0 \vee \Diamond_*(\mathtt{Best}(x) \wedge \forall y.(\mathtt{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\mathtt{Good}(x) \vee \mathtt{Best}(x))$$

<u>Building the exponential structure from a model $\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$:</u>

- Add one rigid unary predicate E_i per monodic sub-formula $\Diamond_* \psi_i$
  (Set the same extension in structure)





Permutational Representatives

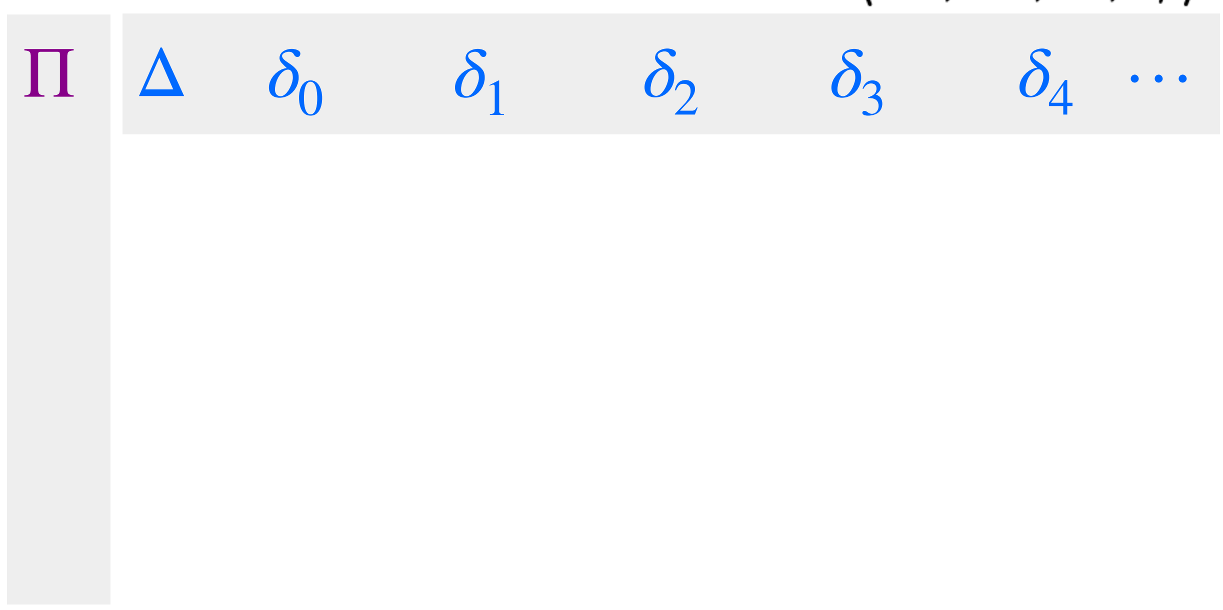
For any satisfiable ϕ, we construct an (exp.) structure that yields a model

$$\phi = \exists^{=1}x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$
$$E_0 = \Box_*(\mathtt{Good}(x))$$
$$E_1 = E_0 \vee \Diamond_*(\mathtt{Best}(x)) \wedge \forall y.(\mathtt{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\mathtt{Good}(x) \vee \mathtt{Best}(x))$$

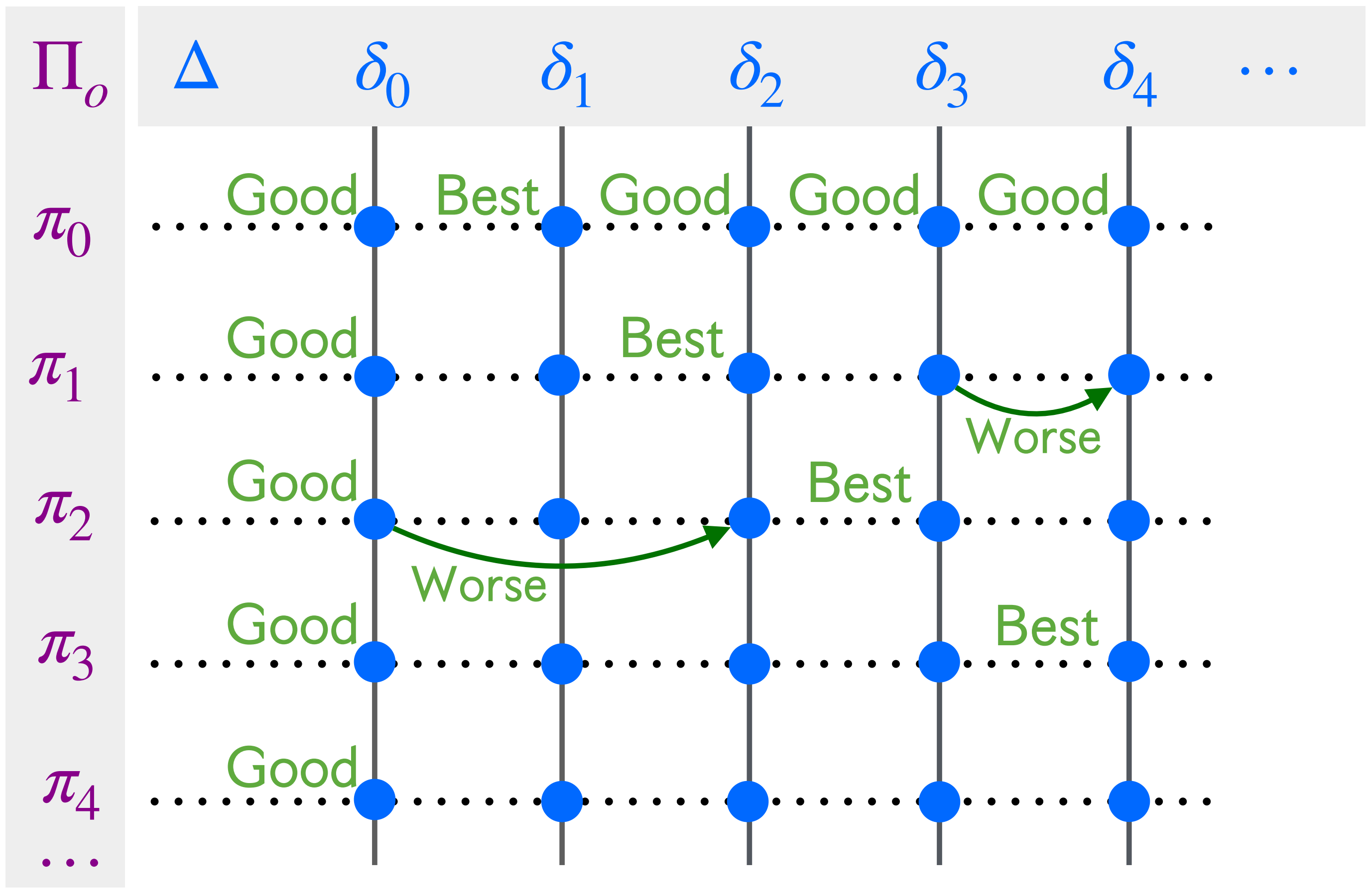
<u>Building the exponential structure from a model $\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$:</u>

- Add one rigid unary predicate E_i per monodic sub-formula $\Diamond_* \psi_i$
  (Set the same extension in structure)

$\mathfrak{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$



$\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$



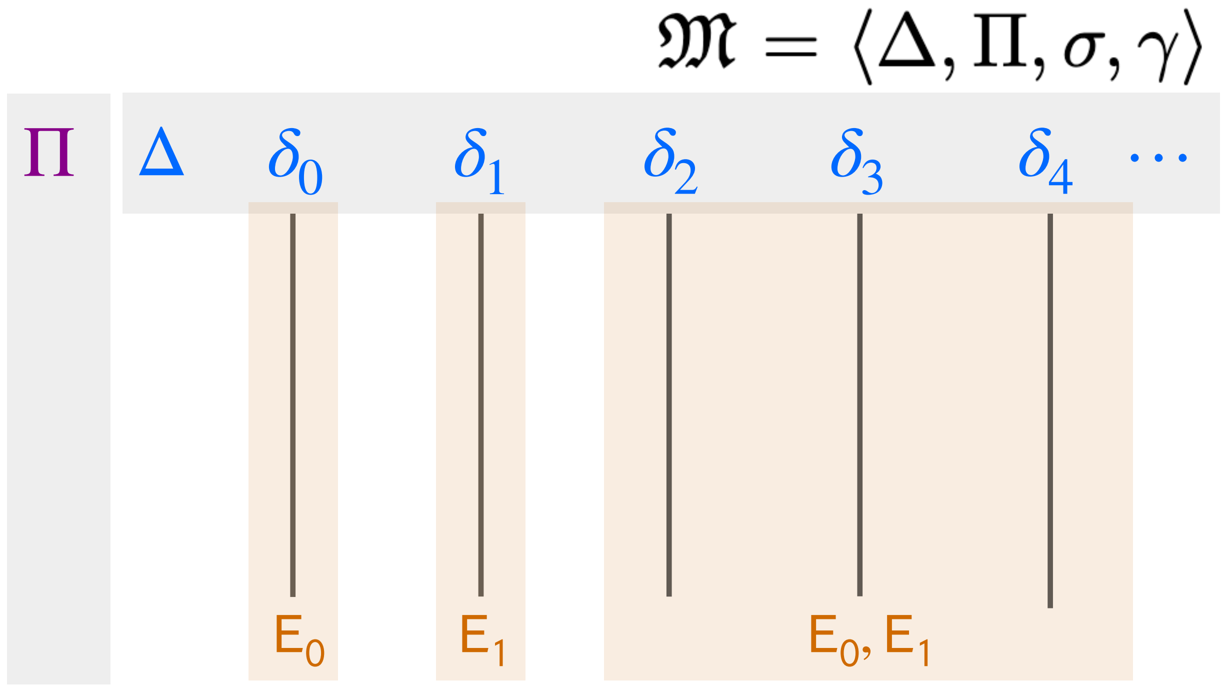
Permutational Representatives

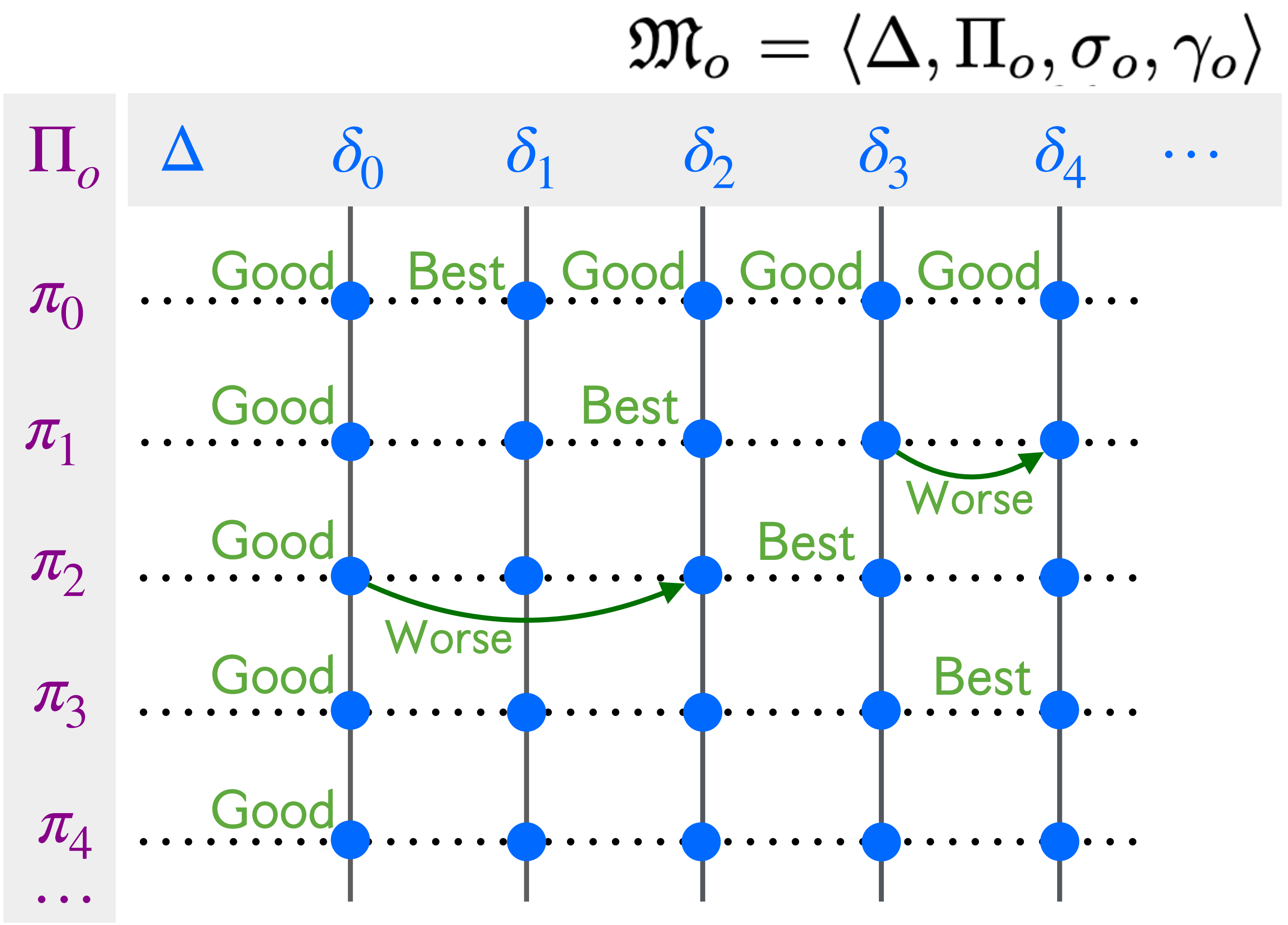
For any satisfiable ϕ, we construct an (exp.) structure that yields a model

$$\phi = \exists^{=1}x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$
$$E_0 = \Box_*(\mathtt{Good}(x))$$
$$E_1 = E_0 \vee \Diamond_*(\mathtt{Best}(x)) \wedge \forall y.(\mathtt{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\mathtt{Good}(x) \vee \mathtt{Best}(x))$$

Building the exponential structure from a model $\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$:

- Add one rigid unary predicate E_i per monodic sub-formula $\Diamond_* \psi_i$
  (Set the same extension in structure)





Permutational Representatives

For any satisfiable ϕ, we construct an (exp.) structure that yields a model

$$\phi = \exists^{=1}x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$

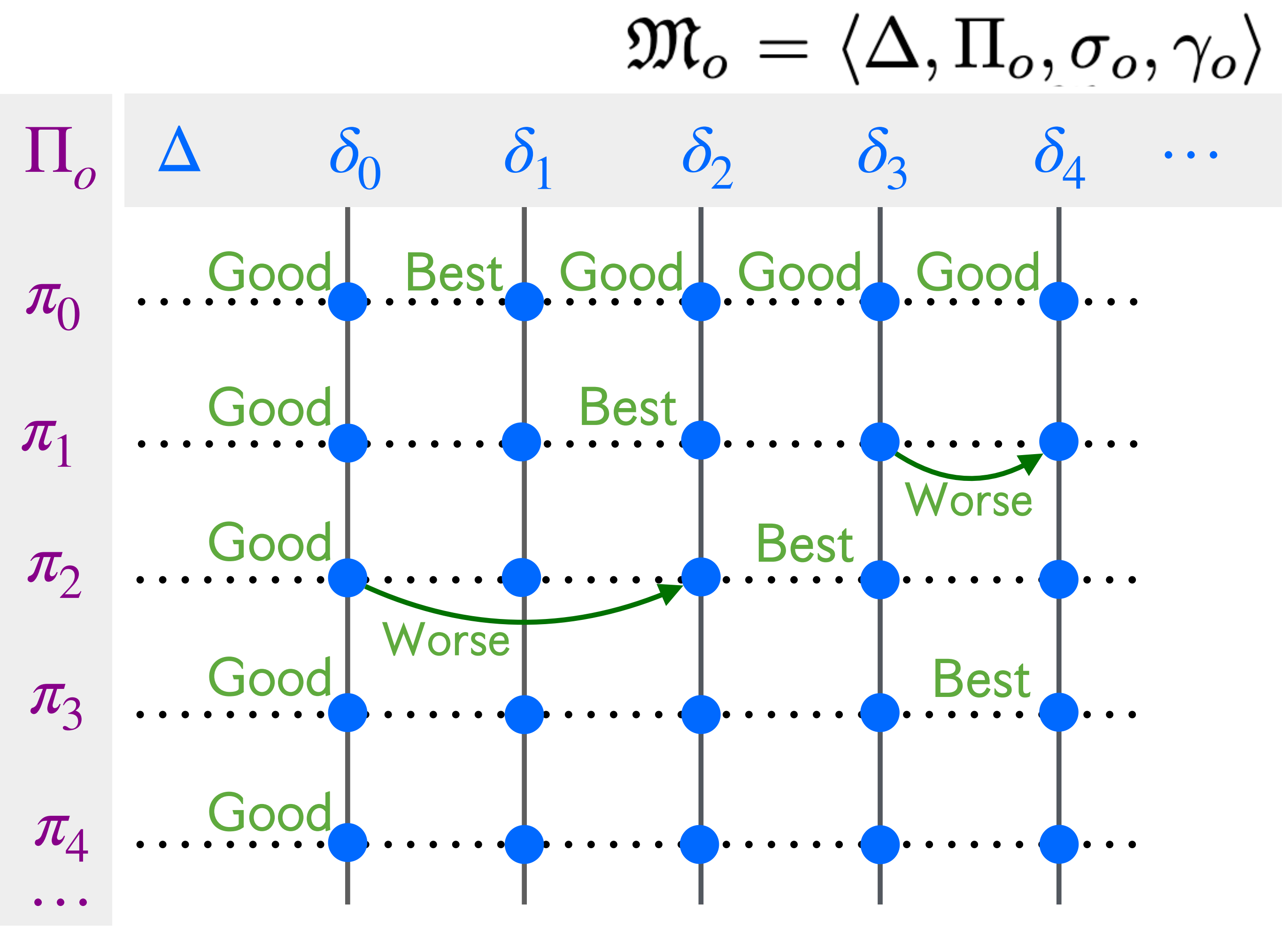
$$E_0 = \Box_*(\texttt{Good}(x))$$

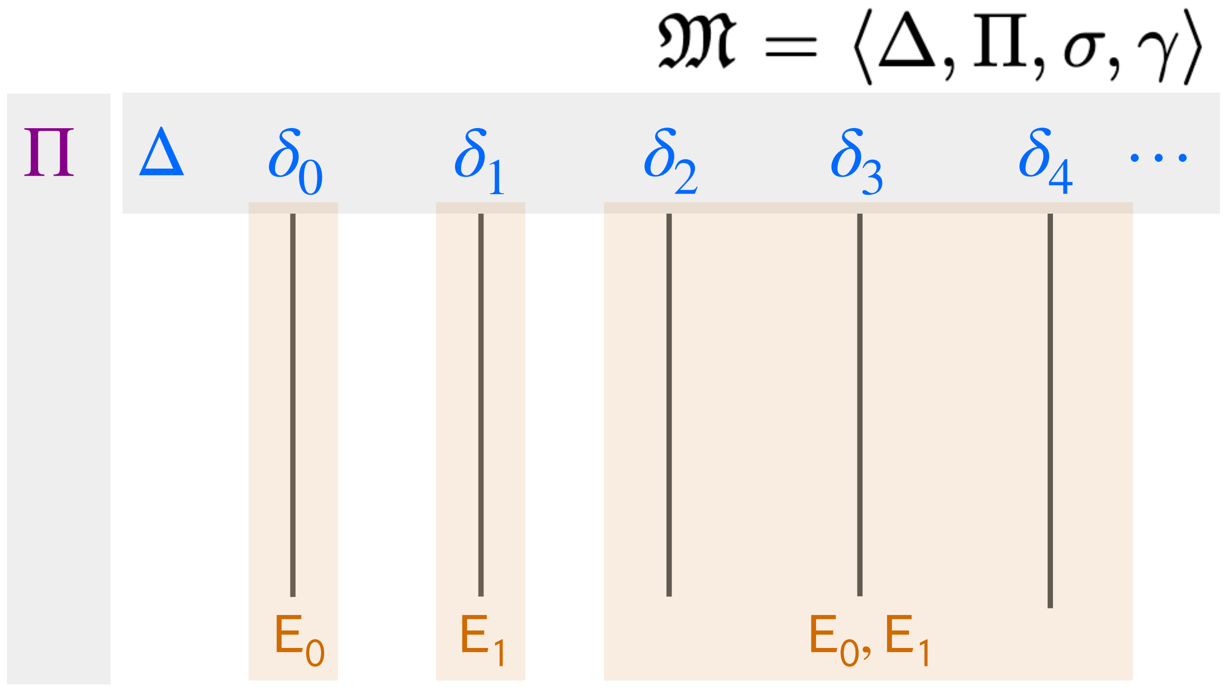
$$E_1 = E_0 \vee \Diamond_*(\texttt{Best}(x) \wedge \forall y.(\texttt{Best}(y) \leftrightarrow x \doteq y))$$

$$E_2 = \Diamond_*(\forall x.\texttt{Good}(x) \vee \texttt{Best}(x))$$

Building the exponential structure from a model $\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$:

- Add one rigid unary predicate E_i per monodic sub-formula $\Diamond_* \psi_i$ (Set the same extension in structure)
- Select from Π_o





Permutational Representatives

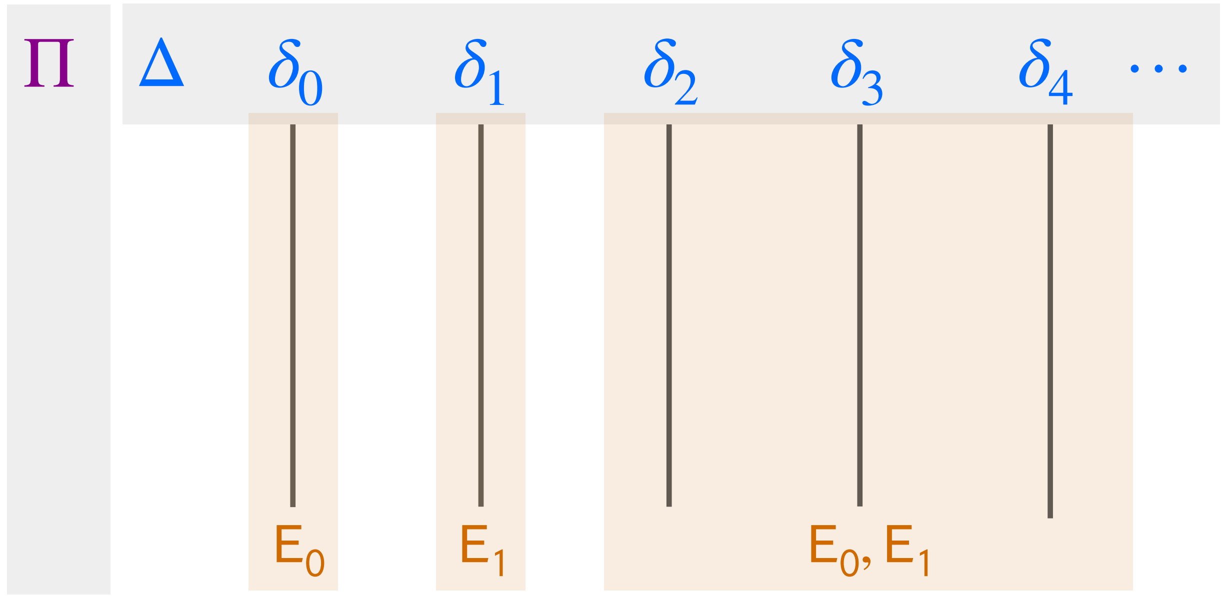
For any satisfiable ϕ, we construct an (exp.) structure that yields a model

$$\phi = \exists^{=1}x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$
$$E_0 = \Box_*(\mathtt{Good}(x))$$
$$E_1 = E_0 \vee \Diamond_*(\mathtt{Best}(x) \wedge \forall y.(\mathtt{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\mathtt{Good}(x) \vee \mathtt{Best}(x))$$

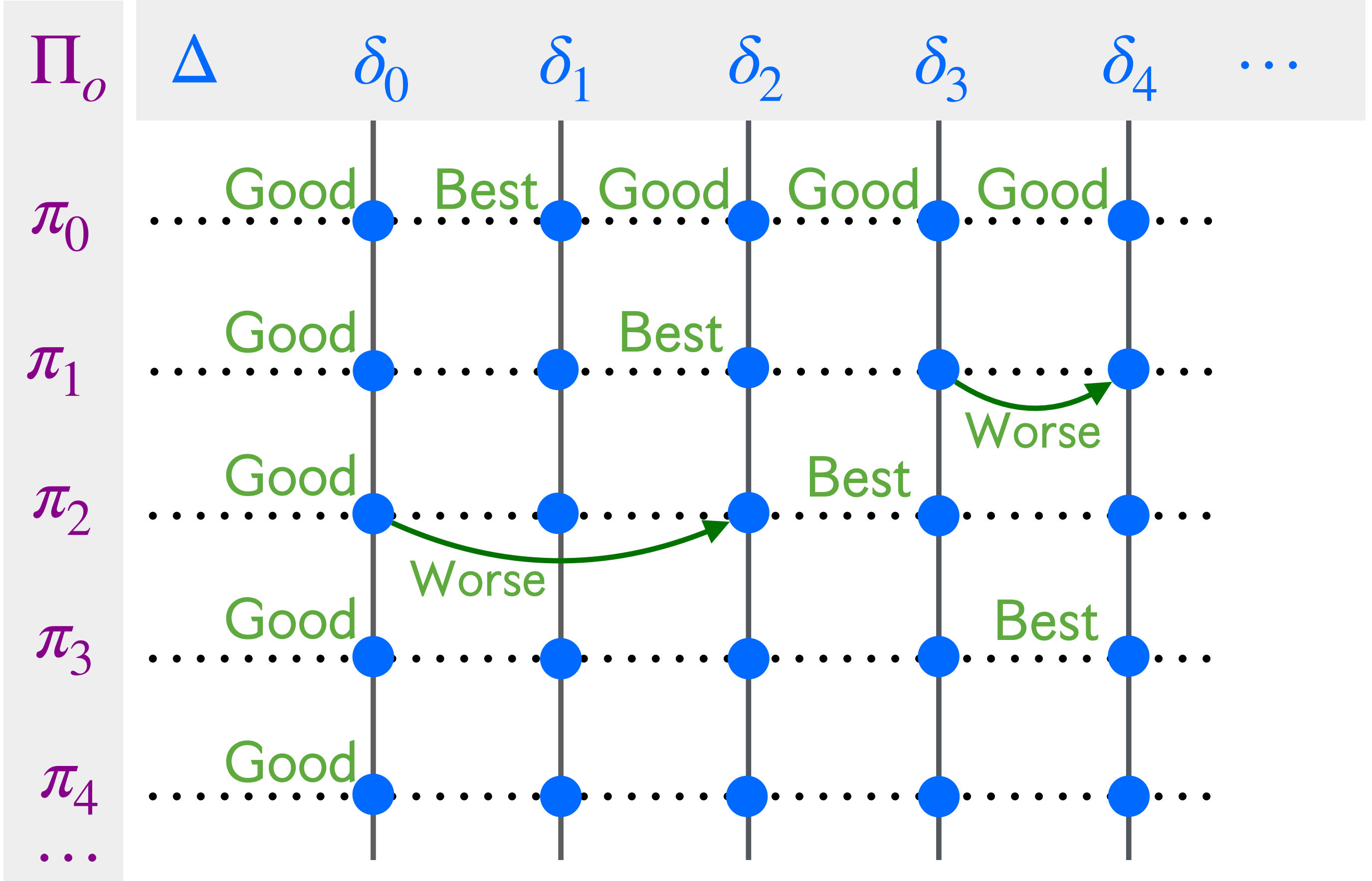
<u>Building the exponential structure from a model $\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$:</u>

- Add one rigid unary predicate E_i per monodic sub-formula $\Diamond_* \psi_i$
  (Set the same extension in structure)
- Select from Π_o
  - a π if no $\Diamond$-subformulas; otherwise:

$\mathfrak{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$



$\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$



Permutational Representatives

For any satisfiable ϕ, we construct an (exp.) structure that yields a model

$$\phi = \exists^{=1}x.(E_0) \land \forall x.(E_1) \land E_2 \text{ with}$$

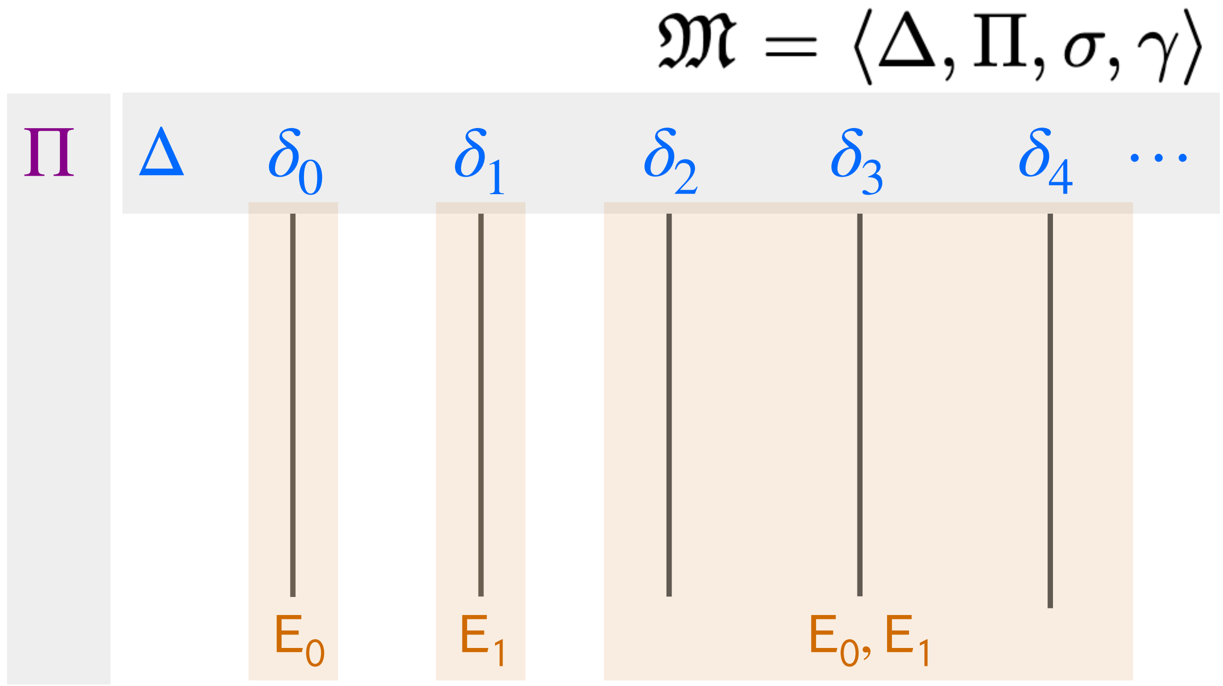
$$E_0 = \Box_*(\texttt{Good}(x))$$

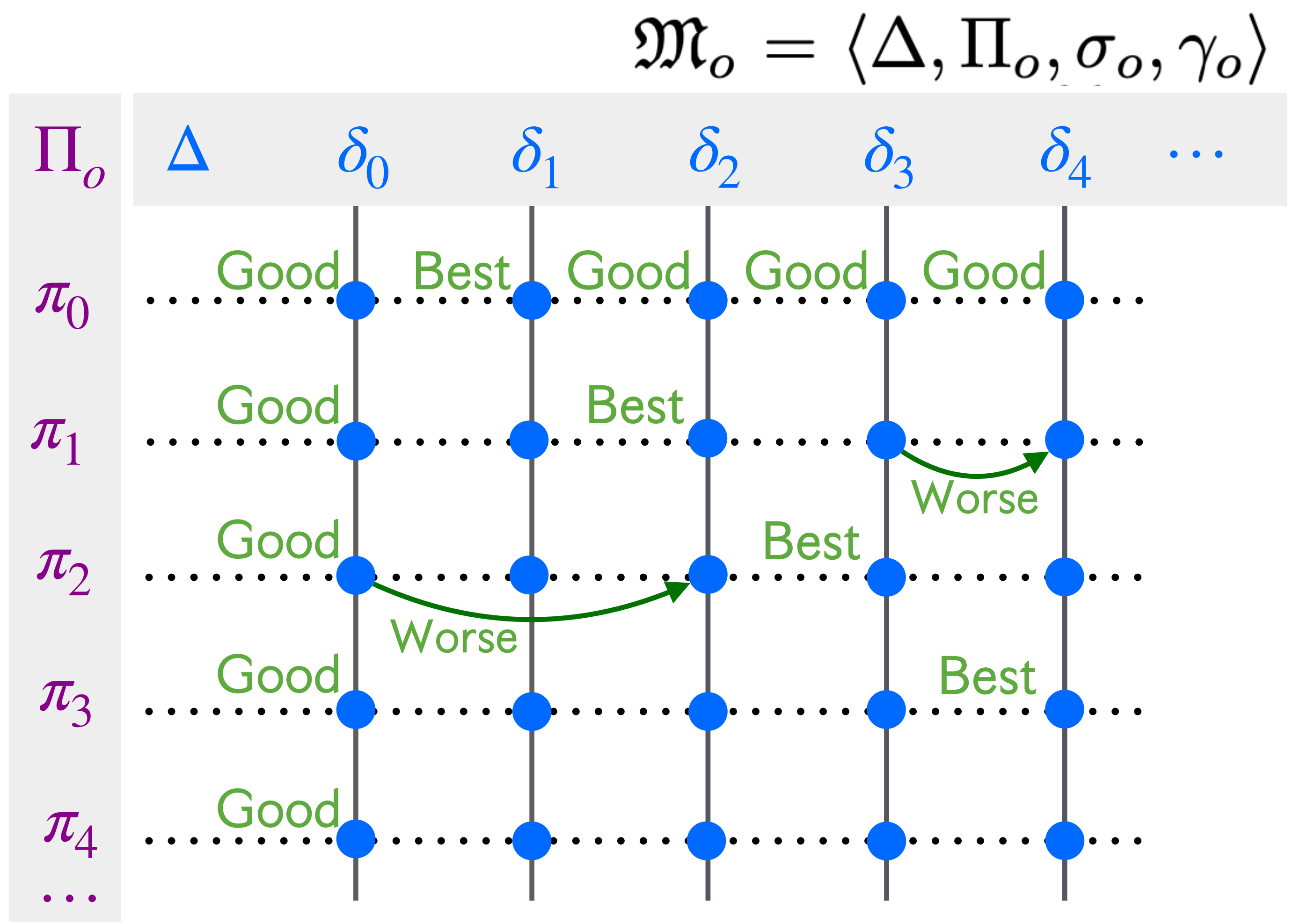
$$E_1 = E_0 \lor \Diamond_*(\texttt{Best}(x) \land \forall y.(\texttt{Best}(y) \leftrightarrow x \doteq y))$$

$$E_2 = \Diamond_*(\forall x.\texttt{Good}(x) \lor \texttt{Best}(x))$$

Building the exponential structure from a model $\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$:

- Add one rigid unary predicate E_i per monodic sub-formula $\Diamond_* \psi_i$
  (Set the same extension in structure)
- Select from Π_o
  - a π if no $\Diamond$-subformulas; otherwise:
  - a π for each sentential $\Diamond_* \psi$ subformula





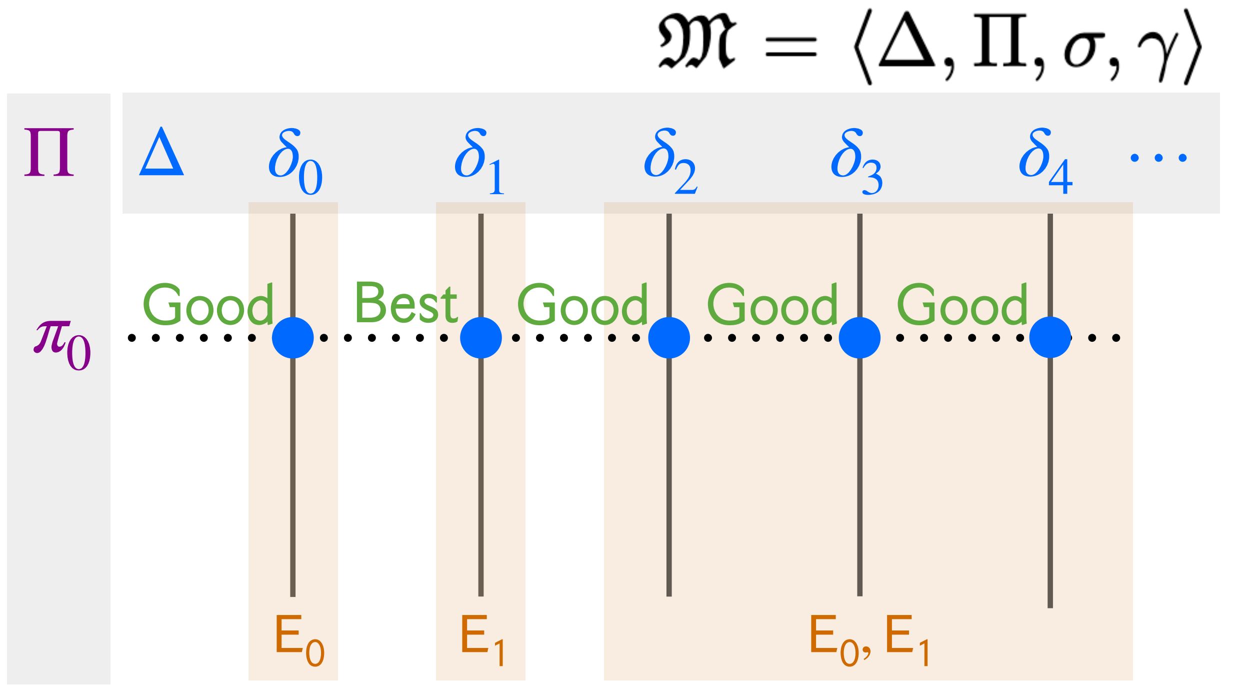
Permutational Representatives

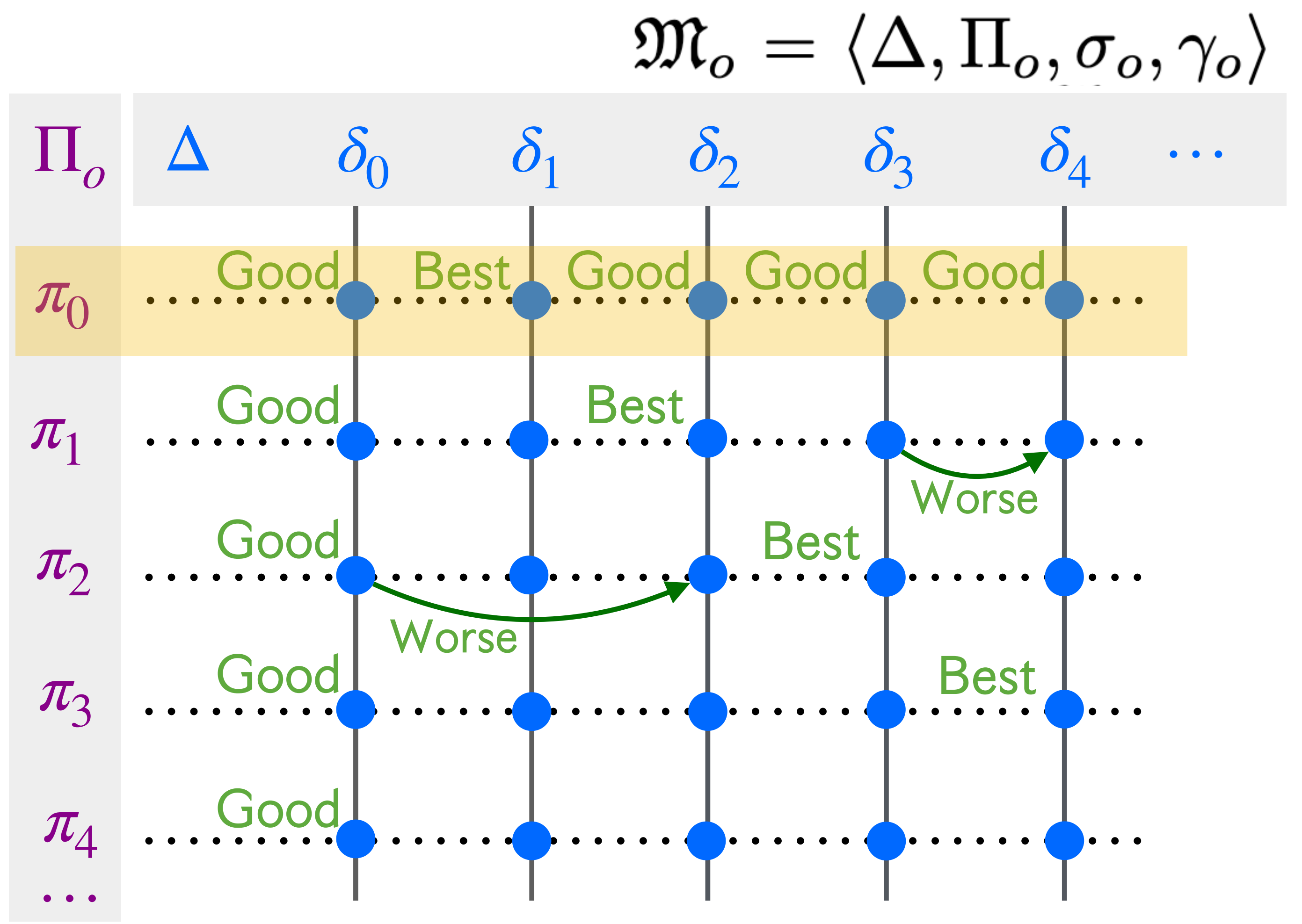
For any satisfiable ϕ, we construct an (exp.) structure that yields a model

$$\phi = \exists^{=1}x.(E_0) \land \forall x.(E_1) \land E_2 \text{ with}$$
$$E_0 = \Box_*(\mathtt{Good}(x))$$
$$E_1 = E_0 \lor \Diamond_*(\mathtt{Best}(x) \land \forall y.(\mathtt{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\mathtt{Good}(x) \lor \mathtt{Best}(x))$$

<u>Building the exponential structure from a model $\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$:</u>

- Add one rigid unary predicate E_i per monodic sub-formula $\Diamond_* \psi_i$
  (Set the same extension in structure)
- Select from Π_o
  - a π if no $\Diamond$-subformulas; otherwise:
  - a π for each sentential $\Diamond_* \psi$ subformula





Permutational Representatives

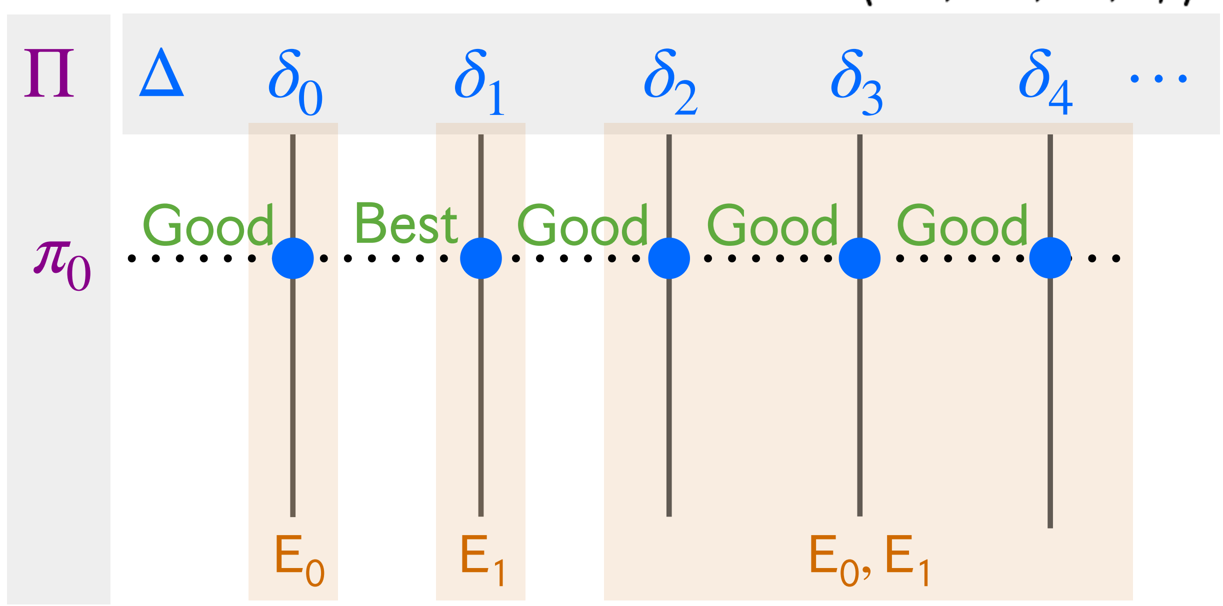
For any satisfiable ϕ, we construct an (exp.) structure that yields a model

$$\phi = \exists^{=1}x.(E_0) \land \forall x.(E_1) \land E_2 \text{ with}$$
$$E_0 = \Box_*(\mathtt{Good}(x))$$
$$E_1 = E_0 \lor \Diamond_*(\mathtt{Best}(x) \land \forall y.(\mathtt{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\mathtt{Good}(x) \lor \mathtt{Best}(x))$$

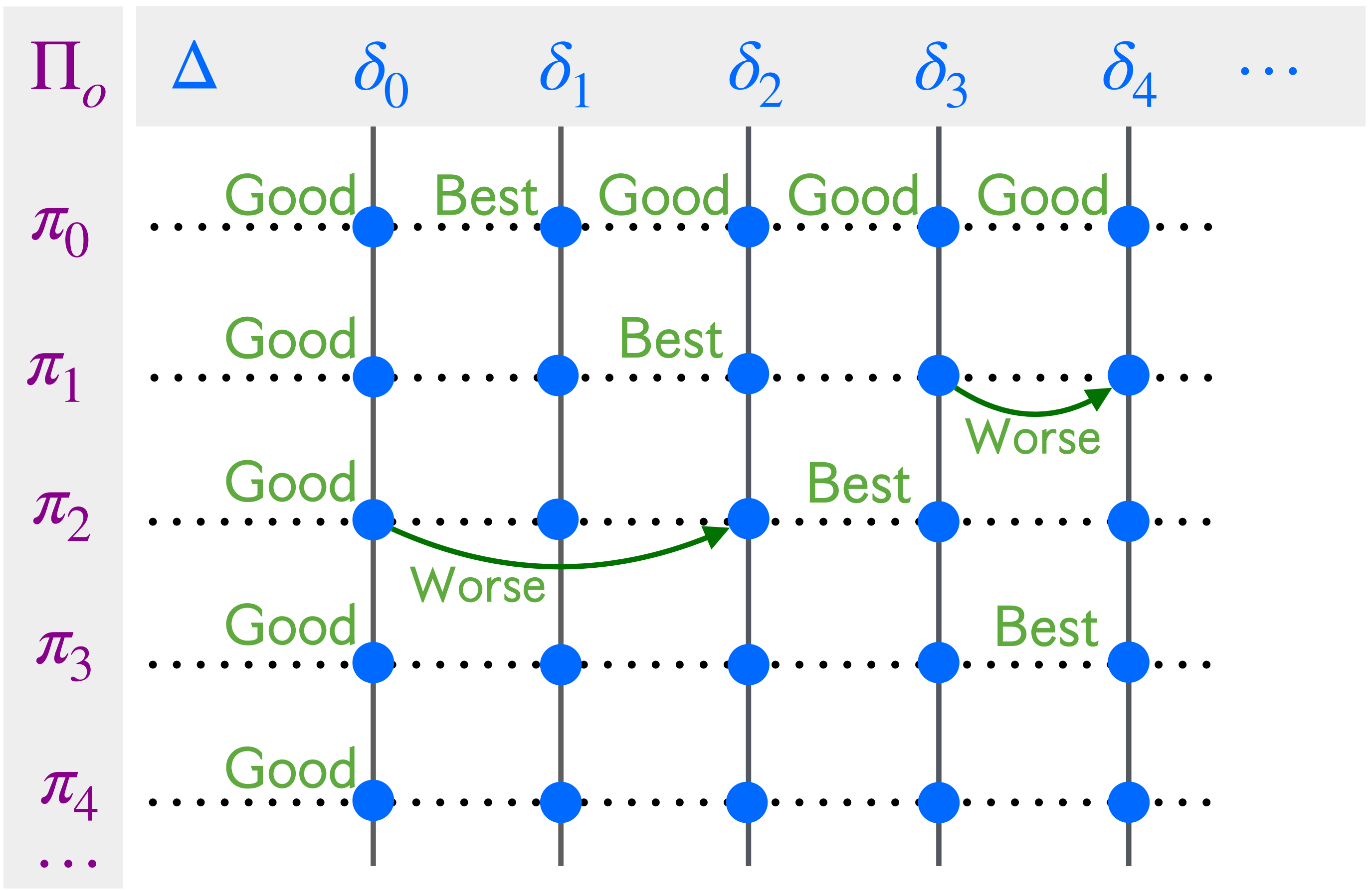
Building the exponential structure from a model $\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$:

- Add one rigid unary predicate E_i per monodic sub-formula $\Diamond_* \psi_i$
  (Set the same extension in structure)
- Select from Π_o
  - a π if no $\Diamond$-subformulas; otherwise:
  - a π for each sentential $\Diamond_* \psi$ subformula

$\mathfrak{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$



$\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$



Permutational Representatives

For any satisfiable ϕ, we construct an (exp.) structure that yields a model

$$\phi = \exists^{=1}x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$

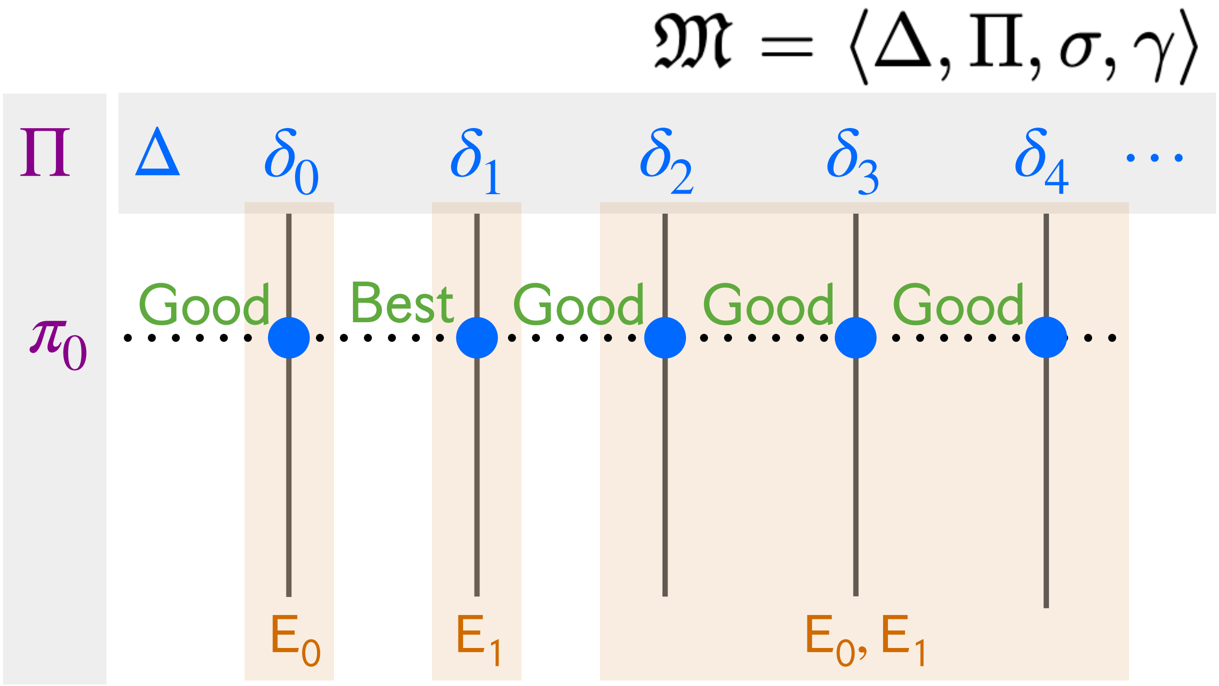
$$E_0 = \Box_*(\mathtt{Good}(x))$$

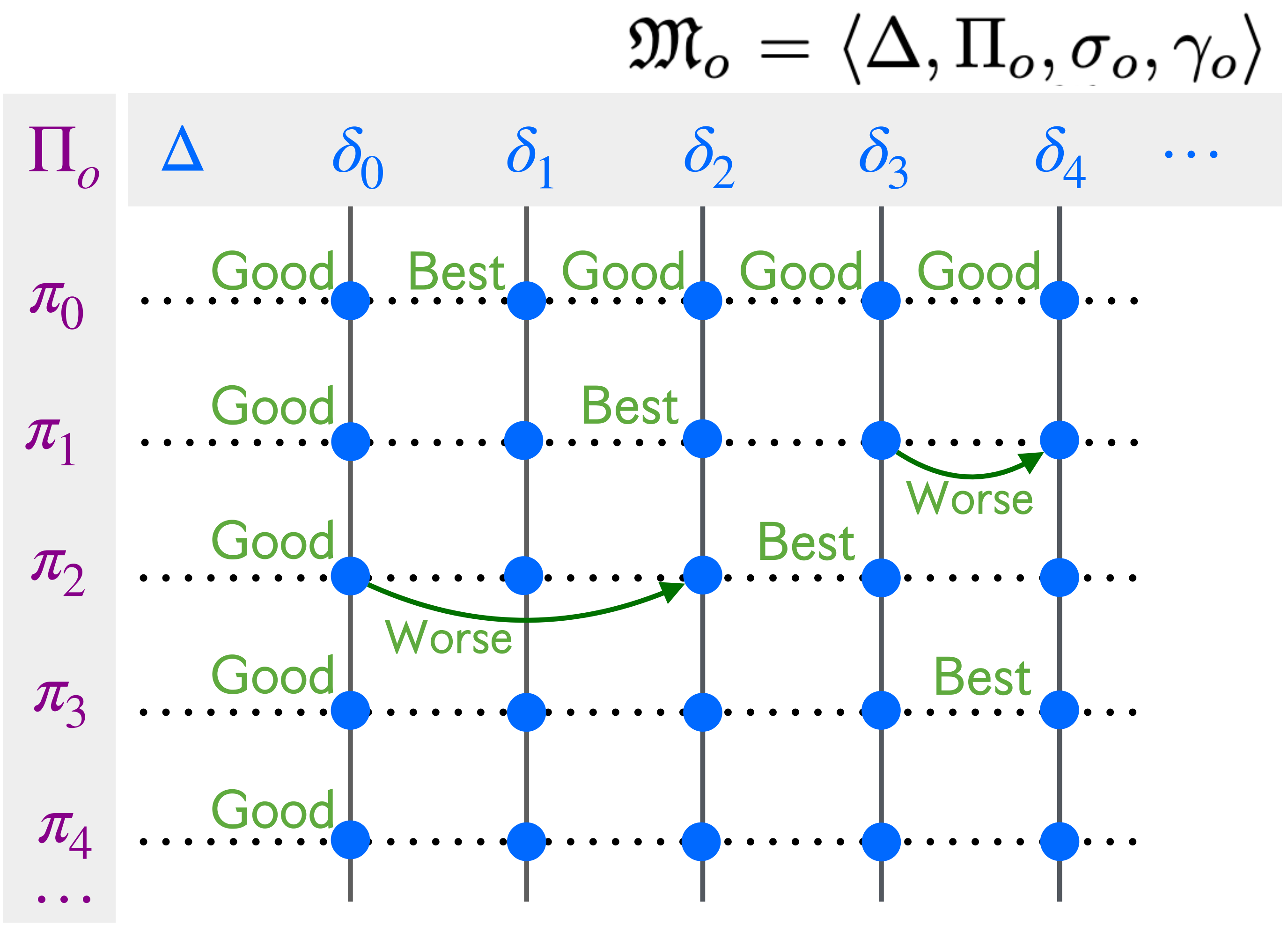
$$E_1 = E_0 \vee \Diamond_*(\mathtt{Best}(x) \wedge \forall y.(\mathtt{Best}(y) \leftrightarrow x \doteq y))$$

$$E_2 = \Diamond_*(\forall x.\mathtt{Good}(x) \vee \mathtt{Best}(x))$$

<u>Building the exponential structure from a model $\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$:</u>

- Add one rigid unary predicate E_i per monodic sub-formula $\Diamond_* \psi_i$
  (Set the same extension in structure)
- Select from Π_o
  - a π if no $\Diamond$-subformulas; otherwise:
  - a π for each sentential $\Diamond_* \psi$ subformula
  - for each realised E-type T,
    pick δ of type T
    a π satisfying ψ_i for δ per $E_i \in T$





Permutational Representatives

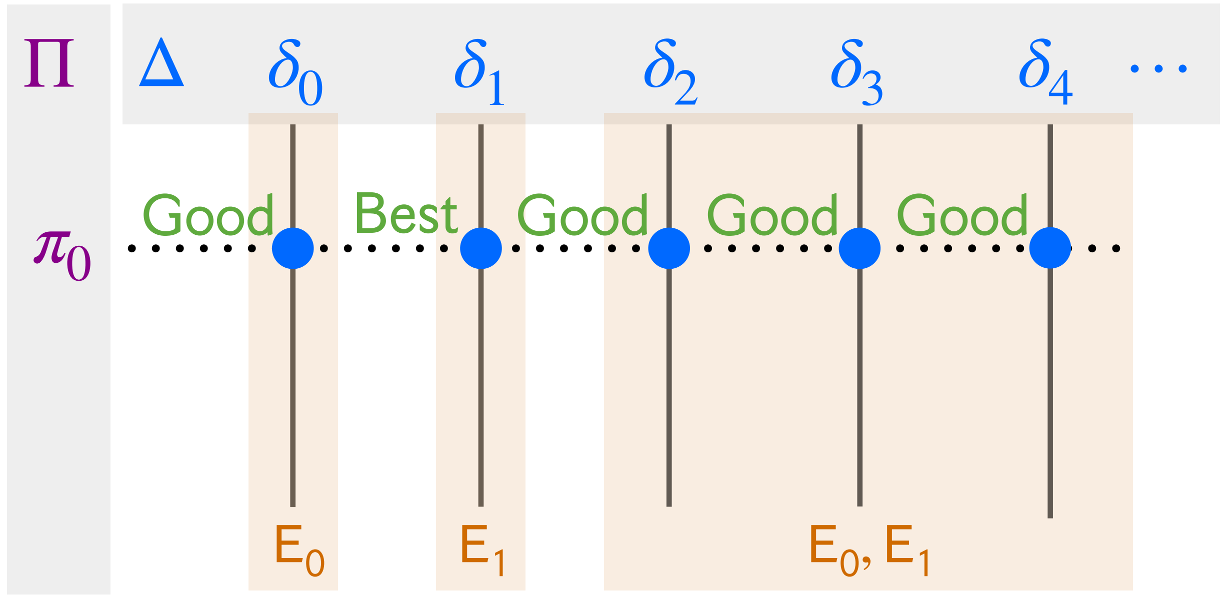
For any satisfiable ϕ, we construct an (exp.) structure that yields a model

$$\phi = \exists^{=1} x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$
$$E_0 = \Box_*(\texttt{Good}(x))$$
$$E_1 = E_0 \vee \Diamond_*(\texttt{Best}(x) \wedge \forall y.(\texttt{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\texttt{Good}(x) \vee \texttt{Best}(x))$$

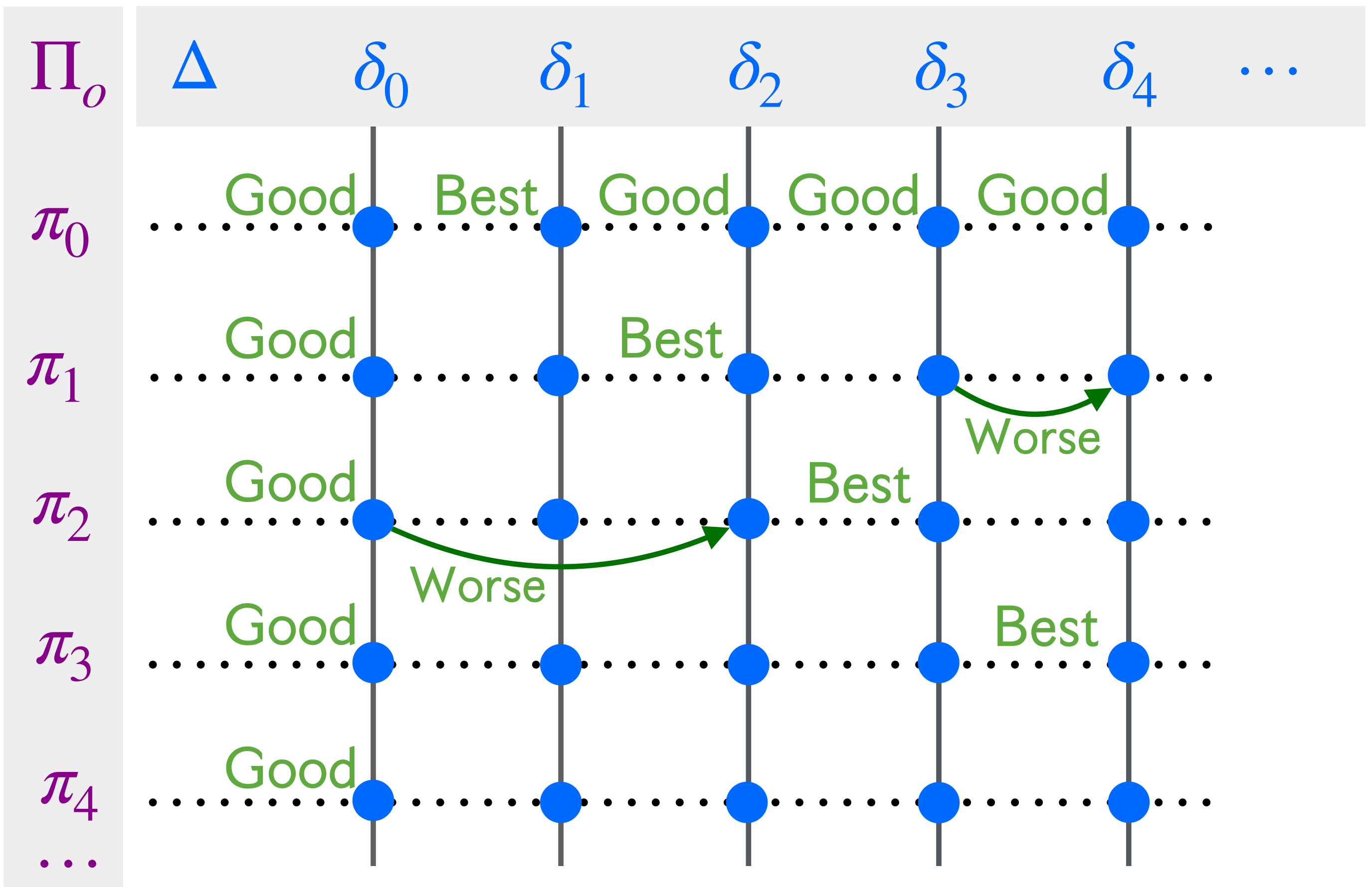
<u>Building the exponential structure from a model $\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$:</u>

- Add one rigid unary predicate E_i per monodic sub-formula $\Diamond_* \psi_i$
  (Set the same extension in structure)
- Select from Π_o
  - a π if no $\Diamond$-subformulas; otherwise:
  - a π for each sentential $\Diamond_* \psi$ subformula
  - for each realised E-type T, $\{\mathsf{E}_0\}, \{\mathsf{E}_1\}, \{\mathsf{E}_0, \mathsf{E}_1\}$
    pick δ of type T
    a π satisfying ψ_i for δ per $\mathsf{E}_i \in T$

$\mathfrak{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$



$\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$



Permutational Representatives

For any satisfiable ϕ, we construct an (exp.) structure that yields a model

$$\phi = \exists^{=1} x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$

$$E_0 = \Box_*(\texttt{Good}(x))$$

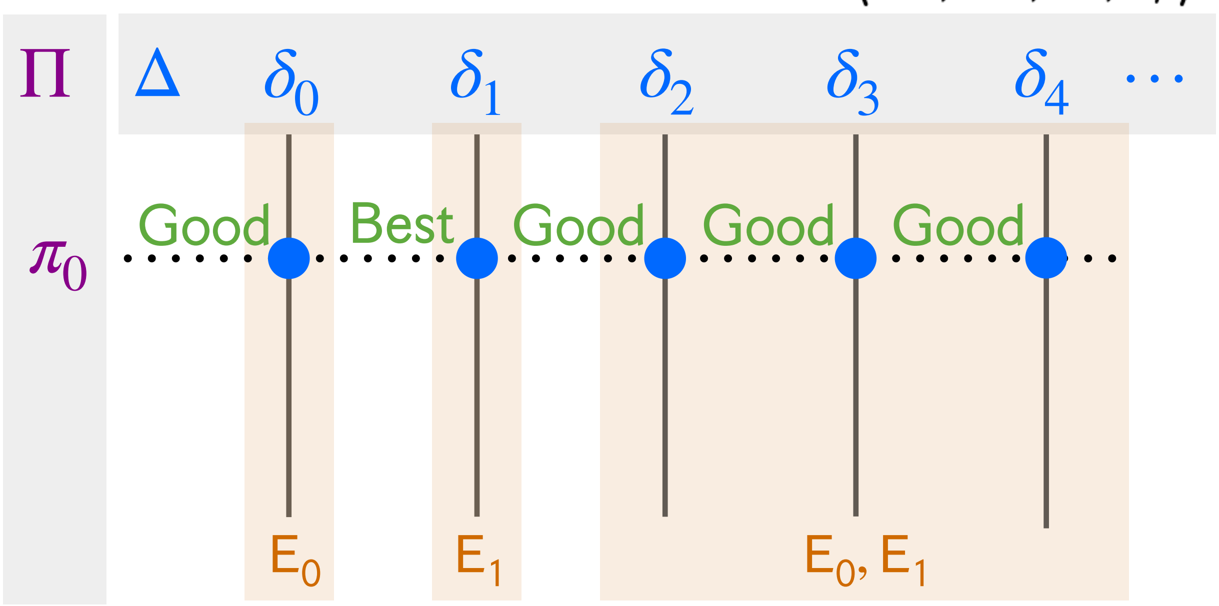
$$E_1 = E_0 \vee \Diamond_*(\texttt{Best}(x) \wedge \forall y.(\texttt{Best}(y) \leftrightarrow x \doteq y))$$

$$E_2 = \Diamond_*(\forall x.\texttt{Good}(x) \vee \texttt{Best}(x))$$

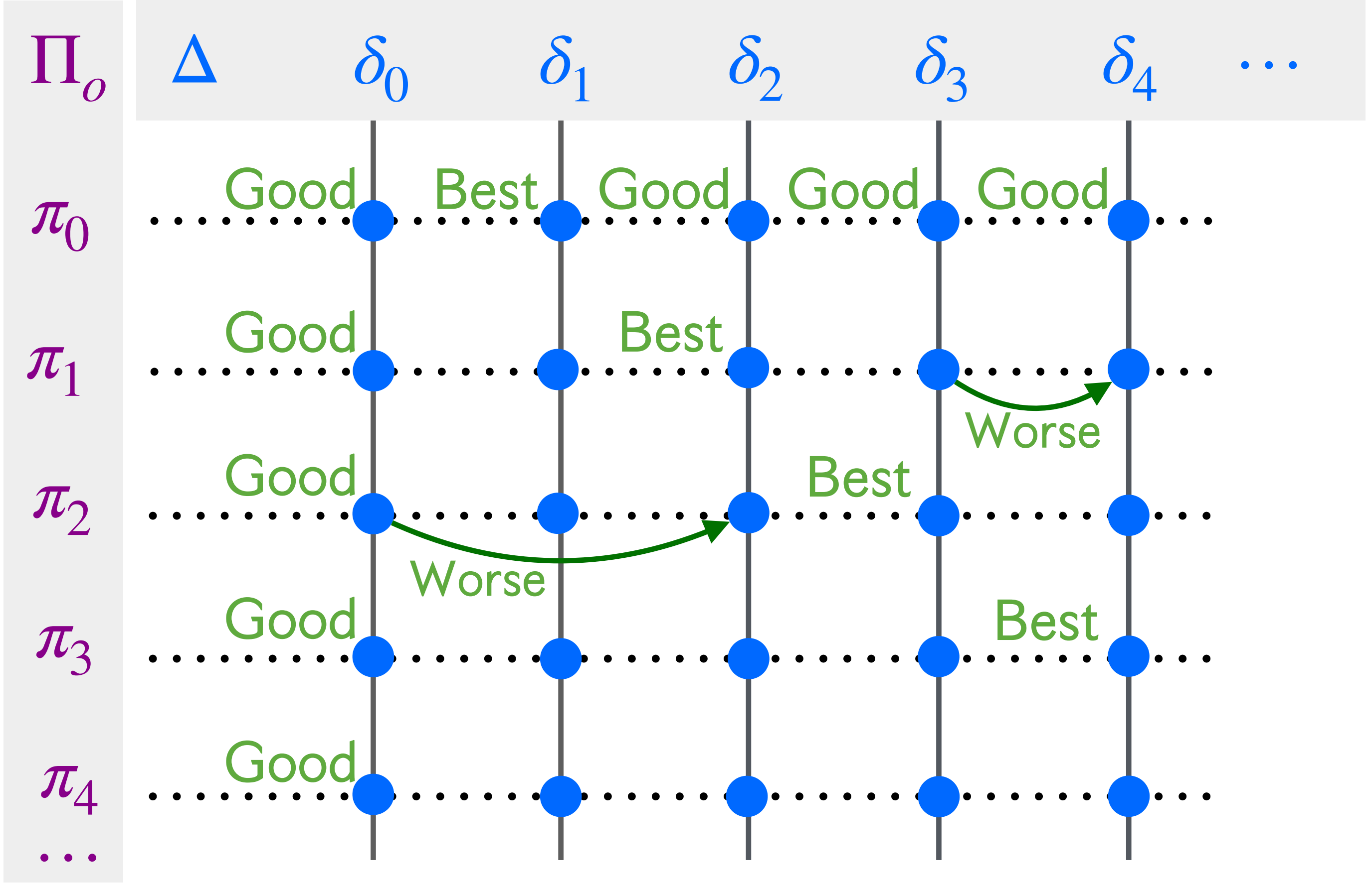
<u>Building the exponential structure from a model $\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$:</u>

- Add one rigid unary predicate E_i per monodic sub-formula $\Diamond_* \psi_i$ (Set the same extension in structure)
- Select from Π_o
  - a π if no $\Diamond$-subformulas; otherwise:
  - a π for each sentential $\Diamond_* \psi$ subformula
  - for each realised E-type T, $\{\mathsf{E}_0\}, \{\mathsf{E}_1\}, \{\mathsf{E}_0, \mathsf{E}_1\}$
    pick δ of type T
    a π satisfying ψ_i for δ per $\mathsf{E}_i \in T$

$\mathfrak{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$



$\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$



Permutational Representatives

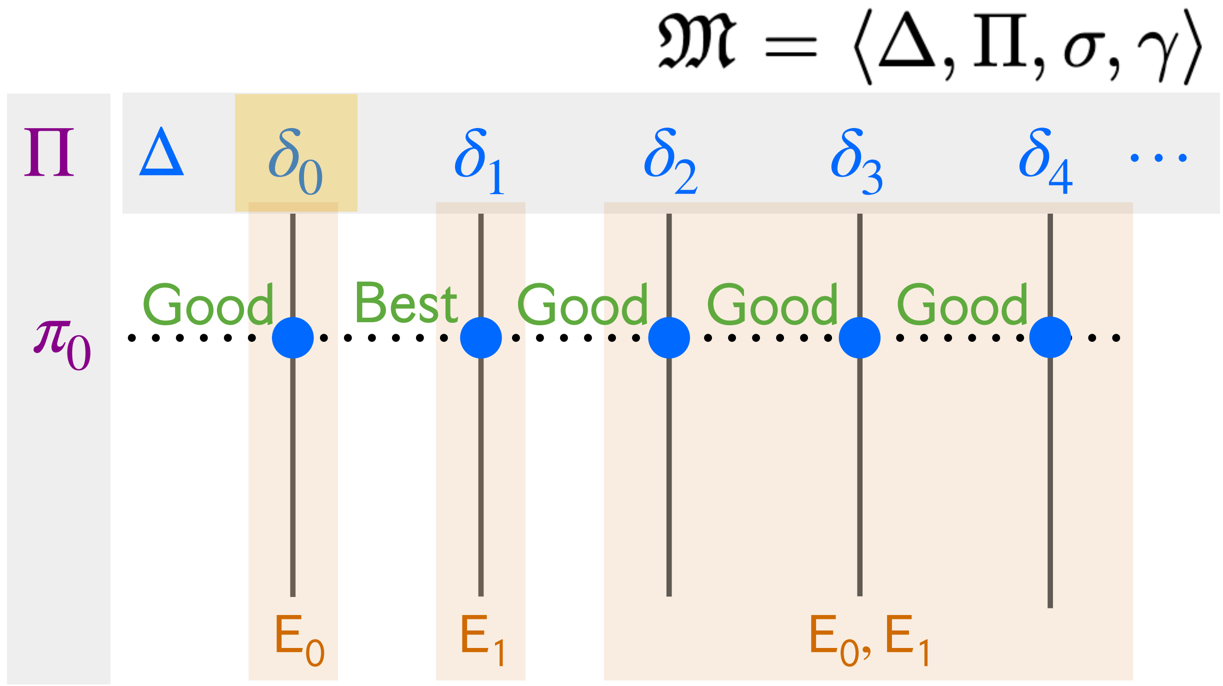
For any satisfiable ϕ, we construct an (exp.) structure that yields a model

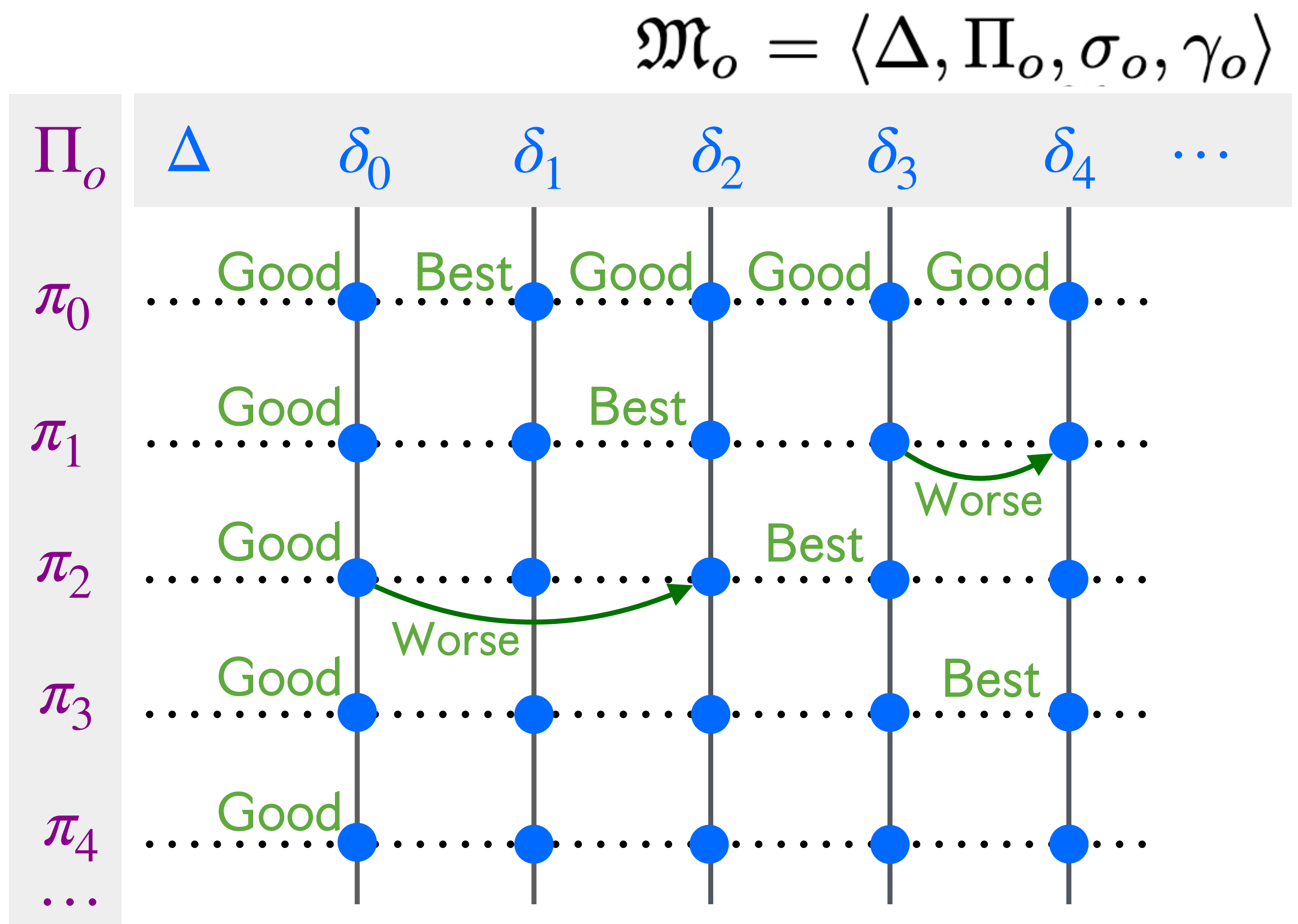
$$\phi = \exists^{=1}x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$
$$E_0 = \Box_*(\texttt{Good}(x))$$
$$E_1 = E_0 \vee \Diamond_*(\texttt{Best}(x) \wedge \forall y.(\texttt{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\texttt{Good}(x) \vee \texttt{Best}(x))$$

<u>Building the exponential structure from a model $\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$:</u>

- Add one rigid unary predicate E_i per monodic sub-formula $\Diamond_* \psi_i$ (Set the same extension in structure)

- Select from Π_o
  - a π if no $\Diamond$-subformulas; otherwise:
  - a π for each sentential $\Diamond_* \psi$ subformula
  - for each realised E-type T, $\{E_0\}, \{E_1\}, \{E_0, E_1\}$
    pick δ of type T
    a π satisfying ψ_i for δ per $E_i \in T$





Permutational Representatives

For any satisfiable ϕ, we construct an (exp.) structure that yields a model

$$\phi = \exists^{=1}x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$

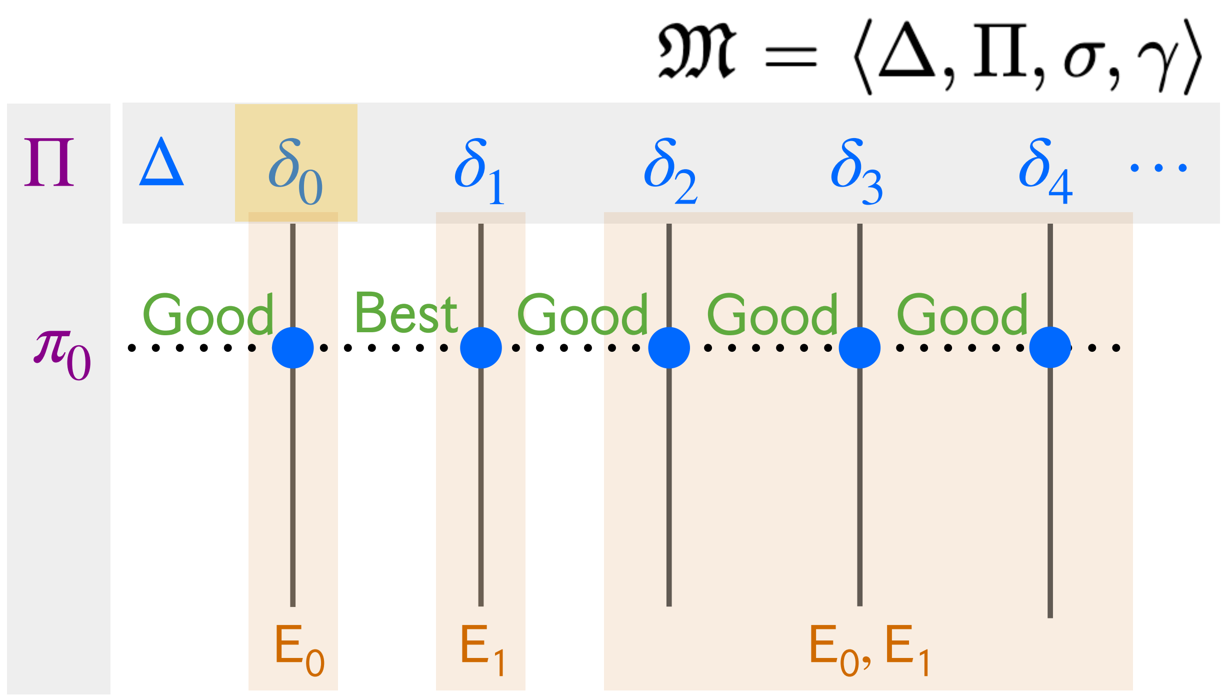
$$E_0 = \Box_*(\texttt{Good}(x))$$

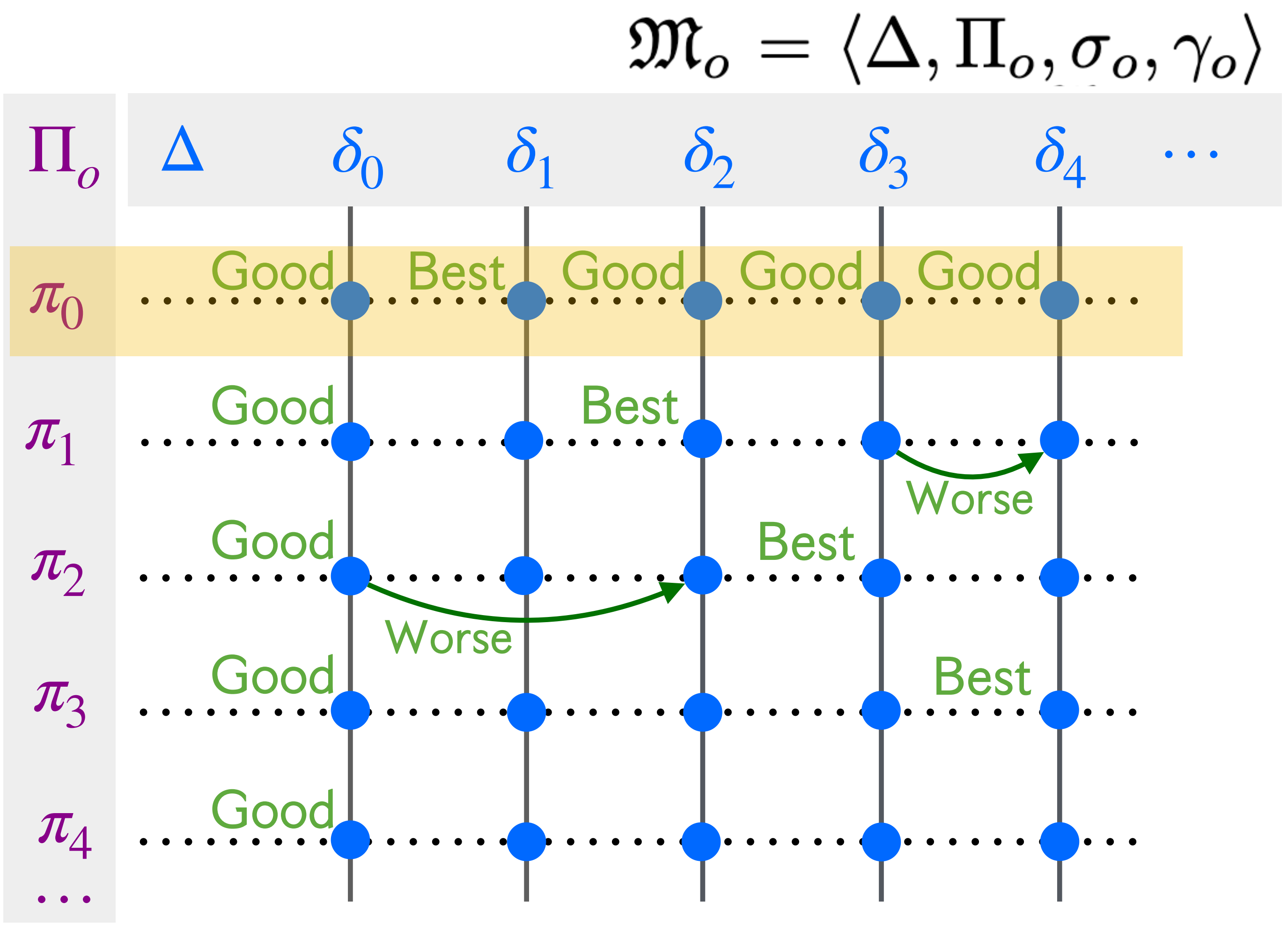
$$E_1 = E_0 \vee \Diamond_*(\texttt{Best}(x) \wedge \forall y.(\texttt{Best}(y) \leftrightarrow x \doteq y))$$

$$E_2 = \Diamond_*(\forall x.\texttt{Good}(x) \vee \texttt{Best}(x))$$

Building the exponential structure from a model $\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$:

- Add one rigid unary predicate E_i per monodic sub-formula $\Diamond_* \psi_i$ (Set the same extension in structure)
- Select from Π_o
  - a π if no $\Diamond$-subformulas; otherwise:
  - a π for each sentential $\Diamond_* \psi$ subformula
  - for each realised E-type T, $\{\mathsf{E}_0\}, \{\mathsf{E}_1\}, \{\mathsf{E}_0, \mathsf{E}_1\}$
    pick δ of type T
    a π satisfying ψ_i for δ per $\mathsf{E}_i \in T$





Permutational Representatives

For any satisfiable ϕ, we construct an (exp.) structure that yields a model

$$\phi = \exists^{=1} x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$

$$E_0 = \Box_*(\texttt{Good}(x))$$

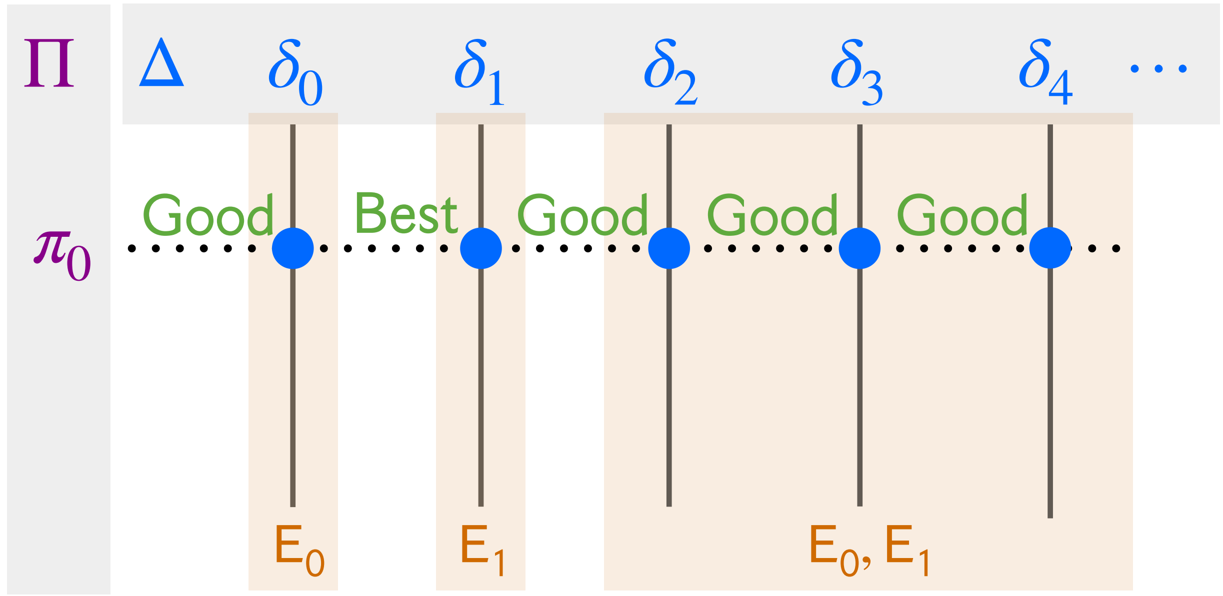
$$E_1 = E_0 \vee \Diamond_*(\texttt{Best}(x) \wedge \forall y.(\texttt{Best}(y) \leftrightarrow x \doteq y))$$

$$E_2 = \Diamond_*(\forall x.\texttt{Good}(x) \vee \texttt{Best}(x))$$

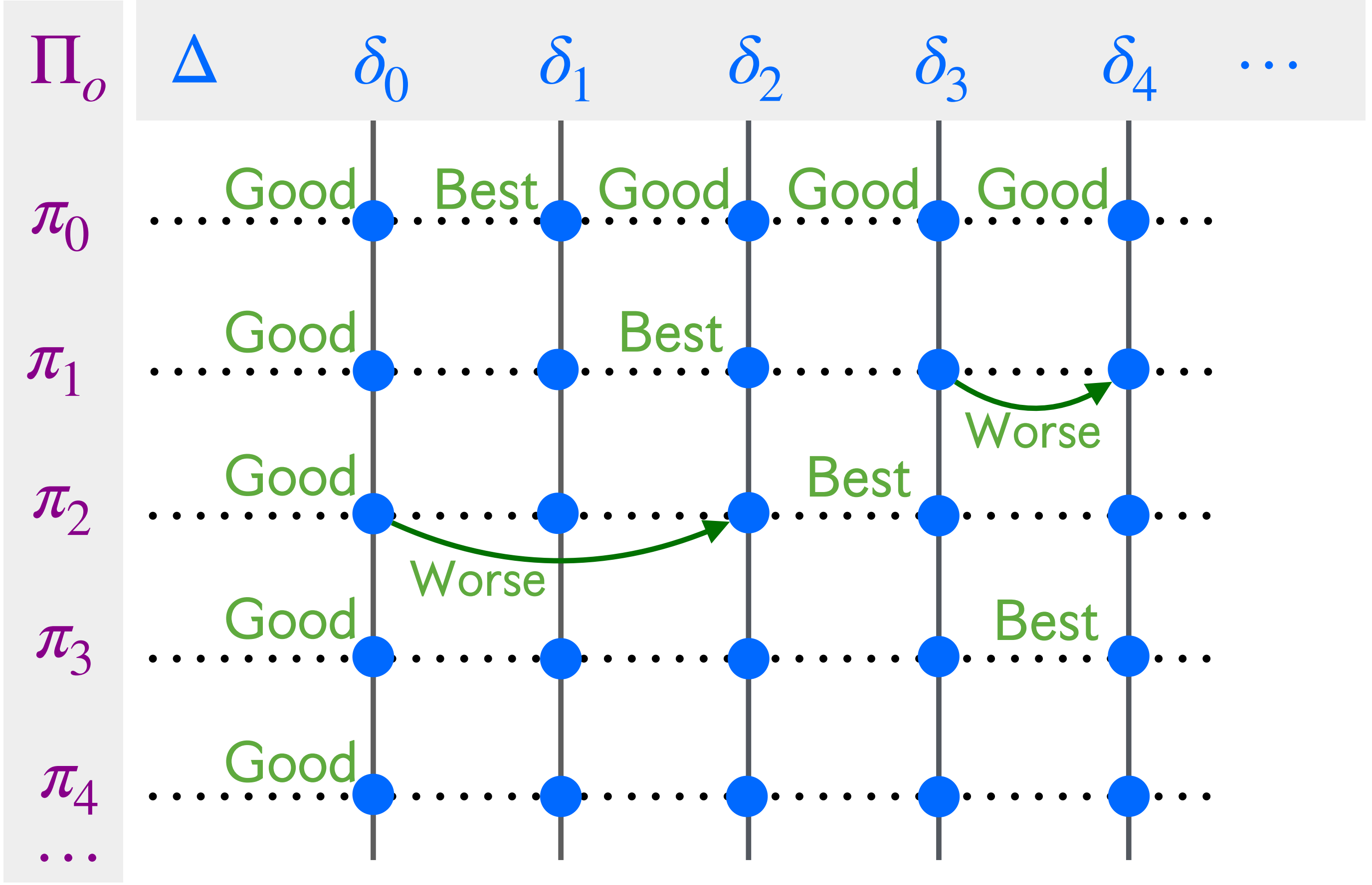
<u>Building the exponential structure from a model</u> $\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$:

- Add one rigid unary predicate E_i per monodic sub-formula $\Diamond_* \psi_i$
  (Set the same extension in structure)
- Select from Π_o
  - a π if no $\Diamond$-subformulas; otherwise:
  - a π for each sentential $\Diamond_* \psi$ subformula
  - for each realised E-type T, $\{\mathsf{E}_0\}, \{\mathsf{E}_1\}, \{\mathsf{E}_0, \mathsf{E}_1\}$
    pick δ of type T
    a π satisfying ψ_i for δ per $\mathsf{E}_i \in T$

$\mathfrak{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$



$\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$



Permutational Representatives

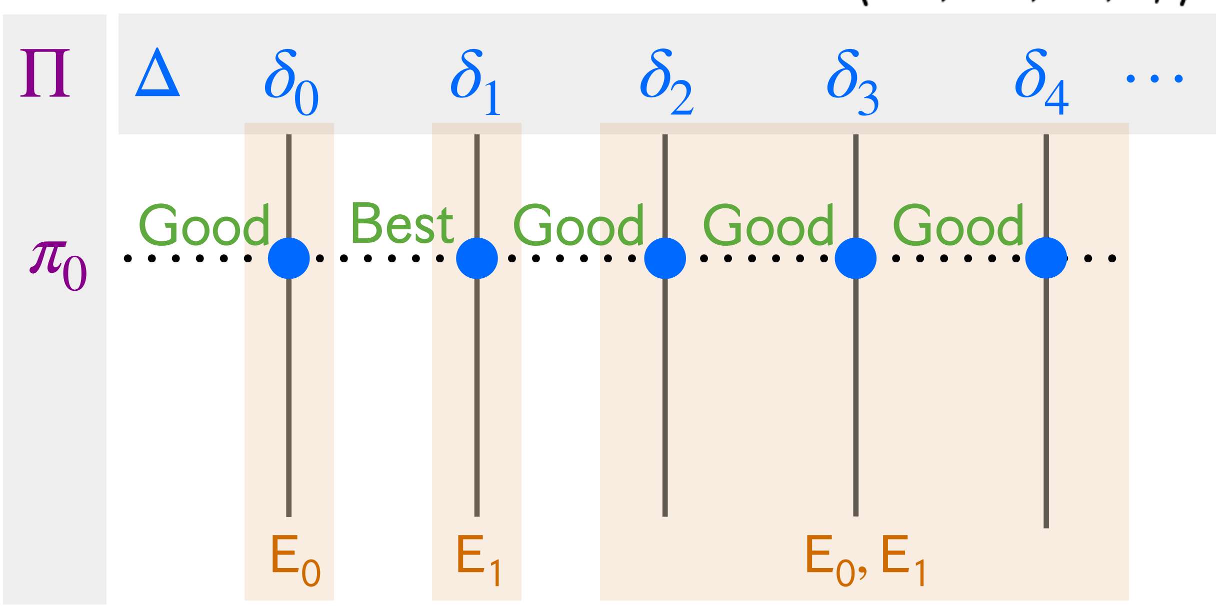
For any satisfiable ϕ, we construct an (exp.) structure that yields a model

$$\phi = \exists^{=1}x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$
$$E_0 = \Box_*(\mathtt{Good}(x))$$
$$E_1 = E_0 \vee \Diamond_*(\mathtt{Best}(x) \wedge \forall y.(\mathtt{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\mathtt{Good}(x) \vee \mathtt{Best}(x))$$

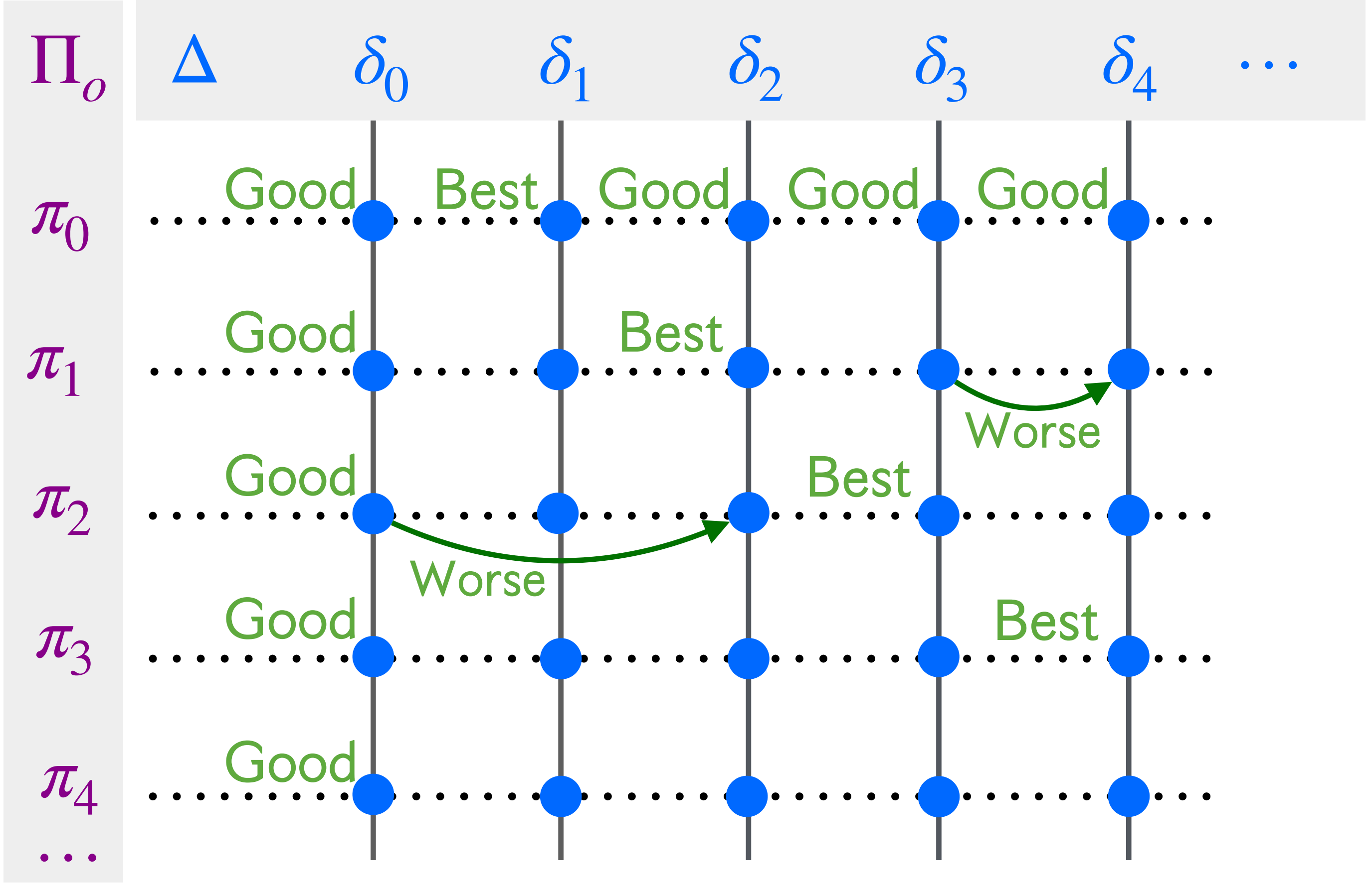
<u>Building the exponential structure from a model $\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$:</u>

- Add one rigid unary predicate E_i per monodic sub-formula $\Diamond_* \psi_i$ (Set the same extension in structure)
- Select from Π_o
  - a π if no $\Diamond$-subformulas; otherwise:
  - a π for each sentential $\Diamond_* \psi$ subformula
  - for each realised E-type T, $\{\mathsf{E}_0\}, \{\mathsf{E}_1\}, \{\mathsf{E}_0, \mathsf{E}_1\}$
    pick δ of type T
    a π satisfying ψ_i for δ per $\mathsf{E}_i \in T$

$\mathfrak{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$



$\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$



Permutational Representatives

For any satisfiable ϕ, we construct an (exp.) structure that yields a model

$$\phi = \exists^{=1}x.(E_0) \land \forall x.(E_1) \land E_2 \text{ with}$$

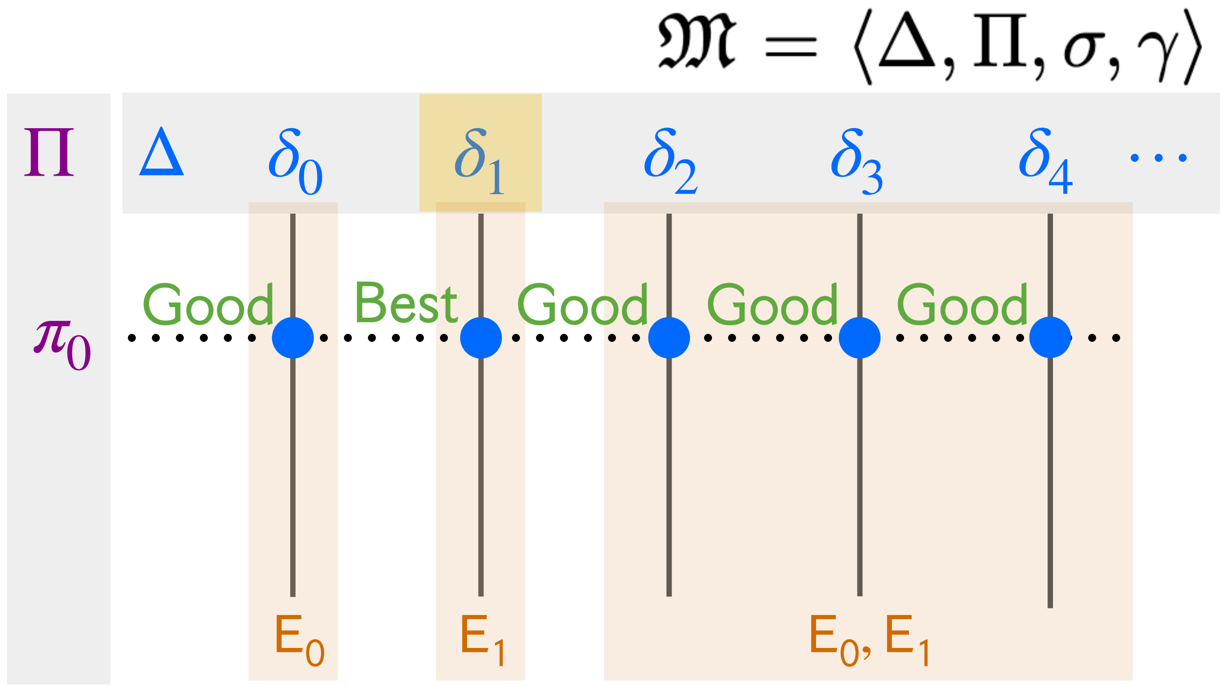
$$E_0 = \Box_*(\mathtt{Good}(x))$$

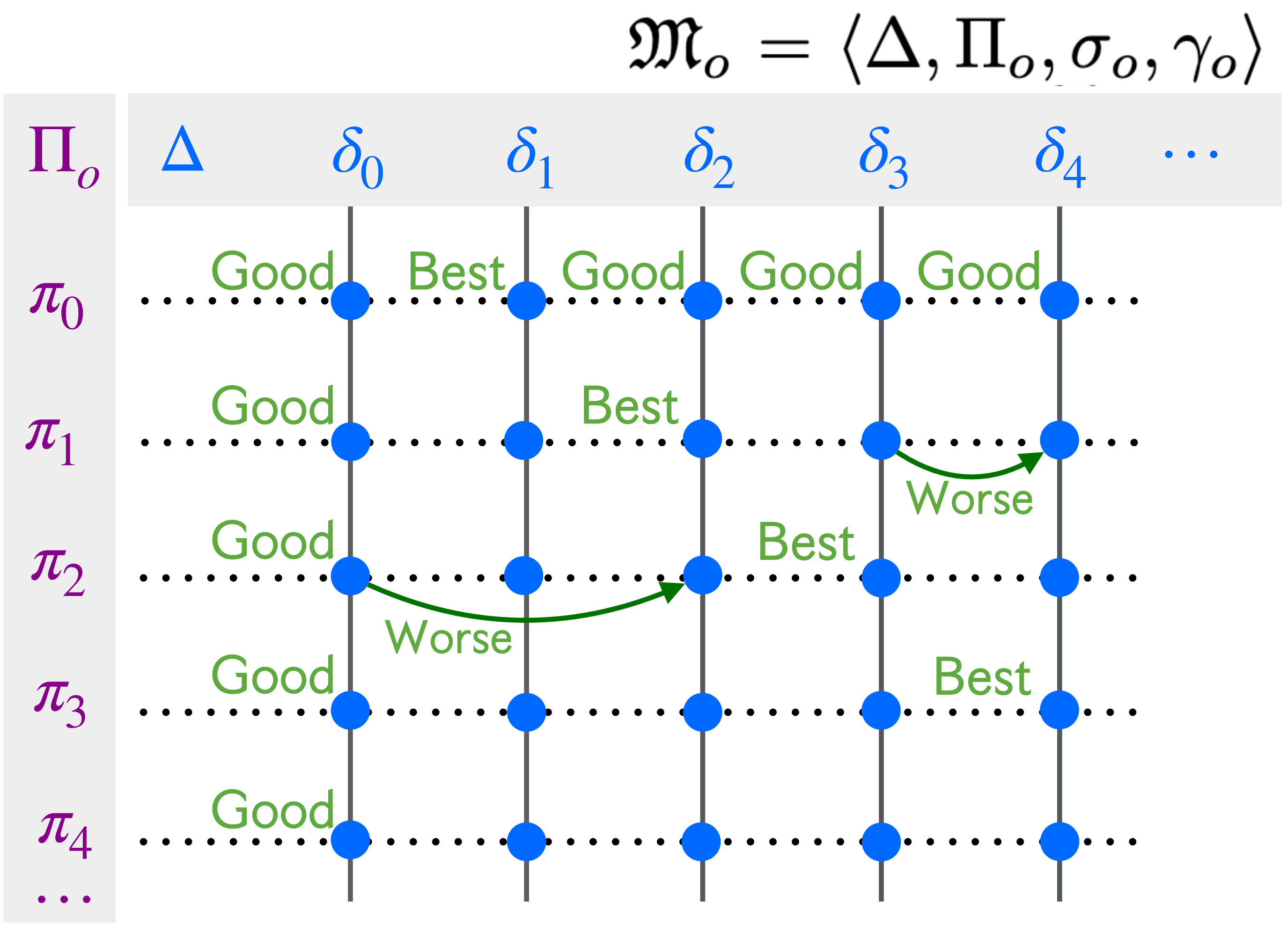
$$E_1 = E_0 \lor \Diamond_*(\mathtt{Best}(x) \land \forall y.(\mathtt{Best}(y) \leftrightarrow x \doteq y))$$

$$E_2 = \Diamond_*(\forall x.\mathtt{Good}(x) \lor \mathtt{Best}(x))$$

<u>Building the exponential structure from a model $\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$:</u>

- Add one rigid unary predicate E_i per monodic sub-formula $\Diamond_* \psi_i$ (Set the same extension in structure)
- Select from Π_o
  - a π if no $\Diamond$-subformulas; otherwise:
  - a π for each sentential $\Diamond_* \psi$ subformula
  - for each realised E-type T, $\{E_0\}, \{E_1\}, \{E_0, E_1\}$
    pick δ of type T
    a π satisfying ψ_i for δ per $E_i \in T$





Permutational Representatives

For any satisfiable ϕ, we construct an (exp.) structure that yields a model

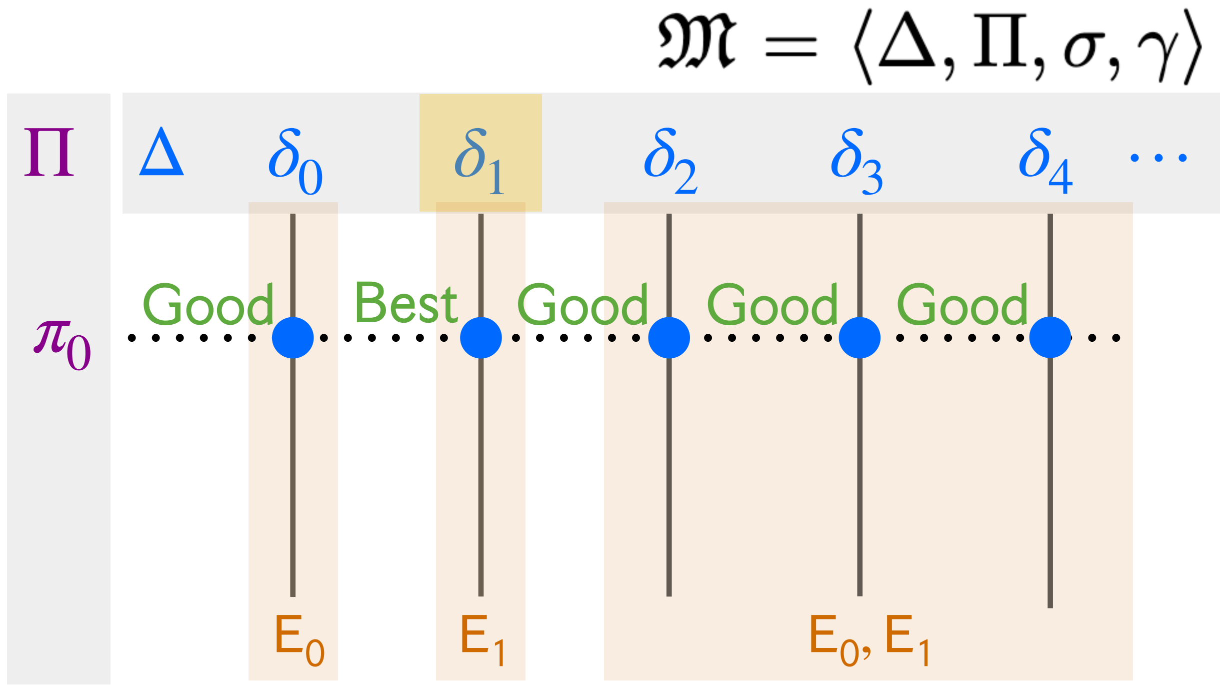
$$\phi = \exists^{=1}x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$

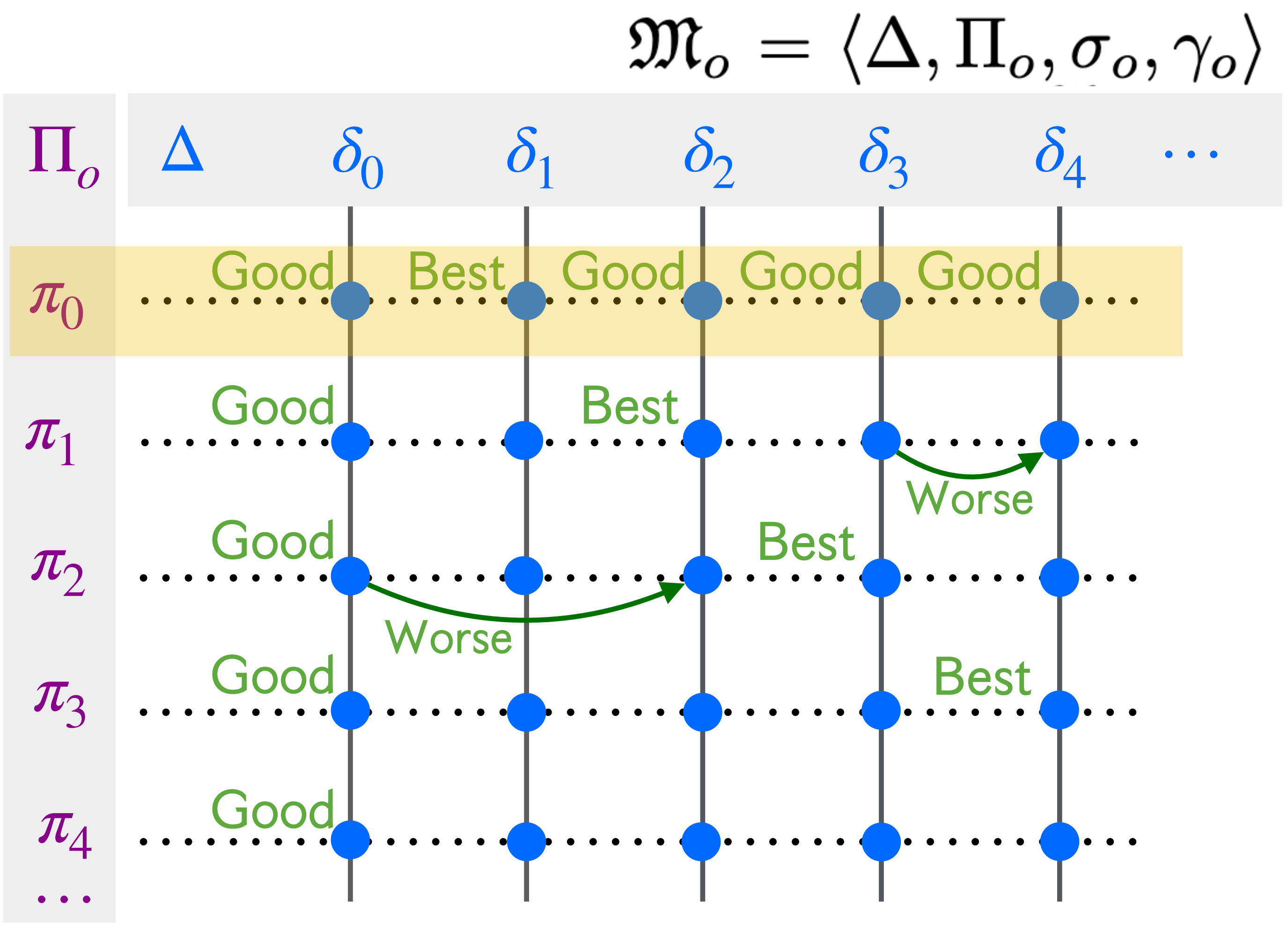
$$E_0 = \Box_*(\mathtt{Good}(x))$$
$$E_1 = E_0 \vee \Diamond_*(\mathtt{Best}(x) \wedge \forall y.(\mathtt{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\mathtt{Good}(x) \vee \mathtt{Best}(x))$$

<u>Building the exponential structure from a model</u> $\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$:

- Add one rigid unary predicate E_i per monodic sub-formula $\Diamond_* \psi_i$
  (Set the same extension in structure)

- Select from Π_o
  - a π if no $\Diamond$-subformulas; otherwise:
  - a π for each sentential $\Diamond_* \psi$ subformula
  - for each realised E-type T, $\{\mathsf{E}_0\}, \{\mathsf{E}_1\}, \{\mathsf{E}_0, \mathsf{E}_1\}$
    pick δ of type T
    a π satisfying ψ_i for δ per $\mathsf{E}_i \in T$





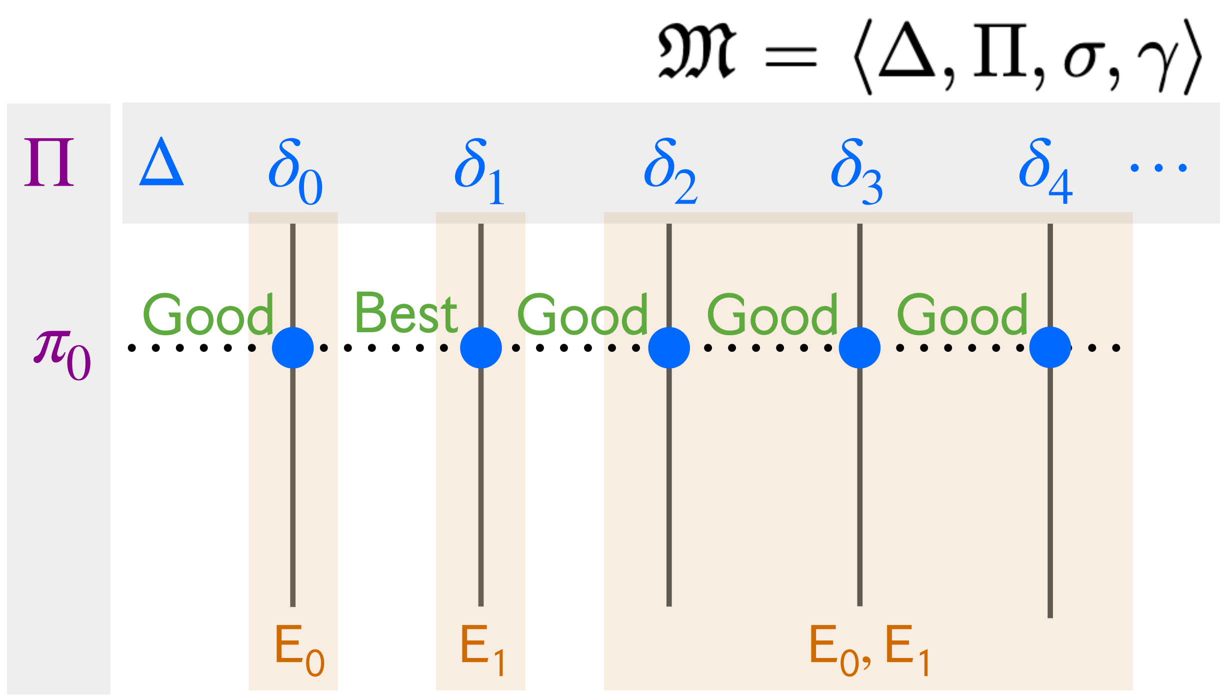
Permutational Representatives

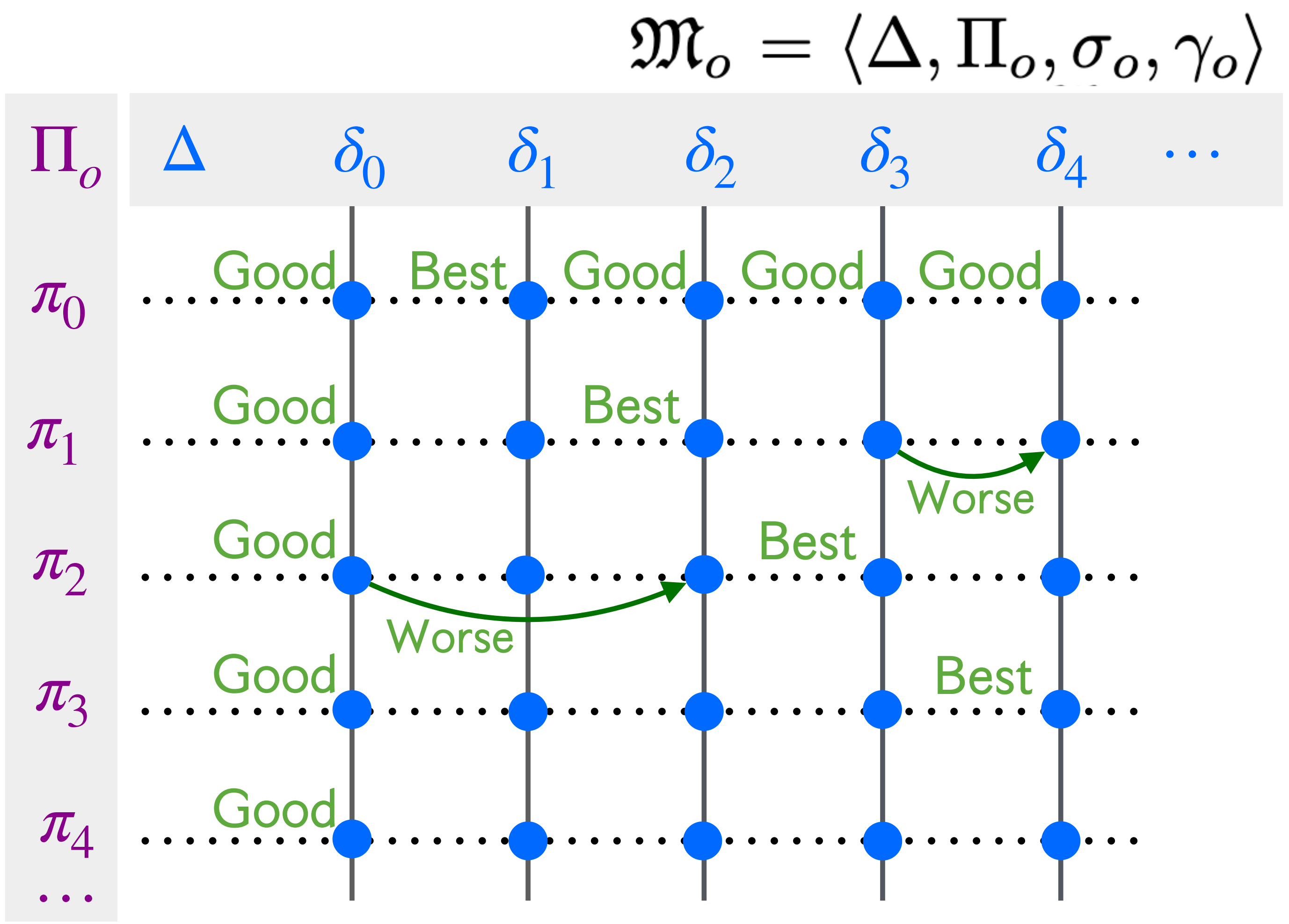
For any satisfiable ϕ, we construct an (exp.) structure that yields a model

$$\phi = \exists^{=1}x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$
$$E_0 = \Box_*(\mathtt{Good}(x))$$
$$E_1 = E_0 \vee \Diamond_*(\mathtt{Best}(x) \wedge \forall y.(\mathtt{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\mathtt{Good}(x) \vee \mathtt{Best}(x))$$

Building the exponential structure from a model $\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$:

- Add one rigid unary predicate E_i per monodic sub-formula $\Diamond_* \psi_i$ (Set the same extension in structure)
- Select from Π_o
  - a π if no $\Diamond$-subformulas; otherwise:
  - a π for each sentential $\Diamond_* \psi$ subformula
  - for each realised E-type T, $\{E_0\}, \{E_1\}, \{E_0, E_1\}$
    pick δ of type T
    a π satisfying ψ_i for δ per $E_i \in T$





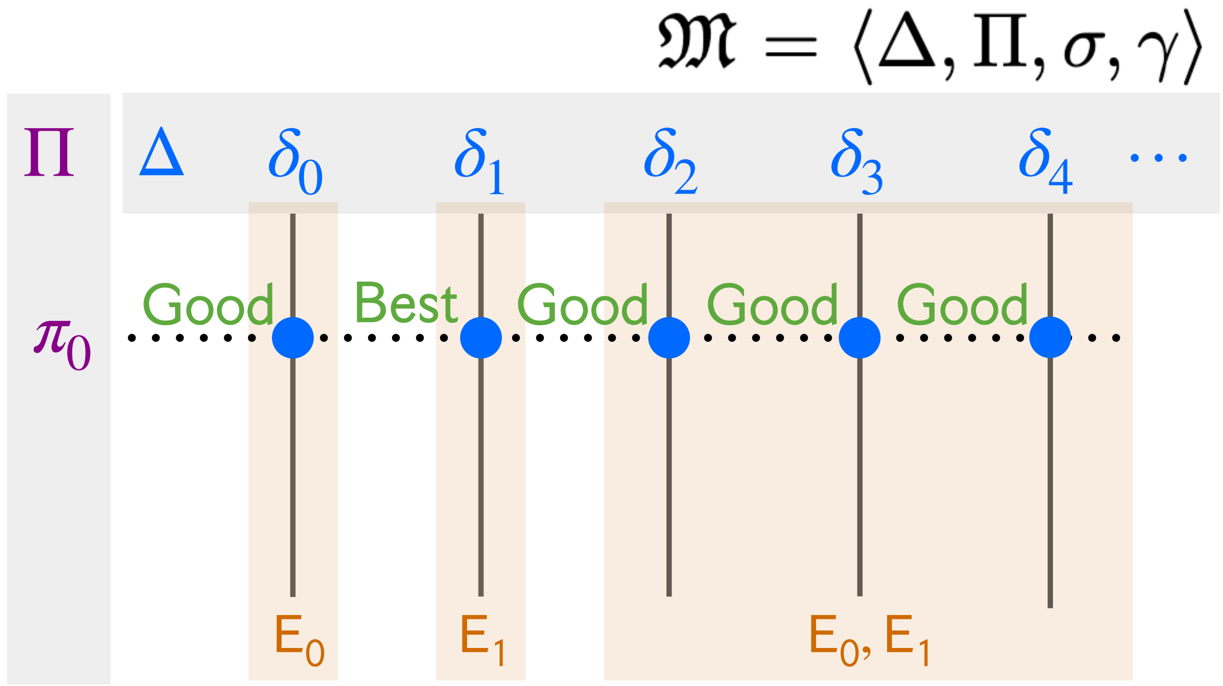
Permutational Representatives

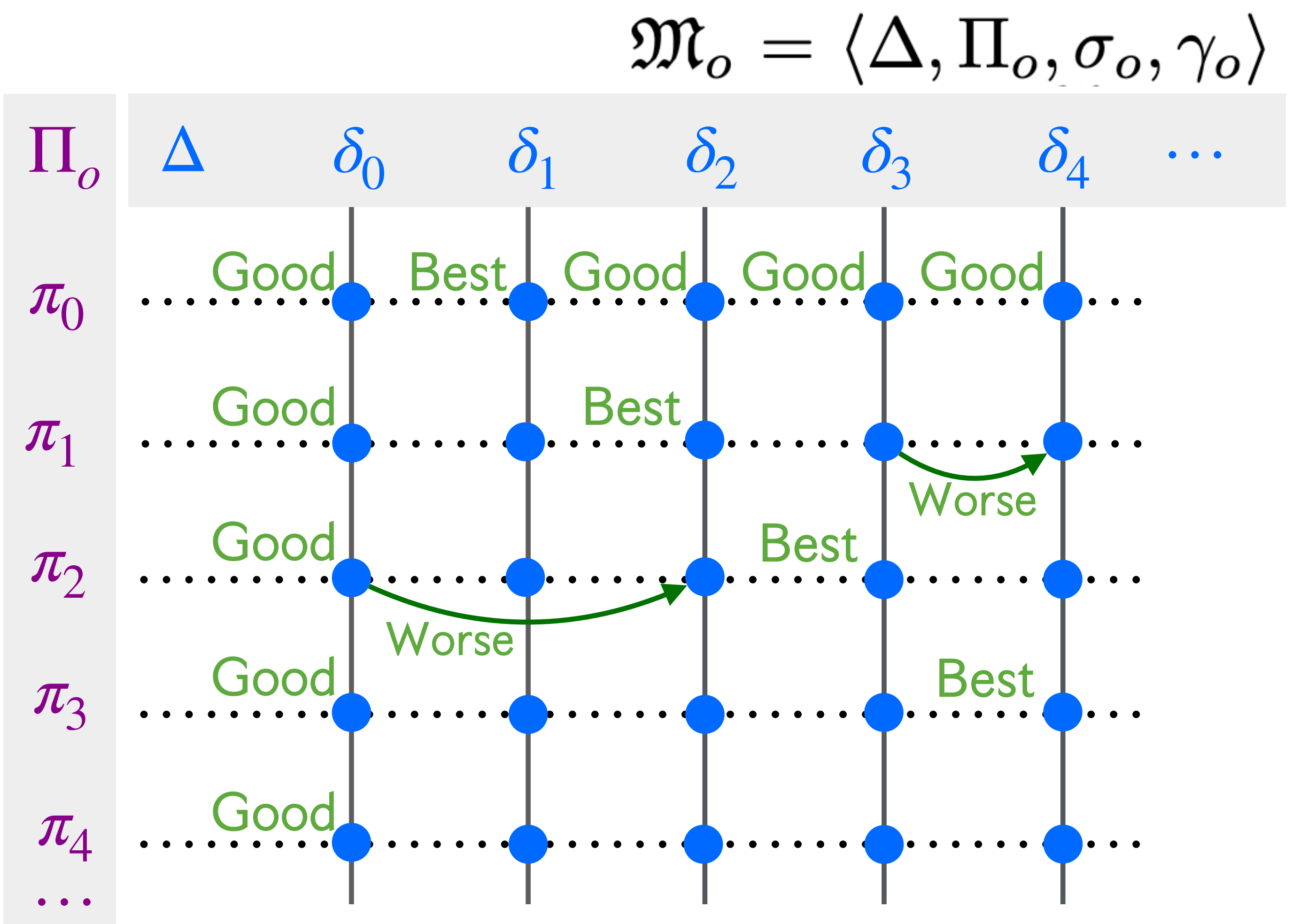
For any satisfiable ϕ, we construct an (exp.) structure that yields a model

$$\phi = \exists^{=1}x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$
$$E_0 = \Box_*(\mathtt{Good}(x))$$
$$E_1 = E_0 \vee \Diamond_*(\mathtt{Best}(x) \wedge \forall y.(\mathtt{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\mathtt{Good}(x) \vee \mathtt{Best}(x))$$

Building the exponential structure from a model $\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$:

- Add one rigid unary predicate E_i per monodic sub-formula $\Diamond_* \psi_i$ (Set the same extension in structure)
- Select from Π_o
  - a π if no $\Diamond$-subformulas; otherwise:
  - a π for each sentential $\Diamond_* \psi$ subformula
  - for each realised E-type T, $\{E_0\}, \{E_1\}, \{E_0, E_1\}$
    pick δ of type T
    a π satisfying ψ_i for δ per $E_i \in T$





Permutational Representatives

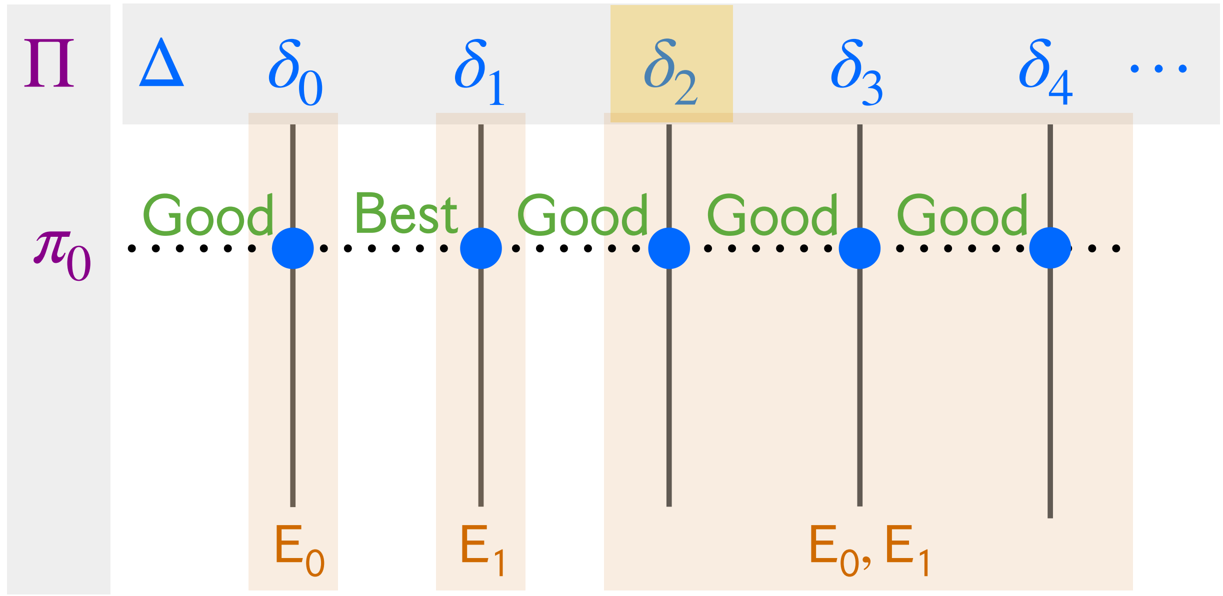
For any satisfiable ϕ, we construct an (exp.) structure that yields a model

$$\phi = \exists^{=1} x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$
$$E_0 = \Box_*(\mathtt{Good}(x))$$
$$E_1 = E_0 \vee \Diamond_*(\mathtt{Best}(x) \wedge \forall y.(\mathtt{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\mathtt{Good}(x) \vee \mathtt{Best}(x))$$

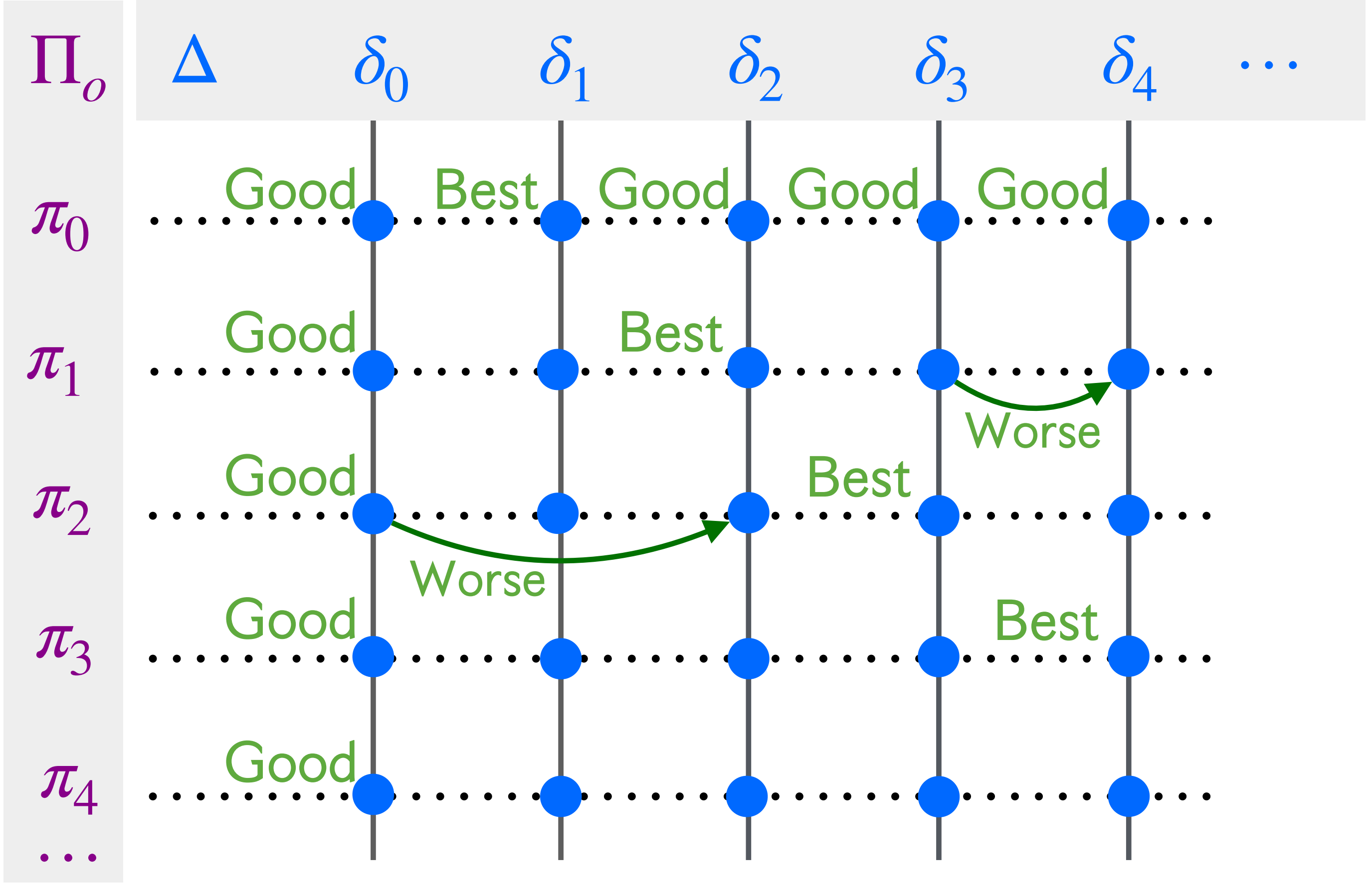
Building the exponential structure from a model $\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$:

- Add one rigid unary predicate E_i per monodic sub-formula $\Diamond_* \psi_i$ (Set the same extension in structure)
- Select from Π_o
  - a π if no $\Diamond$-subformulas; otherwise:
  - a π for each sentential $\Diamond_* \psi$ subformula
  - for each realised E-type T, $\{E_0\}, \{E_1\}, \{E_0, E_1\}$
    pick δ of type T
    a π satisfying ψ_i for δ per $E_i \in T$

$\mathfrak{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$



$\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$



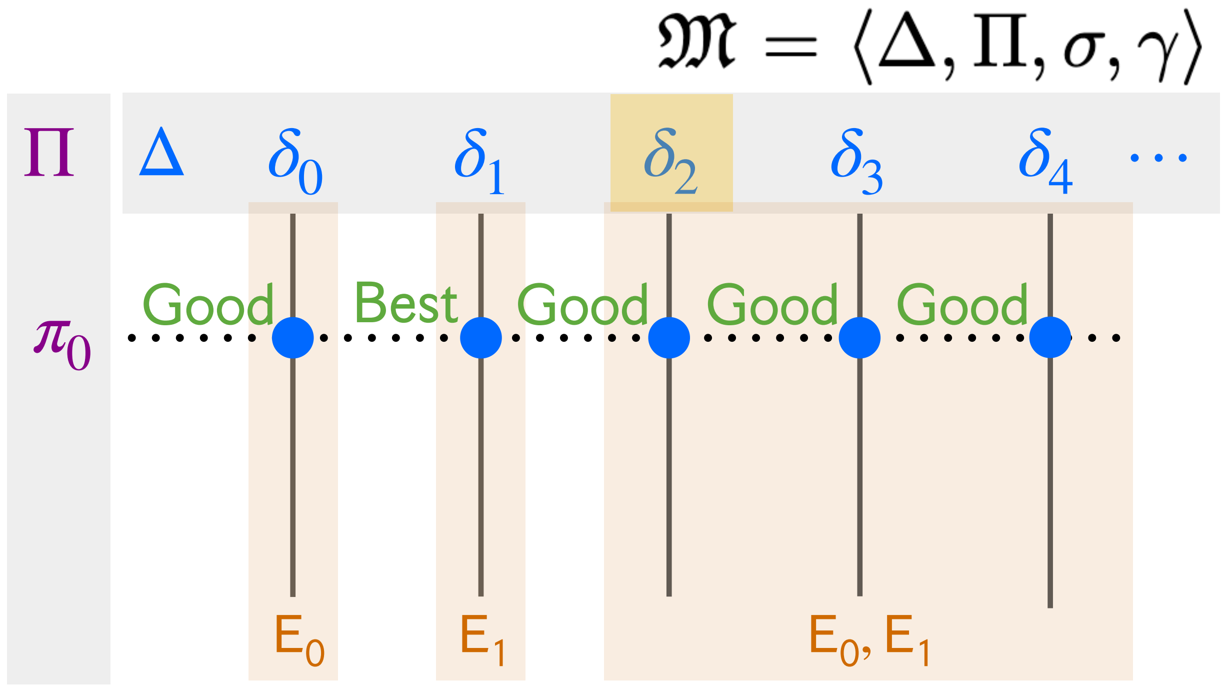
Permutational Representatives

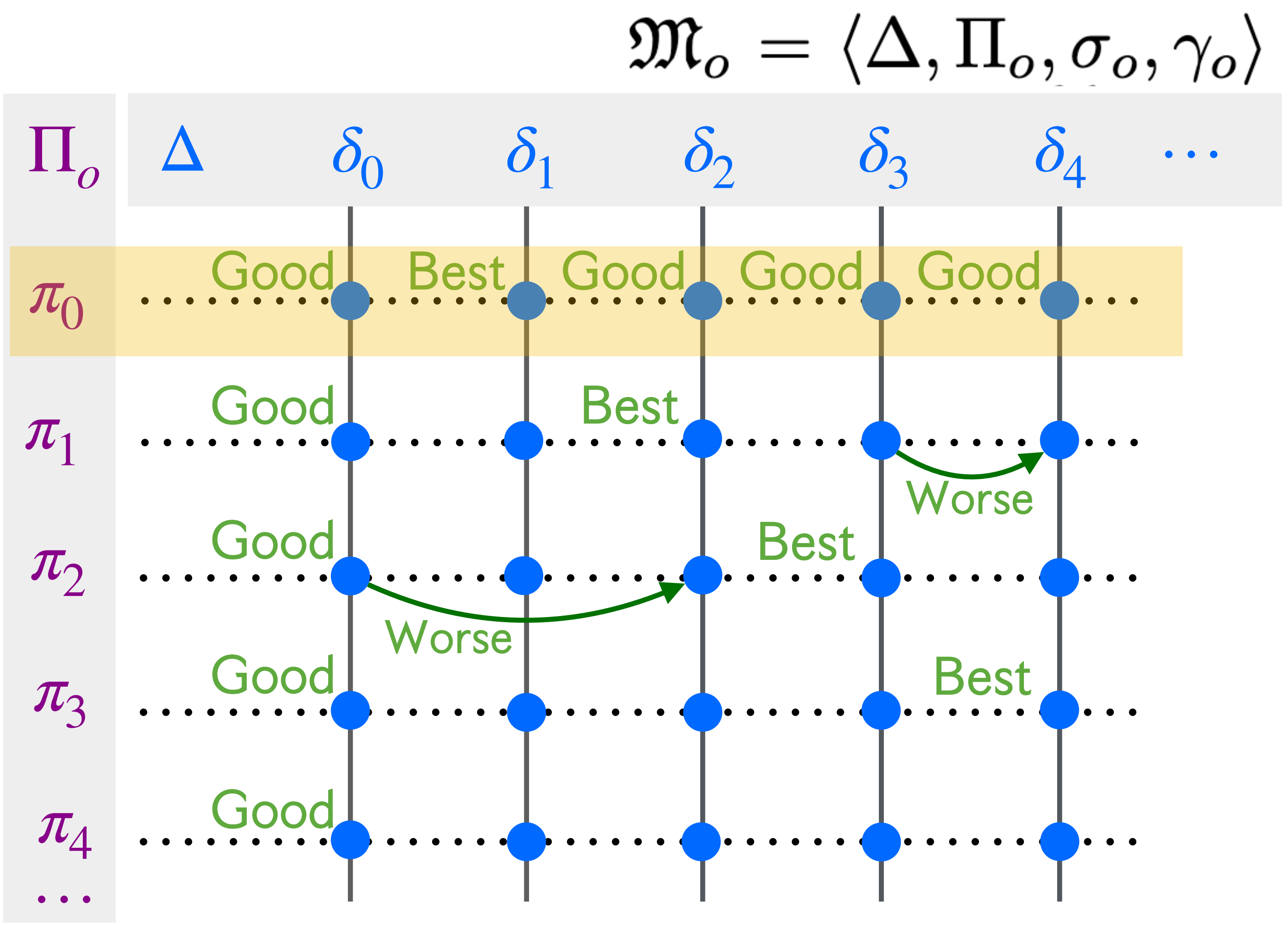
For any satisfiable ϕ, we construct an (exp.) structure that yields a model

$$\phi = \exists^{=1} x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$
$$E_0 = \Box_*(\texttt{Good}(x))$$
$$E_1 = E_0 \vee \Diamond_*(\texttt{Best}(x) \wedge \forall y.(\texttt{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\texttt{Good}(x) \vee \texttt{Best}(x))$$

<u>Building the exponential structure from a model $\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$:</u>

- Add one rigid unary predicate E_i per monodic sub-formula $\Diamond_* \psi_i$
  (Set the same extension in structure)
- Select from Π_o
  - a π if no $\Diamond$-subformulas; otherwise:
  - a π for each sentential $\Diamond_* \psi$ subformula
  - for each realised E-type T, $\{\mathsf{E}_0\}, \{\mathsf{E}_1\}, \{\mathsf{E}_0, \mathsf{E}_1\}$
    pick δ of type T
    a π satisfying ψ_i for δ per $\mathsf{E}_i \in T$





Permutational Representatives

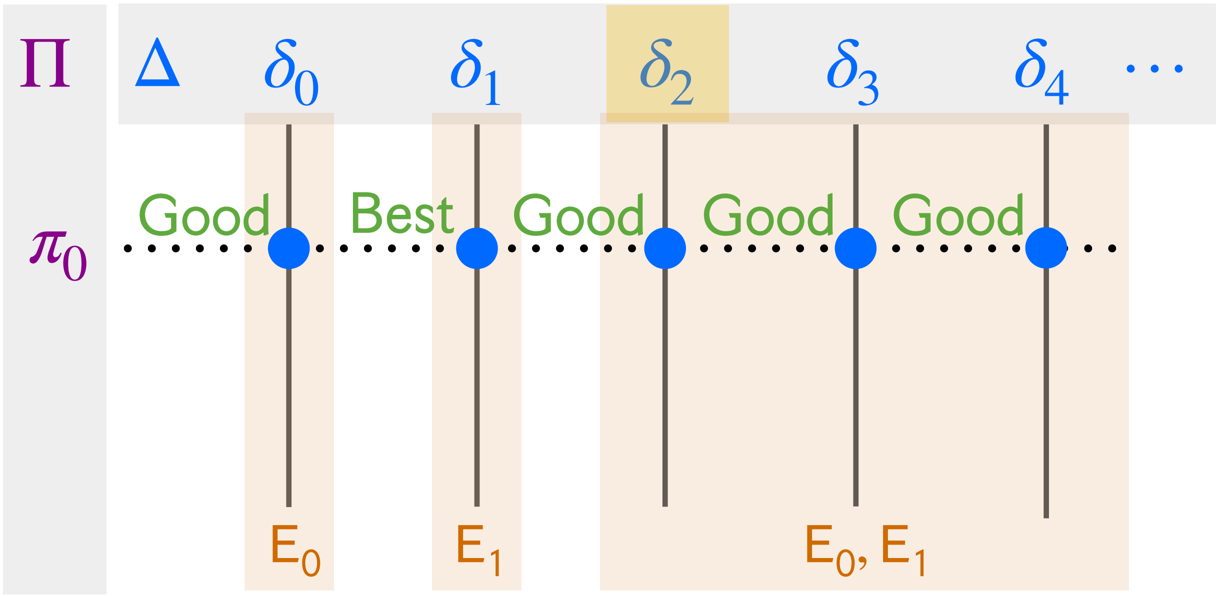
For any satisfiable ϕ, we construct an (exp.) structure that yields a model

$$\phi = \exists^{=1}x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$
$$E_0 = \Box_*(\texttt{Good}(x))$$
$$E_1 = E_0 \vee \Diamond_*(\texttt{Best}(x) \wedge \forall y.(\texttt{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\texttt{Good}(x) \vee \texttt{Best}(x))$$

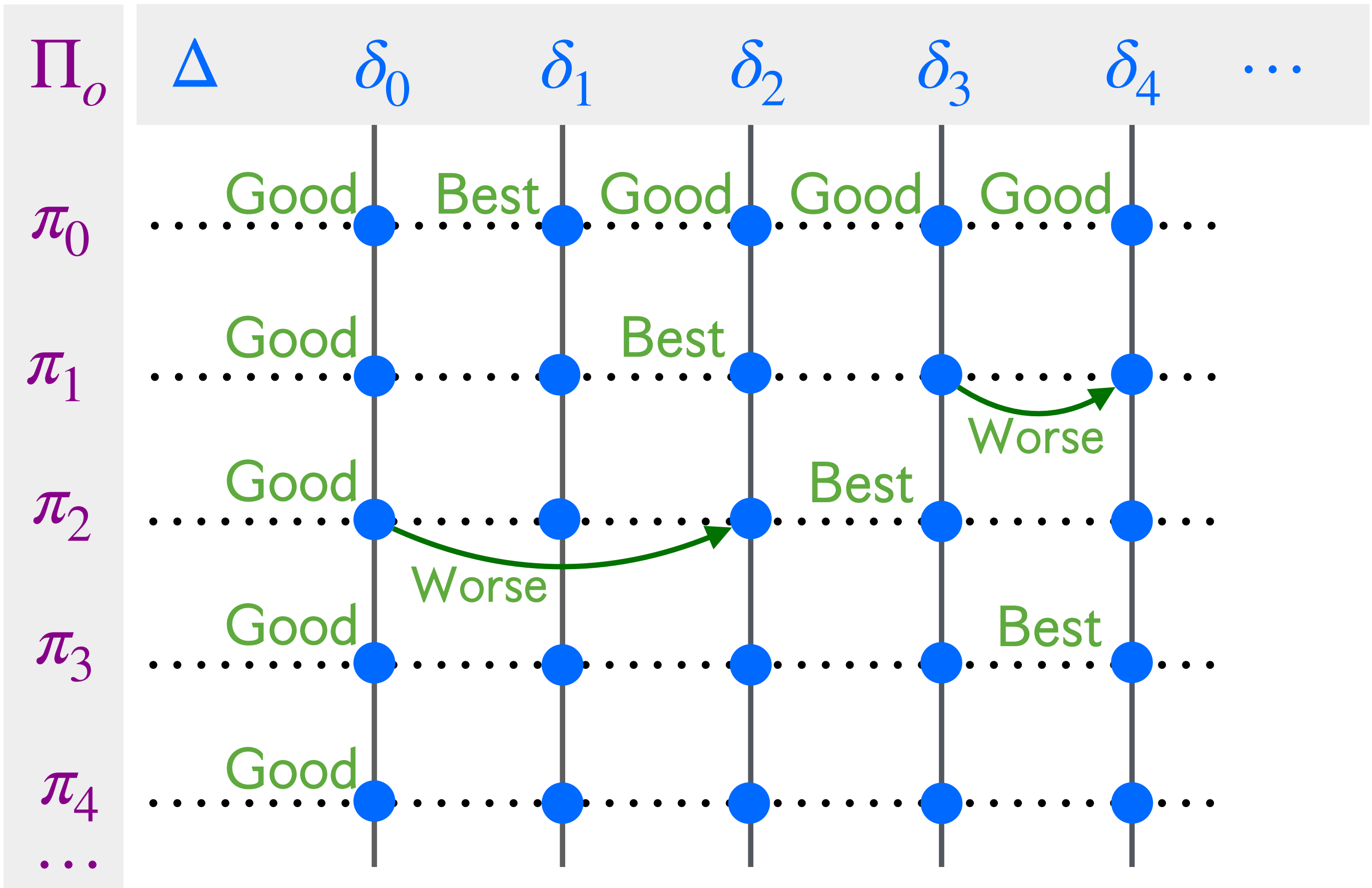
<u>Building the exponential structure from a model</u> $\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$:

- Add one rigid unary predicate E_i per monodic sub-formula $\Diamond_* \psi_i$
  (Set the same extension in structure)
- Select from Π_o
  - a π if no $\Diamond$-subformulas; otherwise:
  - a π for each sentential $\Diamond_* \psi$ subformula
  - for each realised E-type T, $\{\mathsf{E}_0\}, \{\mathsf{E}_1\}, \{\mathsf{E}_0, \mathsf{E}_1\}$
    pick δ of type T
    a π satisfying ψ_i for δ per $\mathsf{E}_i \in T$

$\mathfrak{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$



$\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$



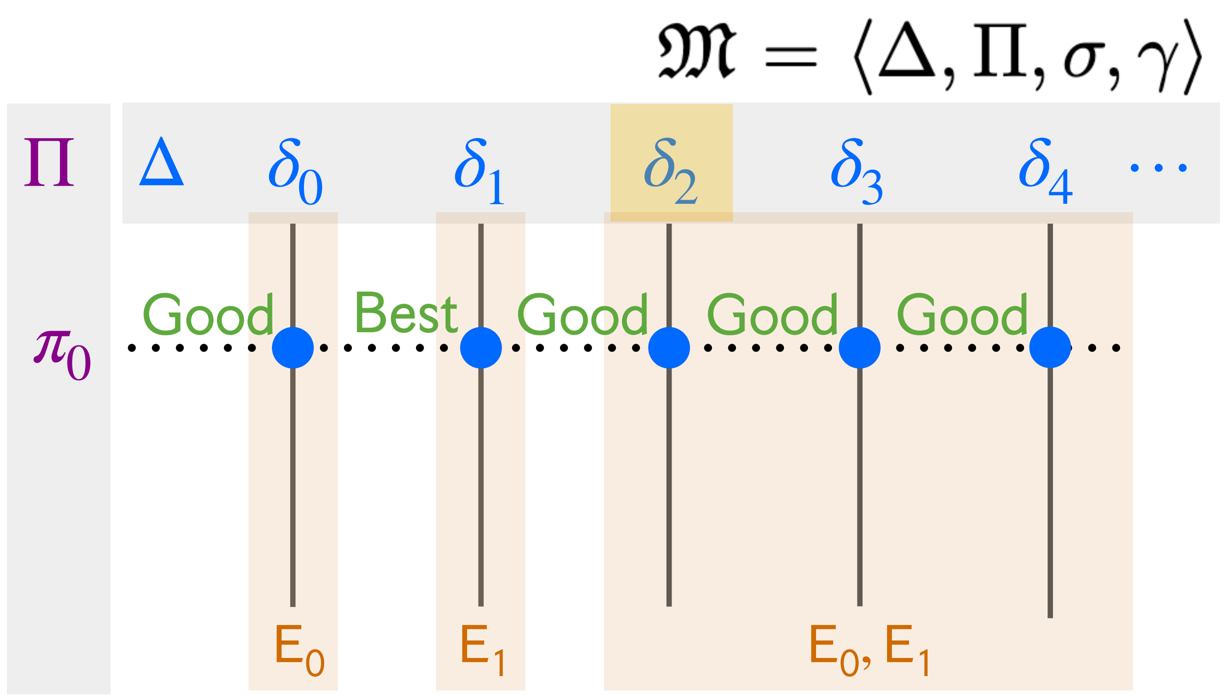
Permutational Representatives

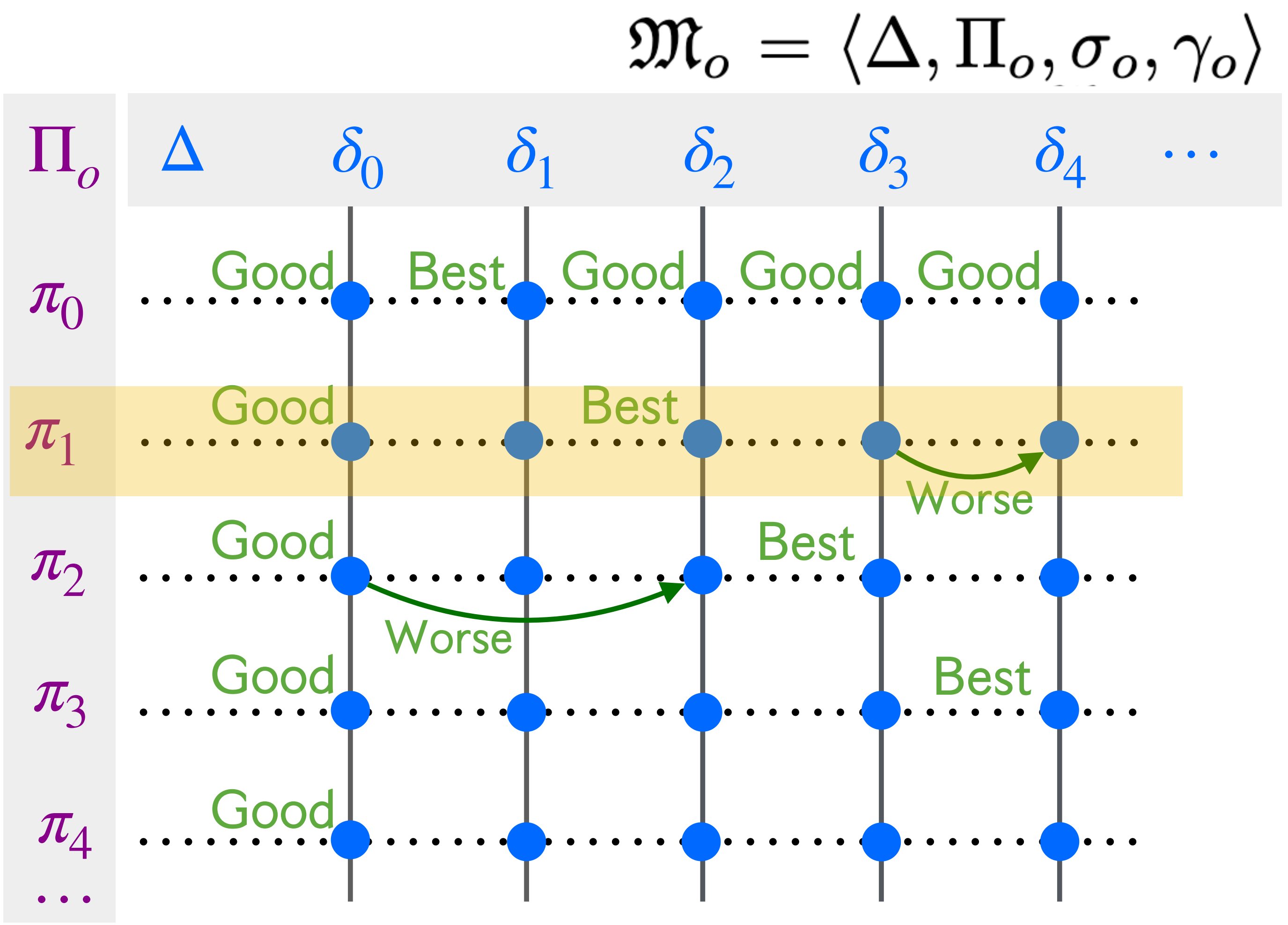
For any satisfiable ϕ, we construct an (exp.) structure that yields a model

$$\phi = \exists^{=1} x.(E_0) \land \forall x.(E_1) \land E_2 \text{ with}$$
$$E_0 = \Box_*(\texttt{Good}(x))$$
$$E_1 = E_0 \lor \Diamond_*(\texttt{Best}(x) \land \forall y.(\texttt{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\texttt{Good}(x) \lor \texttt{Best}(x))$$

<u>Building the exponential structure from a model $\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$:</u>

- Add one rigid unary predicate E_i per monodic sub-formula $\Diamond_* \psi_i$ (Set the same extension in structure)
- Select from Π_o
  - a π if no $\Diamond$-subformulas; otherwise:
  - a π for each sentential $\Diamond_* \psi$ subformula
  - for each realised E-type T, $\{E_0\}, \{E_1\}, \{E_0, E_1\}$
    pick δ of type T
    a π satisfying ψ_i for δ per $E_i \in T$





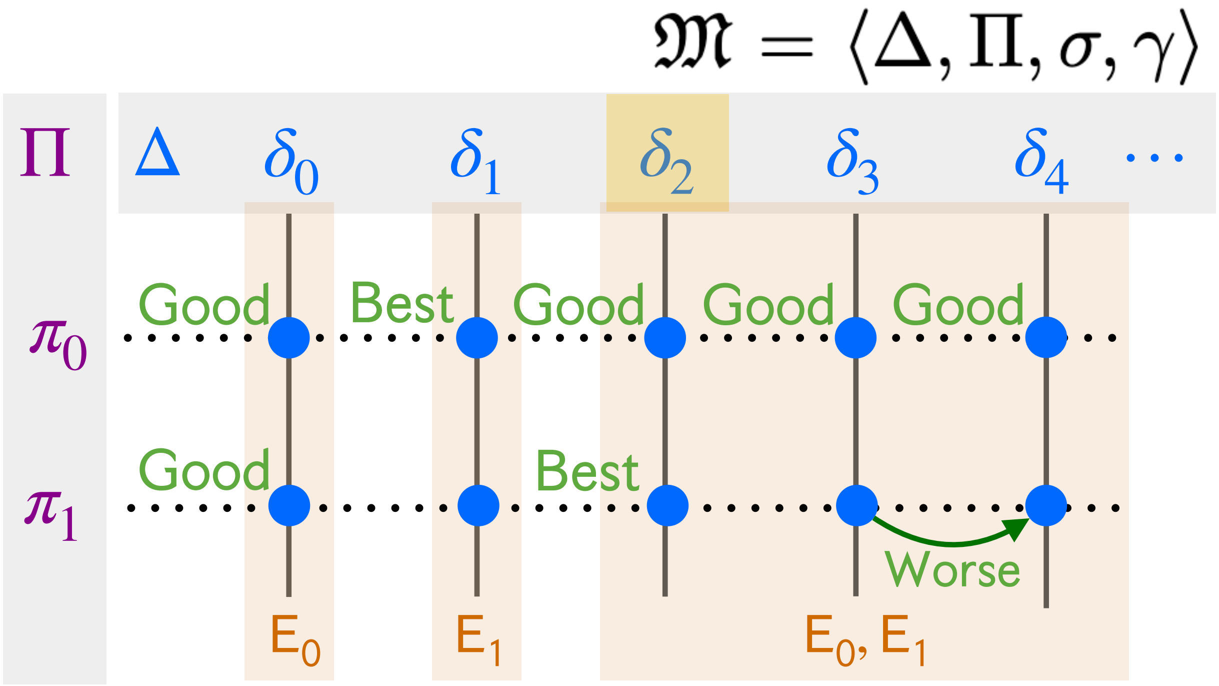
Permutational Representatives

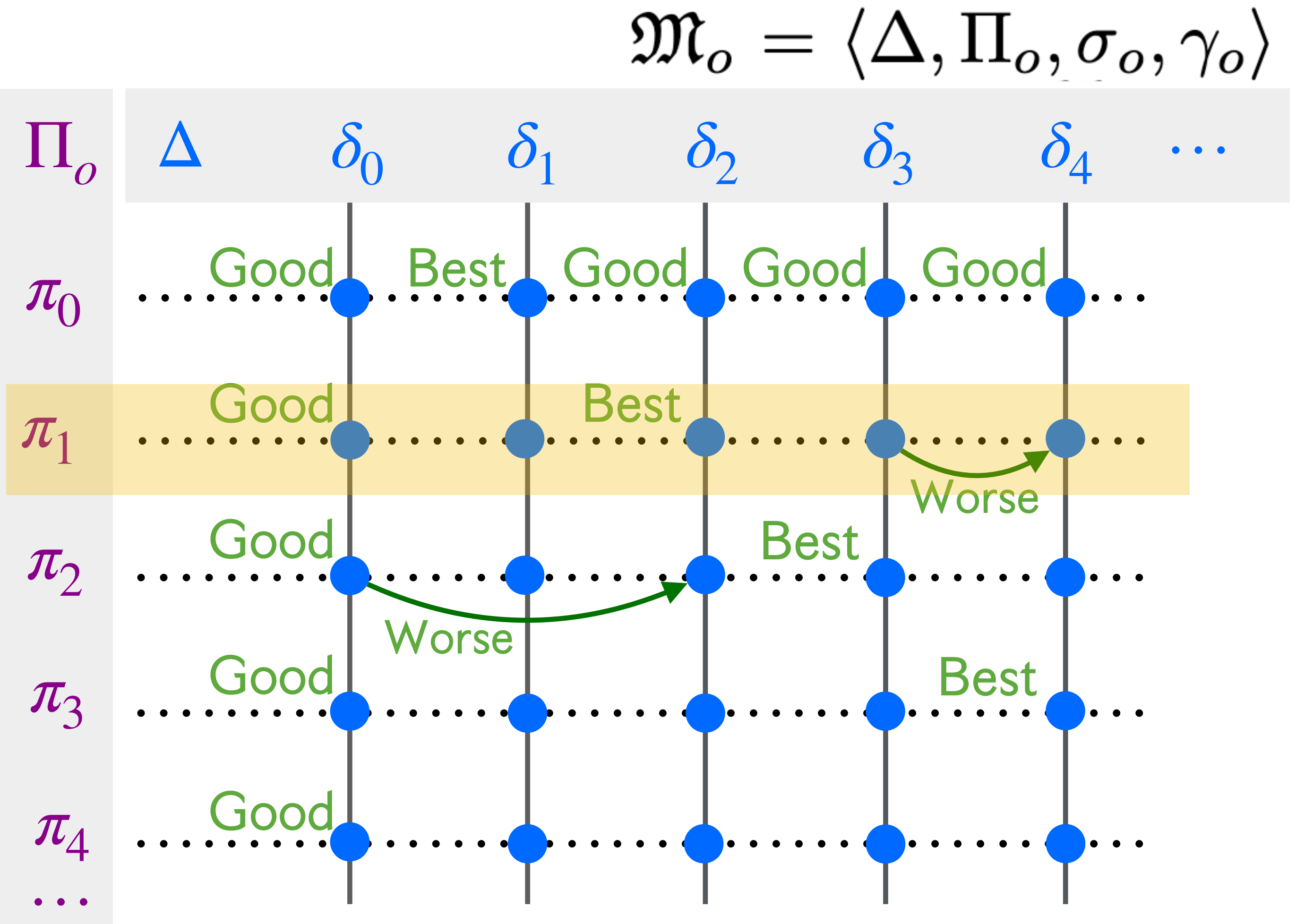
For any satisfiable ϕ, we construct an (exp.) structure that yields a model

$$\phi = \exists^{=1}x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$
$$E_0 = \Box_*(\texttt{Good}(x))$$
$$E_1 = E_0 \vee \Diamond_*(\texttt{Best}(x) \wedge \forall y.(\texttt{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\texttt{Good}(x) \vee \texttt{Best}(x))$$

<u>Building the exponential structure from a model $\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$:</u>

- Add one rigid unary predicate E_i per monodic sub-formula $\Diamond_* \psi_i$
  (Set the same extension in structure)
- Select from Π_o
  - a π if no $\Diamond$-subformulas; otherwise:
  - a π for each sentential $\Diamond_* \psi$ subformula
  - for each realised E-type T, $\{E_0\}, \{E_1\}, \{E_0, E_1\}$
    pick δ of type T
    a π satisfying ψ_i for δ per $E_i \in T$





Permutational Representatives

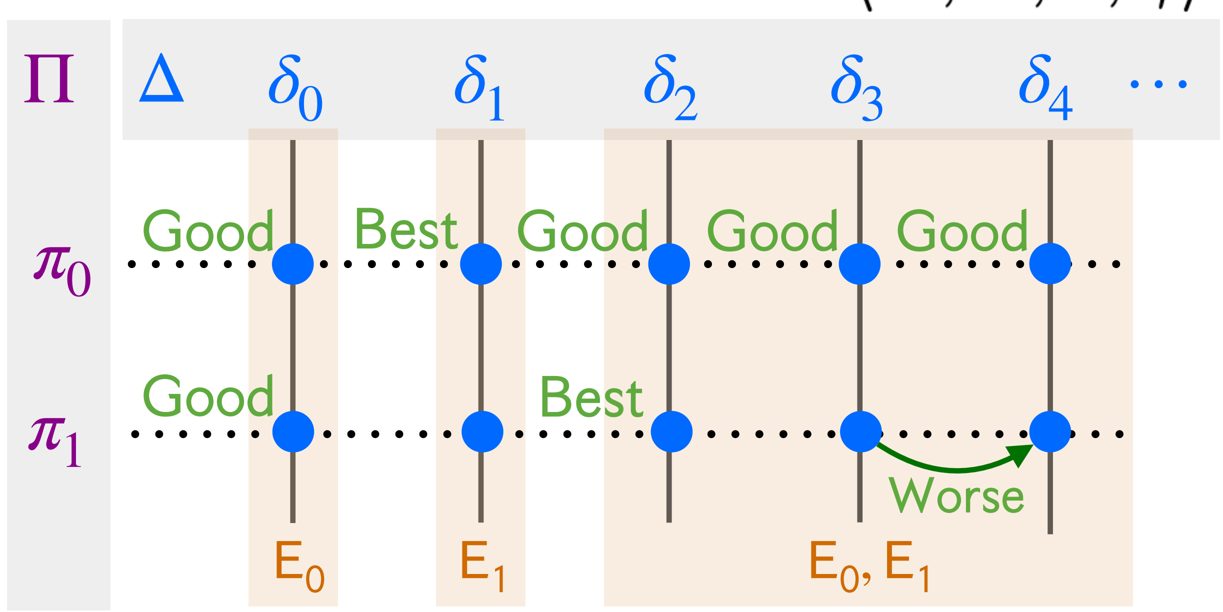
For any satisfiable ϕ, we construct an (exp.) structure that yields a model

$$\phi = \exists^{=1}x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$
$$E_0 = \Box_*(\mathtt{Good}(x))$$
$$E_1 = E_0 \vee \Diamond_*(\mathtt{Best}(x) \wedge \forall y.(\mathtt{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\mathtt{Good}(x) \vee \mathtt{Best}(x))$$

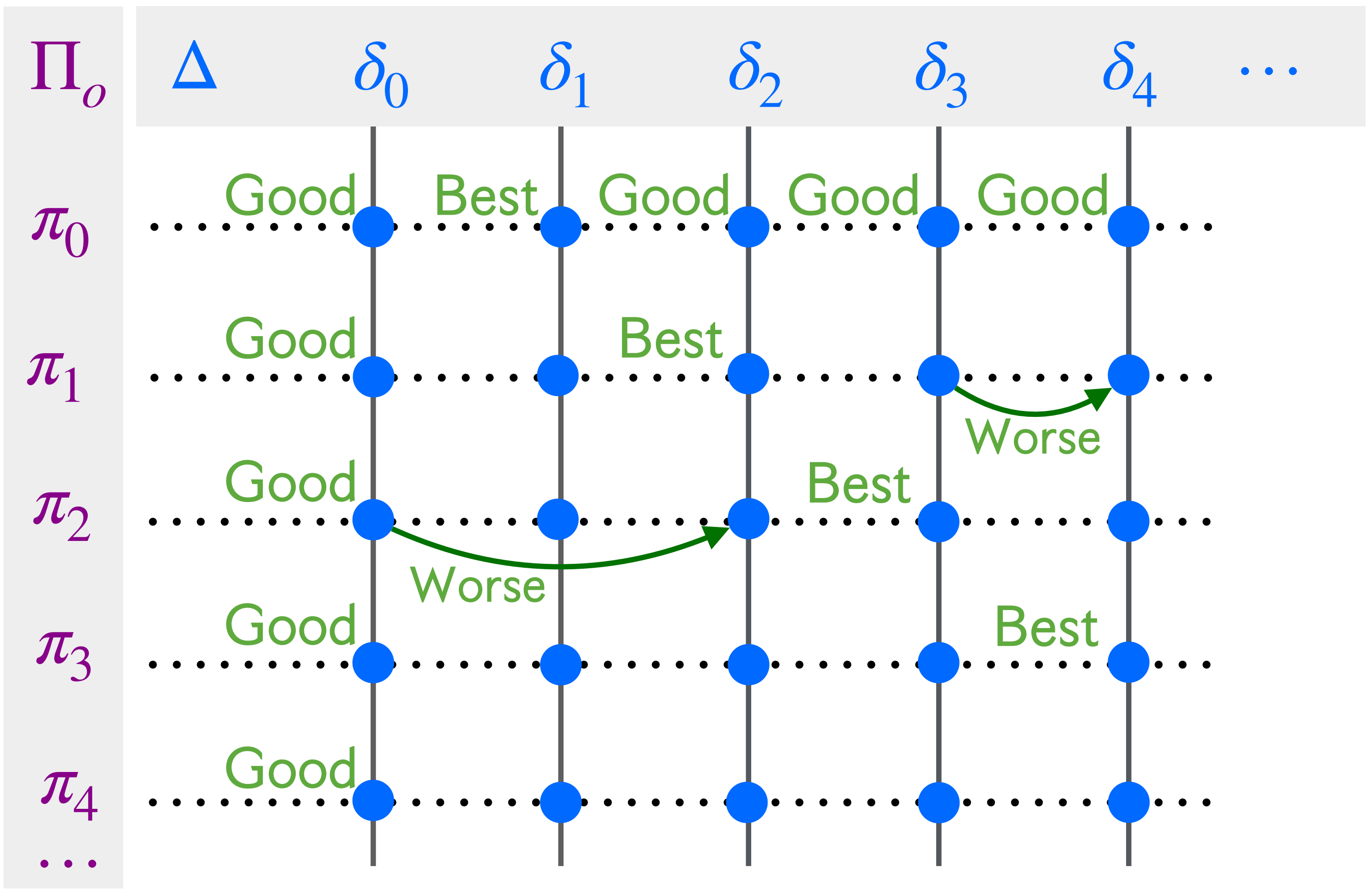
<u>Building the exponential structure from a model $\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$:</u>

- Add one rigid unary predicate E_i per monodic sub-formula $\Diamond_* \psi_i$
  (Set the same extension in structure)
- Select from Π_o
  - a π if no $\Diamond$-subformulas; otherwise:
  - a π for each sentential $\Diamond_* \psi$ subformula
  - for each realised E-type T, $\{\mathsf{E}_0\}, \{\mathsf{E}_1\}, \{\mathsf{E}_0, \mathsf{E}_1\}$
    pick δ of type T
    a π satisfying ψ_i for δ per $\mathsf{E}_i \in T$

$\mathfrak{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$

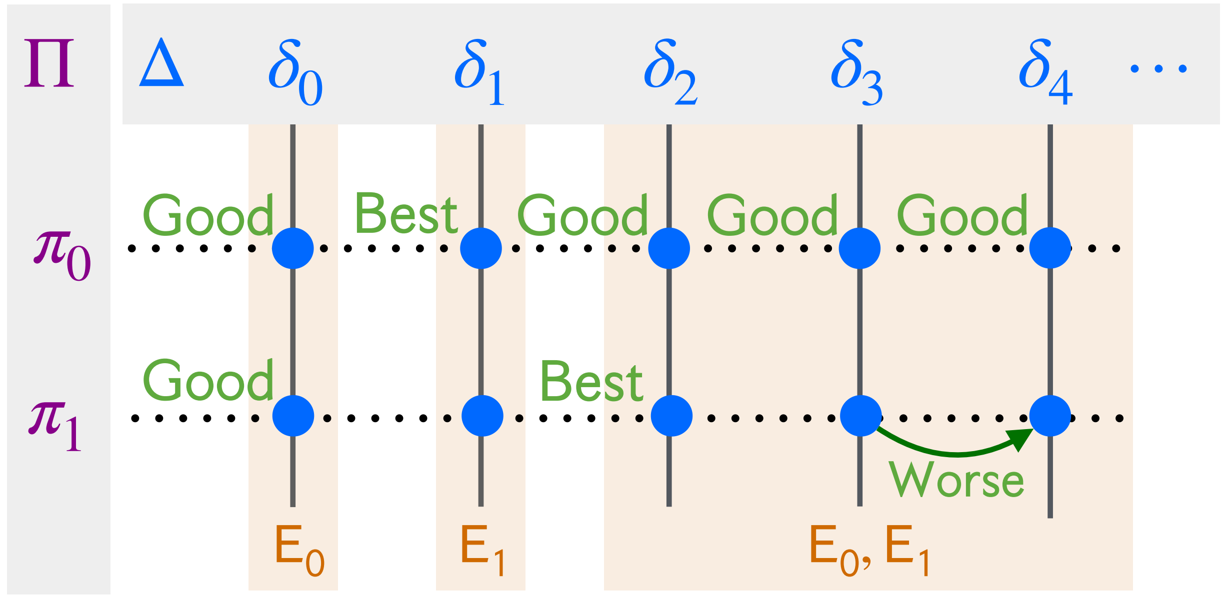


$\mathfrak{M}_o = \langle \Delta, \Pi_o, \sigma_o, \gamma_o \rangle$



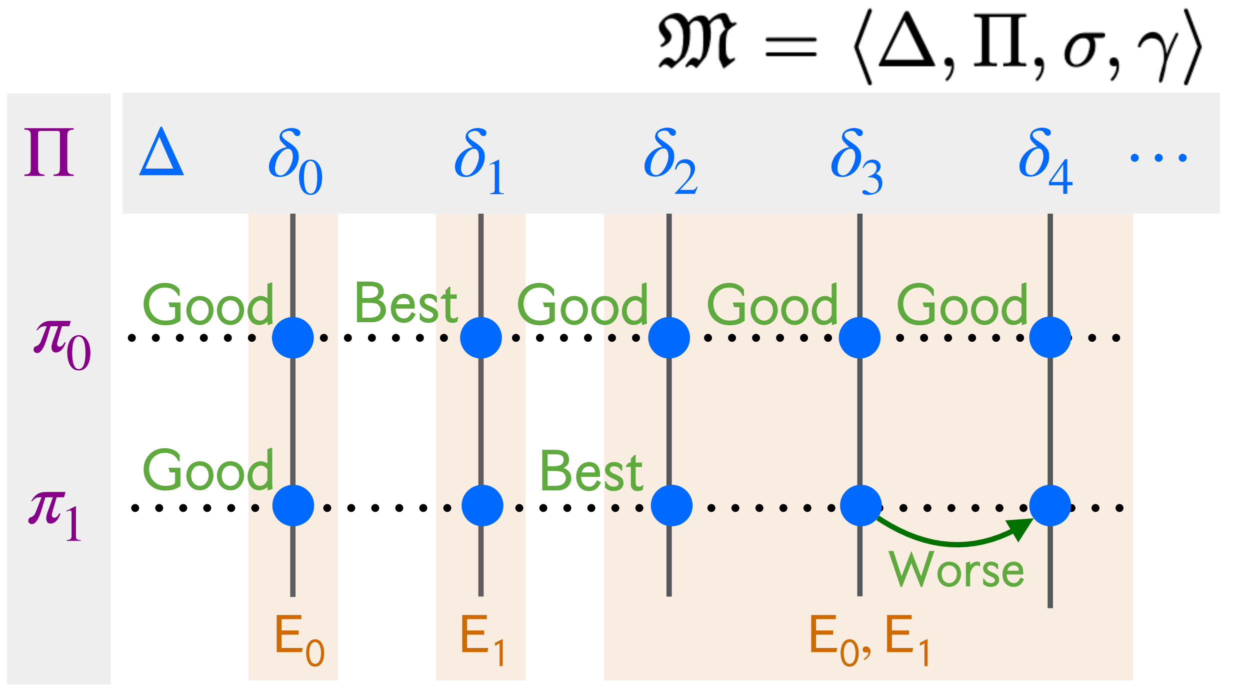
Permutational Representatives

$\mathfrak{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$



Permutational Representatives

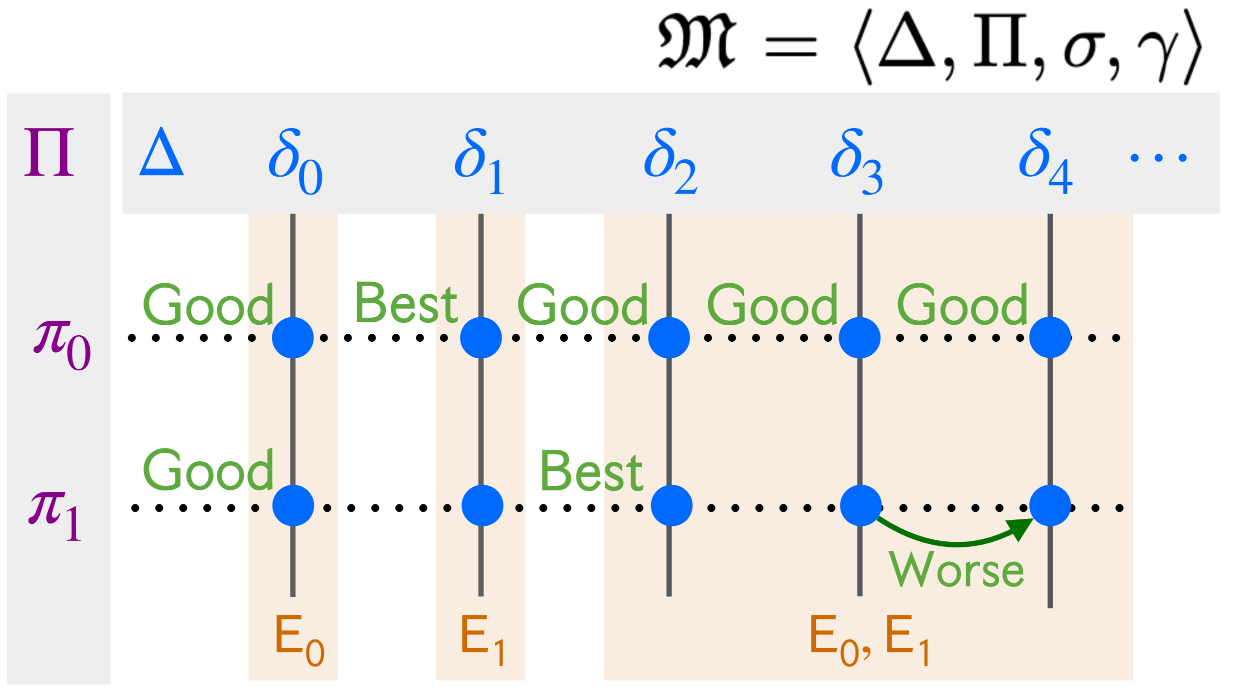
For any satisfiable ϕ, we construct an (exp.) structure that yields a model



Permutational Representatives

For any satisfiable ϕ, we construct an (exp.) structure that yields a model

The $\mathbf{P}_{\mathrm{E}}$-stable permutational closure of $\mathfrak{M}$



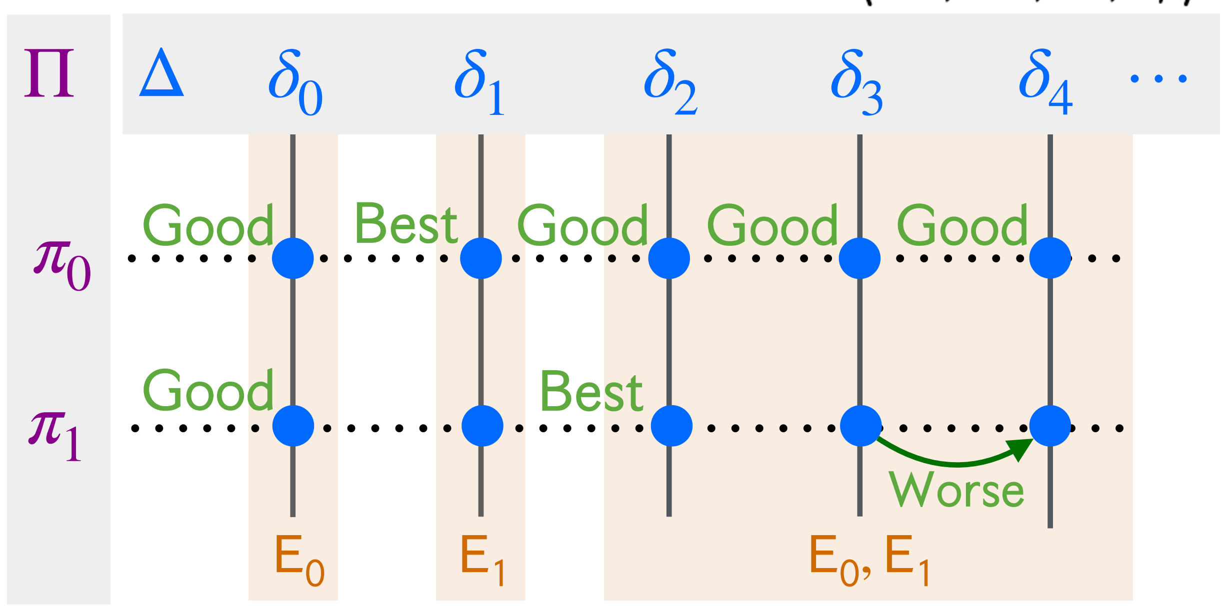
Permutational Representatives

For any satisfiable ϕ, we construct an (exp.) structure that yields a model

<u>The $\mathbf{P}_{\mathsf{E}}$-stable permutational closure of $\mathfrak{M}$</u>

- $\Pi' = \Pi \times \mathbb{P}_{\mathsf{E}}$
  - $\mathbb{P}_{\mathsf{E}}$: set of permutations $f : \Delta \to \Delta$ that preserve E_i membership

$\mathfrak{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$



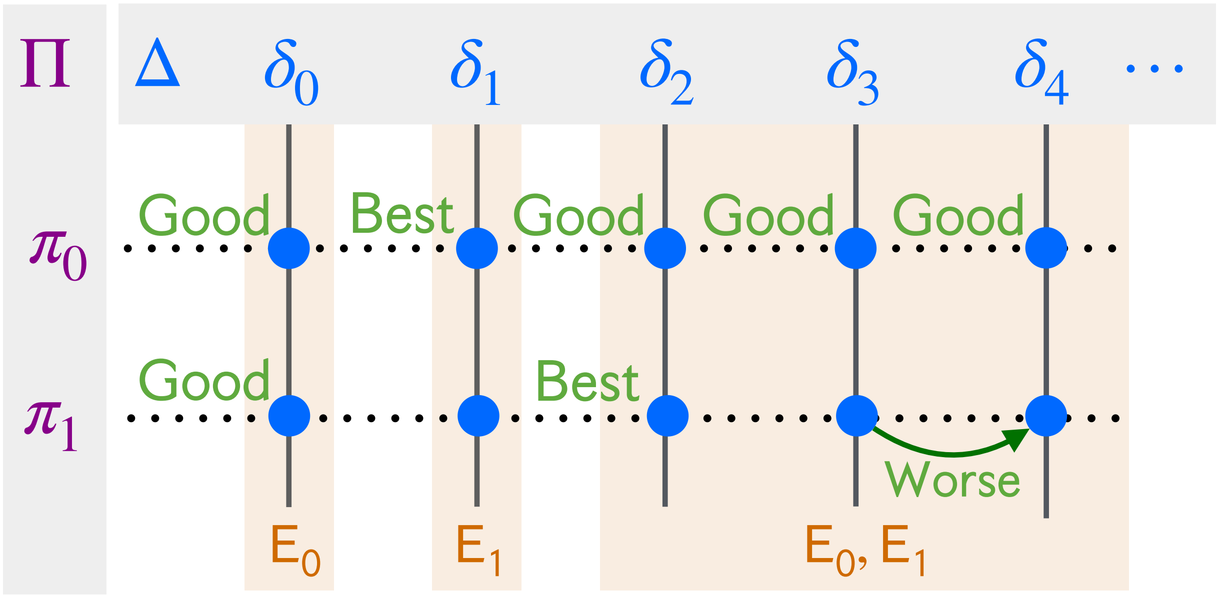
Permutational Representatives

For any satisfiable ϕ, we construct an (exp.) structure that yields a model

The $\mathbf{P}_{\text{E}}$-stable permutational closure of $\mathfrak{M}$

- $\Pi' = \Pi \times \mathbb{P}_{\text{E}}$
  - $\mathbb{P}_{\text{E}}$: set of permutations $f : \Delta \rightarrow \Delta$ that preserve E_i membership
- $\text{P}^{\gamma'(\pi,f)} = \{f(\delta) \mid \delta \in \text{P}^{\gamma(\pi)}\}$
- $\text{P}^{\gamma'(\pi,f)} = \{(f(\delta_1), f(\delta_2)) \mid (\delta_1, \delta_2) \in \text{P}^{\gamma(\pi)}\}$

$\mathfrak{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$

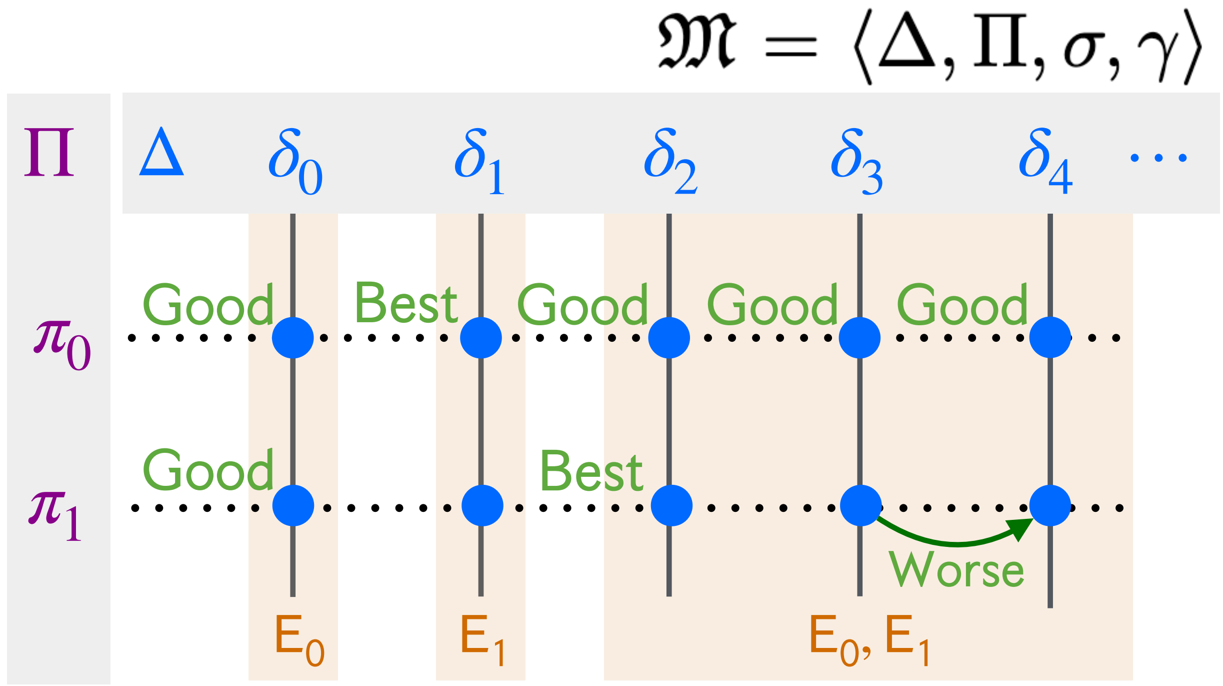


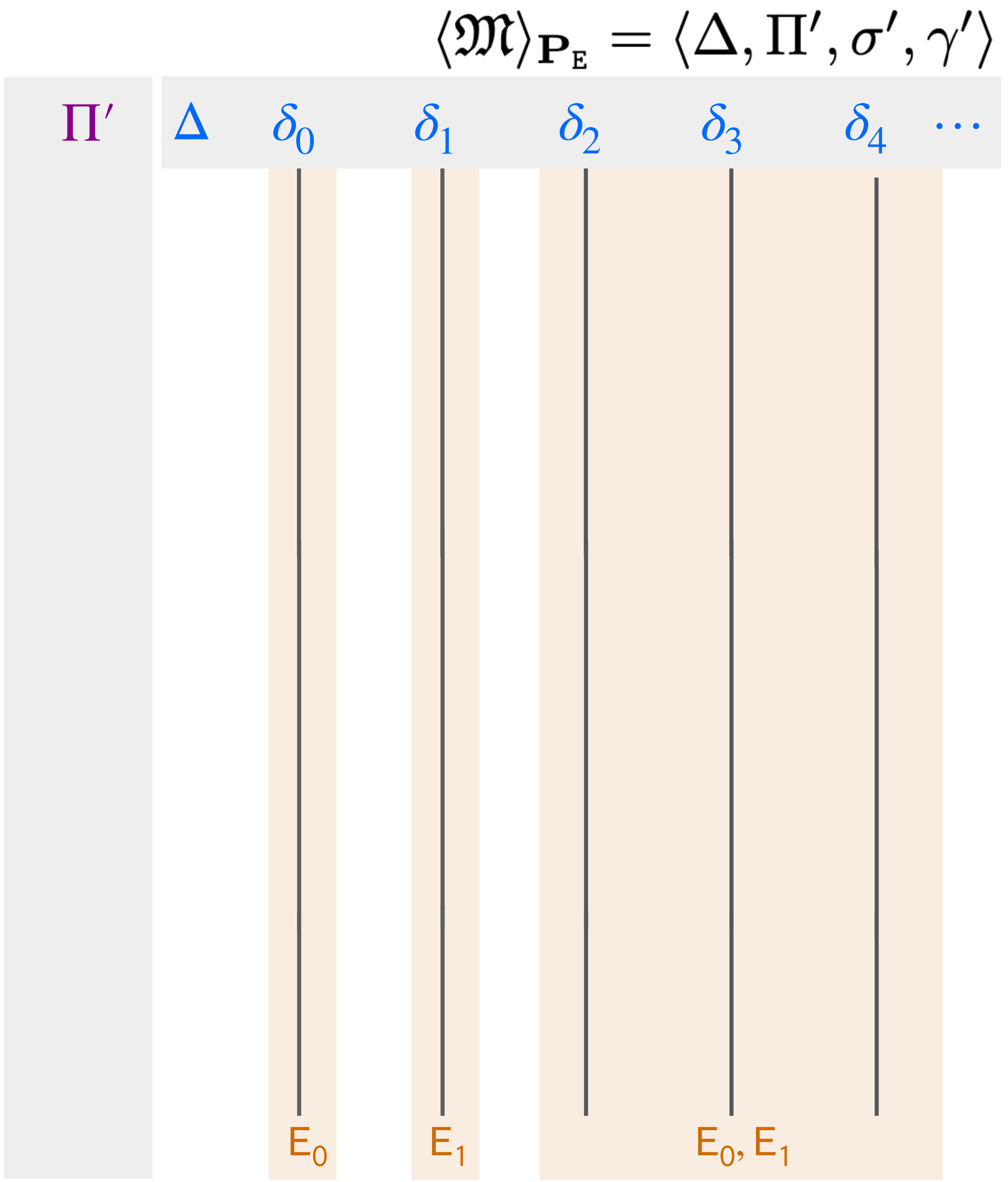
Permutational Representatives

For any satisfiable ϕ, we construct an (exp.) structure that yields a model

<u>The $\mathbf{P}_{\mathsf{E}}$-stable permutational closure of $\mathfrak{M}$</u>

- $\Pi' = \Pi \times \mathbb{P}_{\mathsf{E}}$
  - $\mathbb{P}_{\mathsf{E}}$: set of permutations $f : \Delta \to \Delta$ that preserve E_i membership
- $\mathsf{P}^{\gamma'(\pi,f)} = \{f(\delta) \mid \delta \in \mathsf{P}^{\gamma(\pi)}\}$
- $\mathsf{P}^{\gamma'(\pi,f)} = \{(f(\delta_1), f(\delta_2)) \mid (\delta_1, \delta_2) \in \mathsf{P}^{\gamma(\pi)}\}$





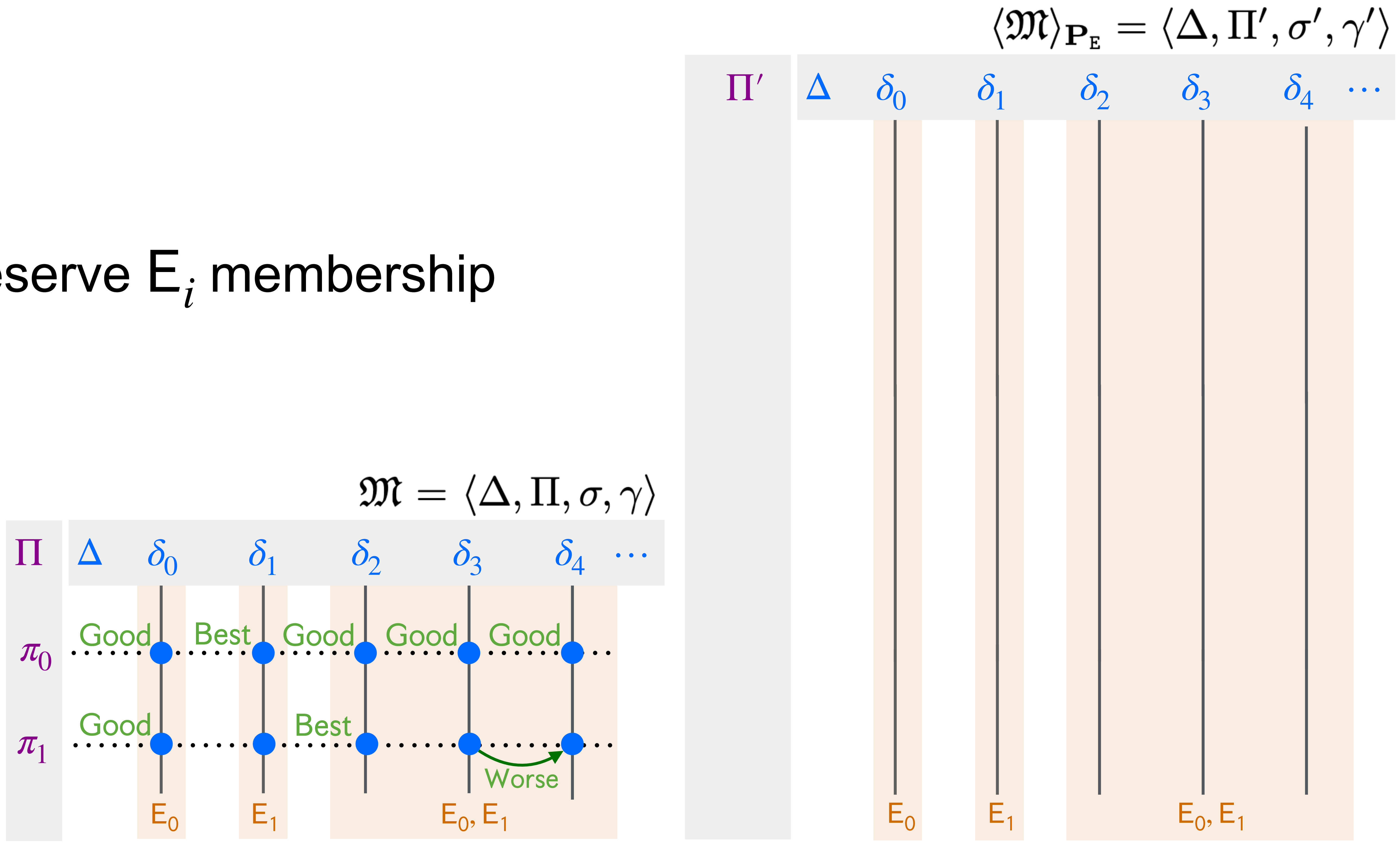
Permutational Representatives

For any satisfiable ϕ, we construct an (exp.) structure that yields a model

The $\mathbf{P}_\mathsf{E}$-stable permutational closure of $\mathfrak{M}$

- $\Pi' = \Pi \times \mathbb{P}_\mathsf{E}$
  - $\mathbb{P}_\mathsf{E}$: set of permutations $f : \Delta \to \Delta$ that preserve E_i membership
- $\mathrm{P}^{\gamma'(\pi,f)} = \{f(\delta) \mid \delta \in \mathrm{P}^{\gamma(\pi)}\}$
- $\mathrm{P}^{\gamma'(\pi,f)} = \{(f(\delta_1), f(\delta_2)) \mid (\delta_1, \delta_2) \in \mathrm{P}^{\gamma(\pi)}\}$

Elements swapped around,
preserving world internal structure



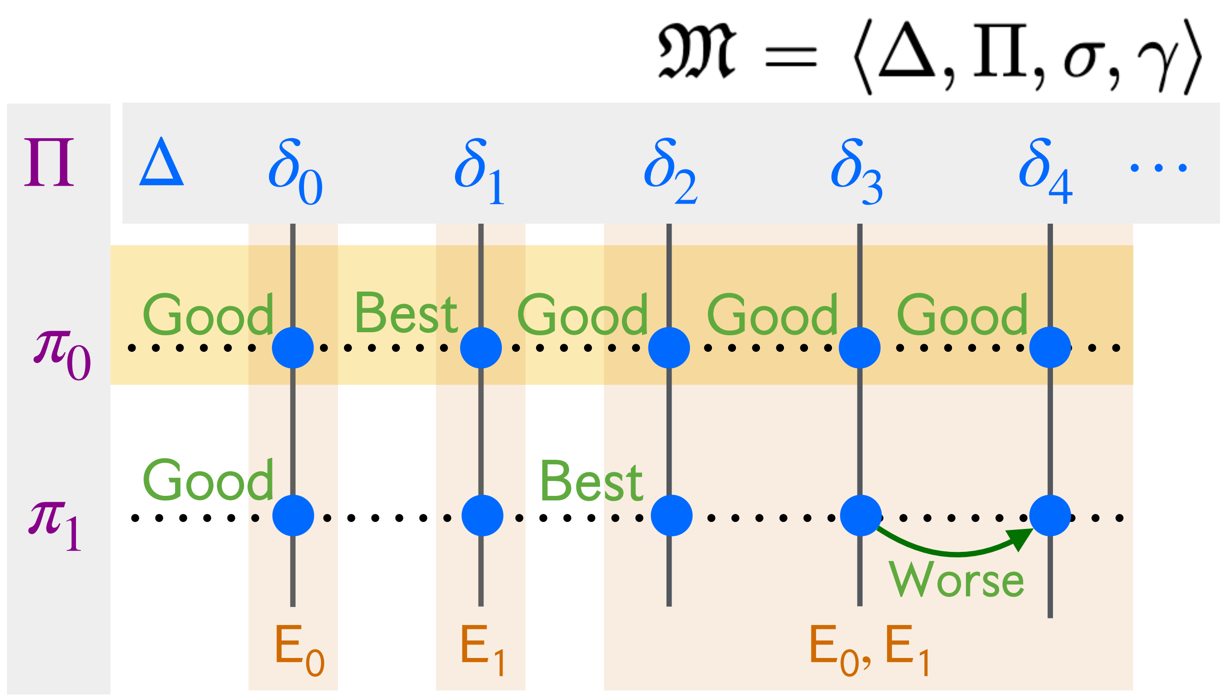
Permutational Representatives

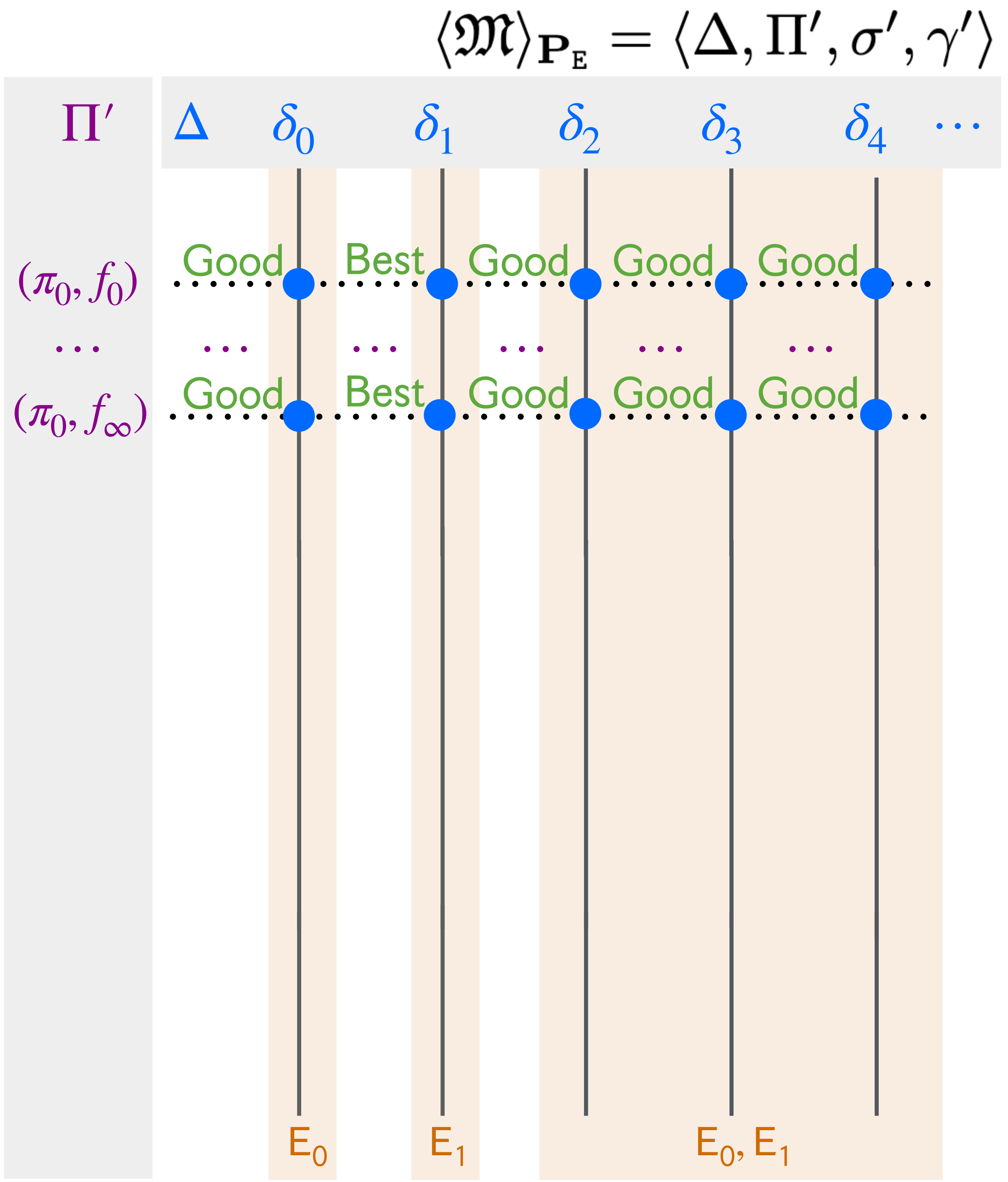
For any satisfiable ϕ, we construct an (exp.) structure that yields a model

<u>The $\mathbf{P}_E$-stable permutational closure of $\mathfrak{M}$</u>

- $\Pi' = \Pi \times \mathbb{P}_E$
  - $\mathbb{P}_E$: set of permutations $f : \Delta \to \Delta$ that preserve E_i membership
- $P^{\gamma'(\pi,f)} = \{f(\delta) \mid \delta \in P^{\gamma(\pi)}\}$
- $P^{\gamma'(\pi,f)} = \{(f(\delta_1), f(\delta_2)) \mid (\delta_1, \delta_2) \in P^{\gamma(\pi)}\}$

Elements swapped around,
preserving world internal structure





Permutational Representatives

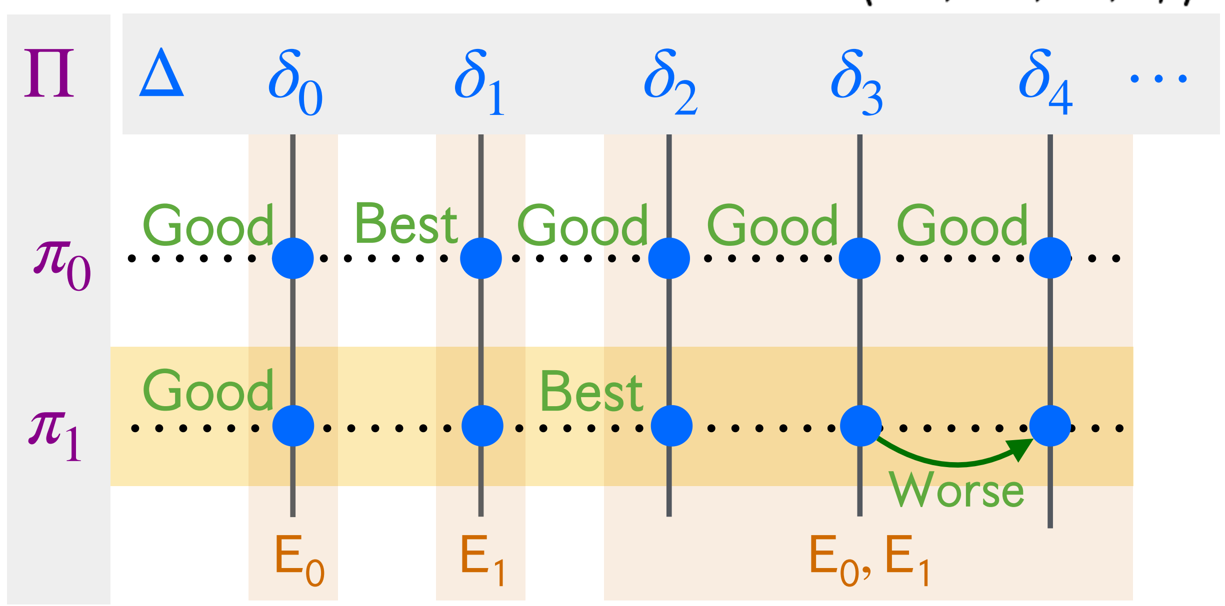
For any satisfiable ϕ, we construct an (exp.) structure that yields a model

<u>The $\mathbf{P}_E$-stable permutational closure of $\mathfrak{M}$</u>

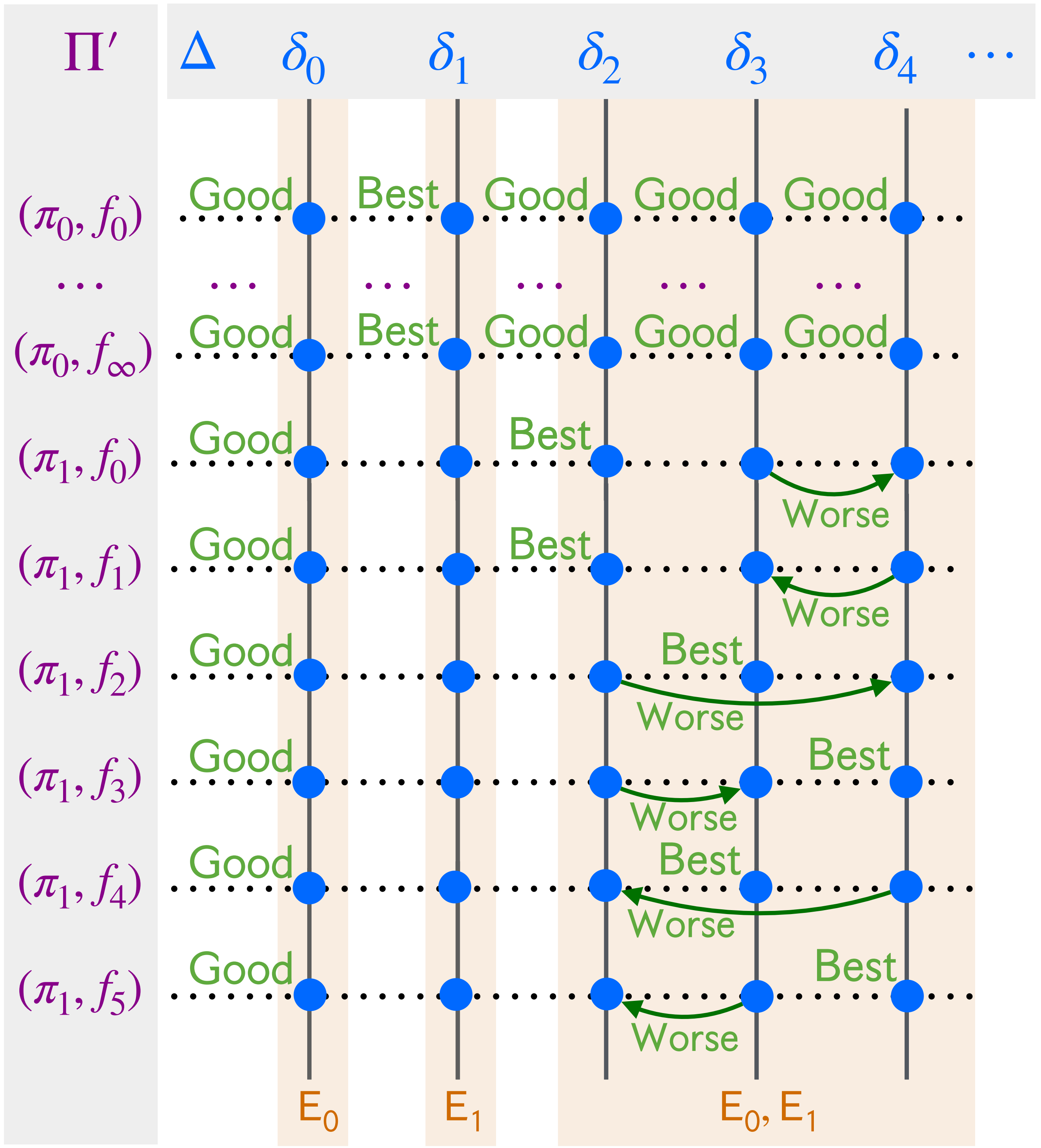
- $\Pi' = \Pi \times \mathbb{P}_E$
  - $\mathbb{P}_E$: set of permutations $f : \Delta \to \Delta$ that preserve E_i membership
- $P^{\gamma'(\pi,f)} = \{f(\delta) \mid \delta \in P^{\gamma(\pi)}\}$
- $P^{\gamma'(\pi,f)} = \{(f(\delta_1), f(\delta_2)) \mid (\delta_1, \delta_2) \in P^{\gamma(\pi)}\}$

Elements swapped around,
preserving world internal structure

$\mathfrak{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$



$\langle \mathfrak{M} \rangle_{\mathbf{P}_E} = \langle \Delta, \Pi', \sigma', \gamma' \rangle$



Permutational Representatives

For any satisfiable ϕ, we construct an (exp.) structure that yields a model

The $\mathbf{P}_\mathsf{E}$-stable permutational closure of $\mathfrak{M}$

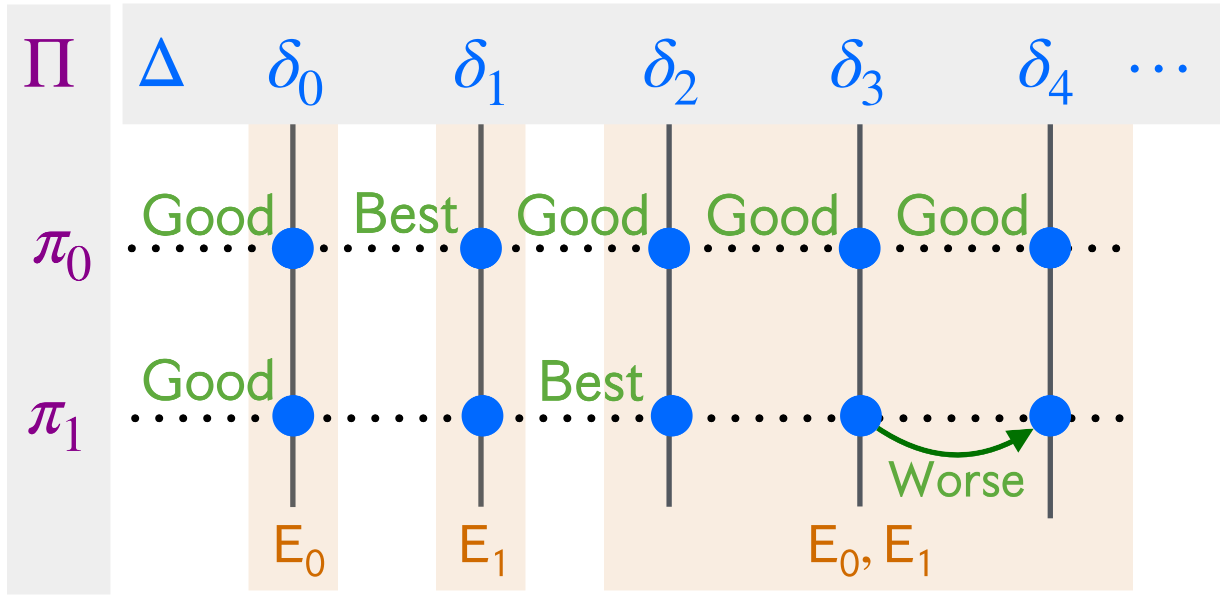
- $\Pi' = \Pi \times \mathbb{P}_\mathsf{E}$
  - $\mathbb{P}_\mathsf{E}$: set of permutations $f : \Delta \to \Delta$ that preserve E_i membership
- $\mathsf{P}^{\gamma'(\pi,f)} = \{f(\delta) \mid \delta \in \mathsf{P}^{\gamma(\pi)}\}$
- $\mathsf{P}^{\gamma'(\pi,f)} = \{(f(\delta_1), f(\delta_2)) \mid (\delta_1, \delta_2) \in \mathsf{P}^{\gamma(\pi)}\}$

Elements swapped around,

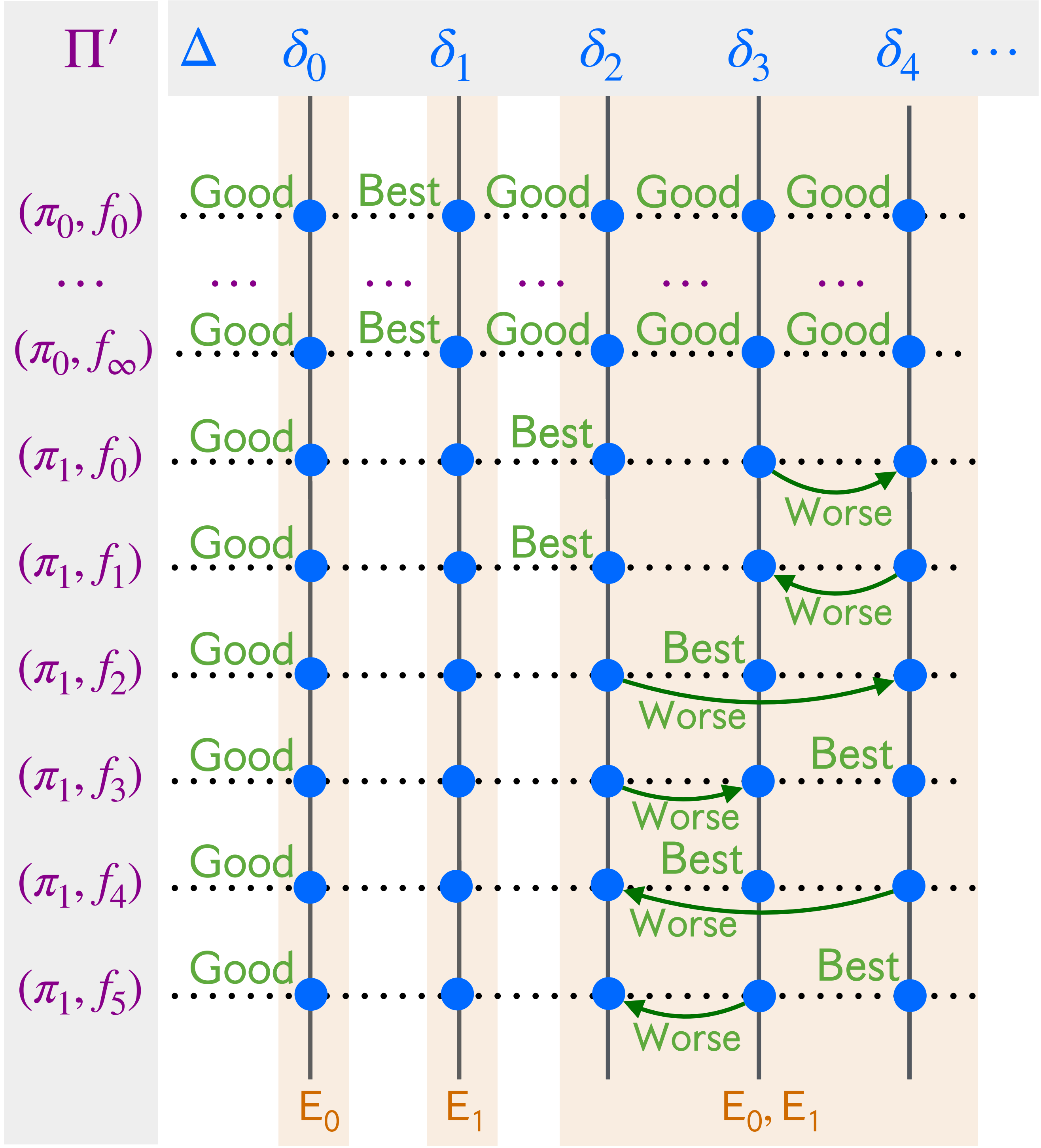
preserving world internal structure

$$\phi = \exists^{=1}x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$
$$E_0 = \Box_*(\mathtt{Good}(x))$$
$$E_1 = E_0 \vee \Diamond_*(\mathtt{Best}(x) \wedge \forall y.(\mathtt{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\mathtt{Good}(x) \vee \mathtt{Best}(x))$$

$\mathfrak{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$



$\langle \mathfrak{M} \rangle_{\mathbf{P}_\mathsf{E}} = \langle \Delta, \Pi', \sigma', \gamma' \rangle$



Permutational Representatives

For any satisfiable ϕ, we construct an (exp.) structure that yields a model

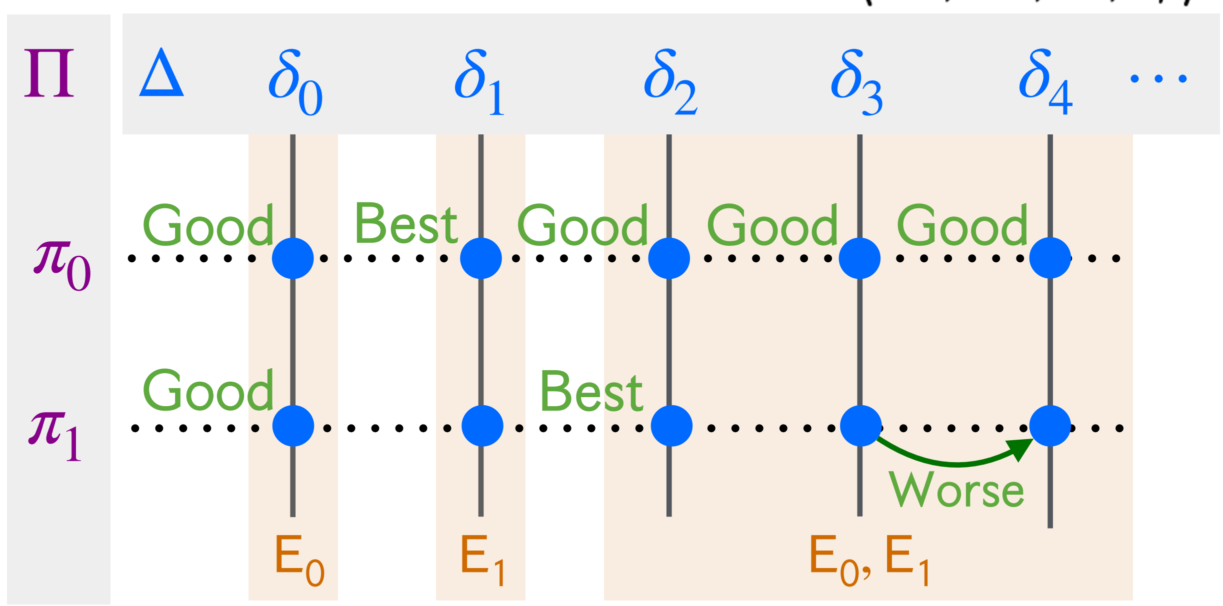
<u>The $\mathbf{P}_\mathsf{E}$-stable permutational closure of $\mathfrak{M}$</u>

- $\Pi' = \Pi \times \mathbb{P}_\mathsf{E}$
  - $\mathbb{P}_\mathsf{E}$: set of permutations $f : \Delta \to \Delta$ that preserve E_i membership
- $\mathsf{P}^{\gamma'(\pi,f)} = \{f(\delta) \mid \delta \in \mathsf{P}^{\gamma(\pi)}\}$
- $\mathsf{P}^{\gamma'(\pi,f)} = \{(f(\delta_1), f(\delta_2)) \mid (\delta_1, \delta_2) \in \mathsf{P}^{\gamma(\pi)}\}$

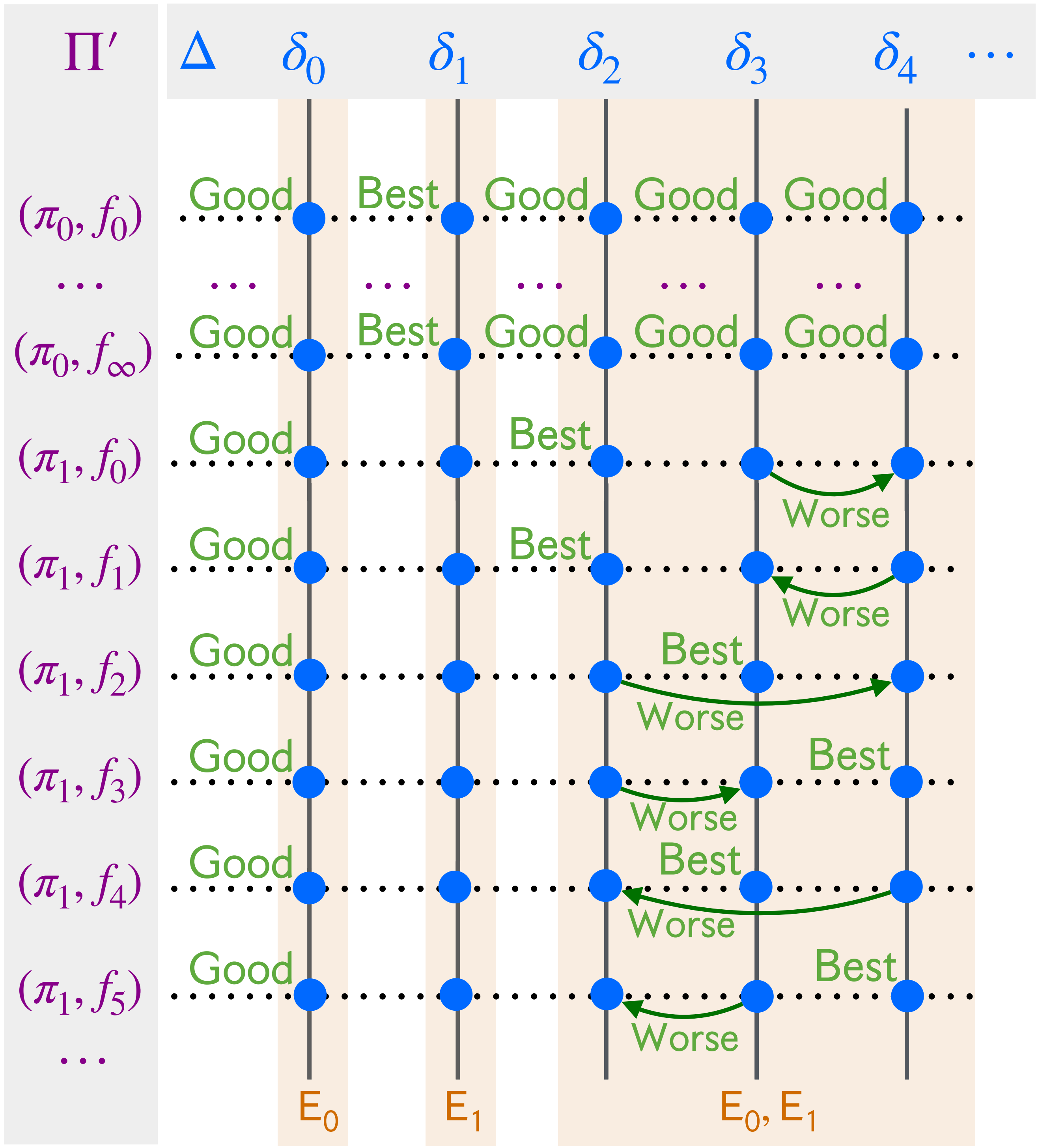
Elements swapped around,
preserving world internal structure

$$\phi = \exists^{=1} x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$
$$E_0 = \Box_*(\mathtt{Good}(x))$$
$$E_1 = E_0 \vee \Diamond_*(\mathtt{Best}(x) \wedge \forall y.(\mathtt{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\mathtt{Good}(x) \vee \mathtt{Best}(x))$$

$\mathfrak{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$



$\langle \mathfrak{M} \rangle_{\mathbf{P}_\mathsf{E}} = \langle \Delta, \Pi', \sigma', \gamma' \rangle$



Stacked Interpretations

$\mathfrak{M}$

Stacked Interpretations

The stacked interpretation of $\mathfrak{M}$ (with $|\Pi| = 2^m$),

Stacked Interpretations

The stacked interpretation of $\mathfrak{M}$ (with $|\Pi| = 2^m$),

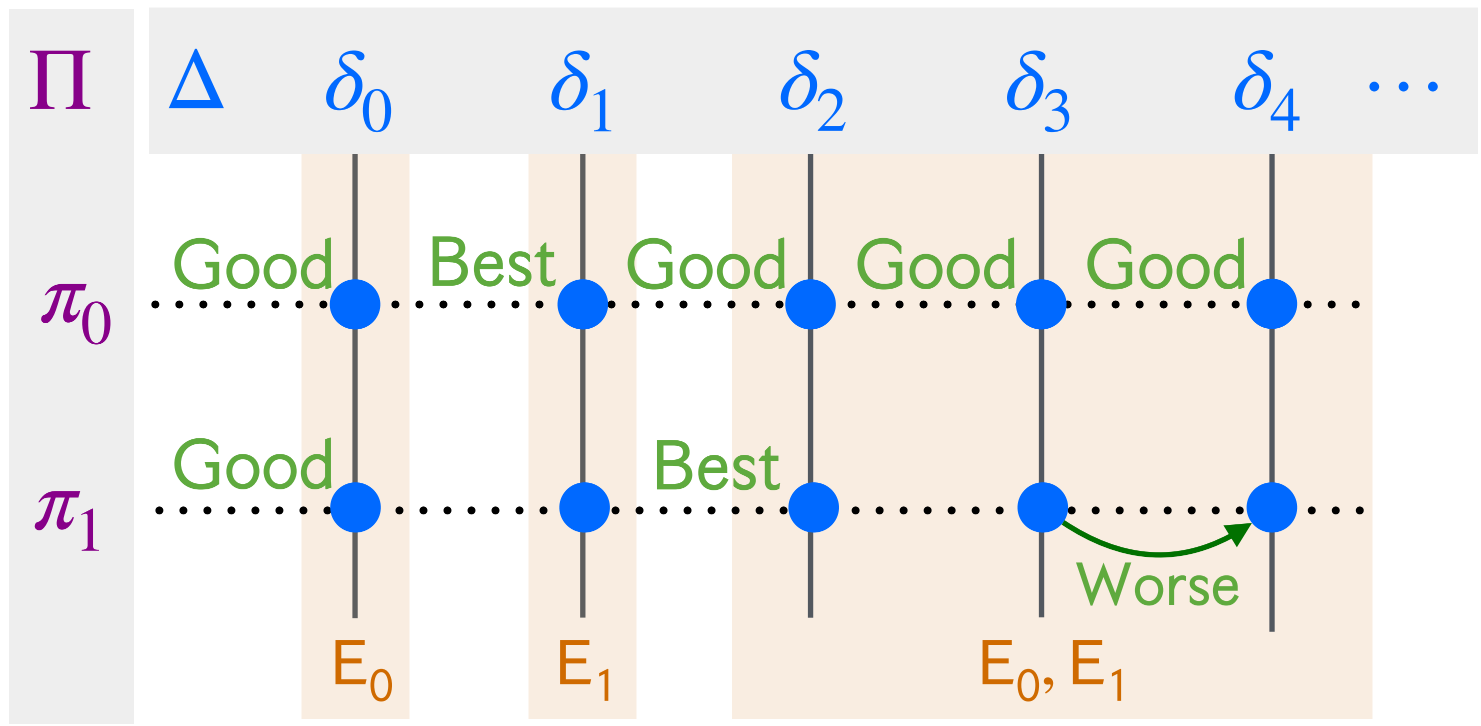
$\mathcal{I}^{\mathfrak{M}} = (\Delta', \cdot^{\mathcal{I}})$ with signature $\langle \mathbf{P} \uplus \{\mathsf{F}, \mathsf{L}_0, \ldots, \mathsf{L}_{m-1}\}, \varnothing \rangle$

Stacked Interpretations

The <u>stacked interpretation</u> of $\mathfrak{M}$ (with $|\Pi| = 2^m$),

$\mathcal{I}^{\mathfrak{M}} = (\Delta', \cdot^{\mathcal{I}})$ with signature $\langle \mathbf{P} \uplus \{\mathsf{F}, \mathsf{L}_0, \ldots, \mathsf{L}_{m-1}\}, \varnothing \rangle$

$\mathfrak{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$



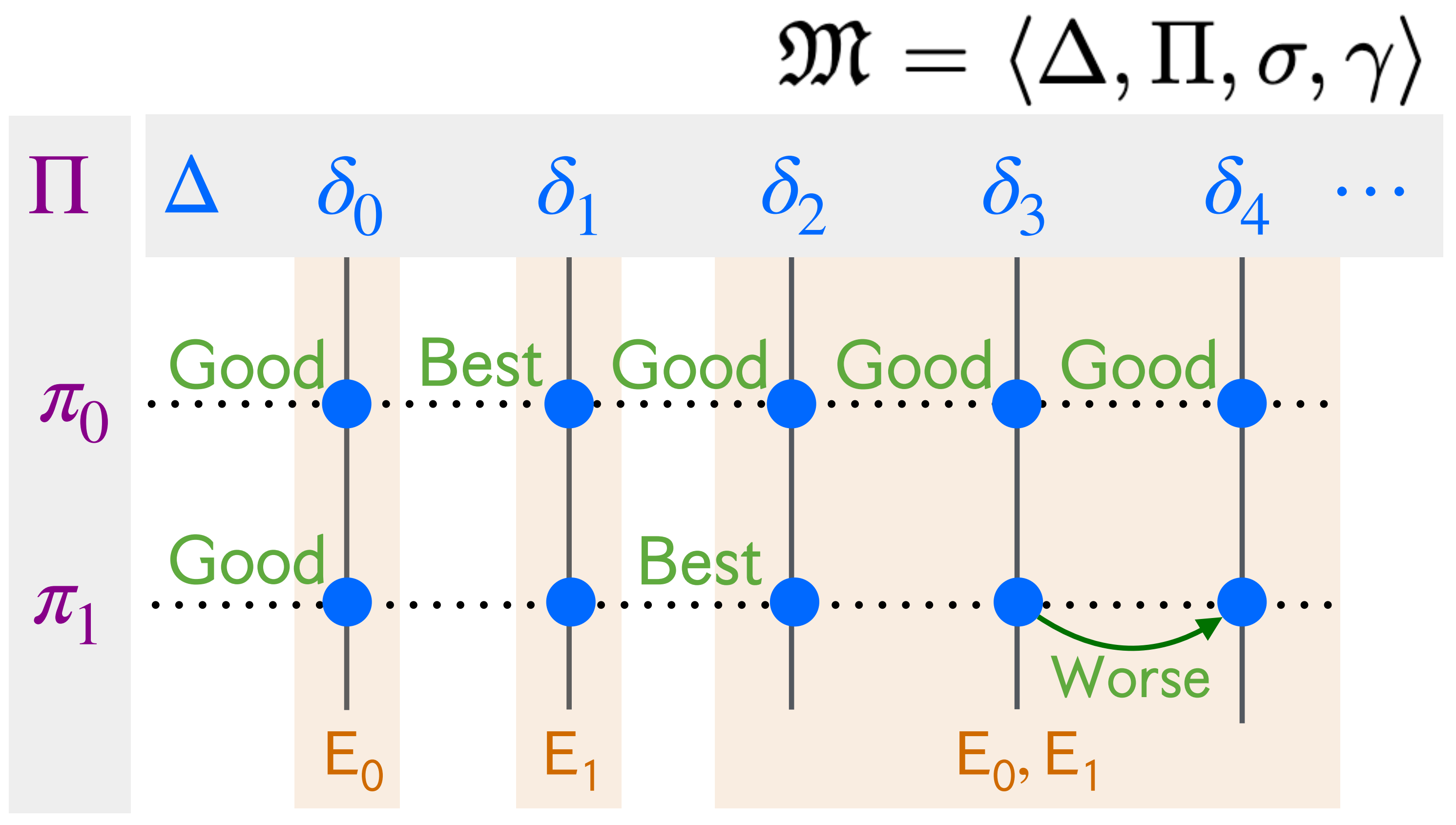
$\mathcal{I}^{\mathfrak{M}} = (\Delta', \cdot^{\mathcal{I}})$

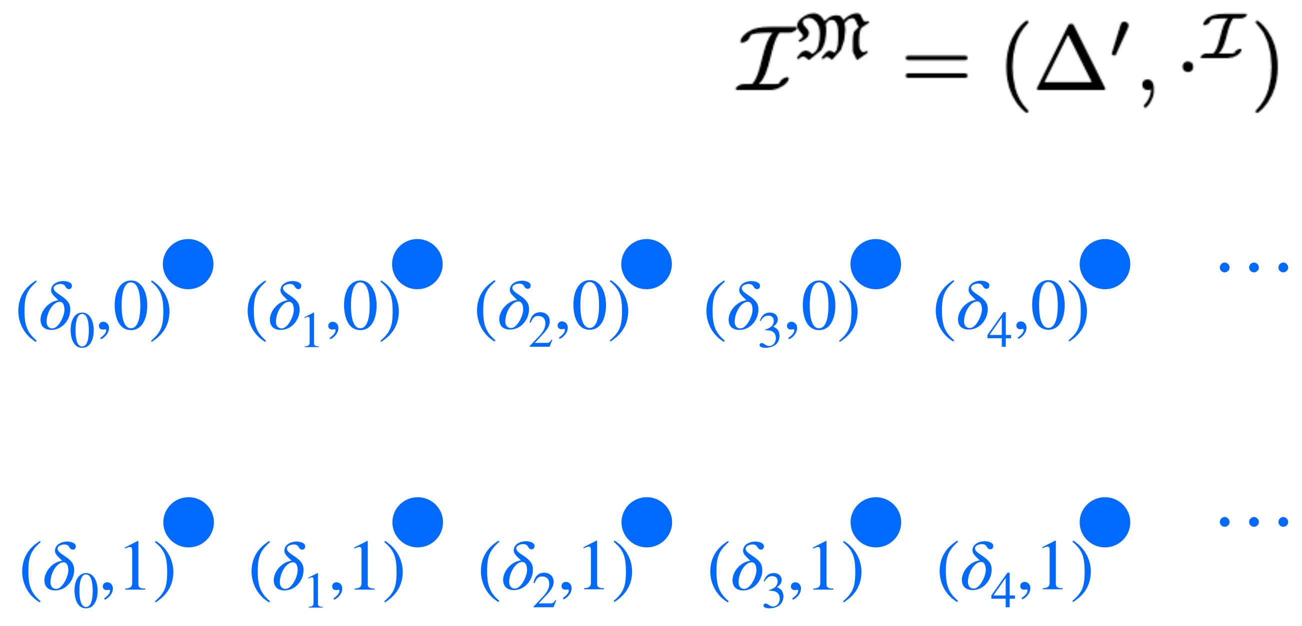
Stacked Interpretations

The <u>stacked interpretation</u> of $\mathfrak{M}$ (with $|\Pi| = 2^m$),

$\mathcal{I}^{\mathfrak{M}} = (\Delta', \cdot^{\mathcal{I}})$ with signature $\langle \mathbf{P} \uplus \{\mathsf{F}, \mathsf{L}_0, \ldots, \mathsf{L}_{m-1}\}, \varnothing \rangle$

- $\Delta' = \Delta \times \{0,\ldots,2^m - 1\}$





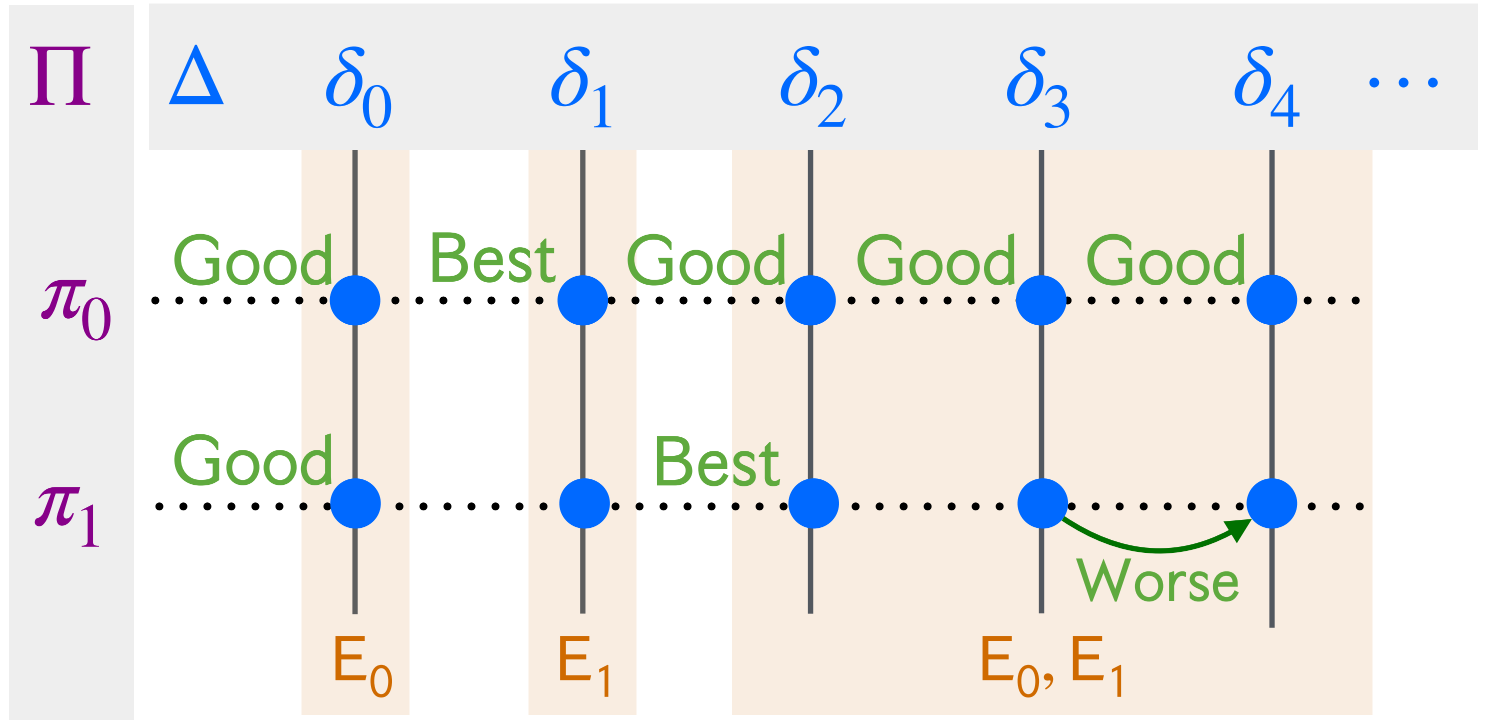
Stacked Interpretations

The <u>stacked interpretation</u> of $\mathfrak{M}$ (with $|\Pi| = 2^m$),

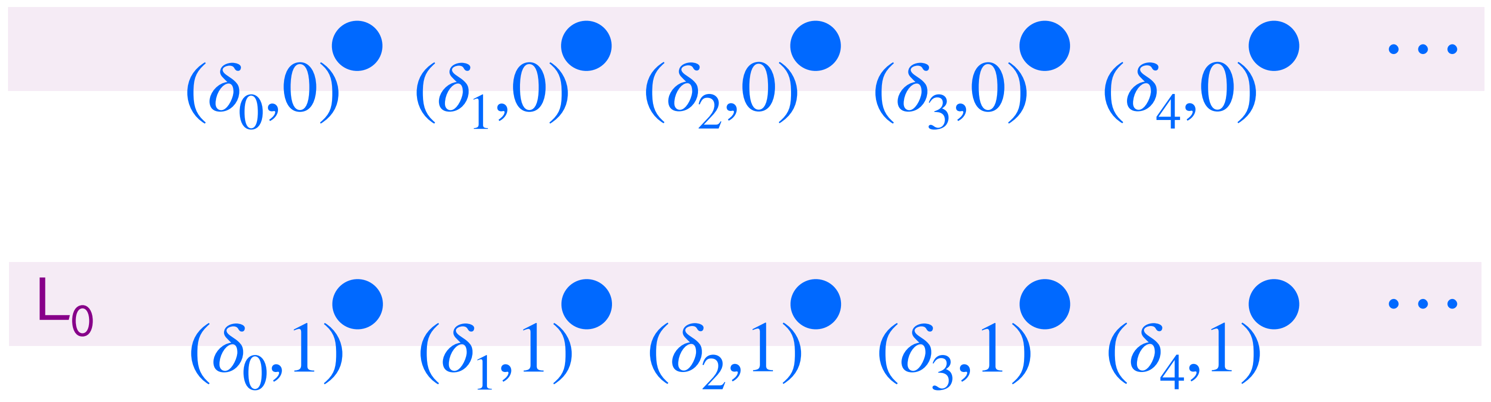
$\mathcal{I}^{\mathfrak{M}} = (\Delta', \cdot^{\mathcal{I}})$ with signature $\langle \mathbf{P} \uplus \{\mathsf{F}, \mathsf{L}_0, \ldots, \mathsf{L}_{m-1}\}, \varnothing \rangle$

- $\Delta' = \Delta \times \{0, \ldots, 2^m - 1\}$
- $\mathsf{L}_j^{\mathcal{I}} = \{(\delta, i) \mid \text{the } j^{th} \text{ bit of bin}(i) \text{ is } 1\}$

$\mathfrak{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$



$\mathcal{I}^{\mathfrak{M}} = (\Delta', \cdot^{\mathcal{I}})$

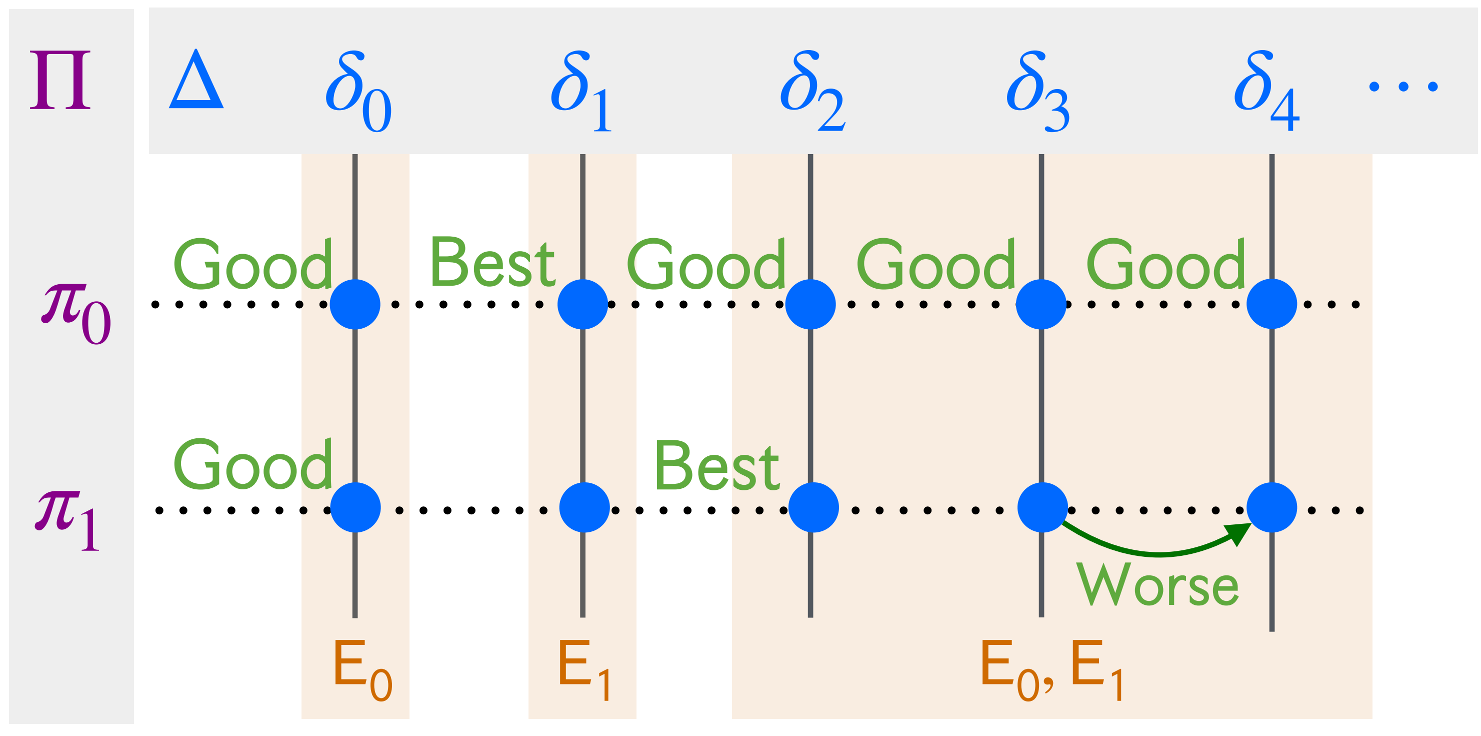


Stacked Interpretations

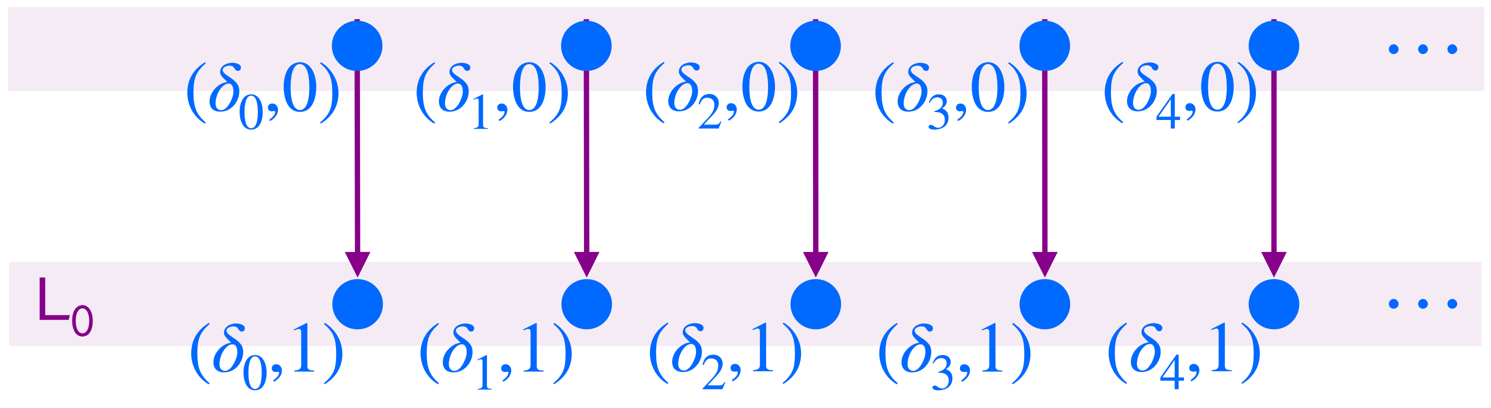
The stacked interpretation of $\mathfrak{M}$ (with $|\Pi| = 2^m$),
$\mathcal{I}^{\mathfrak{M}} = (\Delta', \cdot^{\mathcal{I}})$ with signature $\langle \mathbf{P} \uplus \{\mathsf{F}, \mathsf{L}_0, \ldots, \mathsf{L}_{m-1}\}, \varnothing \rangle$

- $\Delta' = \Delta \times \{0,\ldots,2^m - 1\}$
- $\mathsf{L}_j^{\mathcal{I}} = \{(\delta, i) \mid \text{the } j^{th} \text{ bit of bin}(i) \text{ is } 1\}$
- $\mathsf{F}^{\mathcal{I}} = \{((\delta, i), (\delta, i+1)) \mid \delta \in \Delta, i < 2^m - 1\}$

$\mathfrak{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$



$\mathcal{I}^{\mathfrak{M}} = (\Delta', \cdot^{\mathcal{I}})$

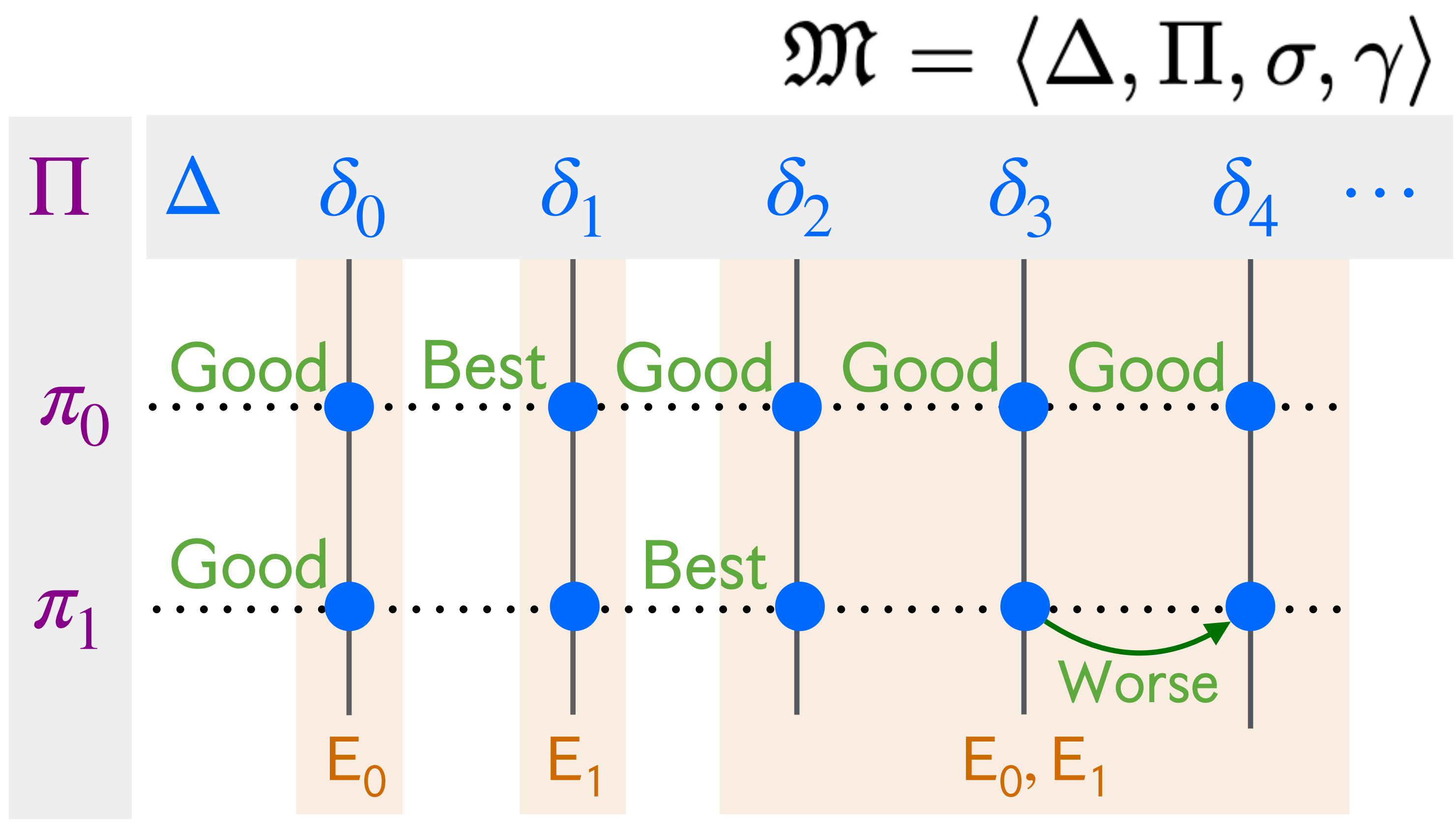


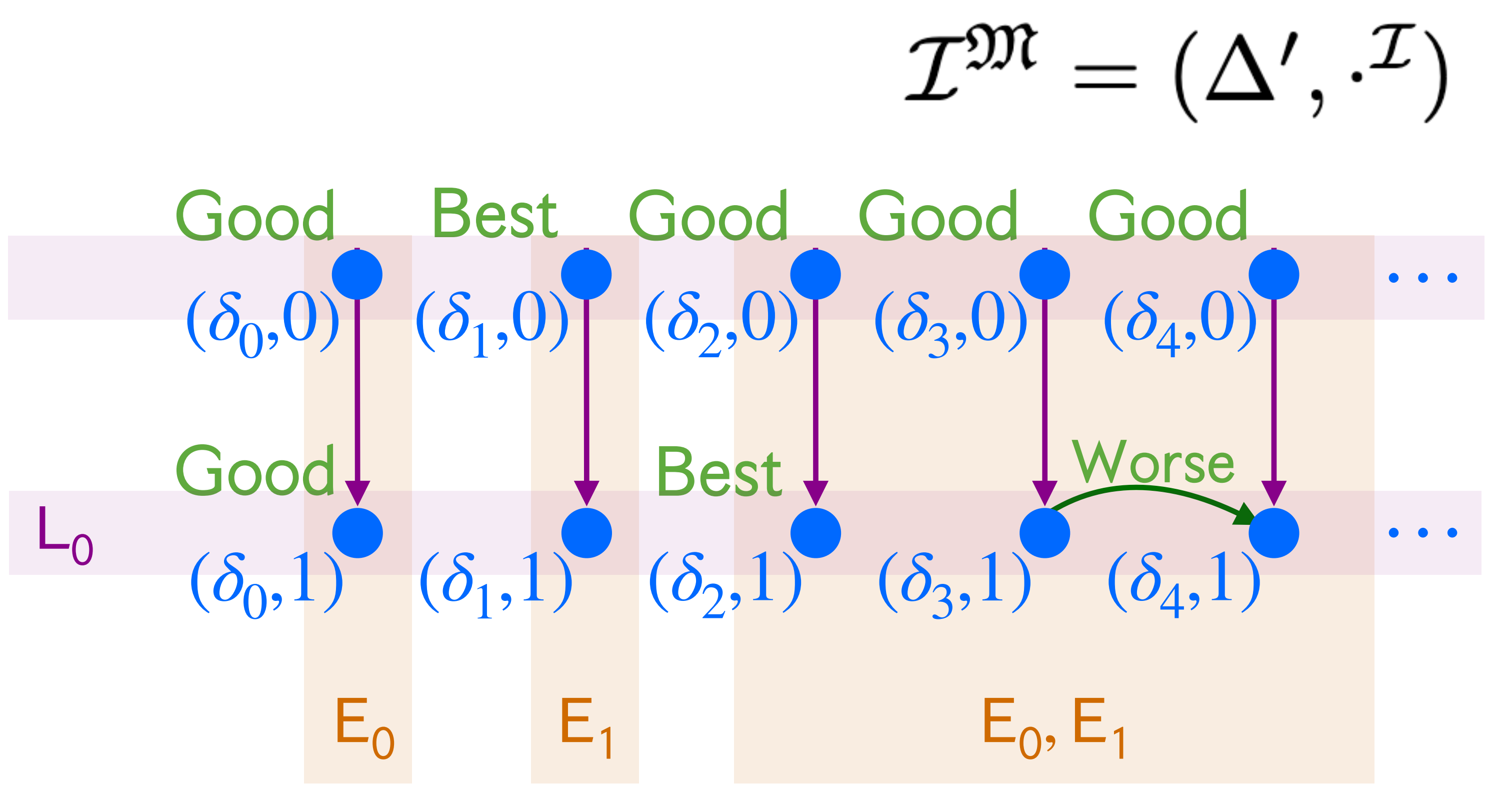
Stacked Interpretations

The <u>stacked interpretation</u> of $\mathfrak{M}$ (with $|\Pi| = 2^m$),

$\mathcal{I}^{\mathfrak{M}} = (\Delta', \cdot^{\mathcal{I}})$ with signature $\langle \mathbf{P} \uplus \{\mathsf{F}, \mathsf{L}_0, \ldots, \mathsf{L}_{m-1}\}, \varnothing \rangle$

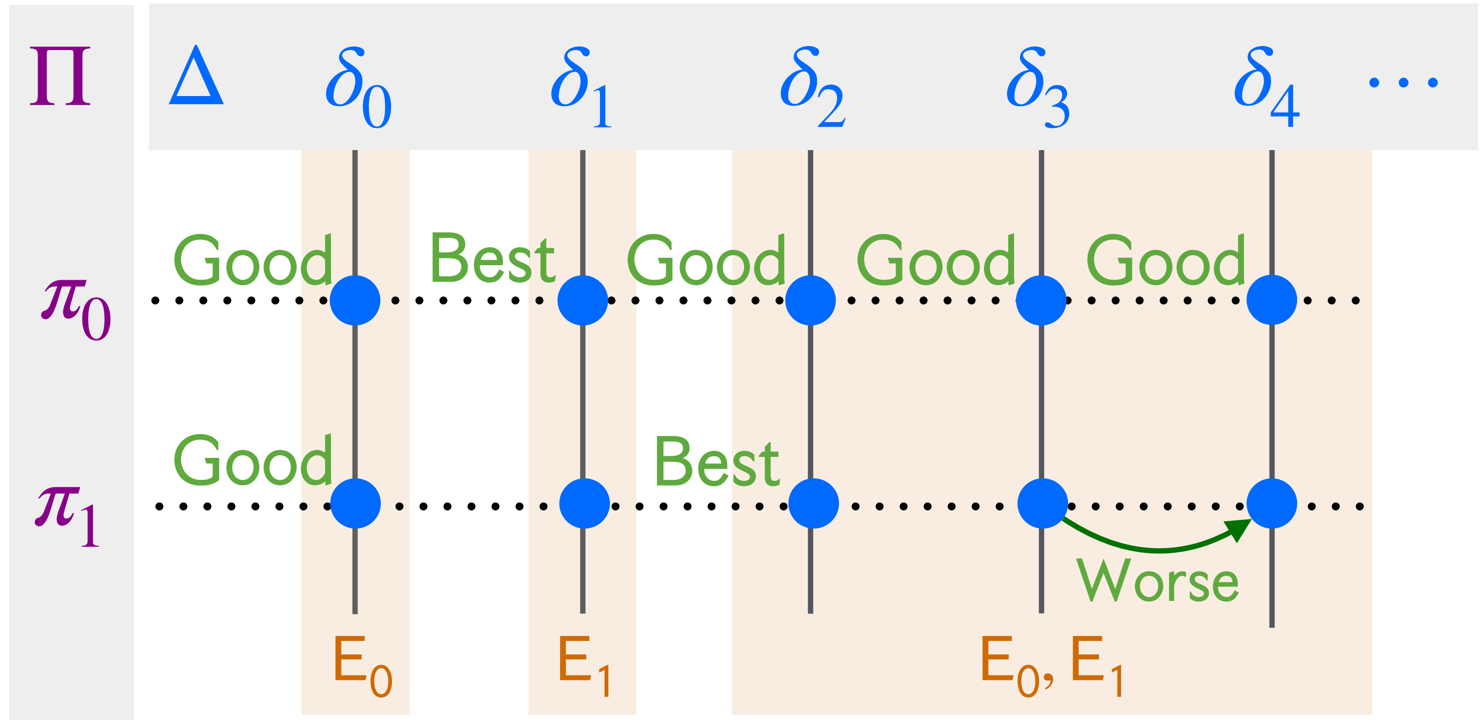
- $\Delta' = \Delta \times \{0,\ldots,2^m - 1\}$
- $\mathsf{L}_j^{\mathcal{I}} = \{(\delta, i) \mid \text{the } j^{th} \text{ bit of } \mathrm{bin}(i) \text{ is } 1\}$
- $\mathsf{F}^{\mathcal{I}} = \{((\delta, i), (\delta, i+1)) \mid \delta \in \Delta, i < 2^m - 1\}$
- $\mathsf{P}^{\mathcal{I}} = \{(\delta, i) \mid i < 2^m, \delta \in \mathsf{P}^{\gamma(\pi_i)}\}$
- $\mathsf{P}^{\mathcal{I}} = \{((\delta_1, i), (\delta_2, i)) \mid i < 2^m, (\delta_1, \delta_2) \in \mathsf{P}^{\gamma(\pi_i)}\}$



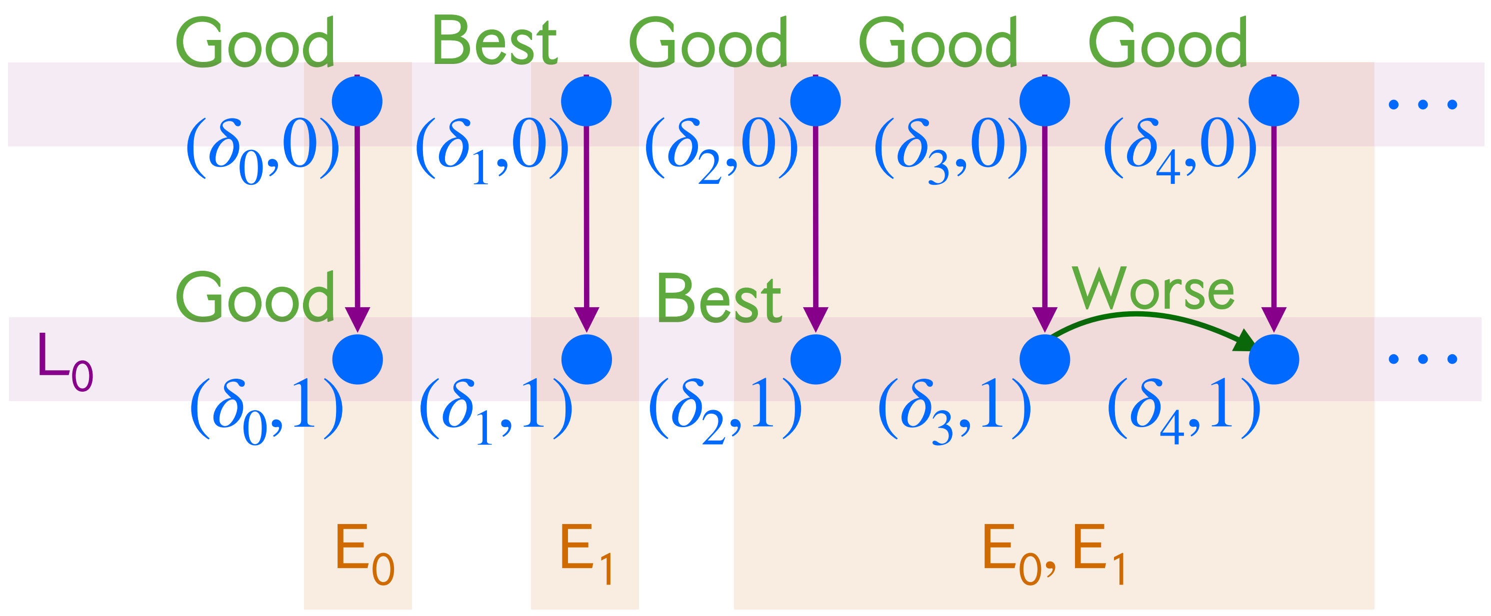


Stacked Interpretations

$$\mathfrak{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$$



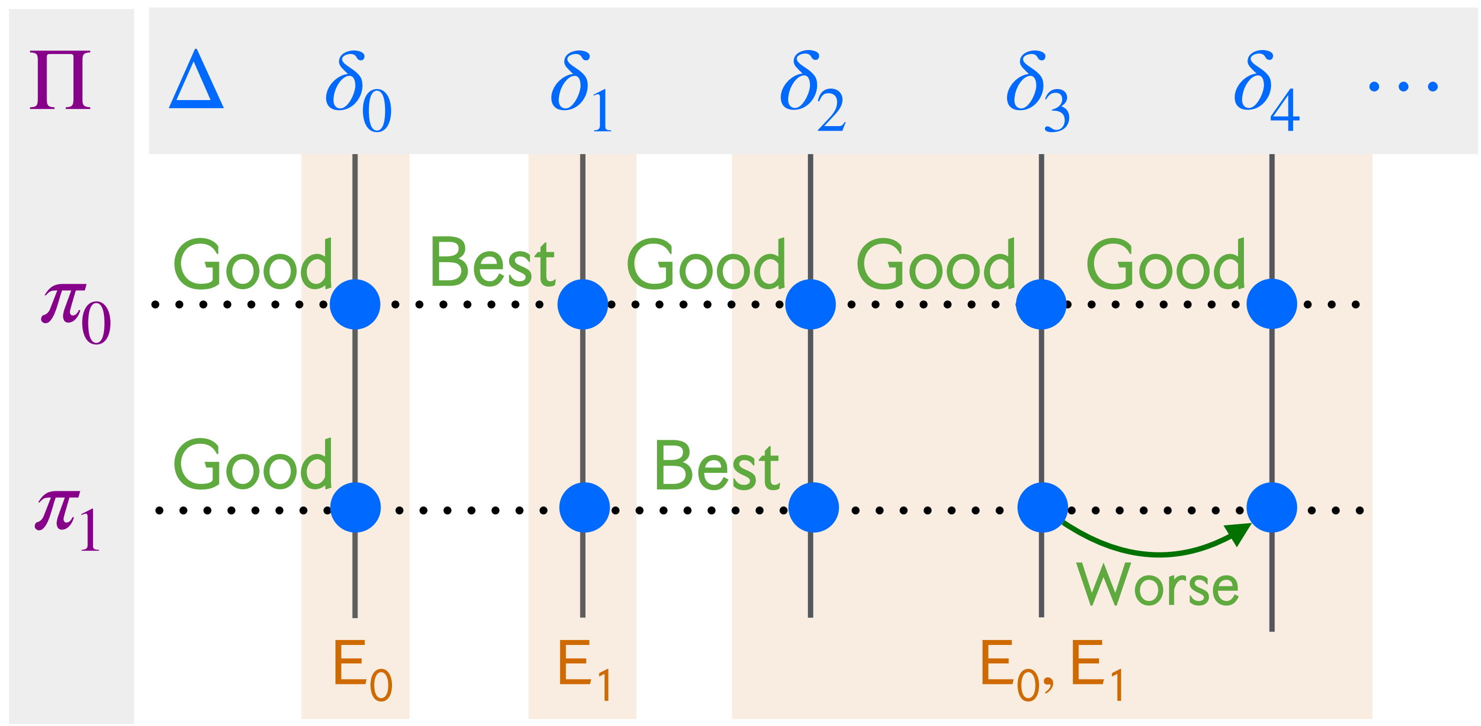
$$\mathcal{I}^{\mathfrak{M}} = (\Delta', \cdot^{\mathcal{I}})$$



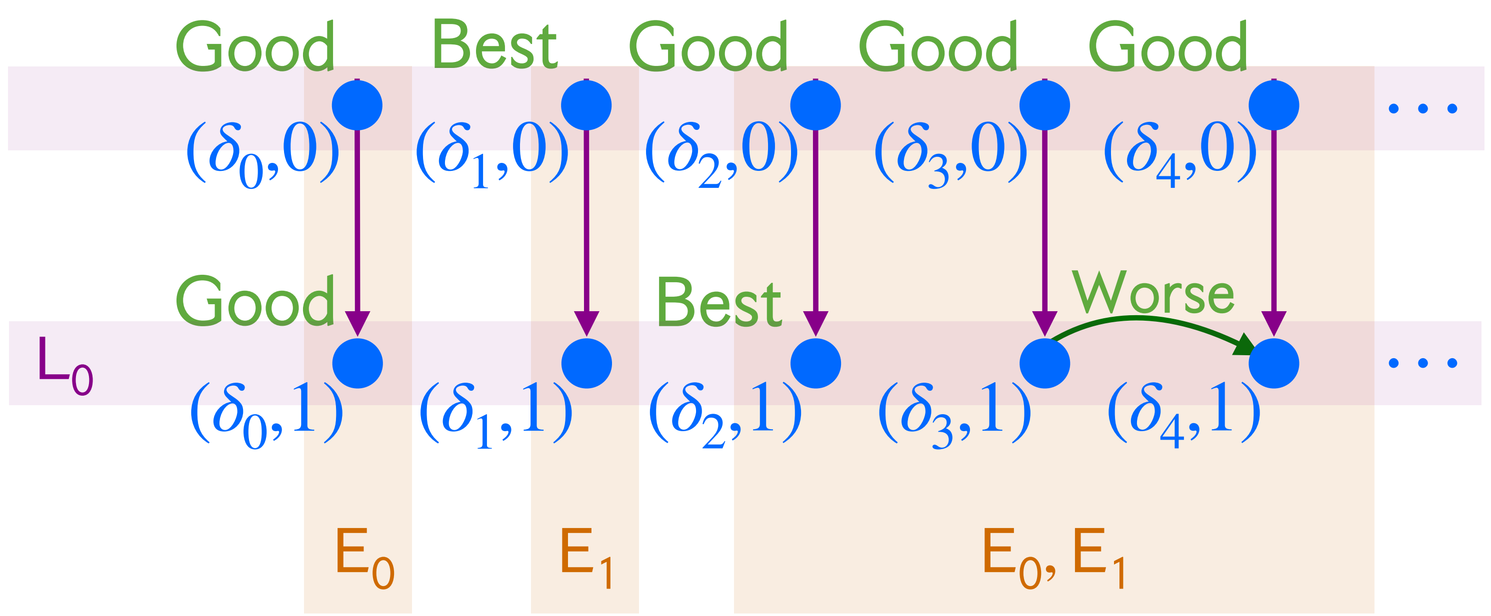
Stacked Interpretations

The C^2 stacked formula ϕ^{m}_{stack} forces all elements to have:

$$\mathfrak{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$$



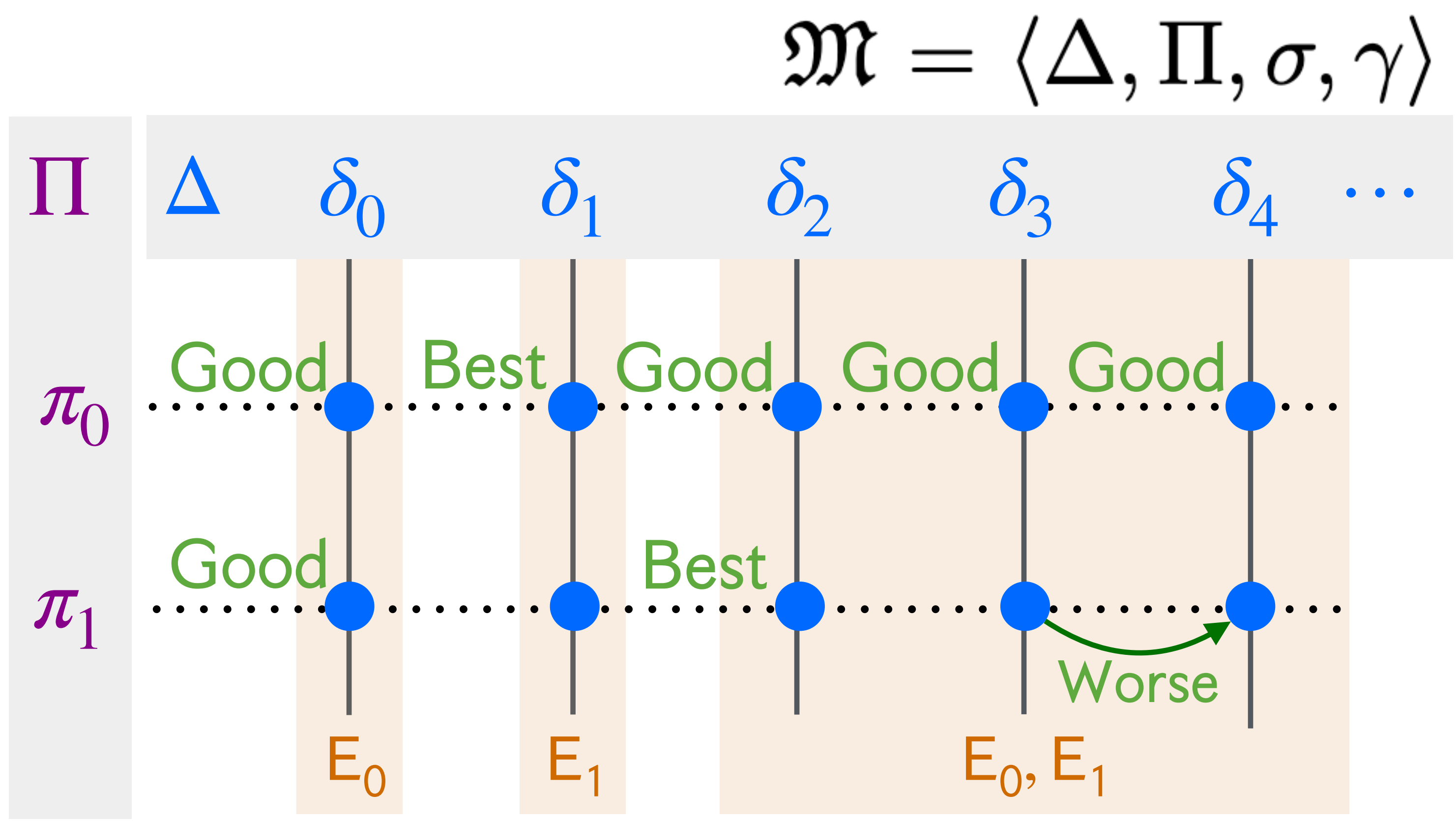
$$\mathcal{I}^{\mathfrak{M}} = (\Delta', \cdot^{\mathcal{I}})$$

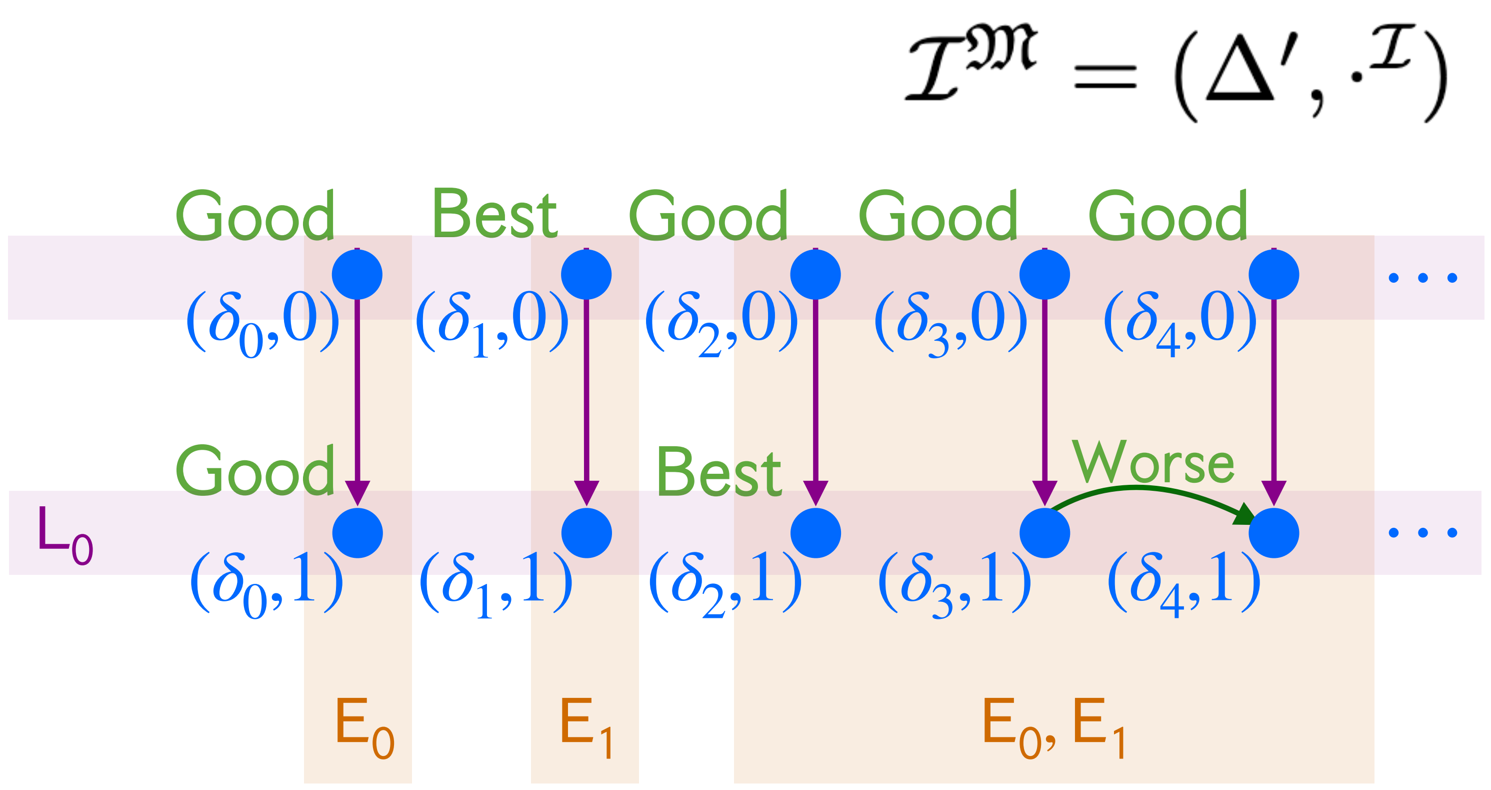


Stacked Interpretations

The C^2 <u>stacked formula</u> ϕ^{m}_{stack} forces all elements to have:

- 1 F-successor unless they have the last index
- 0 F-successors if they have the last index
- 1 F-predecessor unless they have the first index
- 0 F-predecessors if they have the first index



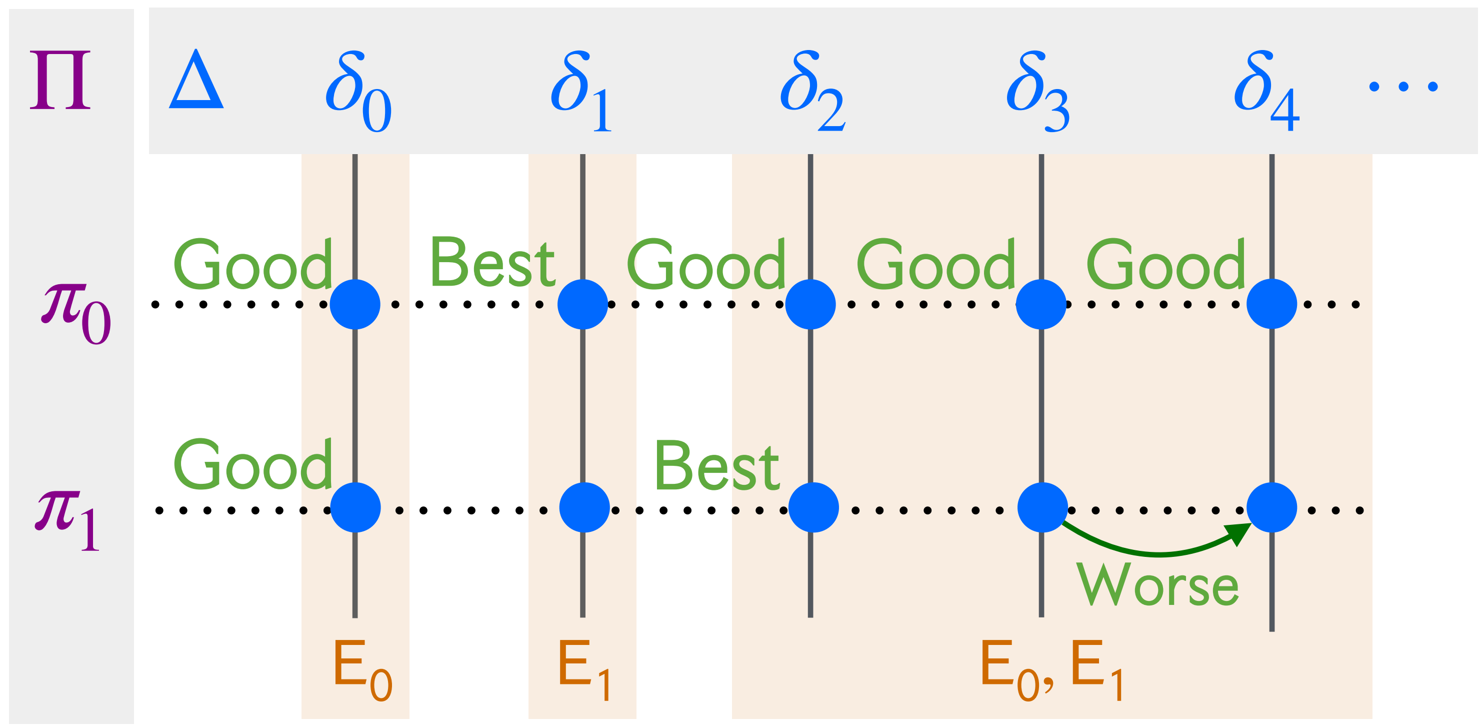


Stacked Interpretations

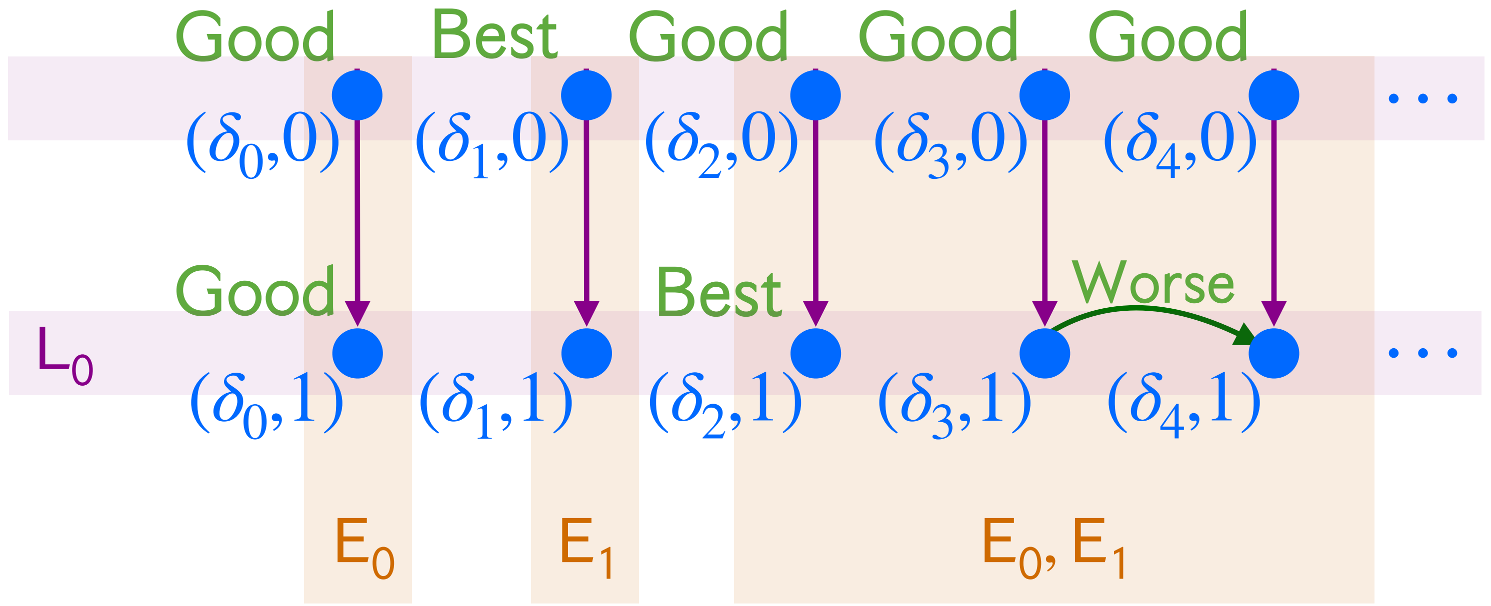
The C^2 stacked formula ϕ^m_{stack} forces all elements to have:

- 1 F-successor unless they have the last index
- 0 F-successors if they have the last index
- 1 F-predecessor unless they have the first index
- 0 F-predecessors if they have the first index
- Any two F-connected elements have consecutive indexes

$\mathfrak{M} = \langle \Delta, \Pi, \sigma, \gamma \rangle$



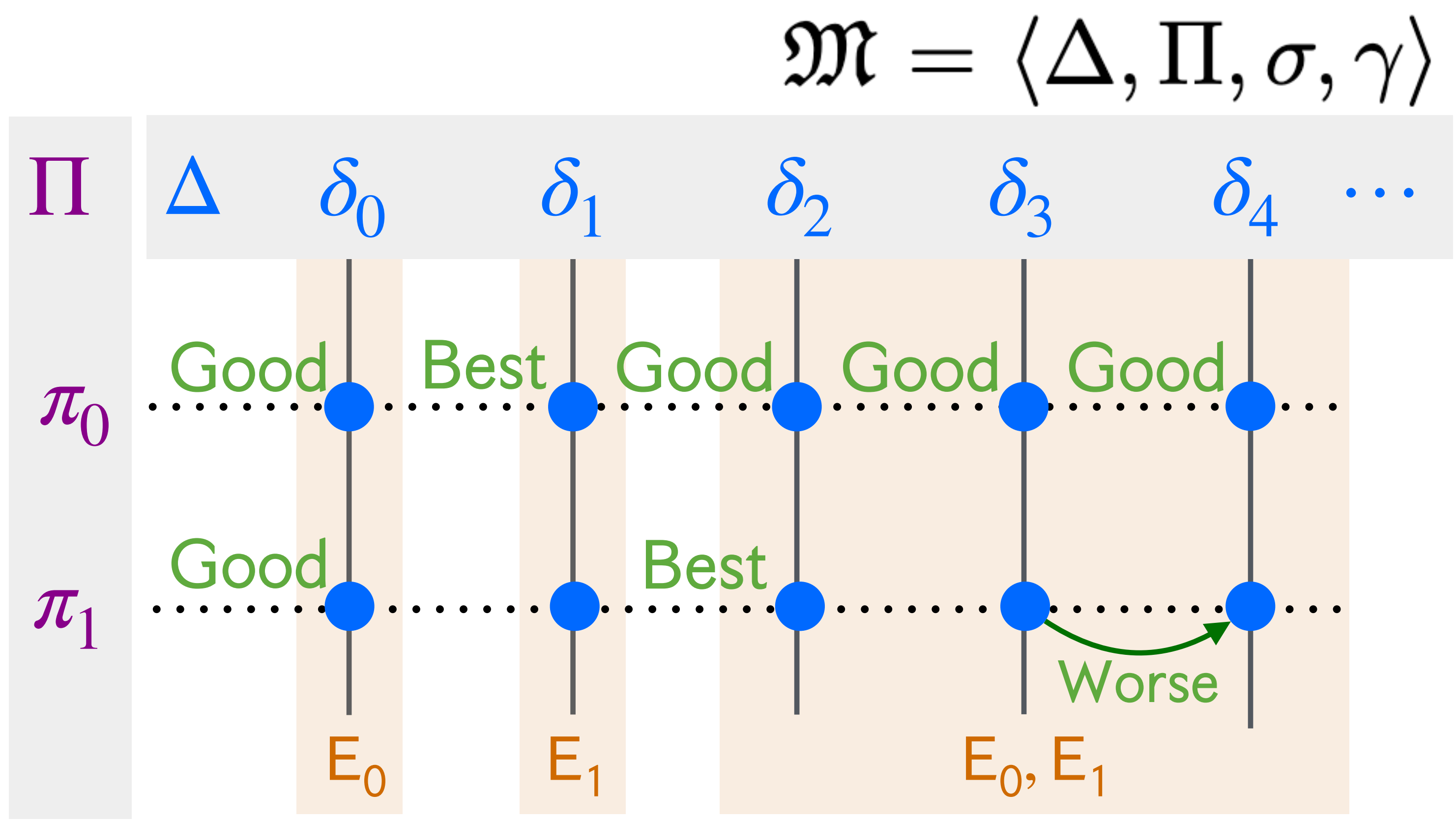
$\mathcal{I}^{\mathfrak{M}} = (\Delta', \cdot^{\mathcal{I}})$

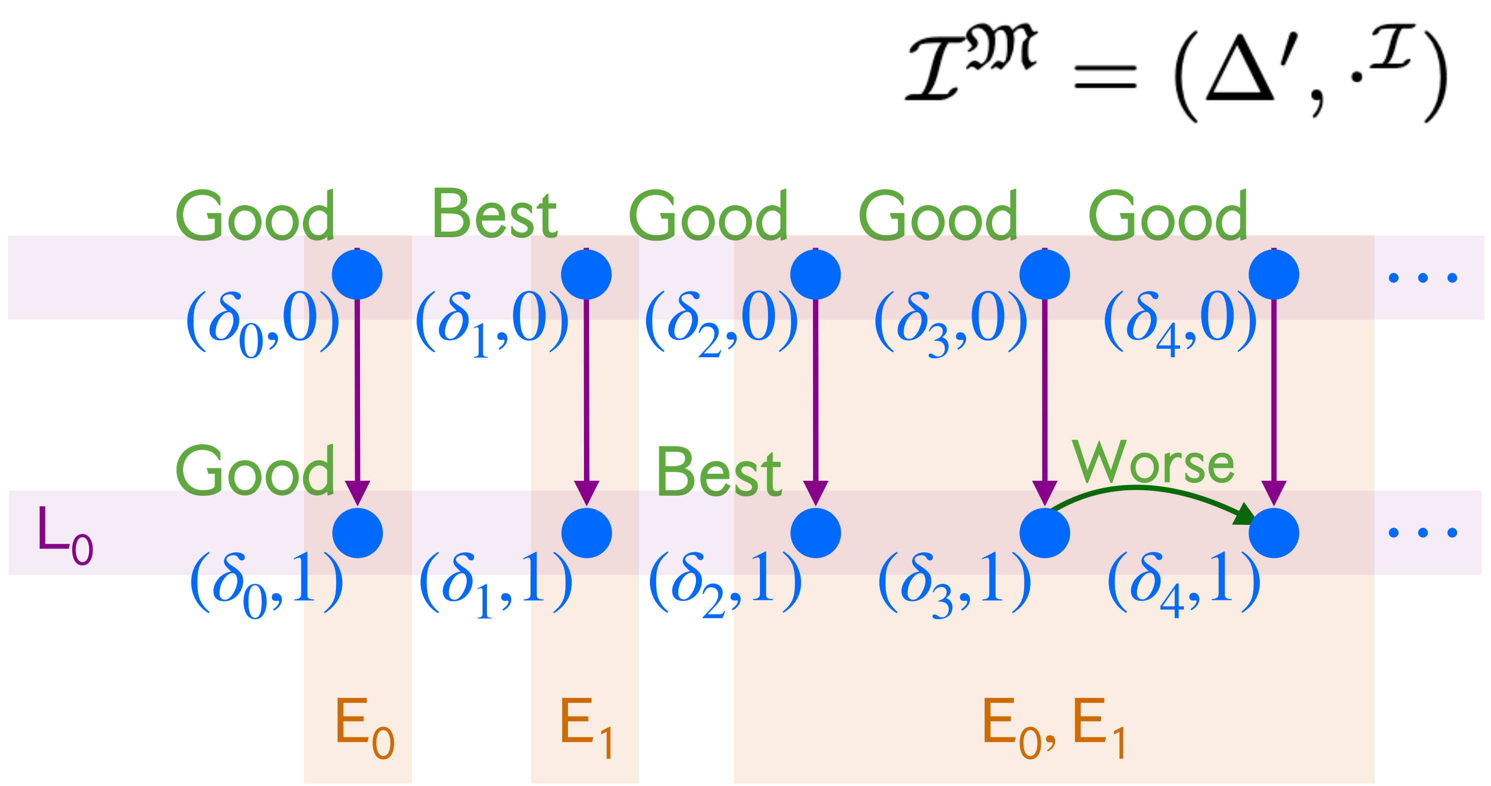


Stacked Interpretations

The C^2 stacked formula ϕ^{m}_{stack} forces all elements to have:

- 1 F-successor unless they have the last index
- 0 F-successors if they have the last index
- 1 F-predecessor unless they have the first index
- 0 F-predecessors if they have the first index
- Any two F-connected elements have consecutive indexes
- Binary predicates (except F) relate elements with matching indices.



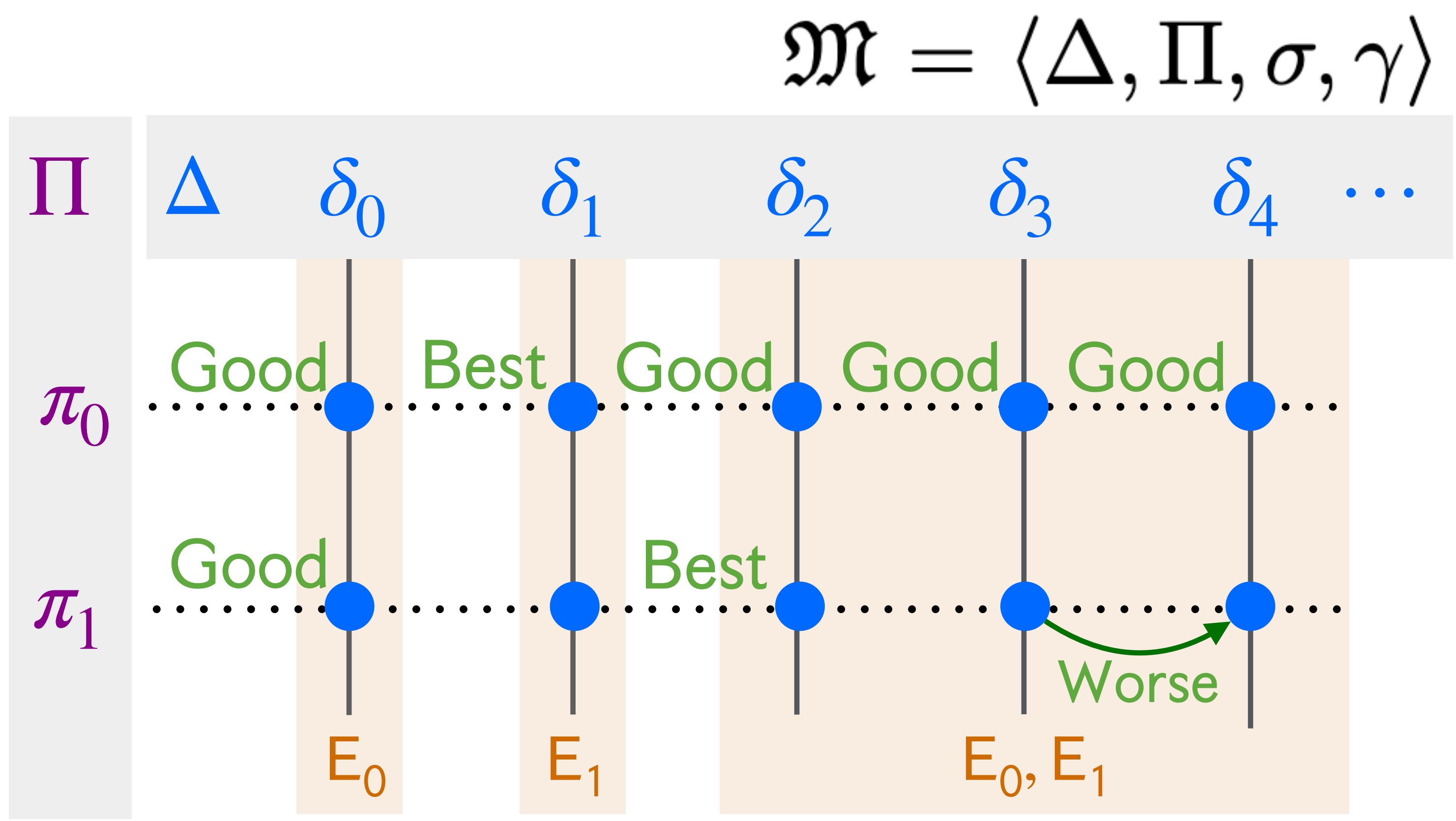


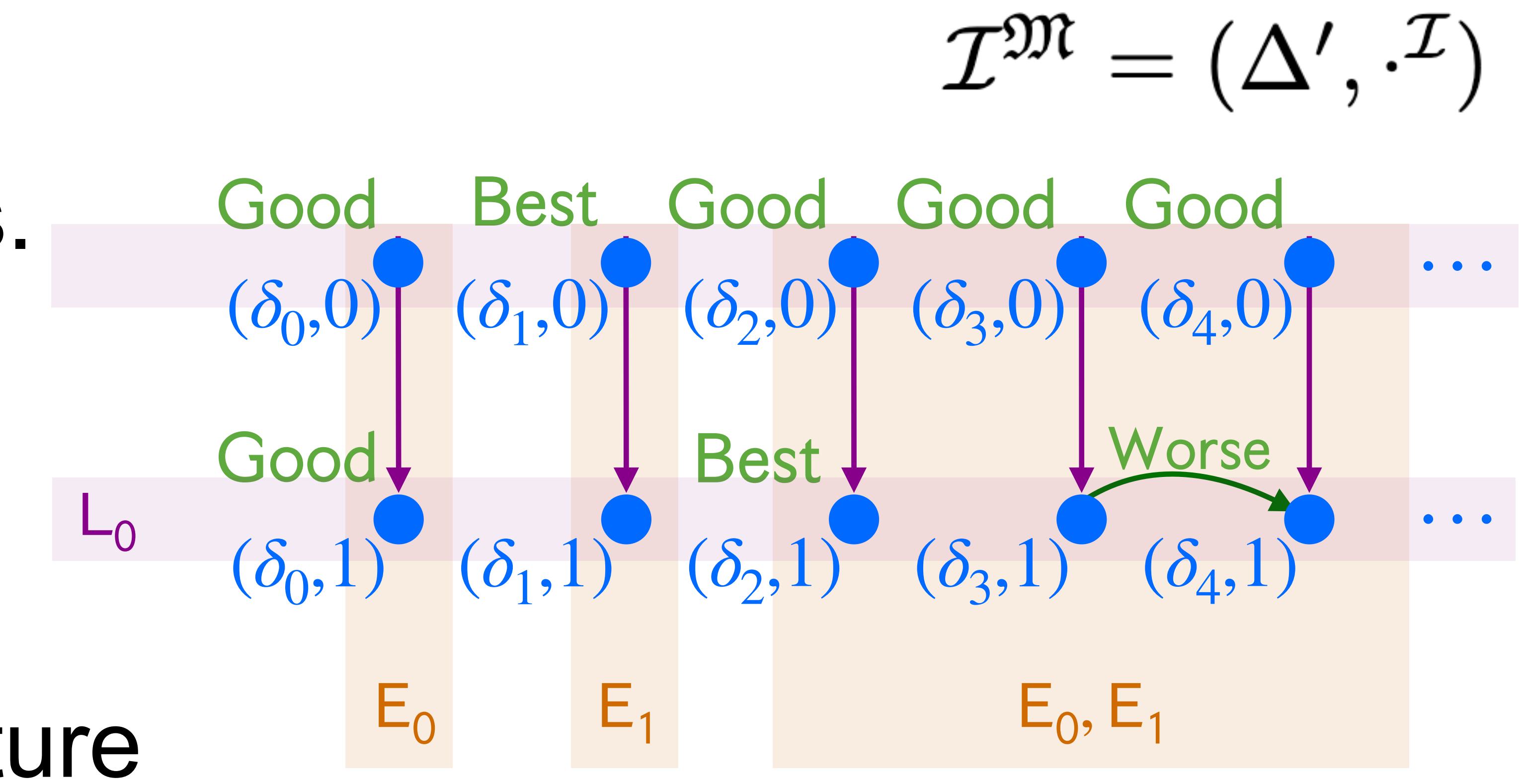
Stacked Interpretations

The C^2 stacked formula ϕ^m_{stack} forces all elements to have:

- 1 F-successor unless they have the last index
- 0 F-successors if they have the last index
- 1 F-predecessor unless they have the first index
- 0 F-predecessors if they have the first index
- Any two F-connected elements have consecutive indexes
- Binary predicates (except F) relate elements with matching indices.

An FO interpretation satisfies ϕ^m_{stack} *iff*
it's isomorphic to a stacked interpretation of an exp. sized FOSL structure





Translating to C^2

Translating to C^2

A translation maps a monodic C^2 FOSL formula ϕ to a plain C^2 formula

Translating to C^2

A translation maps a monodic C^2 FOSL formula ϕ to a plain C^2 formula

- $\exists^{\lhd n} x \,.\, \phi$ is evaluated on elements in the same L-layer

Translating to C^2

A translation maps a monodic C^2 FOSL formula ϕ to a plain C^2 formula

- $\exists^{\triangleleft n} x \,.\, \phi$ is evaluated on elements in the same L-layer
- $\Diamond\phi$ is evaluated on elements with the same E-type

  If an element of an E-type satisfies ϕ, there are permutations so all elements satisfy $\Diamond\phi$

  If $\Diamond\phi$ is sentential, the variable assignment does nothing

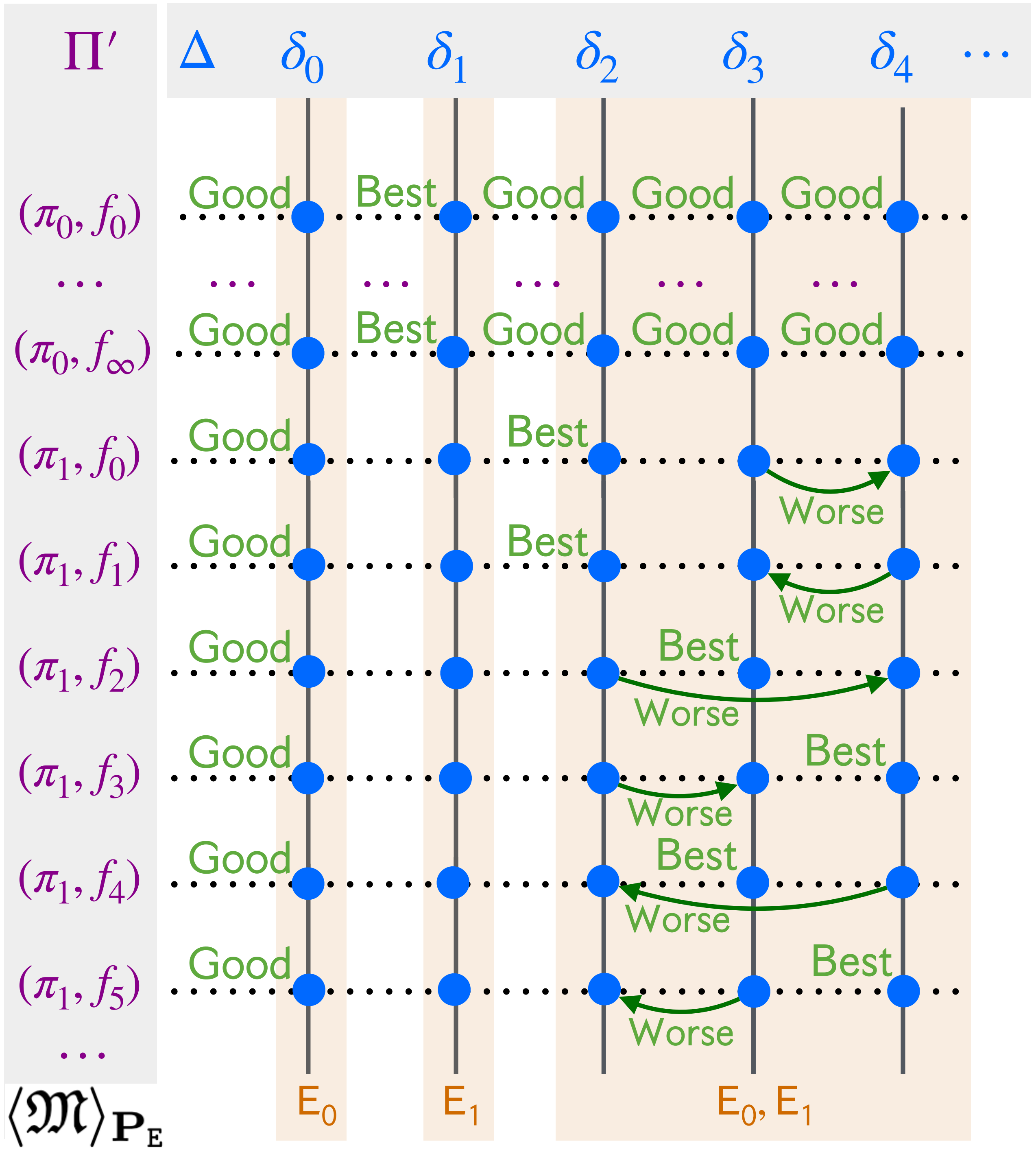
Translating to C^2

A translation maps a monodic C^2 FOSL formula ϕ to a plain C^2 formula

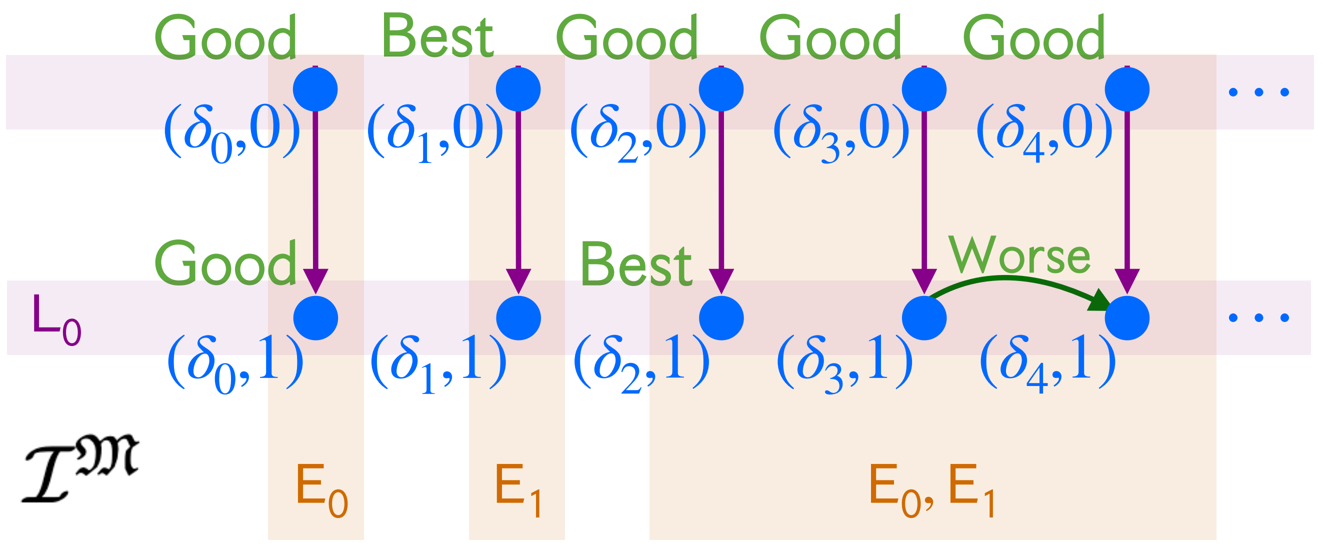
- $\exists^{\triangleleft n} x\,.\,\phi$ is evaluated on elements in the same L-layer
- $\Diamond\phi$ is evaluated on elements with the same E-type

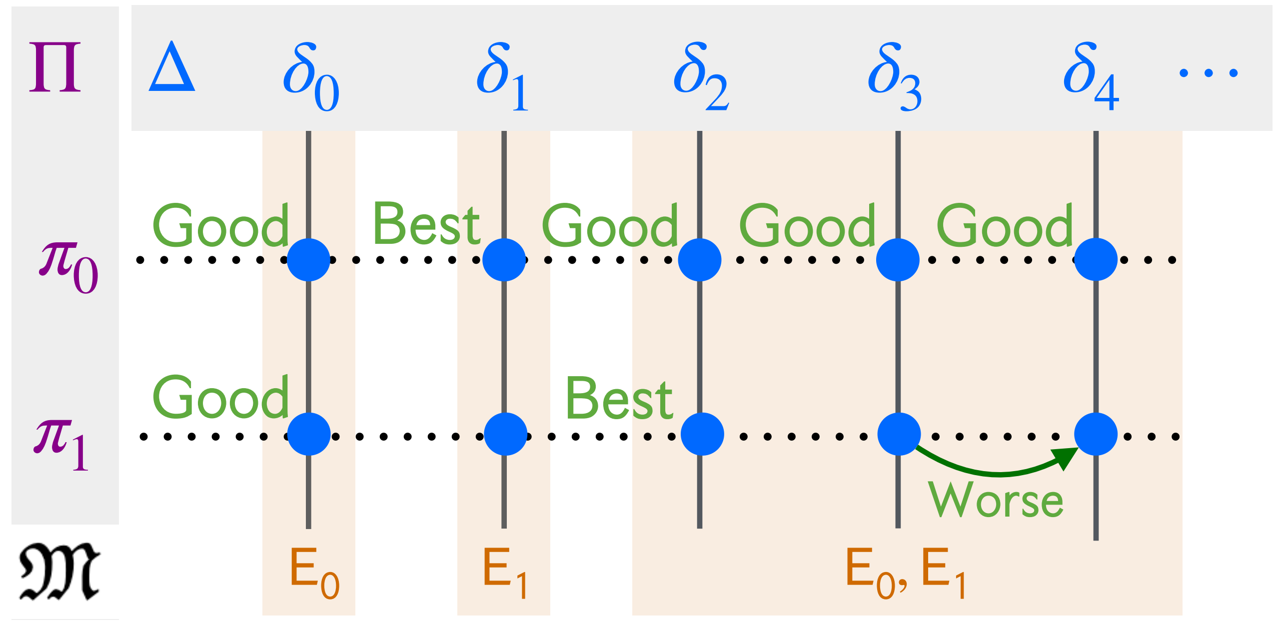
  If an element of an E-type satisfies ϕ, there are permutations so all elements satisfy $\Diamond\phi$

  If $\Diamond\phi$ is sentential, the variable assignment does nothing



$$
\begin{aligned}
\phi &= \exists^{=1}x.(E_0) \land \forall x.(E_1) \land E_2 \text{ with} \\
E_0 &= \Box_*(\mathtt{Good}(x)) \\
E_1 &= E_0 \lor \Diamond_*(\mathtt{Best}(x) \land \forall y.(\mathtt{Best}(y) \leftrightarrow x \doteq y)) \\
E_2 &= \Diamond_*(\forall x.\mathtt{Good}(x) \lor \mathtt{Best}(x))
\end{aligned}
$$





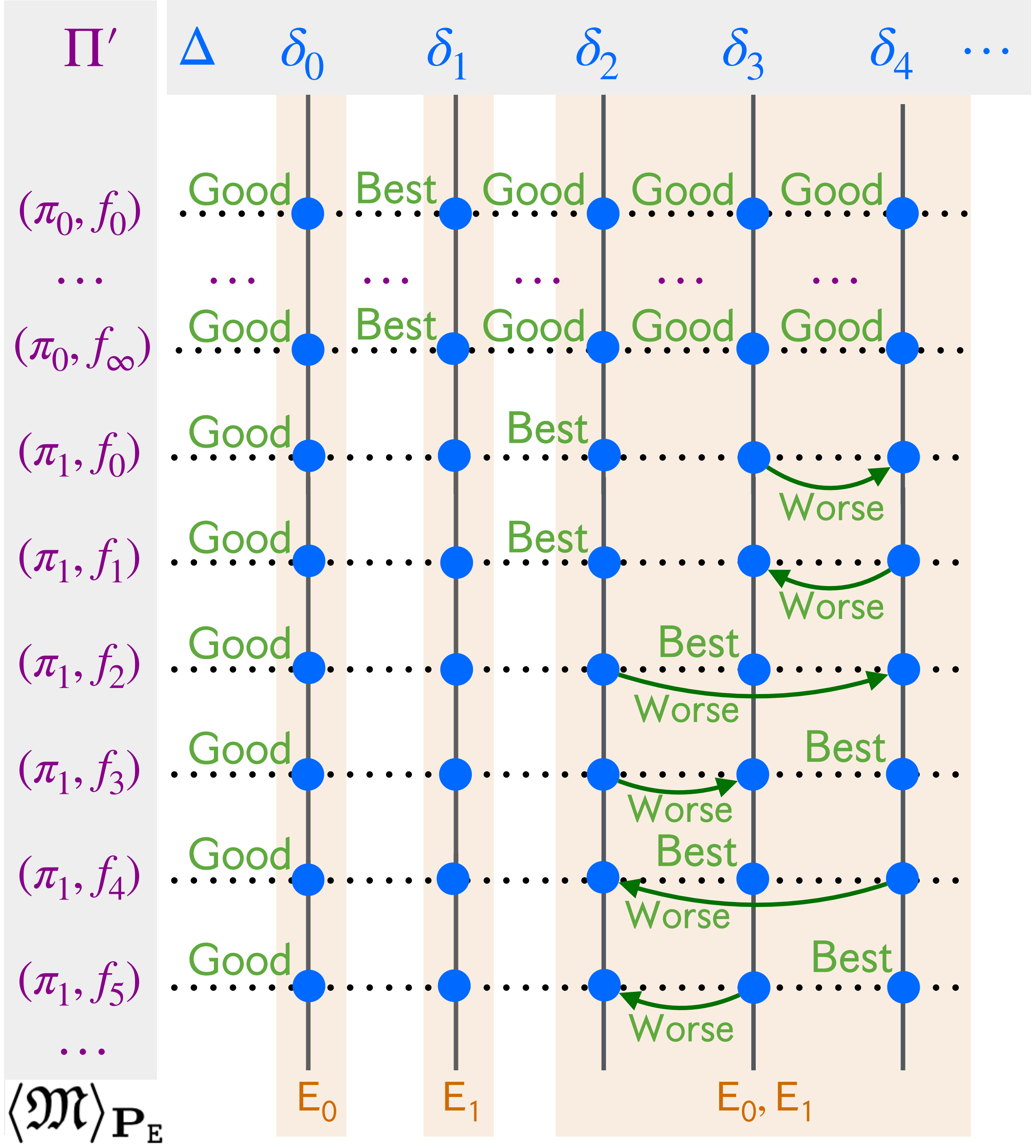
Translating to C^2

A translation maps a monodic C^2 FOSL formula ϕ to a plain C^2 formula

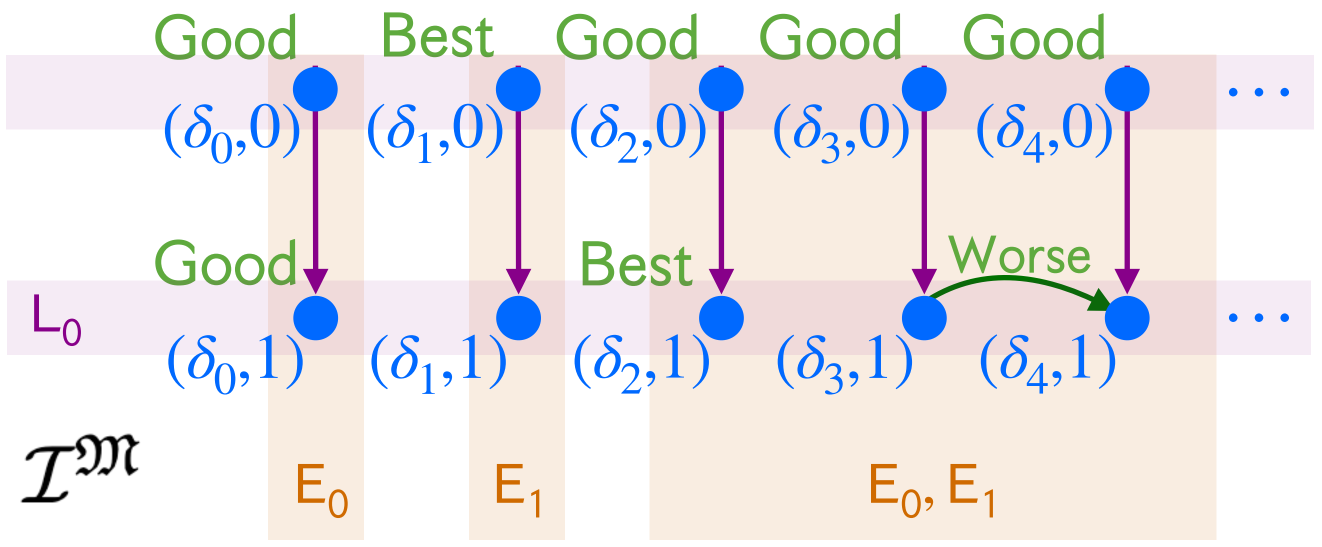
- $\exists^{\triangleleft n} x \,.\, \phi$ is evaluated on elements in the same L-layer
- $\Diamond\phi$ is evaluated on elements with the same E-type

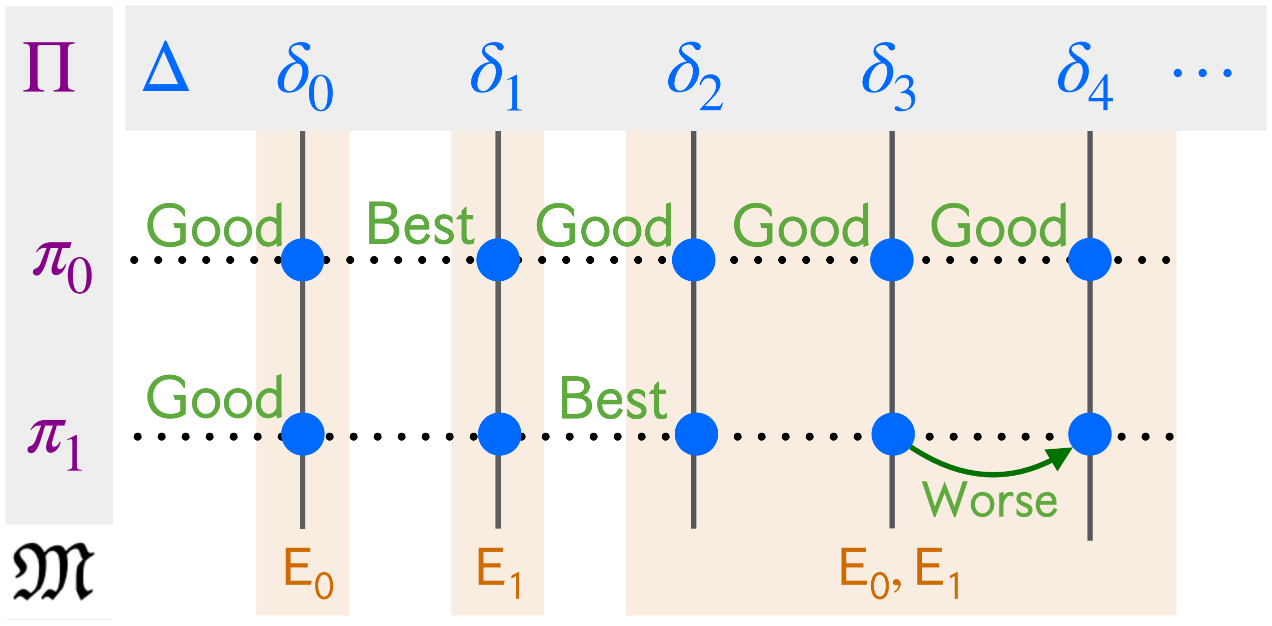
  If an element of an E-type satisfies ϕ, there are permutations so all elements satisfy $\Diamond\phi$

  If $\Diamond\phi$ is sentential, the variable assignment does nothing



$$\phi = \exists^{=1} x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$
$$E_0 = \Box_*(\mathtt{Good}(x))$$
$$E_1 = E_0 \vee \Diamond_*(\mathtt{Best}(x) \wedge \forall y.(\mathtt{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\mathtt{Good}(x) \vee \mathtt{Best}(x))$$





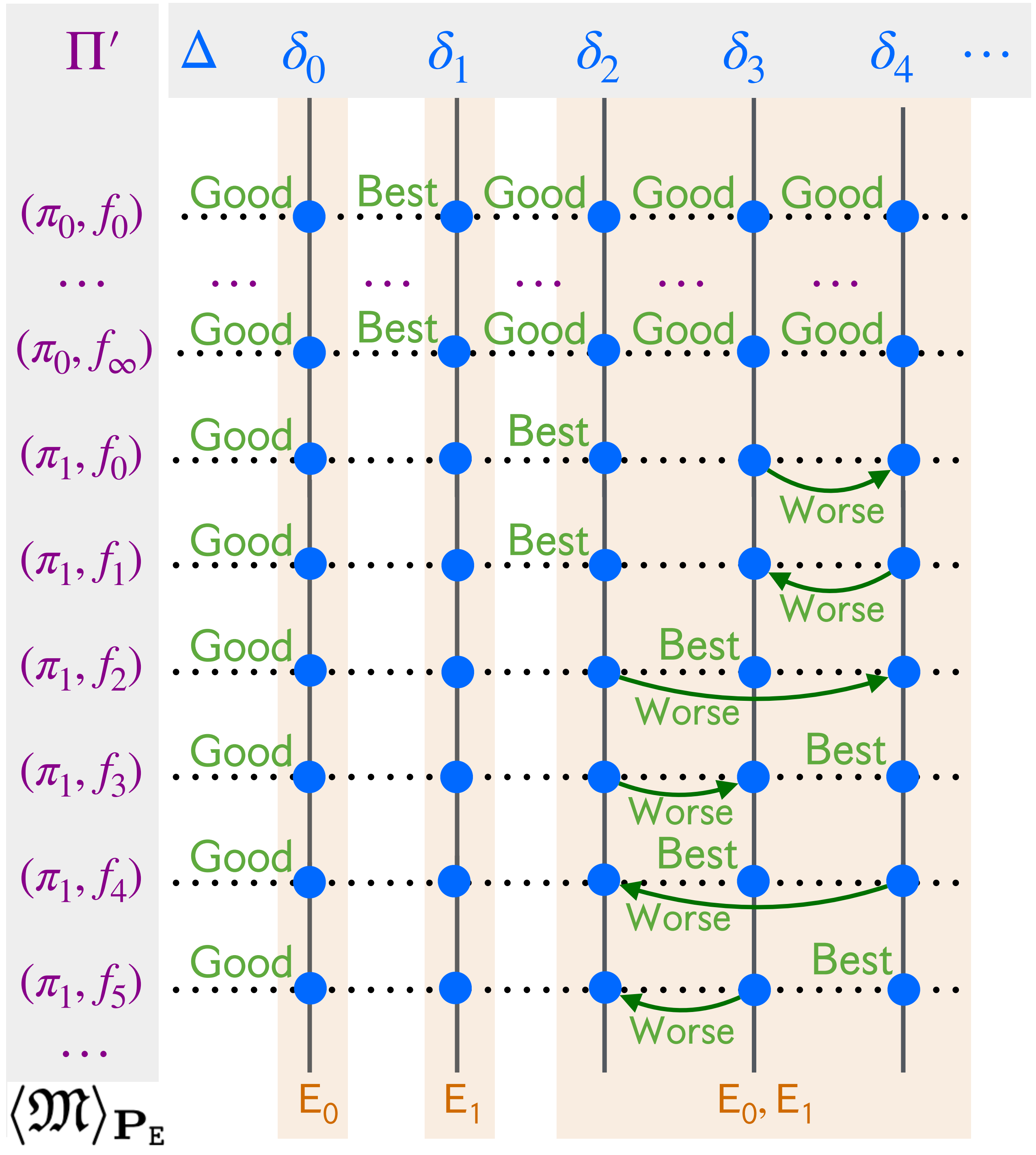
Translating to C^2

A translation maps a monodic C^2 FOSL formula ϕ to a plain C^2 formula

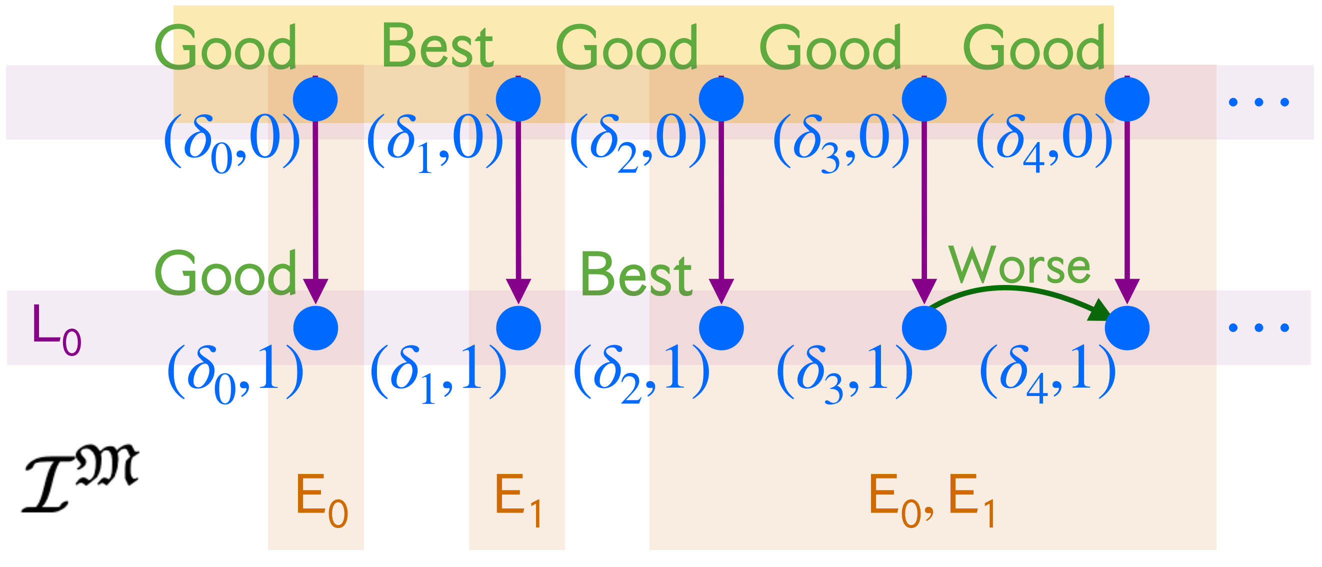
- $\exists^{\lhd n} x \,.\, \phi$ is evaluated on elements in the same L-layer
- $\Diamond\phi$ is evaluated on elements with the same E-type

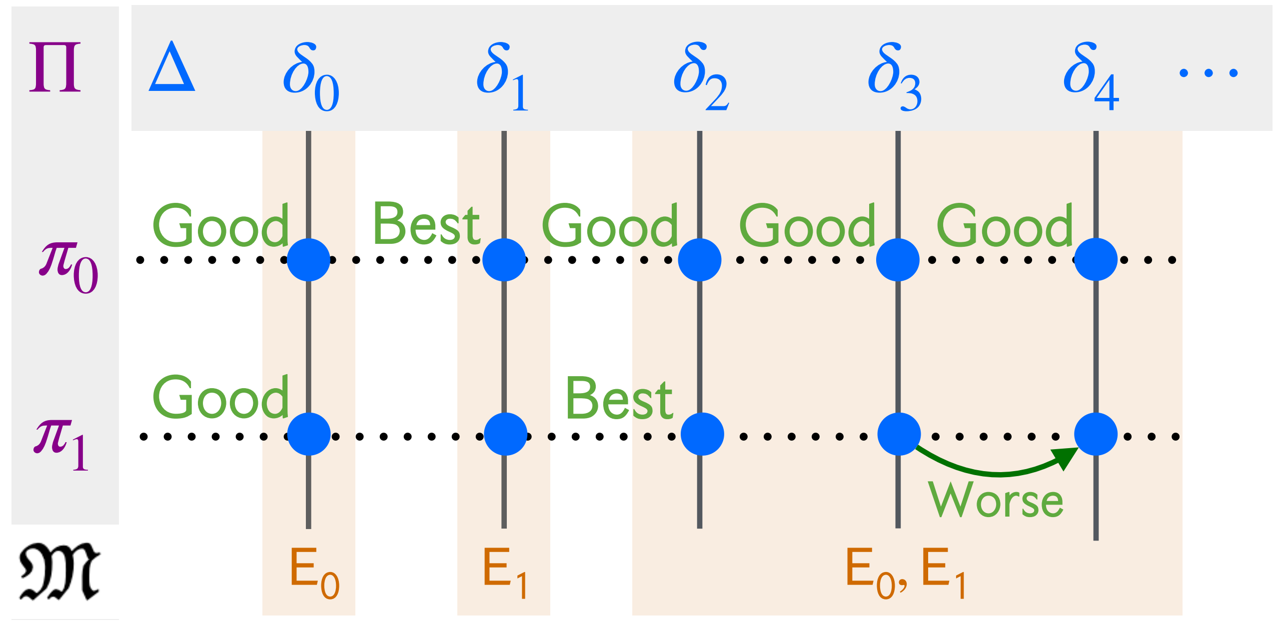
  If an element of an E-type satisfies ϕ, there are permutations so all elements satisfy $\Diamond\phi$

  If $\Diamond\phi$ is sentential, the variable assignment does nothing



$$
\begin{aligned}
\phi &= \exists^{=1}x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with} \\
E_0 &= \Box_*(\mathtt{Good}(x)) \\
E_1 &= E_0 \vee \Diamond_*(\mathtt{Best}(x) \wedge \forall y.(\mathtt{Best}(y) \leftrightarrow x \doteq y)) \\
E_2 &= \Diamond_*(\forall x.\mathtt{Good}(x) \vee \mathtt{Best}(x))
\end{aligned}
$$





Translating to C^2

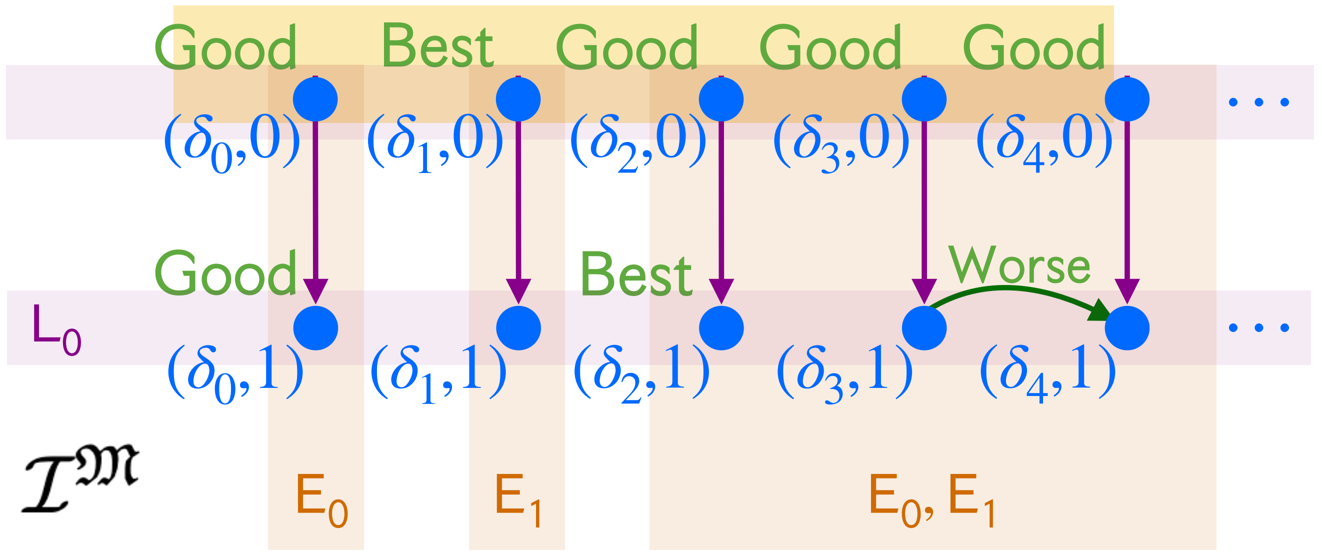
A translation maps a monodic C^2 FOSL formula ϕ to a plain C^2 formula

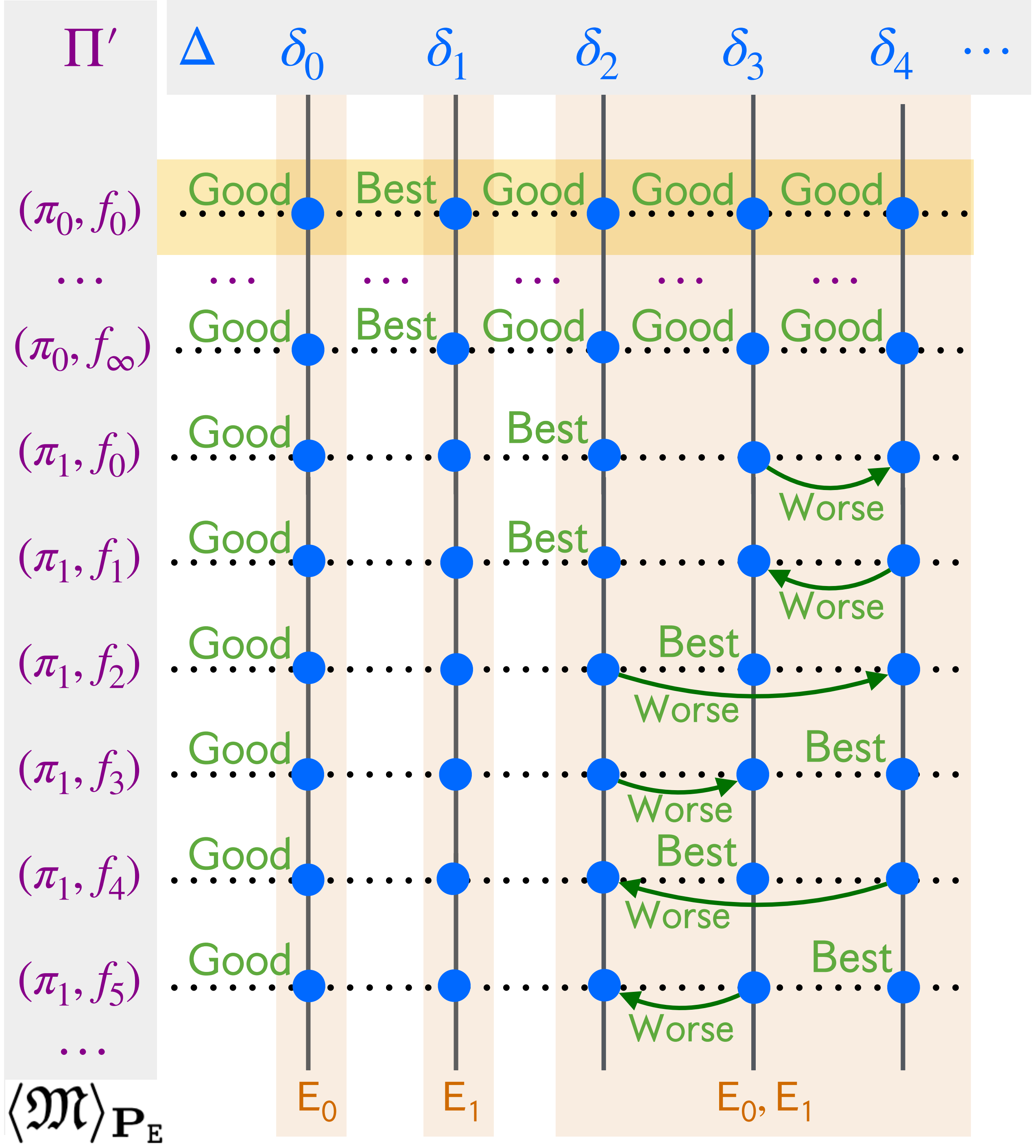
- $\exists^{\lhd n} x \,.\, \phi$ is evaluated on elements in the same L-layer
- $\Diamond\phi$ is evaluated on elements with the same E-type

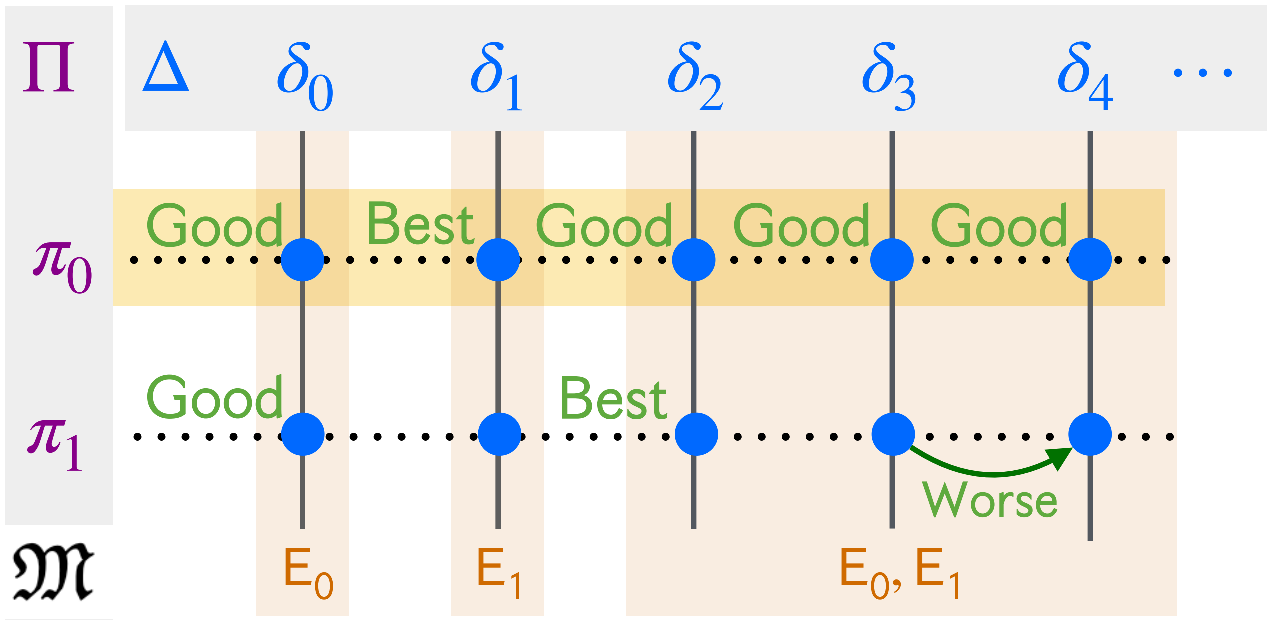
  If an element of an E-type satisfies ϕ, there are permutations so all elements satisfy $\Diamond\phi$

  If $\Diamond\phi$ is sentential, the variable assignment does nothing

$$\phi = \exists^{=1}x.(E_0) \land \forall x.(E_1) \land E_2 \text{ with}$$
$$E_0 = \Box_*(\texttt{Good}(x))$$
$$E_1 = E_0 \lor \Diamond_*(\texttt{Best}(x) \land \forall y.(\texttt{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\texttt{Good}(x) \lor \texttt{Best}(x))$$







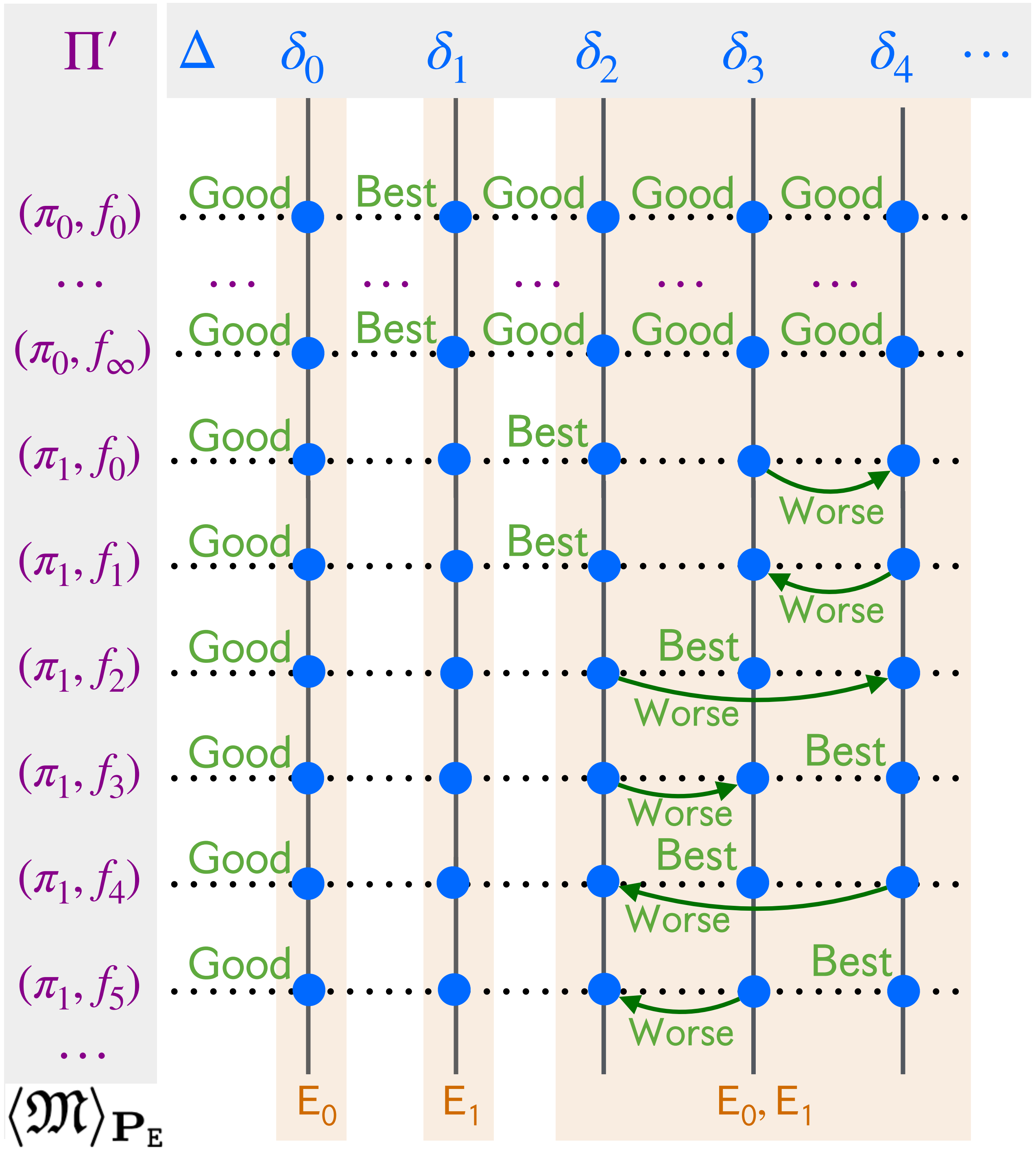
Translating to C^2

A translation maps a monodic C^2 FOSL formula ϕ to a plain C^2 formula

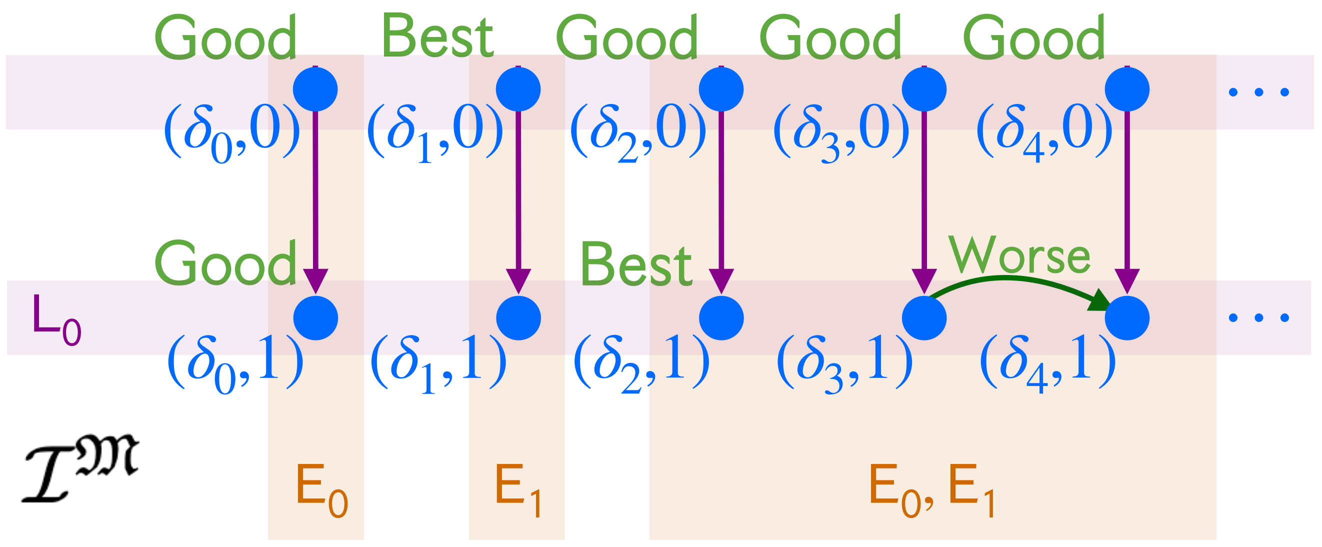
- $\exists^{\triangleleft n} x \,.\, \phi$ is evaluated on elements in the same L-layer
- $\Diamond\phi$ is evaluated on elements with the same E-type

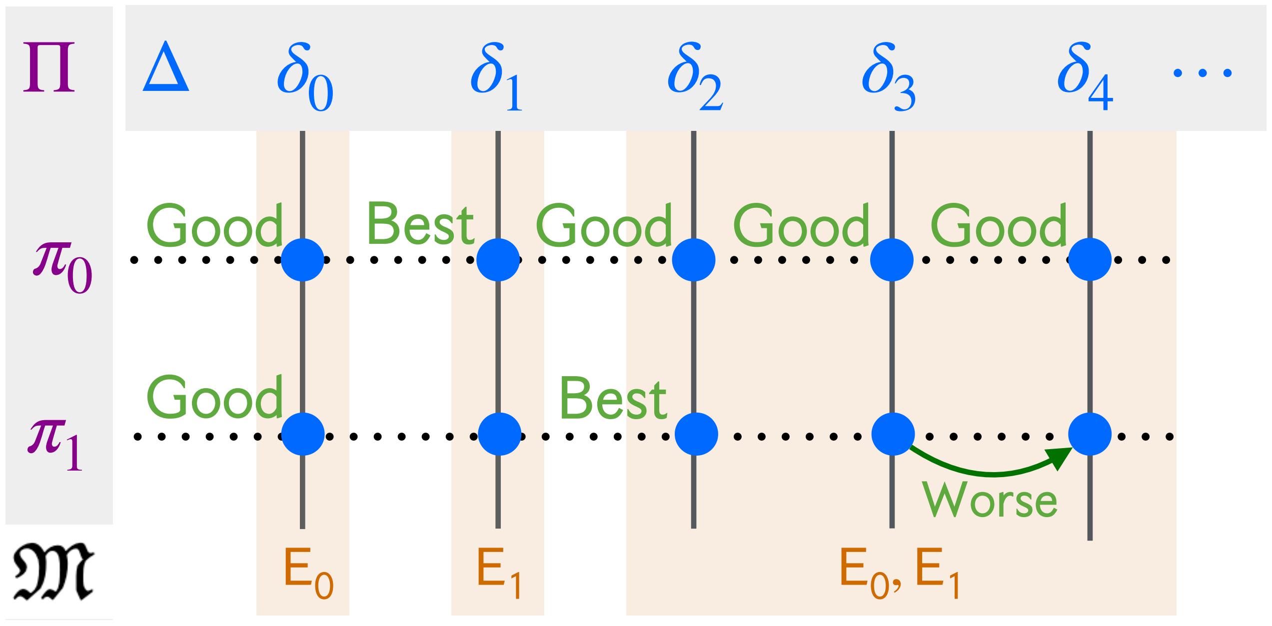
  If an element of an E-type satisfies ϕ, there are permutations so all elements satisfy $\Diamond\phi$

  If $\Diamond\phi$ is sentential, the variable assignment does nothing



$$\phi = \exists^{=1}x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$
$$E_0 = \Box_*(\mathsf{Good}(x))$$
$$E_1 = E_0 \vee \Diamond_*(\mathsf{Best}(x)) \wedge \forall y.(\mathsf{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\mathsf{Good}(x) \vee \mathsf{Best}(x))$$





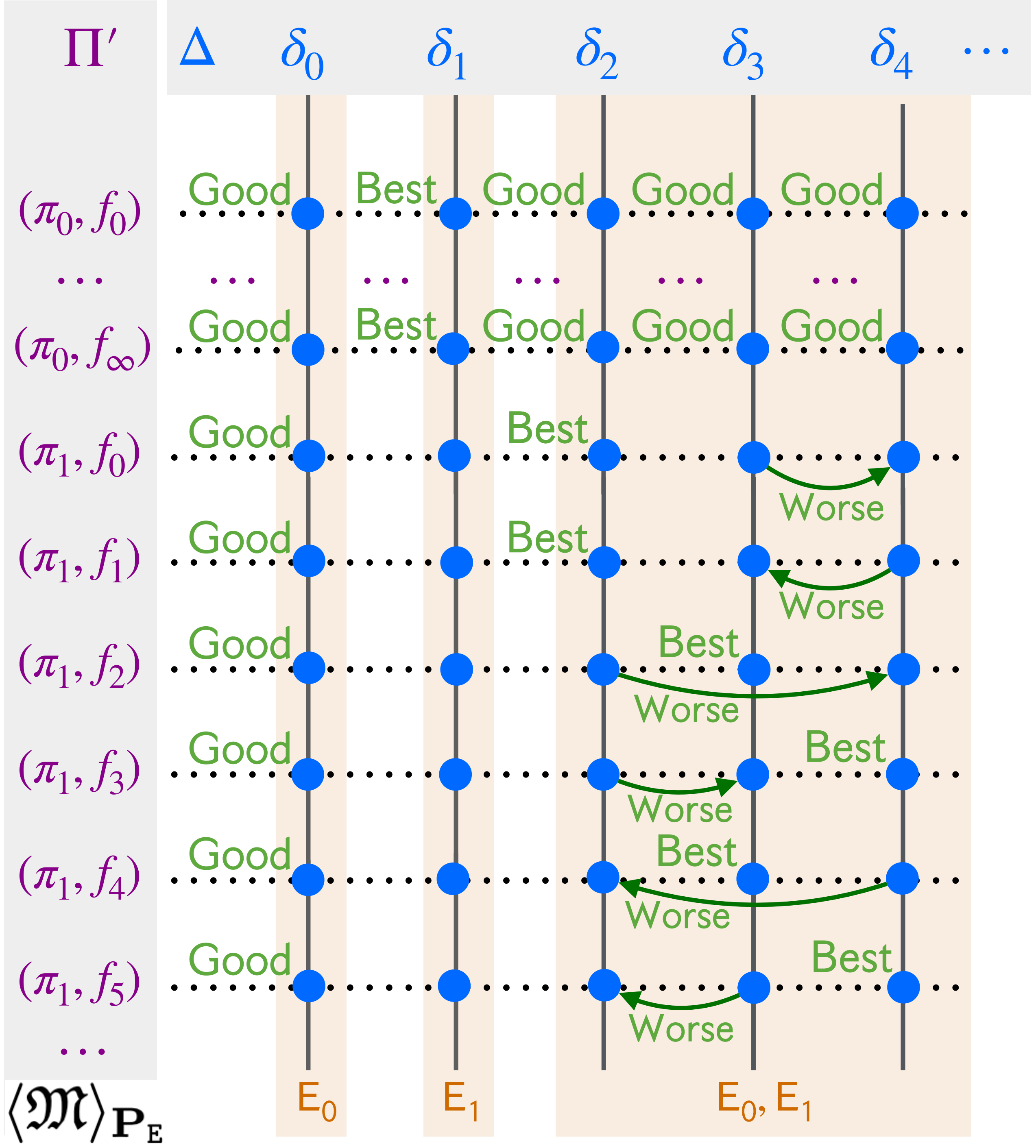
Translating to C^2

A translation maps a monodic C^2 FOSL formula ϕ to a plain C^2 formula

- $\exists^{\triangleleft n} x \,.\, \phi$ is evaluated on elements in the same L-layer
- $\Diamond\phi$ is evaluated on elements with the same E-type

  If an element of an E-type satisfies ϕ, there are permutations so all elements satisfy $\Diamond\phi$

  If $\Diamond\phi$ is sentential, the variable assignment does nothing

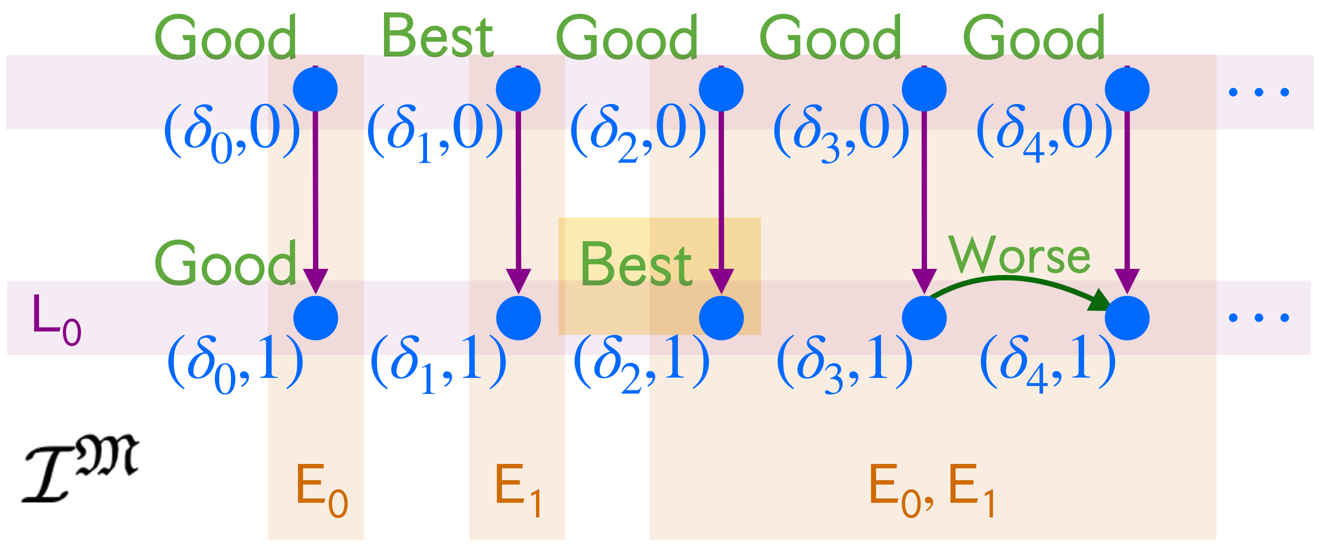


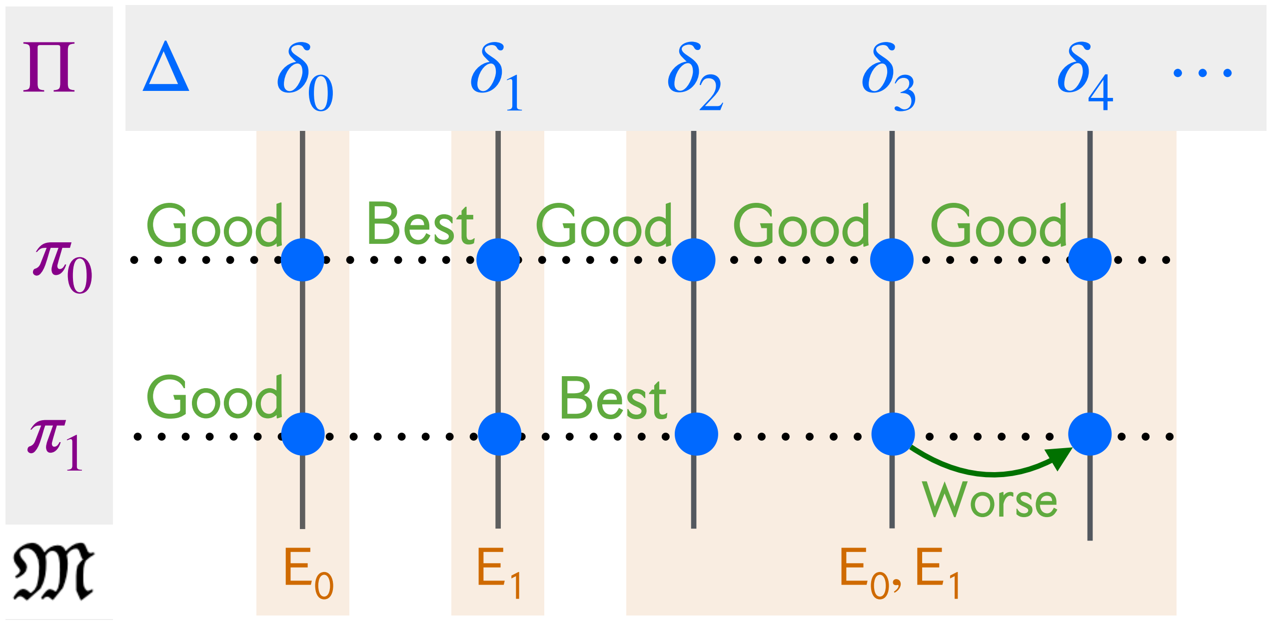
$$\phi = \exists^{=1}x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$

$$E_0 = \Box_*(\mathsf{Good}(x))$$

$$E_1 = E_0 \vee \Diamond_*(\mathsf{Best}(x)) \wedge \forall y.(\mathsf{Best}(y) \leftrightarrow x \doteq y))$$

$$E_2 = \Diamond_*(\forall x.\mathsf{Good}(x) \vee \mathsf{Best}(x))$$





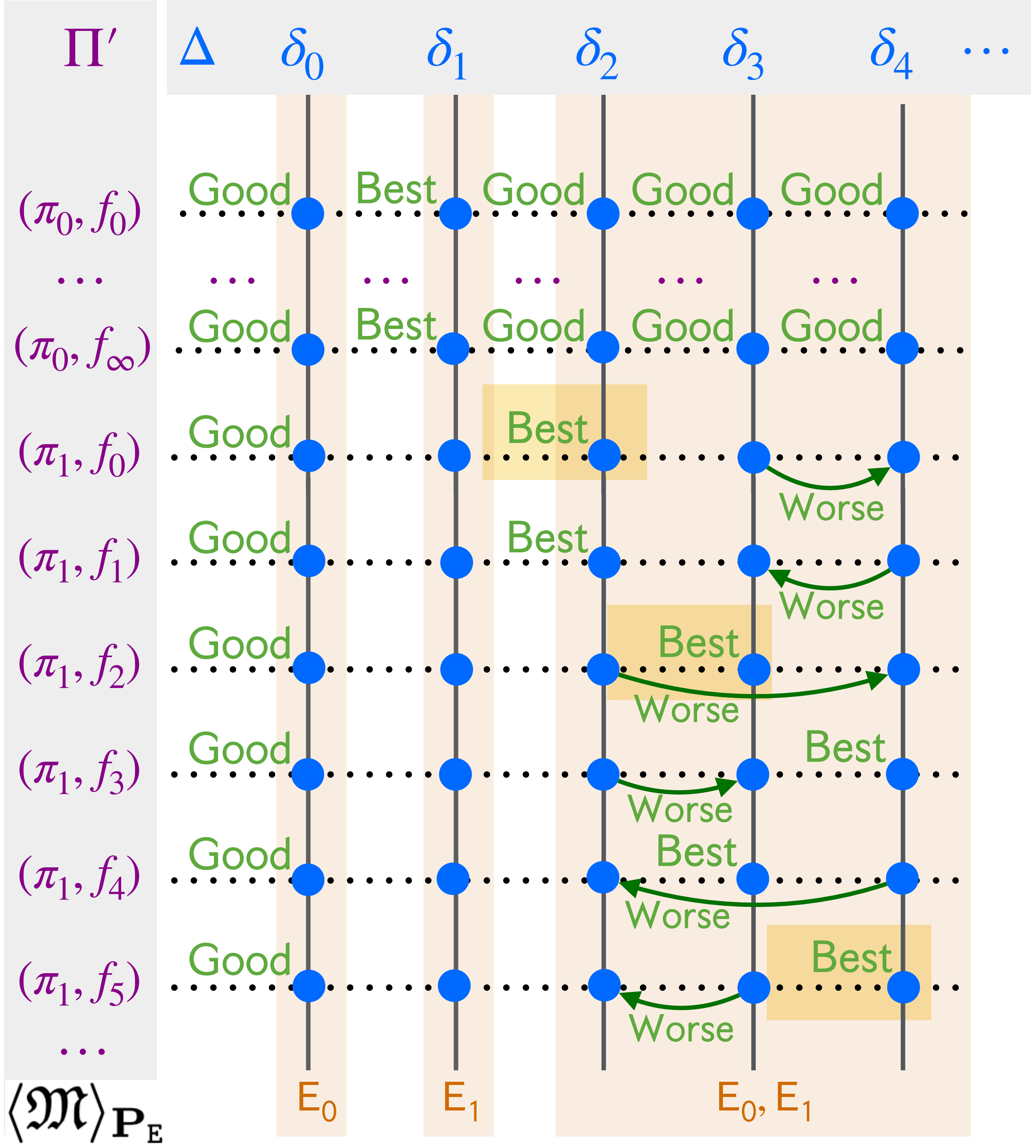
Translating to C^2

A translation maps a monodic C^2 FOSL formula ϕ to a plain C^2 formula

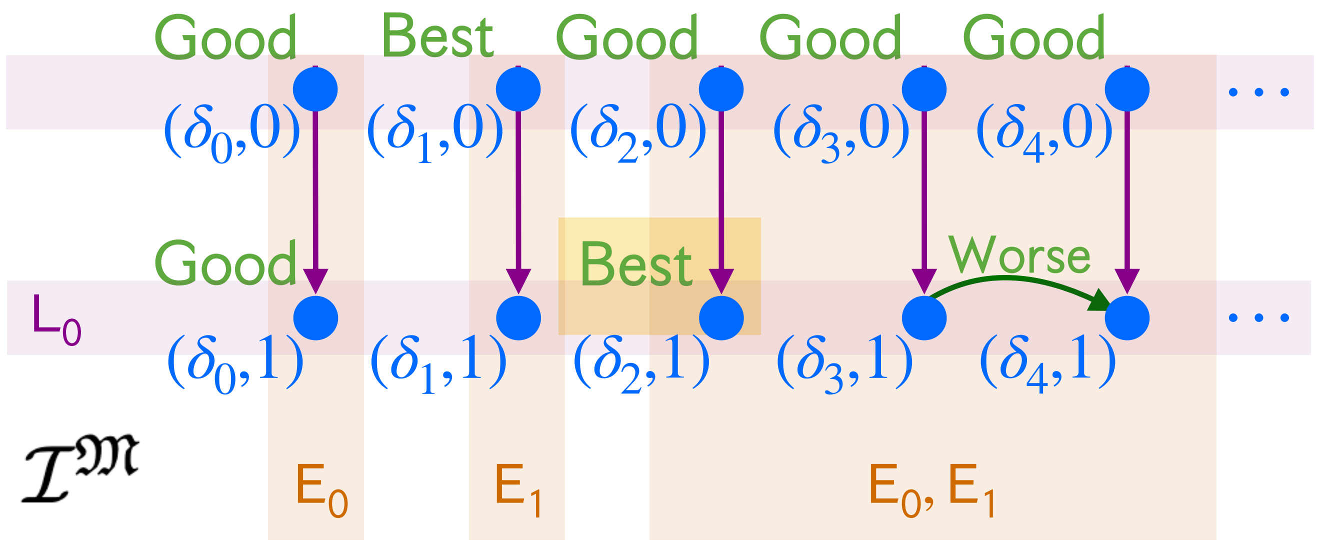
- $\exists^{\triangleleft n} x\,.\,\phi$ is evaluated on elements in the same L-layer
- $\Diamond\phi$ is evaluated on elements with the same E-type

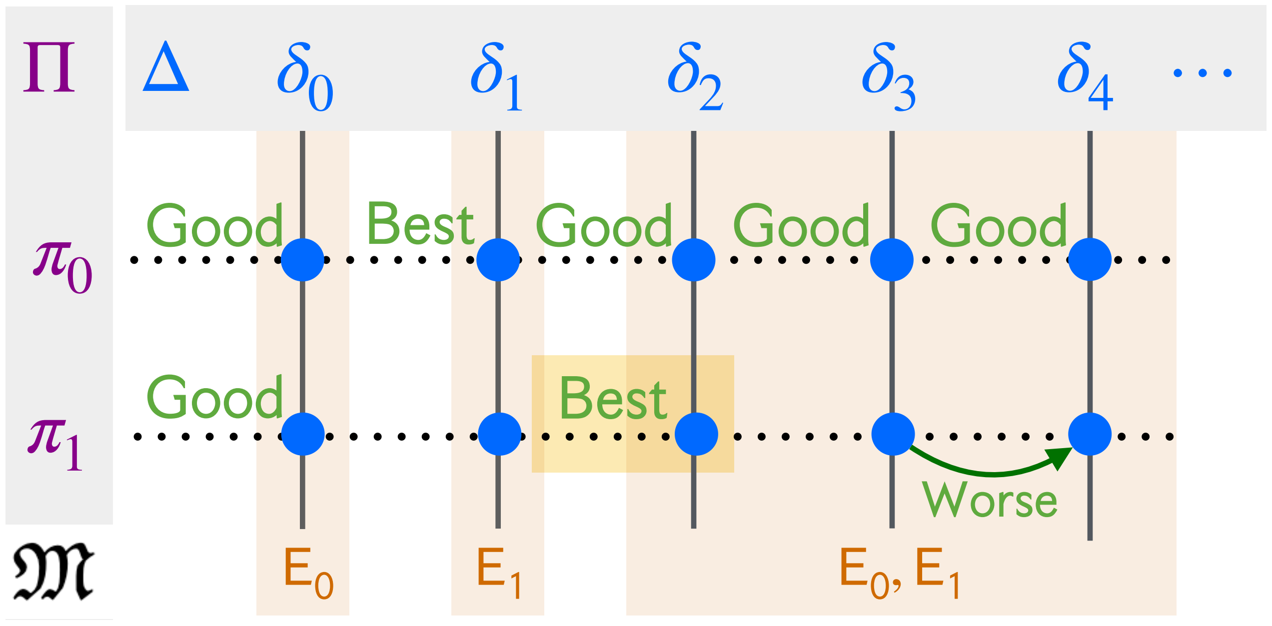
  If an element of an E-type satisfies ϕ, there are permutations so all elements satisfy $\Diamond\phi$

  If $\Diamond\phi$ is sentential, the variable assignment does nothing



$$\phi = \exists^{=1}x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$
$$E_0 = \Box_*(\mathsf{Good}(x))$$
$$E_1 = E_0 \vee \Diamond_*(\mathsf{Best}(x)) \wedge \forall y.(\mathsf{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\mathsf{Good}(x) \vee \mathsf{Best}(x))$$





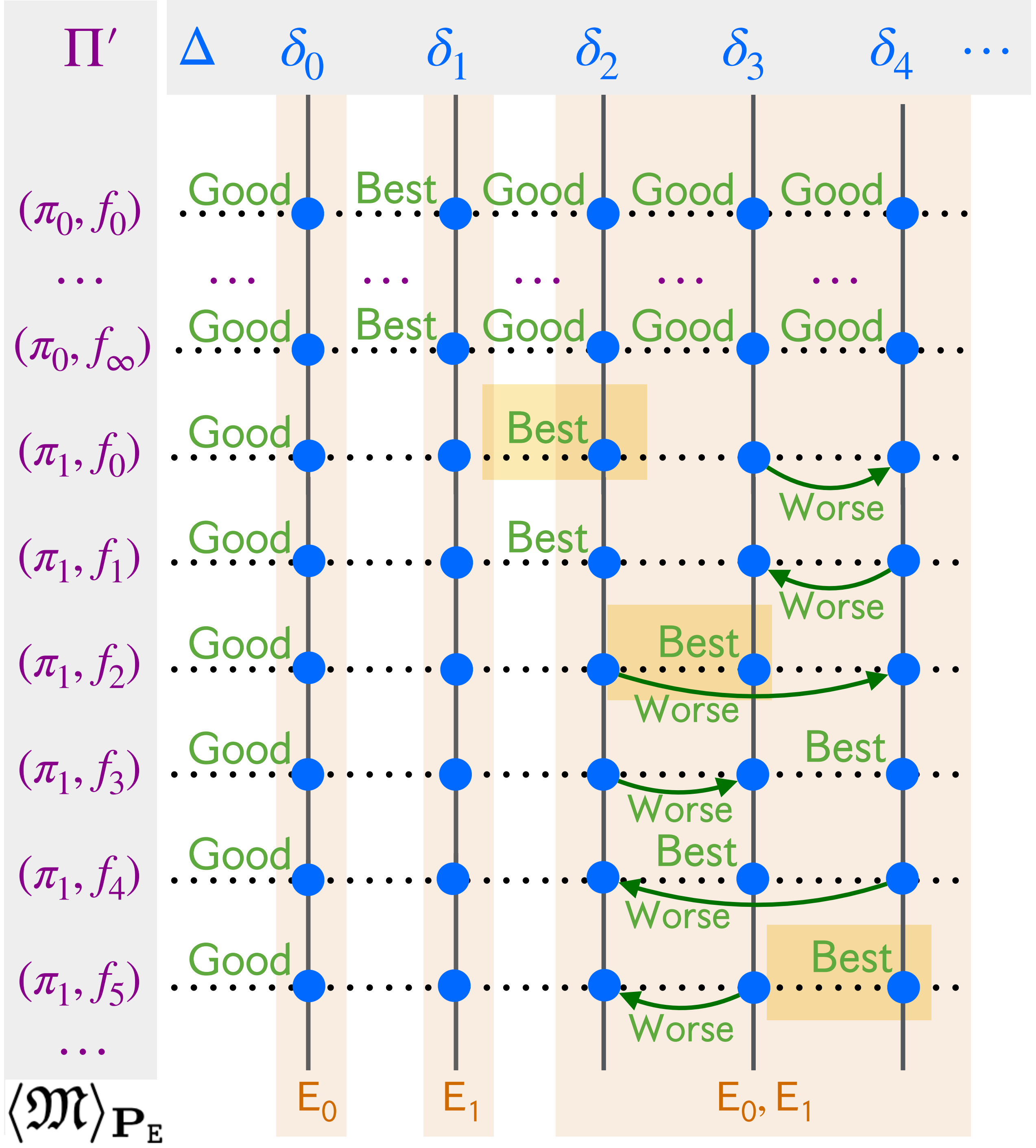
Translating to C^2

A translation maps a monodic C^2 FOSL formula ϕ to a plain C^2 formula

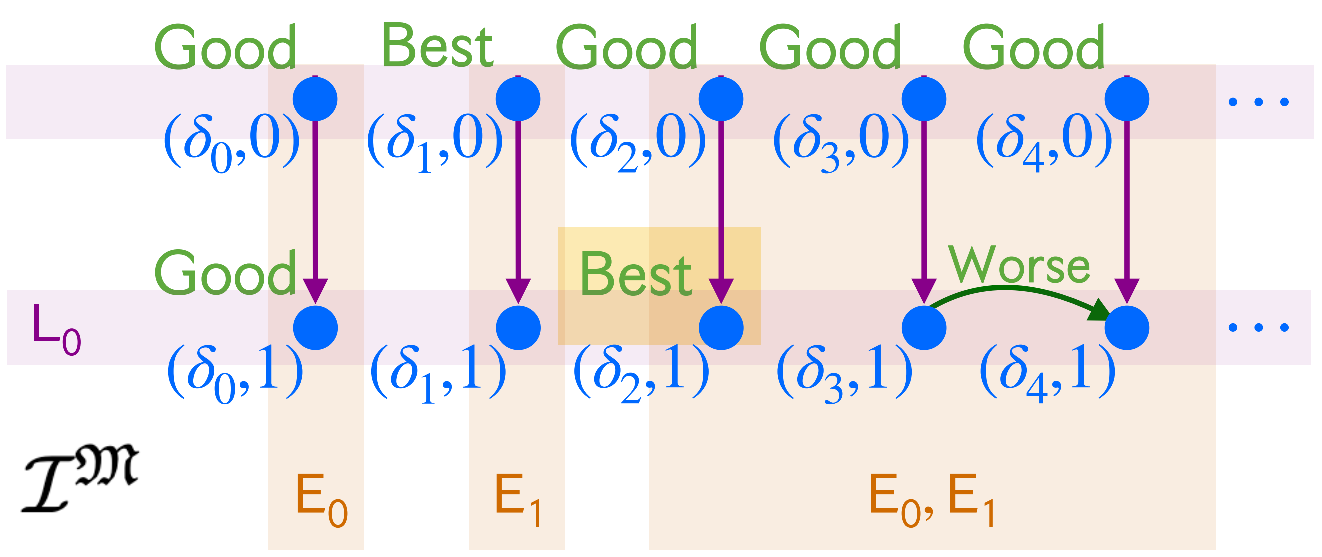
- $\exists^{\lhd n} x \, . \, \phi$ is evaluated on elements in the same L-layer
- $\Diamond \phi$ is evaluated on elements with the same E-type

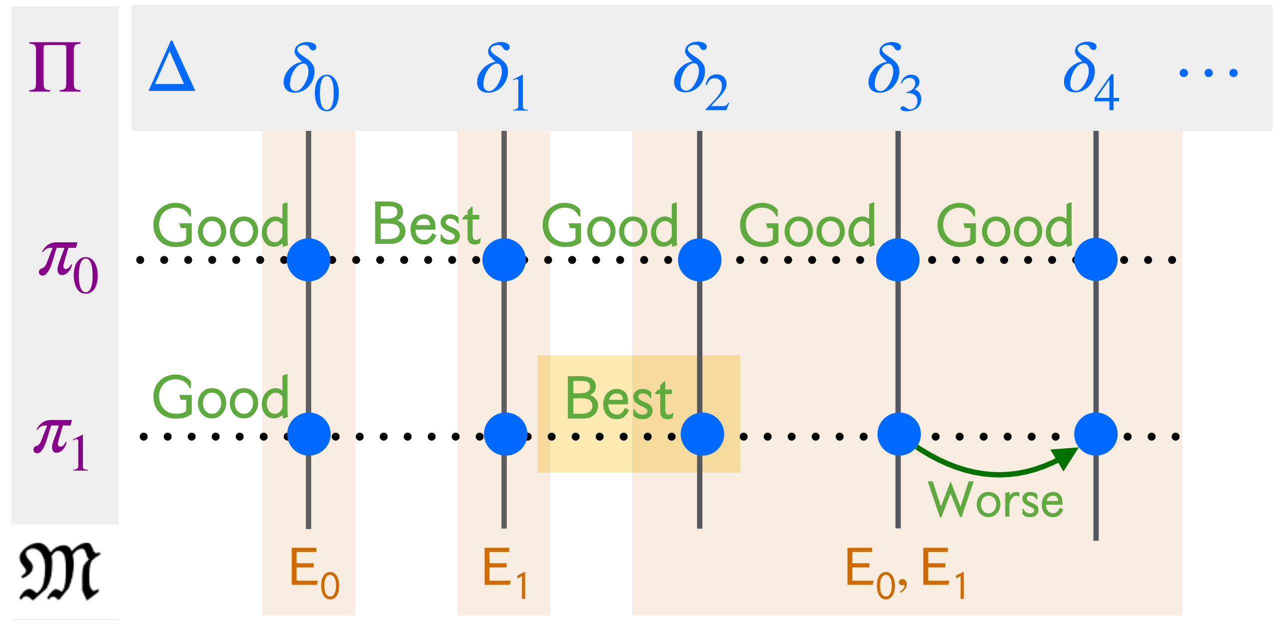
  If an element of an E-type satisfies ϕ, there are permutations so all elements satisfy $\Diamond \phi$

  If $\Diamond \phi$ is sentential, the variable assignment does nothing



$$
\begin{aligned}
\phi \; &= \exists^{=1} x.(E_0) \land \forall x.(E_1) \land E_2 \text{ with} \\
E_0 \; &= \Box_*(\mathtt{Good}(x)) \\
E_1 \; &= E_0 \lor \Diamond_*(\mathtt{Best}(x) \land \forall y.(\mathtt{Best}(y) \leftrightarrow x \doteq y)) \\
E_2 \; &= \Diamond_*(\forall x.\mathtt{Good}(x) \lor \mathtt{Best}(x))
\end{aligned}
$$





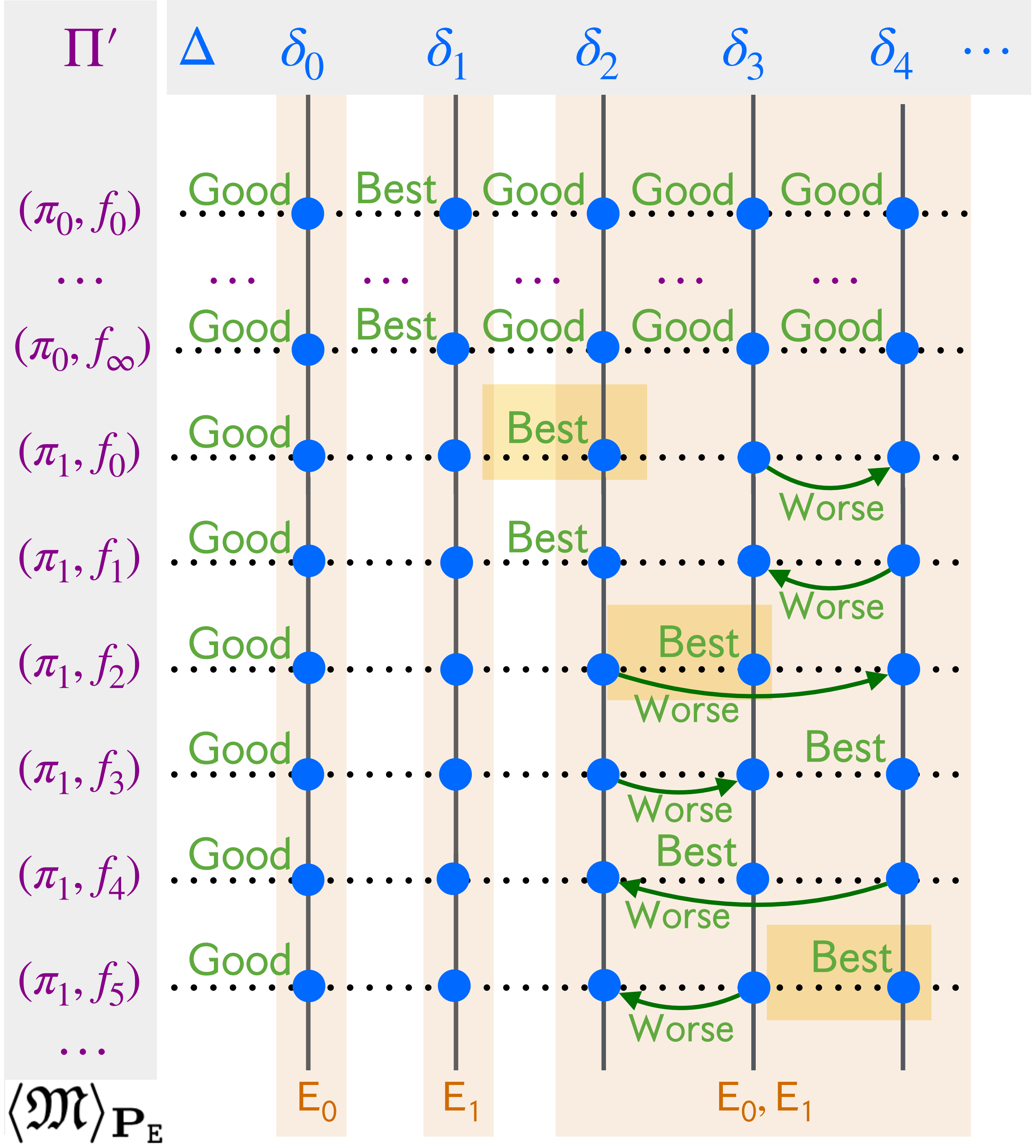
Translating to C^2

A translation maps a monodic C^2 FOSL formula ϕ to a plain C^2 formula

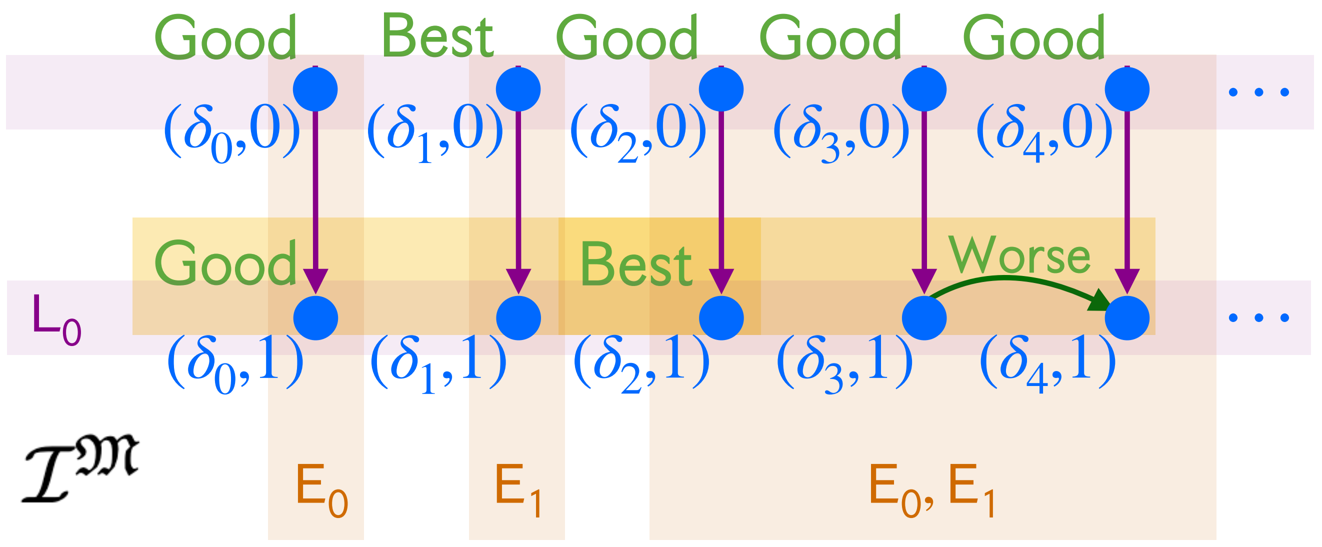
- $\exists^{\triangleleft n} x \,.\, \phi$ is evaluated on elements in the same L-layer
- $\Diamond\phi$ is evaluated on elements with the same E-type

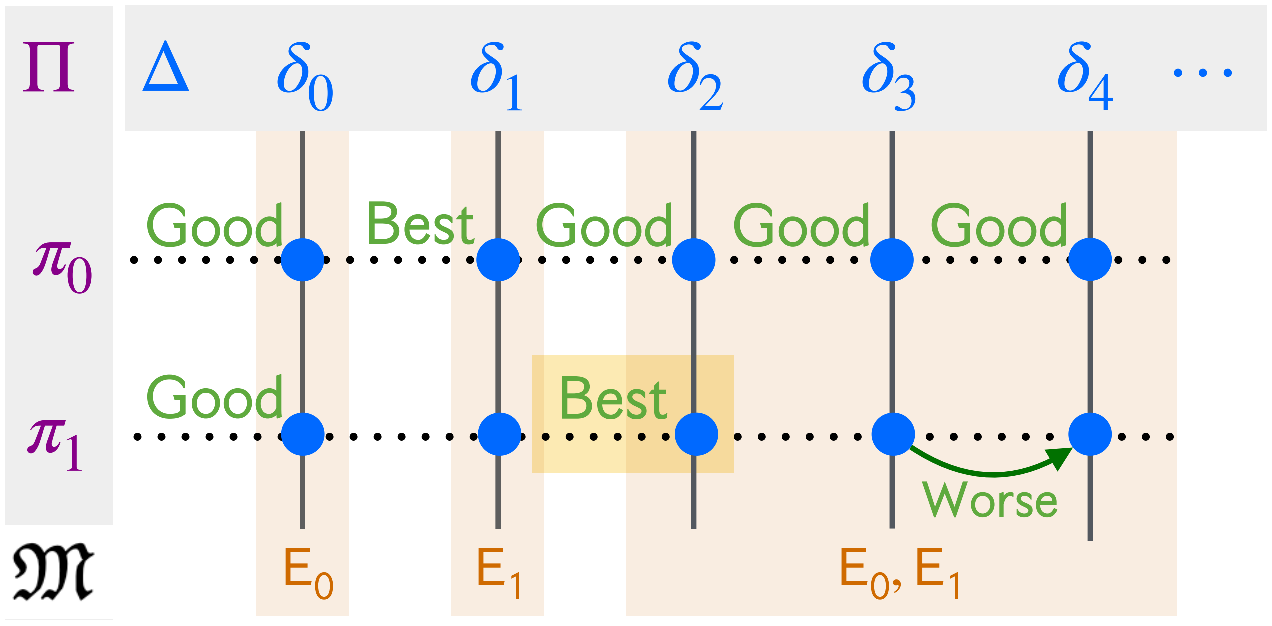
  If an element of an E-type satisfies ϕ, there are permutations so all elements satisfy $\Diamond\phi$

  If $\Diamond\phi$ is sentential, the variable assignment does nothing



$$
\begin{aligned}
\phi &= \exists^{=1}x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with} \\
E_0 &= \Box_*(\mathsf{Good}(x)) \\
E_1 &= E_0 \vee \Diamond_*(\mathtt{Best}(x) \wedge \forall y.(\mathtt{Best}(y) \leftrightarrow x \doteq y)) \\
E_2 &= \Diamond_*(\forall x.\mathsf{Good}(x) \vee \mathtt{Best}(x))
\end{aligned}
$$





Translating to C^2

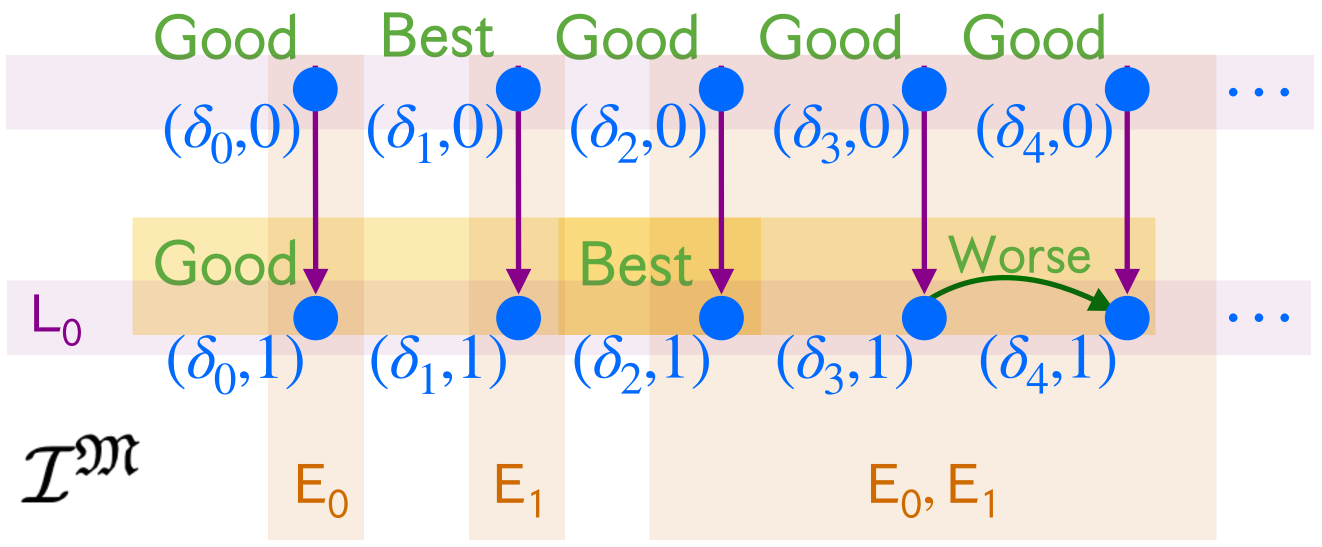
A translation maps a monodic C^2 FOSL formula ϕ to a plain C^2 formula

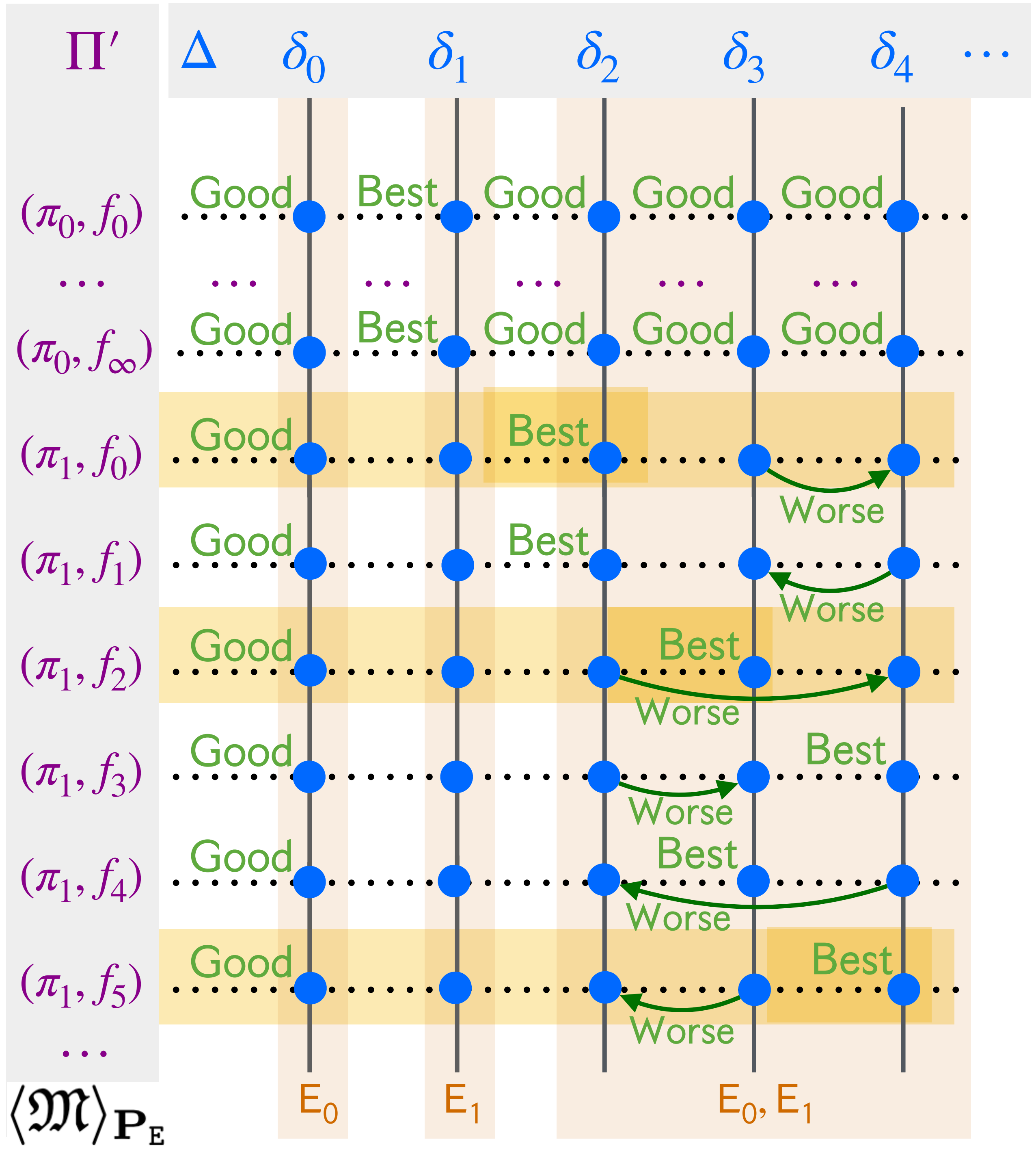
- $\exists^{\lhd n} x \,.\, \phi$ is evaluated on elements in the same L-layer
- $\Diamond\phi$ is evaluated on elements with the same E-type

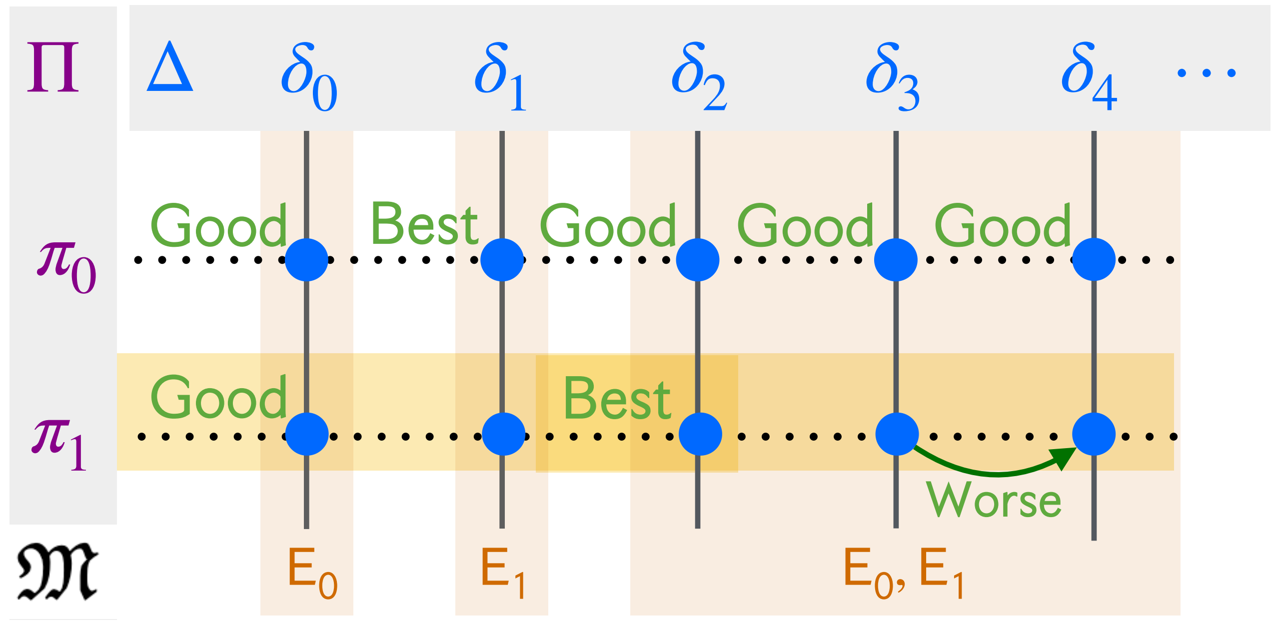
If an element of an E-type satisfies ϕ, there are permutations so all elements satisfy $\Diamond\phi$

If $\Diamond\phi$ is sentential, the variable assignment does nothing

$$\phi = \exists^{=1}x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$
$$E_0 = \Box_*(\mathtt{Good}(x))$$
$$E_1 = E_0 \vee \Diamond_*(\mathtt{Best}(x) \wedge \forall y.(\mathtt{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\mathtt{Good}(x) \vee \mathtt{Best}(x))$$







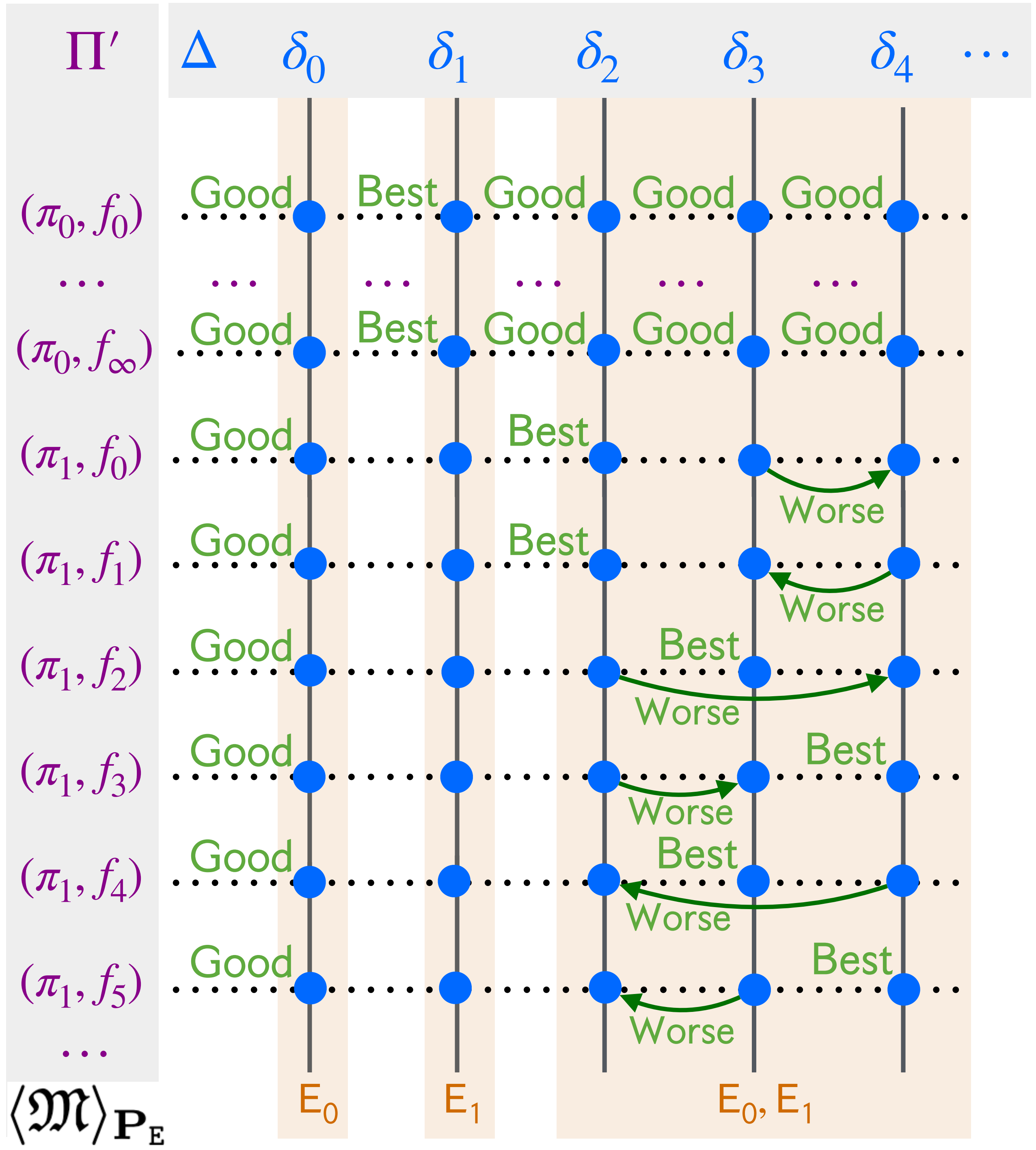
Translating to C^2

A translation maps a monodic C^2 FOSL formula ϕ to a plain C^2 formula

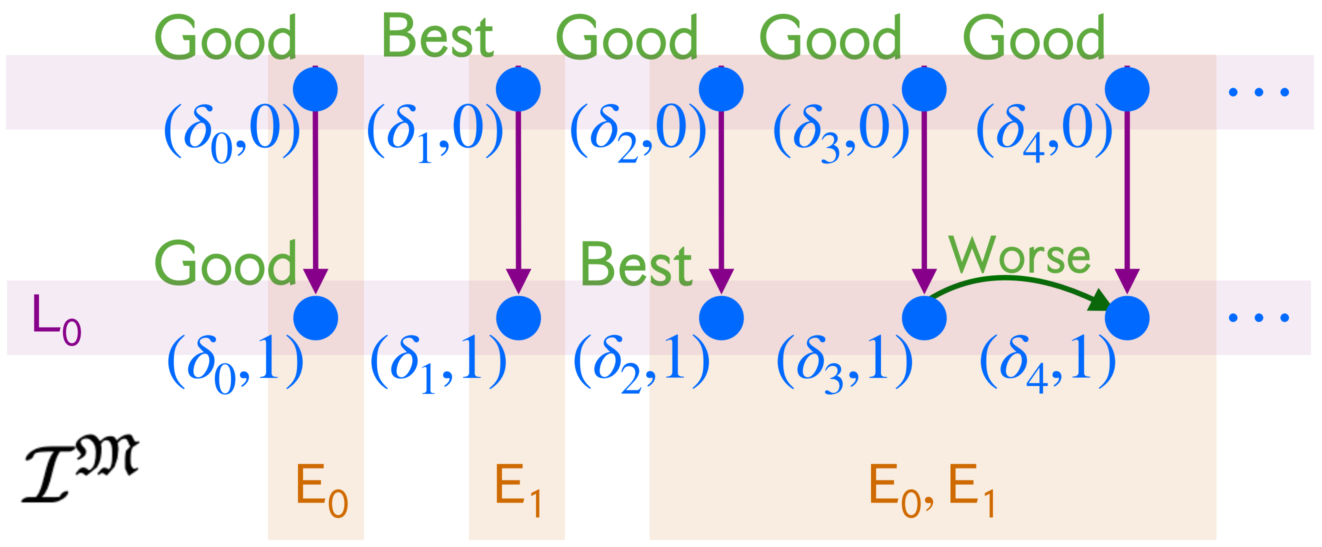
- $\exists^{\triangleleft n} x \,.\, \phi$ is evaluated on elements in the same L-layer
- $\Diamond\phi$ is evaluated on elements with the same E-type

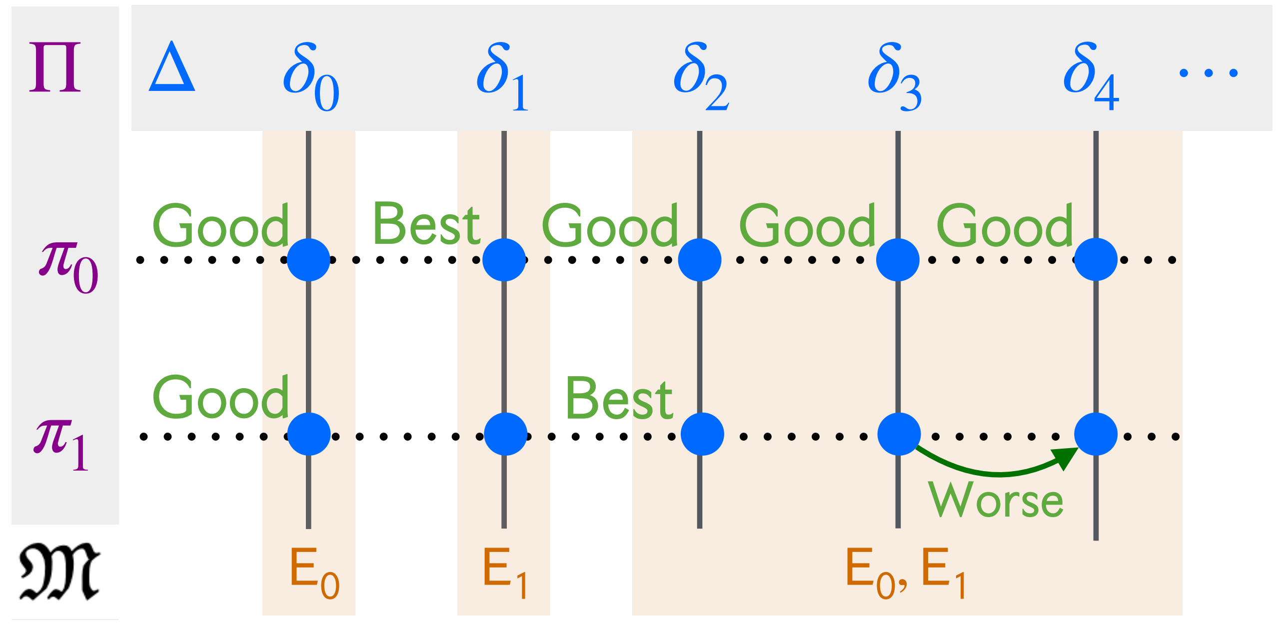
  If an element of an E-type satisfies ϕ, there are permutations so all elements satisfy $\Diamond\phi$

  If $\Diamond\phi$ is sentential, the variable assignment does nothing



$$\phi = \exists^{=1}x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$
$$E_0 = \Box_*(\mathtt{Good}(x))$$
$$E_1 = E_0 \vee \Diamond_*(\mathtt{Best}(x) \wedge \forall y.(\mathtt{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\mathtt{Good}(x) \vee \mathtt{Best}(x))$$





Translating to C^2

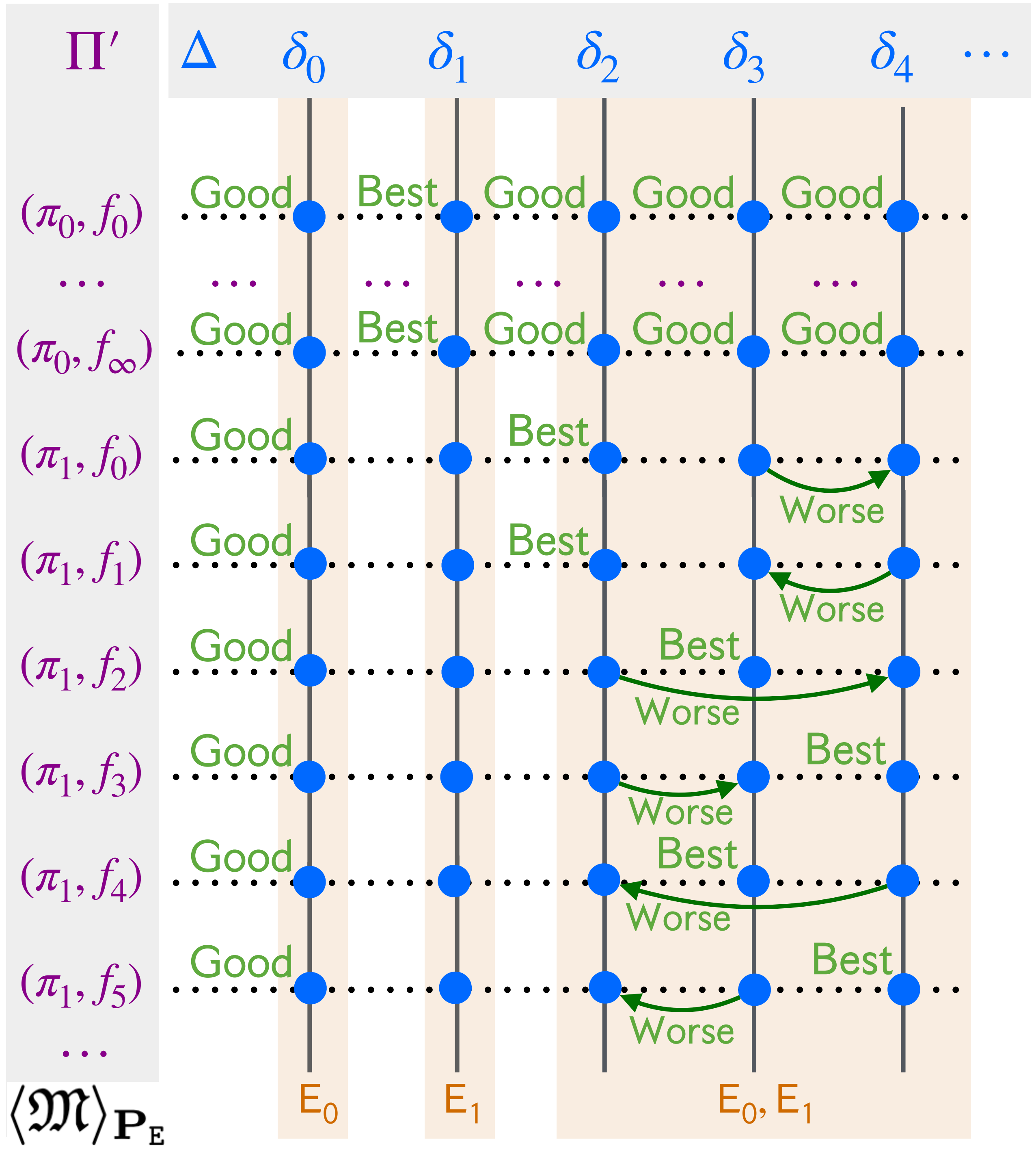
A translation maps a monodic C^2 FOSL formula ϕ to a plain C^2 formula

- $\exists^{\lhd n} x \,.\, \phi$ is evaluated on elements in the same L-layer
- $\Diamond\phi$ is evaluated on elements with the same E-type

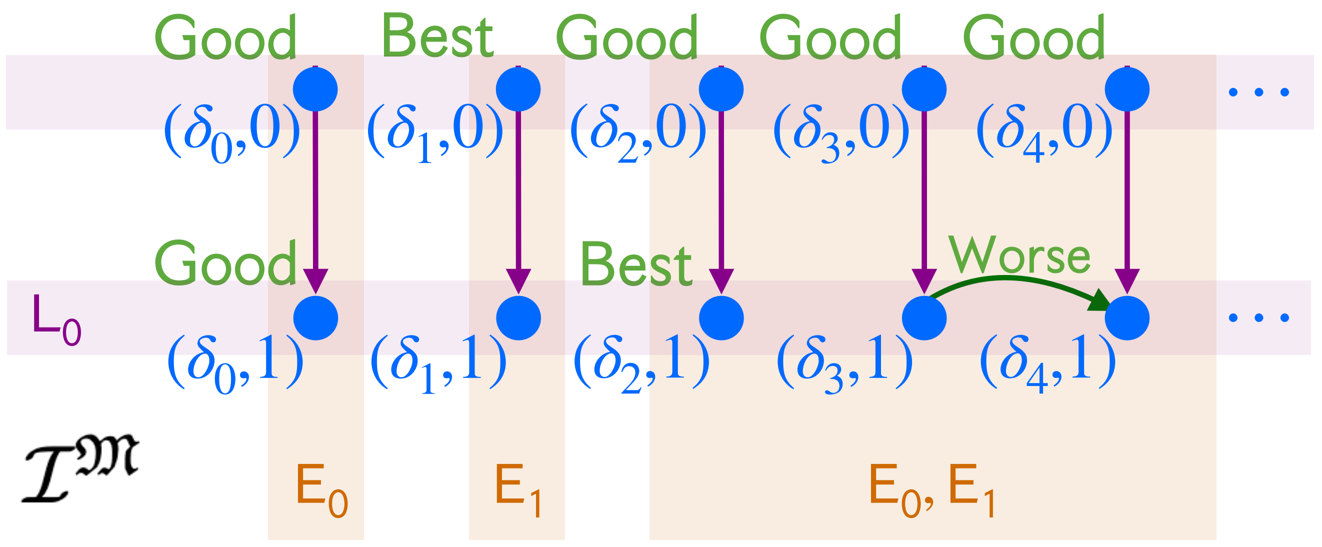
If an element of an E-type satisfies ϕ, there are permutations so all elements satisfy $\Diamond\phi$

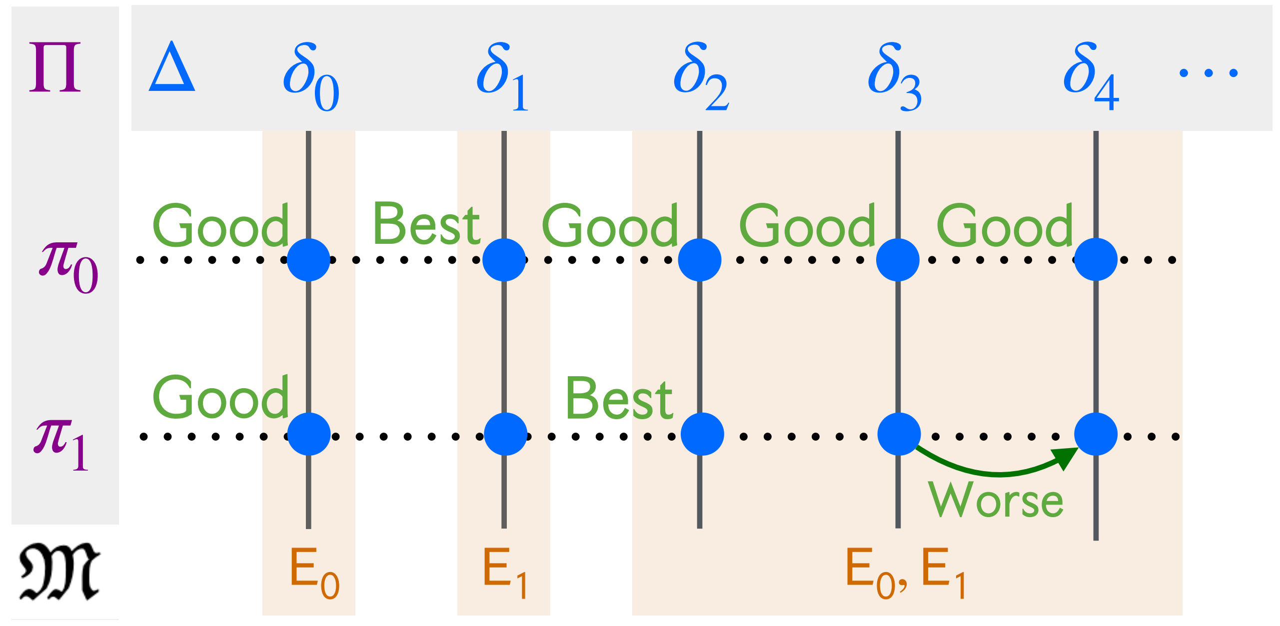
If $\Diamond\phi$ is sentential, the variable assignment does nothing

ϕ and its C^2 translation are equisatisfiable



$$\phi = \exists^{=1} x.(E_0) \wedge \forall x.(E_1) \wedge E_2 \text{ with}$$
$$E_0 = \Box_*(\mathtt{Good}(x))$$
$$E_1 = E_0 \vee \Diamond_*(\mathtt{Best}(x) \wedge \forall y.(\mathtt{Best}(y) \leftrightarrow x \doteq y))$$
$$E_2 = \Diamond_*(\forall x.\mathtt{Good}(x) \vee \mathtt{Best}(x))$$





Translating to C^2

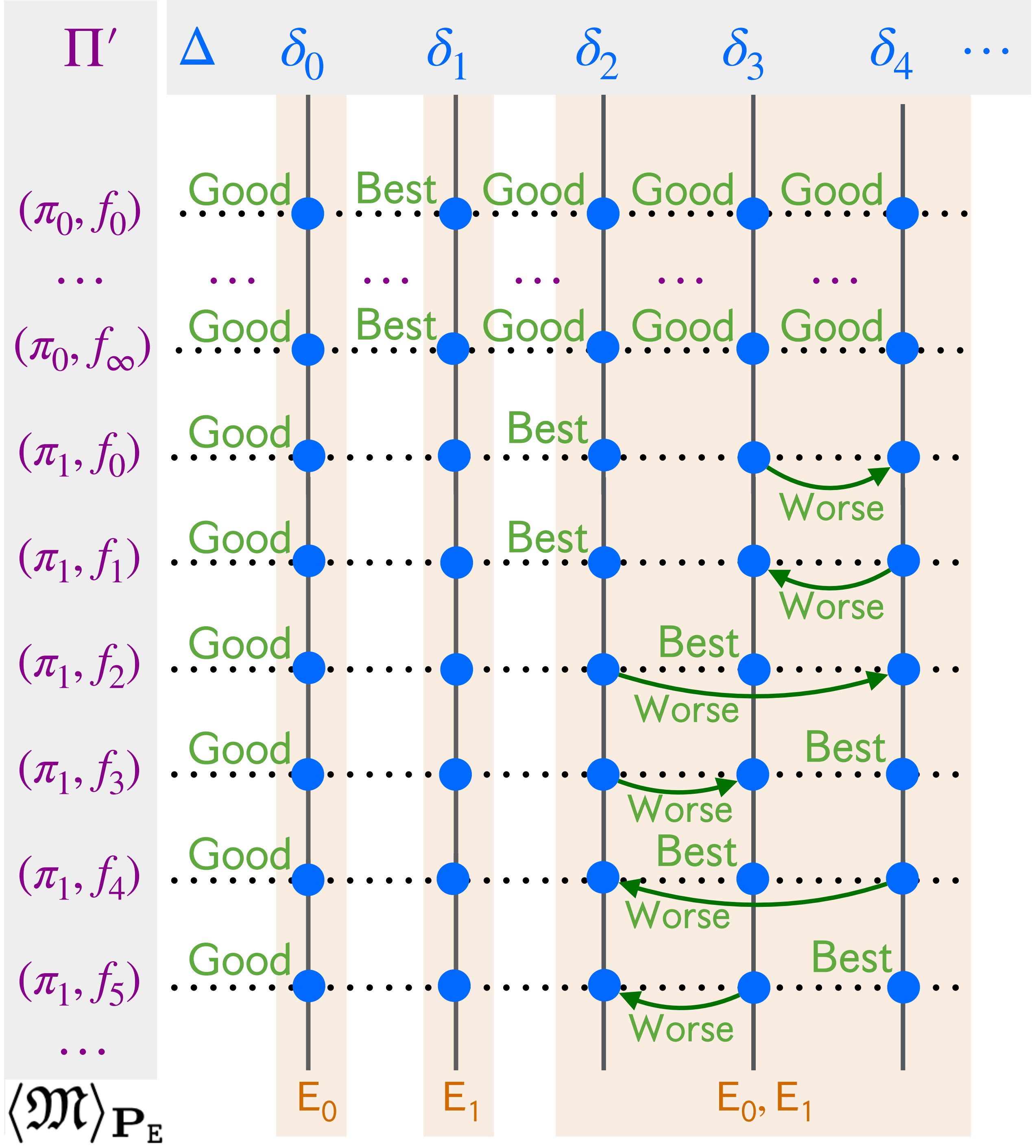
A translation maps a monodic C^2 FOSL formula ϕ to a plain C^2 formula

- $\exists^{\triangleleft n} x \,.\, \phi$ is evaluated on elements in the same L-layer
- $\Diamond\phi$ is evaluated on elements with the same E-type

  If an element of an E-type satisfies ϕ, there are permutations so all elements satisfy $\Diamond\phi$

  If $\Diamond\phi$ is sentential, the variable assignment does nothing

ϕ and its C^2 translation are equisatisfiable



$\phi = \exists^{=1}x.(E_0) \wedge \forall x.(E_1) \wedge$

$E_0 = \Box_*(\mathtt{Good}(x))$

$E_1 = E_0 \vee \Diamond_*(\mathtt{Best}(x) \wedge \forall y.(\mathtt{Bes}$

$E_2 = \Diamond_*(\forall x.\mathtt{Good}(x) \vee \mathtt{Best}(x))$

Corollary: ***Satisfiability in monodic standpoint C^2 is*** NExpTime-***complete***