# Logic-Based Explainable Artificial Intelligence

Joao Marques-Silva

ICREA & Univ. Lleida, Catalunya, Spain

ESSLLI, Bochum, Germany, July 2025

# Lecture 02

# Recapitulate first lecture

- ML models: classification & regression

# Recapitulate first lecture

- ML models: classification & regression
- Glimpse of heuristic XAI

# Recapitulate first lecture

- ML models: classification & regression
- Glimpse of heuristic XAI
- Answers to Why? questions as logic rules

# Recapitulate first lecture

- ML models: classification & regression
- Glimpse of heuristic XAI
- Answers to Why? questions as logic rules
- Logic-based reasoning of ML models

# Recapitulate first lecture

- ML models: classification & regression
- Glimpse of heuristic XAI
- Answers to Why? questions as logic rules
- Logic-based reasoning of ML models
- Apparent difficulties with explaining interpretable models

# Plan for this course

- Lecture 01 – unit(s):
  - #01: Foundations
- Lecture 02 – unit(s):
  - #02: Principles of symbolic XAI – feature selection
  - #03: Tractability in symbolic XAI (& myth of interpretability)
- Lecture 03 – unit(s):
  - #04: Intractability in symbolic XAI (& myth of model-agnostic XAI)
  - #05: Explainability queries
- Lecture 04 – unit(s):
  - #06: Recent, emerging & advanced topics
- Lecture 05 – unit(s):
  - #07: Principles of symbolic XAI – feature attribution (& myth of Shapley values in XAI)
  - #08: Corrected feature attribution – nuSHAP
  - #09: Conclusions & research directions

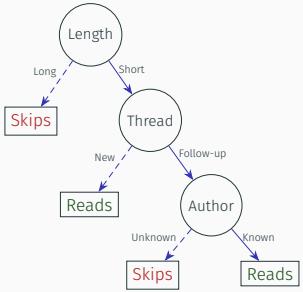
Unit #02

# Principles of Symbolic XAI – Feature Selection

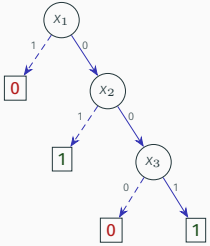
# What is an explanation?

- Notation:

Original DT [PM17]




Rewritten DT



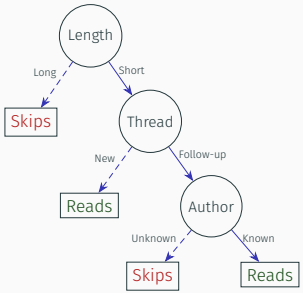

| Mapping |
|---|
| $x_1 = 1$ iff Length $=$ Long |
| $x_2 = 1$ iff Thread $=$ New |
| $x_3 = 1$ iff Author $=$ Known |
| $\kappa(\cdot) = 1$ iff $\kappa'(\cdots) =$ Reads |
| $\kappa(\cdot) = 0$ iff $\kappa'(\cdots) =$ Skips |

- What is an explanation?

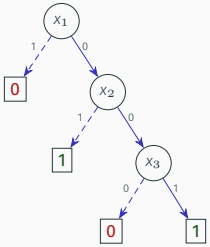
# What is an explanation?

- Notation:

Original DT [PM17]




Rewritten DT



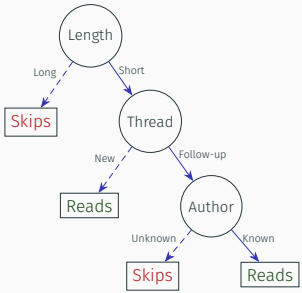

| Mapping |
|---|
| $x_1 = 1$ iff Length $=$ Long |
| $x_2 = 1$ iff Thread $=$ New |
| $x_3 = 1$ iff Author $=$ Known |
| $\kappa(\cdot) = 1$ iff $\kappa'(\cdots) =$ Reads |
| $\kappa(\cdot) = 0$ iff $\kappa'(\cdots) =$ Skips |

- What is an explanation?
  - Answer to question "**Why** (the prediction)?" is a **rule**: IF <COND> THEN $\kappa(\mathbf{x}) = c$

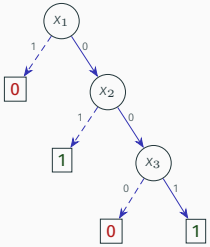
# What is an explanation?

- Notation:

Original DT [PM17]




Rewritten DT



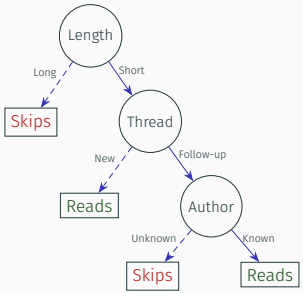

| Mapping |
|---|
| $x_1 = 1$ iff Length = Long |
| $x_2 = 1$ iff Thread = New |
| $x_3 = 1$ iff Author = Known |
| $\kappa(\cdot) = 1$ iff $\kappa'(\cdots)$ = Reads |
| $\kappa(\cdot) = 0$ iff $\kappa'(\cdots)$ = Skips |

- What is an explanation?
  - Answer to question "**Why** (the prediction)?" is a **rule**: IF <COND> THEN $\kappa(\mathbf{x}) = c$
  - **Explanation**: set of literals (or just features) in <COND>; **irreducibility matters!**

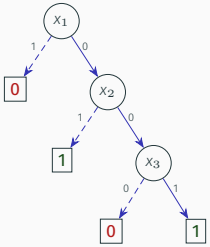
# What is an explanation?

- Notation:

Original DT [PM17]




Rewritten DT



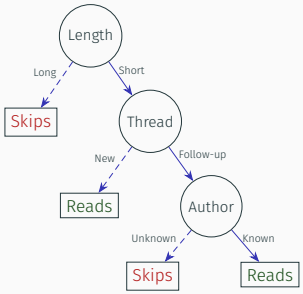

| Mapping |
|---|
| $x_1 = 1$ iff Length $=$ Long |
| $x_2 = 1$ iff Thread $=$ New |
| $x_3 = 1$ iff Author $=$ Known |
| $\kappa(\cdot) = 1$ iff $\kappa'(\cdots) =$ Reads |
| $\kappa(\cdot) = 0$ iff $\kappa'(\cdots) =$ Skips |

- What is an explanation?
  - Answer to question "**Why** (the prediction)?" is a **rule**: IF <COND> THEN $\kappa(\mathbf{x}) = c$
  - **Explanation**: set of literals (or just features) in <COND>; **irreducibility matters!**
  - **E.g.**: explanation for $\mathbf{v} = (\neg x_1, \neg x_2, x_3)$?

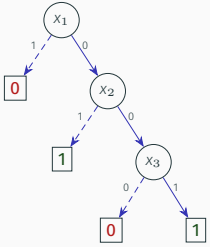
# What is an explanation?

- Notation:

Original DT [PM17]




Rewritten DT



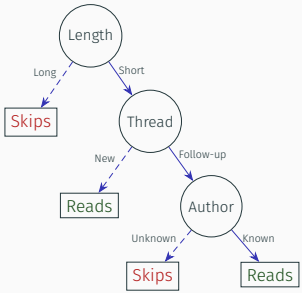

| Mapping |
|---|
| $x_1 = 1$ iff Length $=$ Long |
| $x_2 = 1$ iff Thread $=$ New |
| $x_3 = 1$ iff Author $=$ Known |
| $\kappa(\cdot) = 1$ iff $\kappa'(\cdots) =$ Reads |
| $\kappa(\cdot) = 0$ iff $\kappa'(\cdots) =$ Skips |

- What is an explanation?
  - Answer to question "**Why** (the prediction)?" is a **rule**: IF <COND> THEN $\kappa(\mathbf{x}) = c$
  - **Explanation**: set of literals (or just features) in <COND>; **irreducibility matters!**
  - **E.g.**: explanation for $\mathbf{v} = (\neg x_1, \neg x_2, x_3)$?
    - It is the case that, IF $\neg x_1 \land \neg x_2 \land x_3$ THEN $\kappa(\mathbf{x}) = 1$

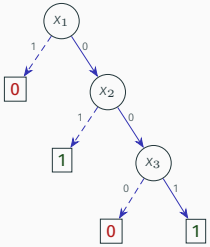
# What is an explanation?

- Notation:

Original DT [PM17]




Rewritten DT




| Mapping |
|---|
| $x_1 = 1$ iff Length $=$ Long |
| $x_2 = 1$ iff Thread $=$ New |
| $x_3 = 1$ iff Author $=$ Known |
| $\kappa(\cdot) = 1$ iff $\kappa'(\cdots) =$ Reads |
| $\kappa(\cdot) = 0$ iff $\kappa'(\cdots) =$ Skips |

- What is an explanation?
  - Answer to question "**Why** (the prediction)?" is a **rule**: IF <COND> THEN $\kappa(\mathbf{x}) = c$
  - **Explanation**: set of literals (or just features) in <COND>; **irreducibility matters!**
  - **E.g.**: explanation for $\mathbf{v} = (\neg x_1, \neg x_2, x_3)$?
    - It is the case that, IF $\neg x_1 \land \neg x_2 \land x_3$ THEN $\kappa(\mathbf{x}) = 1$
    - One possible explanation is $\{\neg x_1, \neg x_2, x_3\}$ or simply $\{1, 2, 3\}$

# The similarity predicate

[Mar24]

- Recall ML models for classification & regression:
  - Classification: $\mathcal{M}_C = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$
  - Regression: $\mathcal{M}_R = (\mathcal{F}, \mathbb{F}, \mathbb{V}, \rho)$
  - General: $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathbb{T}, \tau)$

# The similarity predicate

[Mar24]

- Recall ML models for classification & regression:
  - Classification: $\mathcal{M}_C = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$
  - Regression: $\mathcal{M}_R = (\mathcal{F}, \mathbb{F}, \mathbb{V}, \rho)$
  - General: $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathbb{T}, \tau)$

- Similarity predicate: $\sigma : \mathbb{F} \to \{\top, \bot\}$
  - Classification: $\sigma(\mathbf{x}) \coloneqq [\kappa(\mathbf{x}) = \kappa(\mathbf{v})]$
    - **Obs:** For boolean classifiers, no need for $\sigma$
  - Regression: $\sigma(\mathbf{x}) \coloneqq [|\rho(\mathbf{x}) - \rho(\mathbf{v})| \leqslant \delta]$, where $\delta$ is user-specified

# The similarity predicate

[Mar24]

- Recall ML models for classification & regression:
  - Classification: $\mathcal{M}_C = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$
  - Regression: $\mathcal{M}_R = (\mathcal{F}, \mathbb{F}, \mathbb{V}, \rho)$
  - General: $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathbb{T}, \tau)$

- Similarity predicate: $\sigma : \mathbb{F} \to \{\top, \bot\}$
  - Classification: $\sigma(\mathbf{x}) \coloneqq [\kappa(\mathbf{x}) = \kappa(\mathbf{v})]$
    - **Obs:** For boolean classifiers, no need for $\sigma$
  - Regression: $\sigma(\mathbf{x}) \coloneqq [|\rho(\mathbf{x}) - \rho(\mathbf{v})| \leqslant \delta]$, where $\delta$ is user-specified

- Bottom line:
  Reason about symbolic explainability by abstracting away type of ML model

# Abductive explanations – answering Why? questions

- Instance $(\mathbf{v}, q)$, i.e. $c = \tau(\mathbf{v})$

# Abductive explanations – answering Why? questions

- Instance $(\mathbf{v}, q)$, i.e. $c = \tau(\mathbf{v})$
- **Abductive explanation** (AXp, PI-explanation): [SCD18, INM19a]
  - **Subset-minimal** set of features $\mathcal{X} \subseteq \mathcal{F}$ sufficient for **ensuring** prediction

$$\forall(\mathbf{x} \in \mathbb{F}). \bigwedge\nolimits_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$$

# Abductive explanations – answering Why? questions

- Instance $(\mathbf{v}, q)$, i.e. $c = \tau(\mathbf{v})$
- **Abductive explanation** (AXp, PI-explanation): [SCD18, INM19a]
  - **Subset-minimal** set of features $\mathcal{X} \subseteq \mathcal{F}$ sufficient for **ensuring** prediction

$$\text{WAXp}(\mathcal{X}) \quad := \quad \forall(\mathbf{x} \in \mathbb{F}). \bigwedge\nolimits_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$$

- Defining AXp (from weak AXps, WAXps):

$$\text{AXp}(\mathcal{X}) := \text{WAXp}(\mathcal{X}) \land \forall(\mathcal{X}' \subsetneq \mathcal{X}). \neg\text{WAXp}(\mathcal{X}')$$

# Abductive explanations – answering Why? questions

- Instance $(\mathbf{v}, q)$, i.e. $c = \tau(\mathbf{v})$
- **Abductive explanation** (AXp, PI-explanation): [SCD18, INM19a]
  - **Subset-minimal** set of features $\mathcal{X} \subseteq \mathcal{F}$ sufficient for **ensuring** prediction

$$\text{WAXp}(\mathcal{X}) \quad := \quad \forall(\mathbf{x} \in \mathbb{F}).\ \bigwedge\nolimits_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$$

- Defining AXp (from weak AXps, WAXps):

$$\text{AXp}(\mathcal{X}) := \text{WAXp}(\mathcal{X}) \land \forall(\mathcal{X}' \subsetneq \mathcal{X}).\neg\text{WAXp}(\mathcal{X}')$$

- But, WAXp is **monotone**; hence,

$$\text{AXp}(\mathcal{X}) := \text{WAXp}(\mathcal{X}) \land \forall(t \in \mathcal{X}).\neg\text{WAXp}(\mathcal{X} \backslash \{t\})$$

# Abductive explanations – answering Why? questions

- Instance $(\mathbf{v}, q)$, i.e. $c = \tau(\mathbf{v})$
- **Abductive explanation** (AXp, PI-explanation): [SCD18, INM19a]
  - **Subset-minimal** set of features $\mathcal{X} \subseteq \mathcal{F}$ sufficient for **ensuring** prediction

$$\mathsf{WAXp}(\mathcal{X}) \quad := \quad \forall(\mathbf{x} \in \mathbb{F}).\ \bigwedge\nolimits_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$$

- Defining AXp (from weak AXps, WAXps):

$$\mathsf{AXp}(\mathcal{X}) := \mathsf{WAXp}(\mathcal{X}) \land \forall(\mathcal{X}' \subsetneq \mathcal{X}).\neg\mathsf{WAXp}(\mathcal{X}')$$

- But, WAXp is **monotone**; hence,

$$\mathsf{AXp}(\mathcal{X}) := \mathsf{WAXp}(\mathcal{X}) \land \forall(t \in \mathcal{X}).\neg\mathsf{WAXp}(\mathcal{X}\backslash\{t\})$$

- Finding one AXp (example algorithm; many more exist): [MM20]
  - Let $\mathcal{X} = \mathcal{F}$, i.e. fix all features
  - Invariant: $\mathsf{WAXp}(\mathcal{X})$ must hold. **Why?**
  - Analyze features in any order, one feature $i$ at a time
    - If $\mathsf{WAXp}(\mathcal{X}\backslash\{i\})$ holds, then remove $i$ from $\mathcal{X}$, i.e. $i$ becomes free

# A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

# A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$. AXp?

# A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$. AXp?
- Define $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$

# A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$. AXp?
- Define $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_2 \land \neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$?

Recap weak AXp: $\forall(\mathbf{x} \in \mathbb{F}).\bigwedge_{j \in \mathcal{X}}(x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$

# A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$. AXp?
- Define $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_2 \land \neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes

Recap weak AXp: $\forall(\mathbf{x} \in \mathbb{F}).\ \bigwedge_{j \in \mathcal{X}}(x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$

# A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$. AXp?
- Define $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_2 \land \neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$?

Recap weak AXp: $\forall(\mathbf{x} \in \mathbb{F}).\bigwedge_{j \in \mathcal{X}}(x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$

# A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$. AXp?
- Define $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_2 \land \neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes

Recap weak AXp: $\forall(\mathbf{x} \in \mathbb{F}).\ \bigwedge_{j \in \mathcal{X}}(x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$

# A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$. AXp?
- Define $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_2 \land \neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 3 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$?

Recap weak AXp: $\forall(\mathbf{x} \in \mathbb{F}).\bigwedge_{j \in \mathcal{X}}(x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$

# A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$. AXp?
- Define $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_2 \land \neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 3 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes

Recap weak AXp: $\forall(\mathbf{x} \in \mathbb{F}).\bigwedge_{j \in \mathcal{X}}(x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$

# A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$. AXp?
- Define $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_2 \land \neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 3 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 4 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\top \rightarrow \kappa(x_1, x_2, x_3, x_4)$?

Recap weak AXp: $\forall(\mathbf{x} \in \mathbb{F}).\bigwedge_{j \in \mathcal{X}}(x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$

# A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$. AXp?
- Define $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_2 \land \neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 3 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 4 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\top \rightarrow \kappa(x_1, x_2, x_3, x_4)$? No

Recap weak AXp: $\forall(\mathbf{x} \in \mathbb{F}).\bigwedge_{j \in \mathcal{X}}(x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$

# A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$. AXp?
- Define $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_2 \land \neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 3 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 4 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\top \rightarrow \kappa(x_1, x_2, x_3, x_4)$? No
- AXp $\mathcal{X} = \{4\}$

Recap weak AXp: $\forall(\mathbf{x} \in \mathbb{F}).\bigwedge_{j \in \mathcal{X}}(x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$

# A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee\nolimits_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$. AXp?
- Define $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\forall(\mathbf{x} \in \{0,1\}^4).\neg x_2 \land \neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\forall(\mathbf{x} \in \{0,1\}^4).\neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 3 be removed, i.e. $\forall(\mathbf{x} \in \{0,1\}^4).x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 4 be removed, i.e. $\forall(\mathbf{x} \in \{0,1\}^4).\top \rightarrow \kappa(x_1, x_2, x_3, x_4)$? No
- AXp $\mathcal{X} = \{4\}$

- In general, validity/consistency checked with SAT/SMT/MILP/CP reasoners

Recap weak AXp: $\forall(\mathbf{x} \in \mathbb{F}).\bigwedge_{j \in \mathcal{X}}(x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$

# A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$. AXp?
- Define $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_2 \land \neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 3 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 4 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\top \rightarrow \kappa(x_1, x_2, x_3, x_4)$? No
- AXp $\mathcal{X} = \{4\}$

- In general, validity/consistency checked with SAT/SMT/MILP/CP reasoners
  - **Obs:** for some classes of classifiers, poly-time algorithms exist

Recap weak AXp: $\forall(\mathbf{x} \in \mathbb{F}).\bigwedge_{j \in \mathcal{X}}(x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$

# More notation

- Notation $\mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}$:

$$[\mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] \quad \equiv \quad \bigwedge_{i \in \mathcal{S}} (X_i = v_i)$$

# More notation

- Notation $\mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}$:

$$[\mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] \quad \equiv \quad \bigwedge_{i \in \mathcal{S}} (X_i = v_i)$$

- Definition of $\Upsilon(\mathcal{S})$:

$$\Upsilon(\mathcal{S}) \quad \coloneqq \quad \{\mathbf{x} \in \mathbb{F} \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}\}$$

# More notation

- Notation $\mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}$:

$$[\mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] \quad \equiv \quad \bigwedge_{i \in \mathcal{S}} (X_i = v_i)$$

- Definition of $\Upsilon(\mathcal{S})$:

$$\Upsilon(\mathcal{S}) \quad \coloneqq \quad \{\mathbf{x} \in \mathbb{F} \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}\}$$

- Expected value, non-real-valued features:

$$\mathbf{E}[\tau(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] \quad \coloneqq \quad {}^{1}/_{|\Upsilon(\mathcal{S};\mathbf{v})|} \sum\nolimits_{\mathbf{x} \in \Upsilon(\mathcal{S};\mathbf{v})} \tau(\mathbf{x})$$

# More notation

- Notation $\mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}$:

$$[\mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] \quad \equiv \quad \bigwedge_{i \in \mathcal{S}} (X_i = v_i)$$

- Definition of $\Upsilon(\mathcal{S})$:

$$\Upsilon(\mathcal{S}) \quad \coloneqq \quad \{\mathbf{x} \in \mathbb{F} \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}\}$$

- Expected value, non-real-valued features:

$$\mathbf{E}[\tau(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] \quad \coloneqq \quad {}^{1}/_{|\Upsilon(\mathcal{S};\mathbf{v})|} \sum\nolimits_{\mathbf{x} \in \Upsilon(\mathcal{S};\mathbf{v})} \tau(\mathbf{x})$$

- Expected value, real-valued features:

$$\mathbf{E}[\tau(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] \quad \coloneqq \quad {}^{1}/_{|\Upsilon(\mathcal{S};\mathbf{v})|} \int_{\Upsilon(\mathcal{S};\mathbf{v})} \tau(\mathbf{x}) d\mathbf{x}$$

# Other definitions of WAXps/AXps

- Using probabilities, non-real-valued features: [WMHK21, IHI+22, ABOS22, IHI+23]

$$\text{WAXp}(\mathcal{S}) \quad := \quad \Pr(\sigma(\mathbf{x}) \,|\, \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}) = 1$$

# Other definitions of WAXps/AXps

- Using probabilities, non-real-valued features: [WMHK21, IHI+22, ABOS22, IHI+23]

$$\mathsf{WAXp}(\mathcal{S}) \quad := \quad \mathsf{Pr}(\sigma(\mathbf{x}) \,|\, \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}) = 1$$

- Using expected values:

$$\mathsf{WAXp}(\mathcal{S}) \quad := \quad \mathbf{E}[\sigma(\mathbf{x}) \,|\, \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] = 1$$

# Other definitions of WAXps/AXps

- Using probabilities, non-real-valued features: [WMHK21, IHI+22, ABOS22, IHI+23]

$$\mathsf{WAXp}(\mathcal{S}) \quad := \quad \mathsf{Pr}(\sigma(\mathbf{x}) \,|\, \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}) = 1$$

- Using expected values:

$$\mathsf{WAXp}(\mathcal{S}) \quad := \quad \mathbf{E}[\sigma(\mathbf{x}) \,|\, \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] = 1$$

- Definition of AXp remains unchanged
  - This is true when comparing against 1

# Constrastive explanations – answering Why not? questions

- Instance $(\mathbf{v}, c)$, i.e. $c = \kappa(\mathbf{v})$

# Constrastive explanations – answering Why not? questions

- Instance $(\mathbf{v}, c)$, i.e. $c = \kappa(\mathbf{v})$
- Contrastive explanation (CXp): [Mil19, INAM20]
  - Subset-minimal set of features $\mathcal{Y} \subseteq \mathcal{F}$ sufficient for changing prediction

$$\exists(\mathbf{x} \in \mathbb{F}). \bigwedge\nolimits_{j \notin \mathcal{Y}} (x_j = v_j) \land (\neg\sigma(\mathbf{x}))$$

# Constrastive explanations – answering Why not? questions

- Instance $(\mathbf{v}, c)$, i.e. $c = \kappa(\mathbf{v})$
- Contrastive explanation (CXp): [Mil19, INAM20]
  - Subset-minimal set of features $\mathcal{Y} \subseteq \mathcal{F}$ sufficient for changing prediction

$$\mathsf{WCXp}(\mathcal{Y}) \quad := \quad \exists(\mathbf{x} \in \mathbb{F}).\ \bigwedge\nolimits_{j \notin \mathcal{Y}} (x_j = v_j) \land (\neg\sigma(\mathbf{x}))$$

- Defining CXp:

$$\mathsf{CXp}(\mathcal{Y}) := \mathsf{WCXp}(\mathcal{Y}) \land \forall(\mathcal{Y}' \subsetneq \mathcal{Y}).\neg\mathsf{WCXp}(\mathcal{Y}')$$

# Constrastive explanations – answering Why not? questions

- Instance $(\mathbf{v}, c)$, i.e. $c = \kappa(\mathbf{v})$
- Contrastive explanation (CXp): [Mil19, INAM20]
  - Subset-minimal set of features $\mathcal{Y} \subseteq \mathcal{F}$ sufficient for changing prediction

$$\mathsf{WCXp}(\mathcal{Y}) \quad := \quad \exists(\mathbf{x} \in \mathbb{F}).\ \bigwedge\nolimits_{j \notin \mathcal{Y}} (x_j = v_j) \land (\neg\sigma(\mathbf{x}))$$

- Defining CXp:

$$\mathsf{CXp}(\mathcal{Y}) := \mathsf{WCXp}(\mathcal{Y}) \land \forall(\mathcal{Y}' \subsetneq \mathcal{Y}).\neg\mathsf{WCXp}(\mathcal{Y}')$$

- But, WCXp is also monotone; hence,

$$\mathsf{CXp}(\mathcal{Y}) := \mathsf{WCXp}(\mathcal{Y}) \land \forall(t \in \mathcal{Y}).\neg\mathsf{WCXp}(\mathcal{Y}\backslash\{t\})$$

# Constrastive explanations – answering Why not? questions

- Instance $(\mathbf{v}, c)$, i.e. $c = \kappa(\mathbf{v})$
- Contrastive explanation (CXp): [Mil19, INAM20]
  - Subset-minimal set of features $\mathcal{Y} \subseteq \mathcal{F}$ sufficient for changing prediction

$$\text{WCXp}(\mathcal{Y}) \quad := \quad \exists(\mathbf{x} \in \mathbb{F}). \bigwedge\nolimits_{j \notin \mathcal{Y}} (x_j = v_j) \land (\neg\sigma(\mathbf{x}))$$

- Defining CXp:

$$\text{CXp}(\mathcal{Y}) := \text{WCXp}(\mathcal{Y}) \land \forall(\mathcal{Y}' \subsetneq \mathcal{Y}).\neg\text{WCXp}(\mathcal{Y}')$$

- But, WCXp is also monotone; hence,

$$\text{CXp}(\mathcal{Y}) := \text{WCXp}(\mathcal{Y}) \land \forall(t \in \mathcal{Y}).\neg\text{WCXp}(\mathcal{Y}\backslash\{t\})$$

- Finding one CXp: [MM20]
  - Let $\mathcal{Y} = \mathcal{F}$, i.e. free all features
  - Invariant: $\text{WCXp}(\mathcal{Y})$ must hold. Why?
  - Analyze features in any order, one feature $i$ at a time
    - If $\text{WCXp}(\mathcal{Y}\backslash\{i\})$ holds, then remove $i$ from $\mathcal{Y}$, i.e. $i$ is becomes fixed

# A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$
- Define $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$

# A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$
- Define $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg\kappa(x_1, x_2, x_3, x_4)$?

Recap weak CXp: $\exists(\mathbf{x} \in \mathbb{F}).\bigwedge_{j \notin \mathcal{Y}}(x_j = v_j) \land (\neg\sigma(\mathbf{x}))$

# A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$
- Define $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes

Recap weak CXp: $\exists(\mathbf{x} \in \mathbb{F}).\bigwedge_{j \notin \mathcal{Y}}(x_j = v_j) \land (\neg\sigma(\mathbf{x}))$

# A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$
- Define $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg\kappa(x_1, x_2, x_3, x_4)$?

Recap weak CXp: $\exists(\mathbf{x} \in \mathbb{F}).\bigwedge_{j \notin \mathcal{Y}}(x_j = v_j) \land (\neg\sigma(\mathbf{x}))$

# A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$
- Define $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes

Recap weak CXp: $\exists(\mathbf{x} \in \mathbb{F}).\bigwedge_{j \notin \mathcal{Y}}(x_j = v_j) \land (\neg\sigma(\mathbf{x}))$

# A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$
- Define $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 3 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg x_3 \land \neg\kappa(x_1, x_2, x_3, x_4)$?

Recap weak CXp: $\exists(\mathbf{x} \in \mathbb{F}).\bigwedge_{j \notin \mathcal{Y}}(x_j = v_j) \land (\neg\sigma(\mathbf{x}))$

# A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$
- Define $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 3 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg x_3 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes

Recap weak CXp: $\exists(\mathbf{x} \in \mathbb{F}).\bigwedge_{j \notin \mathcal{Y}}(x_j = v_j) \land (\neg\sigma(\mathbf{x}))$

# A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$
- Define $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 3 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg x_3 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 4 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg x_3 \land x_4 \land \neg\kappa(x_1, x_2, x_3, x_4)$?

Recap weak CXp: $\exists(\mathbf{x} \in \mathbb{F}).\bigwedge_{j \notin \mathcal{Y}}(x_j = v_j) \land (\neg\sigma(\mathbf{x}))$

# A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$
- Define $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 3 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg x_3 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 4 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg x_3 \land x_4 \land \neg\kappa(x_1, x_2, x_3, x_4)$? No

Recap weak CXp: $\exists(\mathbf{x} \in \mathbb{F}).\bigwedge_{j \notin \mathcal{Y}}(x_j = v_j) \land (\neg\sigma(\mathbf{x}))$

# A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$
- Define $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 3 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg x_3 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 4 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg x_3 \land x_4 \land \neg\kappa(x_1, x_2, x_3, x_4)$? No
- CXp $\mathcal{Y} = \{4\}$
- **Obs:** AXp is MHS of CXp and vice-versa...

Recap weak CXp: $\exists(\mathbf{x} \in \mathbb{F}).\bigwedge_{j \notin \mathcal{Y}}(x_j = v_j) \land (\neg\sigma(\mathbf{x}))$

# Other definitions of WCXps/CXps

- Using probabilities, non-real-valued features:

$$\mathsf{WCXp}(\mathcal{S}) \quad := \quad \mathsf{Pr}(\sigma(\mathbf{x}) \,|\, \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}) < 1$$

# Other definitions of WCXps/CXps

- Using probabilities, non-real-valued features:

$$\text{WCXp}(\mathcal{S}) \quad := \quad \Pr(\sigma(\mathbf{x}) \,|\, \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}) < 1$$

- Using expected values:

$$\text{WCXp}(\mathcal{S}) \quad := \quad \mathbf{E}[\sigma(\mathbf{x}) \,|\, \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] < 1$$

# Other definitions of WCXps/CXps

- Using probabilities, non-real-valued features:

$$\text{WCXp}(\mathcal{S}) \quad := \quad \Pr(\sigma(\mathbf{x}) \,|\, \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}) < 1$$

- Using expected values:

$$\text{WCXp}(\mathcal{S}) \quad := \quad \mathbf{E}[\sigma(\mathbf{x}) \,|\, \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] < 1$$

- Definition of CXp remains unchanged

# Detour: global explanations

[INM19b]

- AXps and CXps are defined locally (because of $\mathbf{v}$) but hold globally
  - Localized explanations
  - Can be viewed as attempt at formalizing local explanations [RSG16, LL17, RSG18]
- One can define explanations without picking a given point in feature space
  - Let $q \in \mathbb{T}$, and refefine the similarity predicate:
    - Classification: $\sigma(\mathbf{x}) = [\kappa(\mathbf{x}) = q]$
    - Regression: $\sigma(\mathbf{x}) = [|\kappa(\mathbf{x}) - q| \leqslant \delta]$, $\delta$ is user-specified
  - Let $\mathbb{L} = \{(x_i = v_i) \,|\, i \in \mathcal{F} \land v_i \in \mathbb{V}\}$
  - Let $\mathcal{S} \subsetneq \mathbb{L}$ be a subset of literals that does not repeat features, i.e. $\mathcal{S}$ is not inconsistent
  - Then, $\mathcal{S}$ is a global AXp if,

$$\forall(\mathbf{x} \in \mathbb{F}).\ \bigwedge\nolimits_{(x_i = v_i) \in \mathcal{S}} (x_i = v_i) \rightarrow (\sigma(\mathbf{x}))$$

  - Counterexamples are minimal hitting sets of global AXps and vice-versa [INM19b]

# Outline – Unit #02

# Duality in explainability – basic results

[INAM20, Mar22]

# Duality in explainability – basic results

[INAM20, Mar22]

- **Claim:**
  $\mathcal{S} \subseteq \mathcal{F}$ is an AXp iff it is a minimal hitting set (MHS) of the set of CXps

# Duality in explainability – basic results

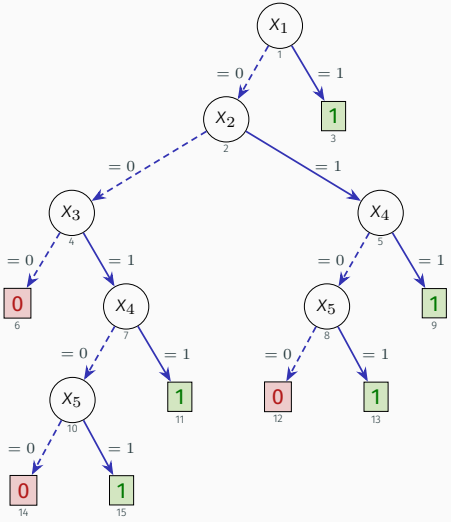
[INAM20, Mar22]

- **Claim:**
  $\mathcal{S} \subseteq \mathcal{F}$ is an AXp iff it is a minimal hitting set (MHS) of the set of CXps
- **Claim:**
  $\mathcal{S} \subseteq \mathcal{F}$ is a CXp iff it is a minimal hitting set (MHS) of the set of AXps

# Duality in explainability – basic results

[INAM20, Mar22]

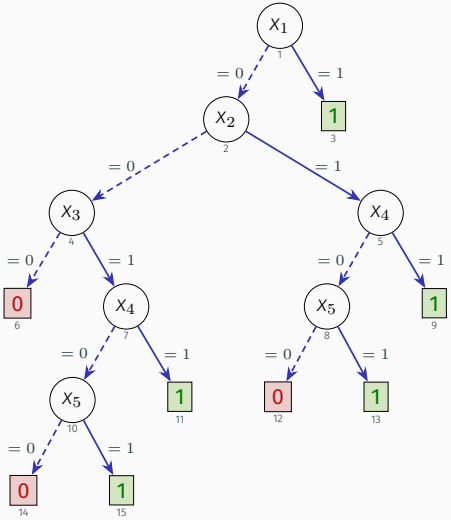
- **Claim:**
  $\mathcal{S} \subseteq \mathcal{F}$ is an AXp iff it is a minimal hitting set (MHS) of the set of CXps
- **Claim:**
  $\mathcal{S} \subseteq \mathcal{F}$ is a CXp iff it is a minimal hitting set (MHS) of the set of AXps
- An example, $(\mathbf{v}, c) = ((0, 0, 1, 0, 1), 1)$:



# Duality in explainability – basic results

[INAM20, Mar22]

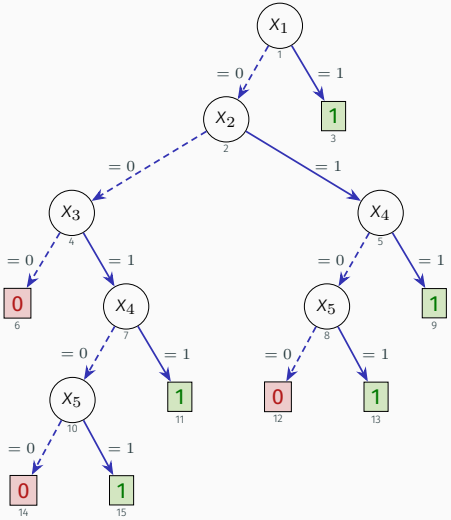
- **Claim:**
  $\mathcal{S} \subseteq \mathcal{F}$ is an AXp iff it is a minimal hitting set (MHS) of the set of CXps
- **Claim:**
  $\mathcal{S} \subseteq \mathcal{F}$ is a CXp iff it is a minimal hitting set (MHS) of the set of AXps
- An example, $(\mathbf{v}, c) = ((0, 0, 1, 0, 1), 1)$:
  - AXps:



# Duality in explainability – basic results

[INAM20, Mar22]

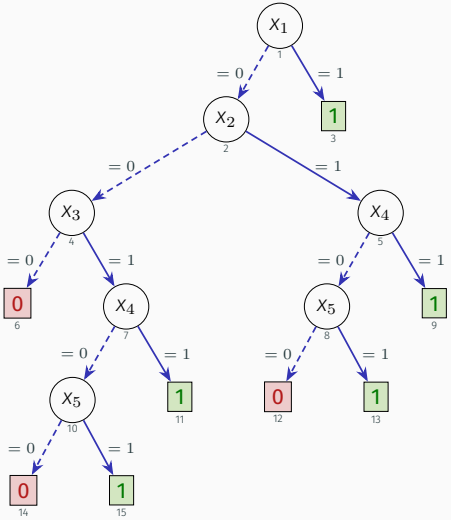
- **Claim:**
  $\mathcal{S} \subseteq \mathcal{F}$ is an AXp iff it is a minimal hitting set (MHS) of the set of CXps
- **Claim:**
  $\mathcal{S} \subseteq \mathcal{F}$ is a CXp iff it is a minimal hitting set (MHS) of the set of AXps
- An example, $(\mathbf{v}, c) = ((0, 0, 1, 0, 1), 1)$:
  - AXps: $\{\{3, 5\}\}$



# Duality in explainability – basic results

[INAM20, Mar22]

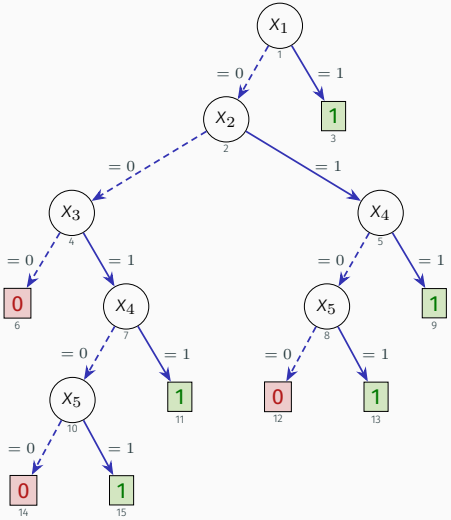
- **Claim:**
  $\mathcal{S} \subseteq \mathcal{F}$ is an AXp iff it is a minimal hitting set (MHS) of the set of CXps
- **Claim:**
  $\mathcal{S} \subseteq \mathcal{F}$ is a CXp iff it is a minimal hitting set (MHS) of the set of AXps
- An example, $(\mathbf{v}, c) = ((0, 0, 1, 0, 1), 1)$:
  - AXps: $\{\{3, 5\}\}$
  - CXps:



# Duality in explainability – basic results

[INAM20, Mar22]

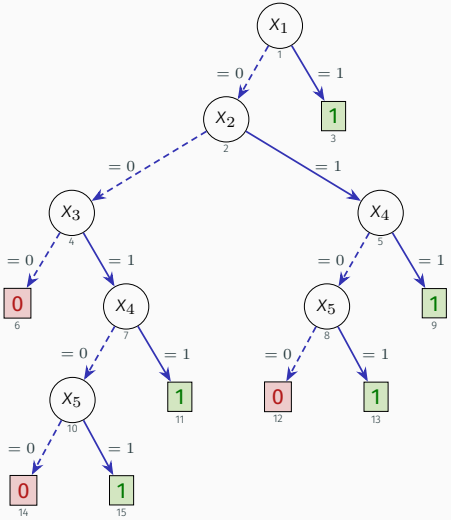
- **Claim:**
  $\mathcal{S} \subseteq \mathcal{F}$ is an AXp iff it is a minimal hitting set (MHS) of the set of CXps
- **Claim:**
  $\mathcal{S} \subseteq \mathcal{F}$ is a CXp iff it is a minimal hitting set (MHS) of the set of AXps
- An example, $(\mathbf{v}, c) = ((0, 0, 1, 0, 1), 1)$:
  - AXps: $\{\{3, 5\}\}$
  - CXps: $\{\{3\}, \{5\}\}$



# Duality in explainability – basic results

[INAM20, Mar22]

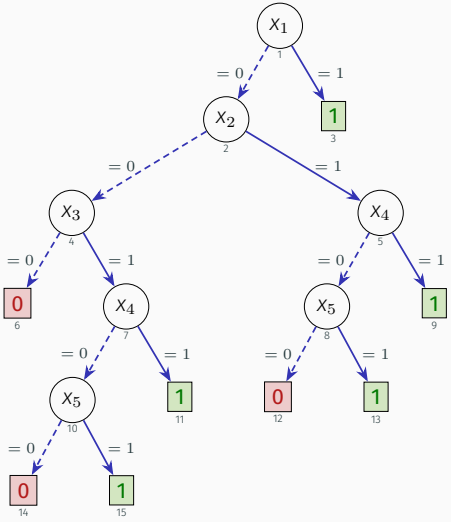
- **Claim:**
  $\mathcal{S} \subseteq \mathcal{F}$ is an AXp iff it is a minimal hitting set (MHS) of the set of CXps
- **Claim:**
  $\mathcal{S} \subseteq \mathcal{F}$ is a CXp iff it is a minimal hitting set (MHS) of the set of AXps
- An example, $(\mathbf{v}, c) = ((0, 0, 1, 0, 1), 1)$:
  - AXps: $\{\{3, 5\}\}$
  - CXps: $\{\{3\}, \{5\}\}$
  - Each AXp is an MHS of the set of CXps
  - Each CXp is an MHS of the set of AXps



# Duality in explainability – basic results

[INAM20, Mar22]

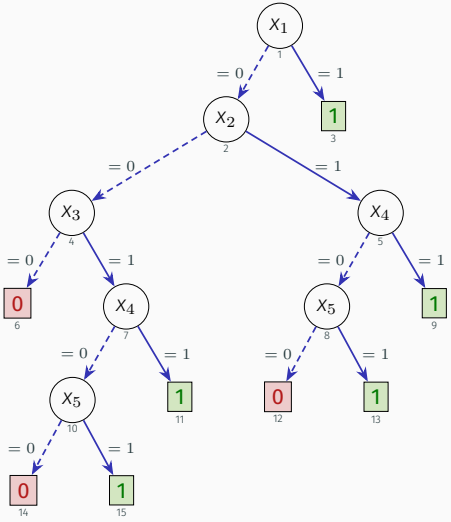
- **Claim:**
  $\mathcal{S} \subseteq \mathcal{F}$ is an AXp iff it is a minimal hitting set (MHS) of the set of CXps
- **Claim:**
  $\mathcal{S} \subseteq \mathcal{F}$ is a CXp iff it is a minimal hitting set (MHS) of the set of AXps
- An example, $(\mathbf{v}, c) = ((0, 0, 1, 0, 1), 1)$:
  - AXps: $\{\{3, 5\}\}$
  - CXps: $\{\{3\}, \{5\}\}$
  - Each AXp is an MHS of the set of CXps
  - Each CXp is an MHS of the set of AXps
  - BTW,
    - $\{2, 5\}$ is not a CXp
    - $\{1, 2, 3, 4, 5\}$, $\{1, 2, 3, 5\}$ and $\{1, 3, 5\}$ are not AXps



# Duality in explainability – basic results

[INAM20, Mar22]

- **Claim:**
  $\mathcal{S} \subseteq \mathcal{F}$ is an AXp iff it is a minimal hitting set (MHS) of the set of CXps
- **Claim:**
  $\mathcal{S} \subseteq \mathcal{F}$ is a CXp iff it is a minimal hitting set (MHS) of the set of AXps
- An example, $(\mathbf{v}, c) = ((0, 0, 1, 0, 1), 1)$:
  - AXps: $\{\{3, 5\}\}$
  - CXps: $\{\{3\}, \{5\}\}$
  - Each AXp is an MHS of the set of CXps
  - Each CXp is an MHS of the set of AXps
  - BTW,
    - $\{2, 5\}$ is not a CXp
    - $\{1, 2, 3, 4, 5\}$, $\{1, 2, 3, 5\}$ and $\{1, 3, 5\}$ are not AXps
    - Why?



# Outline – Unit #02

# Computational problems in (formal) explainability

- Compute one abductive/contrastive explanation

# Computational problems in (formal) explainability

- Compute one abductive/contrastive explanation
- Enumerate all abductive/contrastive explanations

# Computational problems in (formal) explainability

- Compute one abductive/contrastive explanation
- Enumerate all abductive/contrastive explanations
- Decide whether feature included in all abductive/contrastive explanations

# Computational problems in (formal) explainability

- Compute one abductive/contrastive explanation
- Enumerate all abductive/contrastive explanations
- Decide whether feature included in all abductive/contrastive explanations
- Decide whether feature included in some abductive/contrastive explanation

# Computing one AXp/CXp

- Encode classifier into suitable logic representation $\mathcal{T}$ & pick suitable reasoner

# Computing one AXp/CXp

- Encode classifier into suitable logic representation $\mathcal{T}$ & pick suitable reasoner
- For AXp: start from $\mathcal{S} = \mathcal{F}$ and drop (i.e. free) features from $\mathcal{S}$ while WAXp condition holds

# Computing one AXp/CXp

- Encode classifier into suitable logic representation $\mathcal{T}$ & pick suitable reasoner
- For AXp: start from $\mathcal{S} = \mathcal{F}$ and drop (i.e. free) features from $\mathcal{S}$ while WAXp condition holds
- For CXp: start from $\mathcal{S} = \mathcal{F}$ and drop (i.e. fix) features from $\mathcal{S}$ while WCXp condition holds

# Computing one AXp/CXp

- Encode classifier into suitable logic representation $\mathcal{T}$ & pick suitable reasoner
- For AXp: start from $\mathcal{S} = \mathcal{F}$ and drop (i.e. free) features from $\mathcal{S}$ while WAXp condition holds
- For CXp: start from $\mathcal{S} = \mathcal{F}$ and drop (i.e. fix) features from $\mathcal{S}$ while WCXp condition holds
- Monotone predicates for WAXp & WCXp:

$$\mathbb{P}_{\text{axp}}(\mathcal{S}) \triangleq \neg \textsf{CO}\left([\![(\bigwedge\nolimits_{i\in\mathcal{S}}(x_i = v_i)) \land (\neg\sigma(\mathbf{x}))]\!]\right) \qquad \mathbb{P}_{\text{cxp}}(\mathcal{S}) \triangleq \textsf{CO}\left([\![\left(\bigwedge\nolimits_{i\in\mathcal{F}\setminus\mathcal{S}}(x_i = v_i)\right) \land (\neg\sigma(\mathbf{x}))]\!]\right)$$

# Computing one AXp/CXp

- Encode classifier into suitable logic representation $\mathcal{T}$ & pick suitable reasoner
- For AXp: start from $\mathcal{S} = \mathcal{F}$ and drop (i.e. free) features from $\mathcal{S}$ while WAXp condition holds
- For CXp: start from $\mathcal{S} = \mathcal{F}$ and drop (i.e. fix) features from $\mathcal{S}$ while WCXp condition holds
- Monotone predicates for WAXp & WCXp:

$$\mathbb{P}_{\text{axp}}(\mathcal{S}) \triangleq \neg \textsf{CO}\left([\![(\textstyle\bigwedge_{i\in\mathcal{S}}(x_i = v_i)) \wedge (\neg\sigma(\mathbf{x}))]\!]\right) \qquad \mathbb{P}_{\text{cxp}}(\mathcal{S}) \triangleq \textsf{CO}\left([\![\left(\textstyle\bigwedge_{i\in\mathcal{F}\setminus\mathcal{S}}(x_i = v_i)\right) \wedge (\neg\sigma(\mathbf{x}))]\!]\right)$$

**Input**: Predicate $\mathbb{P}$, parameterized by $\mathcal{T}$, $\mathcal{M}$
**Output**: One XP $\mathcal{S}$

1: **procedure** oneXP($\mathbb{P}$)
2: $\quad\mathcal{S} \leftarrow \mathcal{F}$ ▷ Initialization: $\mathbb{P}(\mathcal{S})$ holds
3: $\quad$**for** $i \in \mathcal{F}$ **do** ▷ Loop invariant: $\mathbb{P}(\mathcal{S})$ holds
4: $\quad\quad$**if** $\mathbb{P}(\mathcal{S}\setminus\{i\})$ **then**
5: $\quad\quad\quad\mathcal{S} \leftarrow \mathcal{S}\setminus\{i\}$ ▷ Update $\mathcal{S}$ only if $\mathbb{P}(\mathcal{S}\setminus\{i\})$ holds
6: $\quad$**return** $\mathcal{S}$ ▷ Returned set $\mathcal{S}$: $\mathbb{P}(\mathcal{S})$ holds

# Computing one AXp/CXp

- Encode classifier into suitable logic representation $\mathcal{T}$ & pick suitable reasoner
- For AXp: start from $\mathcal{S} = \mathcal{F}$ and drop (i.e. free) features from $\mathcal{S}$ while WAXp condition holds
- For CXp: start from $\mathcal{S} = \mathcal{F}$ and drop (i.e. fix) features from $\mathcal{S}$ while WCXp condition holds
- Monotone predicates for WAXp & WCXp:

$$\mathbb{P}_{\text{axp}}(\mathcal{S}) \triangleq \neg\,\textsf{CO}\left([\![(\textstyle\bigwedge_{i\in\mathcal{S}}(x_i = v_i)) \wedge (\neg\sigma(\mathbf{x}))]\!]\right) \qquad \mathbb{P}_{\text{cxp}}(\mathcal{S}) \triangleq \textsf{CO}\left([\![\left(\textstyle\bigwedge_{i\in\mathcal{F}\setminus\mathcal{S}}(x_i = v_i)\right) \wedge (\neg\sigma(\mathbf{x}))]\!]\right)$$

**Input**: Predicate $\mathbb{P}$, parameterized by $\mathcal{T}$, $\mathcal{M}$
**Output**: One XP $\mathcal{S}$

```
1: procedure oneXP(ℙ)
2:     𝒮 ← ℱ                                   ▷ Initialization: ℙ(𝒮) holds
3:     for i ∈ ℱ do                            ▷ Loop invariant: ℙ(𝒮) holds
4:         if ℙ(𝒮\{i}) then
5:             𝒮 ← 𝒮\{i}                       ▷ Update 𝒮 only if ℙ(𝒮\{i}) holds
6:     return 𝒮                                ▷ Returned set 𝒮: ℙ(𝒮) holds
```

Exploiting MSMP, i.e. basic algorithm used for different problems.

# Detour: More Connections with Automated Reasoning

# Prime implicants & implicates

- A conjunction of literals $\pi$ (which will be viewed as a set of literals where convenient) is a prime implicant of some function $\varphi$ if,
  1. $\pi \models \varphi$
  2. For any $\pi' \subsetneq \pi$, $\pi' \not\models \varphi$

# Prime implicants & implicates

- A conjunction of literals $\pi$ (which will be viewed as a set of literals where convenient) is a prime implicant of some function $\varphi$ if,
  1. $\pi \models \varphi$
  2. For any $\pi' \subsetneq \pi$, $\pi' \not\models \varphi$

  - Example:
    - $\mathbb{F} = \{0,1\}^3$
    - $\varphi(x_1, x_2, x_3) = x_1 \wedge x_2 \vee x_1 \wedge x_3$
    - Clearly, $x_1 \wedge x_2 \models \varphi$
    - Also, $x_1 \not\models \varphi$ and $x_2 \not\models \varphi$

# Prime implicants & implicates

- A conjunction of literals $\pi$ (which will be viewed as a set of literals where convenient) is a prime implicant of some function $\varphi$ if,
  1. $\pi \models \varphi$
  2. For any $\pi' \subsetneq \pi$, $\pi' \not\models \varphi$
  - Example:
    - $\mathbb{F} = \{0,1\}^3$
    - $\varphi(x_1, x_2, x_3) = x_1 \wedge x_2 \vee x_1 \wedge x_3$
    - Clearly, $x_1 \wedge x_2 \models \varphi$
    - Also, $x_1 \not\models \varphi$ and $x_2 \not\models \varphi$

- A disjunction of literals $\eta$ (also viewed as a set of literals where convenient) is a prime implicate of some function $\varphi$ if
  1. $\varphi \models \eta$
  2. For any $\eta' \subsetneq \eta$, $\varphi \not\models \eta'$

# Reasoning about inconsistency

[MM20]

- Formula $\mathcal{T} = \mathcal{B} \cup \mathcal{S}$, with
  - $\mathcal{B}$: background knowledge (base), i.e. hard constraints
  - $\mathcal{S}$: additional (inconsistent) knowledge, i.e. soft constraints
  - And, $\mathcal{T} \models \bot$
  - E.g. $\mathcal{B} = \{(x_1 \vee x_2), (x_1 \vee \neg x_3)\}, \mathcal{S} = \{(\neg x_1), (\neg x_2), (x_3)\}$

# Reasoning about inconsistency

[MM20]

- Formula $\mathcal{T} = \mathcal{B} \cup \mathcal{S}$, with
  - $\mathcal{B}$: background knowledge (base), i.e. hard constraints
  - $\mathcal{S}$: additional (inconsistent) knowledge, i.e. soft constraints
  - And, $\mathcal{T} \models \bot$
  - E.g. $\mathcal{B} = \{(x_1 \vee x_2), (x_1 \vee \neg x_3)\}, \mathcal{S} = \{(\neg x_1), (\neg x_2), (x_3)\}$
- Minimal unsatisfiable subset (MUS):
  - Subset-minimal set $\mathcal{U} \subseteq \mathcal{S}$, s.t. $\mathcal{B} \cup \mathcal{U} \models \bot$
  - E.g. $\mathcal{U} = \{(\neg x_1), (\neg x_2)\}$

# Reasoning about inconsistency

[MM20]

- Formula $\mathcal{T} = \mathcal{B} \cup \mathcal{S}$, with
  - $\mathcal{B}$: background knowledge (base), i.e. hard constraints
  - $\mathcal{S}$: additional (inconsistent) knowledge, i.e. soft constraints
  - And, $\mathcal{T} \models \bot$
  - E.g. $\mathcal{B} = \{(x_1 \vee x_2), (x_1 \vee \neg x_3)\}$, $\mathcal{S} = \{(\neg x_1), (\neg x_2), (x_3)\}$
- Minimal unsatisfiable subset (MUS):
  - Subset-minimal set $\mathcal{U} \subseteq \mathcal{S}$, s.t. $\mathcal{B} \cup \mathcal{U} \models \bot$
  - E.g. $\mathcal{U} = \{(\neg x_1), (\neg x_2)\}$
- Minimal correction subset (MCS):
  - Subset-minimal set $\mathcal{C} \subseteq \mathcal{S}$, s.t. $\mathcal{B} \cup (\mathcal{S} \setminus \mathcal{C}) \not\models \bot$
  - E.g. $\mathcal{C} = \{(\neg x_1)\}$

# Reasoning about inconsistency

[MM20]

- Formula $\mathcal{T} = \mathcal{B} \cup \mathcal{S}$, with
  - $\mathcal{B}$: background knowledge (base), i.e. hard constraints
  - $\mathcal{S}$: additional (inconsistent) knowledge, i.e. soft constraints
  - And, $\mathcal{T} \models \bot$
  - E.g. $\mathcal{B} = \{(x_1 \vee x_2), (x_1 \vee \neg x_3)\}$, $\mathcal{S} = \{(\neg x_1), (\neg x_2), (x_3)\}$
- Minimal unsatisfiable subset (MUS):
  - Subset-minimal set $\mathcal{U} \subseteq \mathcal{S}$, s.t. $\mathcal{B} \cup \mathcal{U} \models \bot$
  - E.g. $\mathcal{U} = \{(\neg x_1), (\neg x_2)\}$
- Minimal correction subset (MCS):
  - Subset-minimal set $\mathcal{C} \subseteq \mathcal{S}$, s.t. $\mathcal{B} \cup (\mathcal{S} \setminus \mathcal{C}) \not\models \bot$
  - E.g. $\mathcal{C} = \{(\neg x_1)\}$
- Duality:
  - MUSes are minimal-hitting sets (MHSes) of the MCSes, and vice-versa [Rei87]

# Reasoning about inconsistency

[MM20]

- Formula $\mathcal{T} = \mathcal{B} \cup \mathcal{S}$, with
  - $\mathcal{B}$: background knowledge (base), i.e. hard constraints
  - $\mathcal{S}$: additional (inconsistent) knowledge, i.e. soft constraints
  - And, $\mathcal{T} \models \bot$
  - E.g. $\mathcal{B} = \{(x_1 \vee x_2), (x_1 \vee \neg x_3)\}, \mathcal{S} = \{(\neg x_1), (\neg x_2), (x_3)\}$
- Minimal unsatisfiable subset (MUS):
  - Subset-minimal set $\mathcal{U} \subseteq \mathcal{S}$, s.t. $\mathcal{B} \cup \mathcal{U} \models \bot$
  - E.g. $\mathcal{U} = \{(\neg x_1), (\neg x_2)\}$
- Minimal correction subset (MCS):
  - Subset-minimal set $\mathcal{C} \subseteq \mathcal{S}$, s.t. $\mathcal{B} \cup (\mathcal{S} \backslash \mathcal{C}) \not\models \bot$
  - E.g. $\mathcal{C} = \{(\neg x_1)\}$
- Duality:
  - MUSes are minimal-hitting sets (MHSes) of the MCSes, and vice-versa [Rei87]
- Variants:
  - Smallest(-cost) MCS, i.e. complement of maximum(-cost) satisfiability (MaxSAT)
  - Smallest(-cost) MUS

# Computing AXps (resp. CXps) as MUSes (resp. MCSes)

- Recap:

$$\begin{aligned}
\mathrm{WAXp}(\mathcal{X}) &:= \forall(\mathbf{x} \in \mathbb{F}).\ \bigwedge\nolimits_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\sigma(\mathbf{x})) \\
\mathrm{WCXp}(\mathcal{Y}) &:= \exists(\mathbf{x} \in \mathbb{F}).\ \bigwedge\nolimits_{j \notin \mathcal{Y}} (x_j = v_j) \land (\neg\sigma(\mathbf{x}))
\end{aligned}$$

# Computing AXps (resp. CXps) as MUSes (resp. MCSes)

- Recap:

$$\begin{aligned}
\mathsf{WAXp}(\mathcal{X}) &:= \neg\left[\exists(\mathbf{x}\in\mathbb{F}).\bigwedge\nolimits_{j\in\mathcal{X}}(x_j = v_j)\land(\neg\sigma(\mathbf{x}))\right]\\
\mathsf{WCXp}(\mathcal{Y}) &:= \exists(\mathbf{x}\in\mathbb{F}).\bigwedge\nolimits_{j\notin\mathcal{Y}}(x_j = v_j)\land(\neg\sigma(\mathbf{x}))
\end{aligned}$$

# Computing AXps (resp. CXps) as MUSes (resp. MCSes)

- Recap:

$$\begin{aligned}\text{WAXp}(\mathcal{X}) &:= \neg\left[\exists(\mathbf{x}\in\mathbb{F}).\bigwedge\nolimits_{j\in\mathcal{X}}(x_j = v_j)\land(\neg\sigma(\mathbf{x}))\right]\\ \text{WCXp}(\mathcal{Y}) &:= \exists(\mathbf{x}\in\mathbb{F}).\bigwedge\nolimits_{j\notin\mathcal{Y}}(x_j = v_j)\land(\neg\sigma(\mathbf{x}))\end{aligned}$$

- Let,
  - Hard constraints, $\mathcal{B}$:

$$\mathcal{B} := \wedge_{i\in\mathcal{F}}(s_i\rightarrow(x_i = v_i))\land\text{Encode}_{\mathcal{T}}(\neg\sigma(\mathbf{x}))$$

# Computing AXps (resp. CXps) as MUSes (resp. MCSes)

- Recap:

$$\begin{aligned} \mathrm{WAXp}(\mathcal{X}) &:= \neg\left[\exists(\mathbf{x}\in\mathbb{F}).\bigwedge\nolimits_{j\in\mathcal{X}}(x_j = v_j)\land(\neg\sigma(\mathbf{x}))\right] \\ \mathrm{WCXp}(\mathcal{Y}) &:= \exists(\mathbf{x}\in\mathbb{F}).\bigwedge\nolimits_{j\notin\mathcal{Y}}(x_j = v_j)\land(\neg\sigma(\mathbf{x})) \end{aligned}$$

- Let,
  - Hard constraints, $\mathcal{B}$:

$$\mathcal{B} := \land_{i\in\mathcal{F}}(s_i \rightarrow (x_i = v_i)) \land \mathrm{Encode}_{\mathcal{T}}(\neg\sigma(\mathbf{x}))$$

  - Soft constraints: $\mathcal{S} = \{s_i \mid i \in \mathcal{F}\}$

# Computing AXps (resp. CXps) as MUSes (resp. MCSes)

- Recap:

$$\begin{aligned} \text{WAXp}(\mathcal{X}) &:= \neg\left[\exists(\mathbf{x}\in\mathbb{F}).\bigwedge\nolimits_{j\in\mathcal{X}}(x_j = v_j)\land(\neg\sigma(\mathbf{x}))\right] \\ \text{WCXp}(\mathcal{Y}) &:= \exists(\mathbf{x}\in\mathbb{F}).\bigwedge\nolimits_{j\notin\mathcal{Y}}(x_j = v_j)\land(\neg\sigma(\mathbf{x})) \end{aligned}$$

- Let,
  - Hard constraints, $\mathcal{B}$:

$$\mathcal{B} := \wedge_{i\in\mathcal{F}}(s_i \rightarrow (x_i = v_i)) \land \text{Encode}_{\mathcal{T}}(\neg\sigma(\mathbf{x}))$$

  - Soft constraints: $\mathcal{S} = \{s_i \mid i \in \mathcal{F}\}$

- **Claim**: Each MUS of $(\mathcal{B},\mathcal{S})$ is an AXp & each MCS of $(\mathcal{B},\mathcal{S})$ is a CXp

# Computing AXps (resp. CXps) as MUSes (resp. MCSes)

- Recap:

$$\begin{aligned} \text{WAXp}(\mathcal{X}) &:= \neg\left[\exists(\mathbf{x} \in \mathbb{F}).\ \bigwedge\nolimits_{j\in\mathcal{X}}(x_j = v_j) \land (\neg\sigma(\mathbf{x}))\right] \\ \text{WCXp}(\mathcal{Y}) &:= \exists(\mathbf{x} \in \mathbb{F}).\ \bigwedge\nolimits_{j\notin\mathcal{Y}}(x_j = v_j) \land (\neg\sigma(\mathbf{x})) \end{aligned}$$

- Let,
  - Hard constraints, $\mathcal{B}$:

$$\mathcal{B} := \land_{i\in\mathcal{F}}\,(s_i \rightarrow (x_i = v_i)) \land \text{Encode}_{\mathcal{T}}(\neg\sigma(\mathbf{x}))$$

  - Soft constraints: $\mathcal{S} = \{s_i \mid i \in \mathcal{F}\}$

- **Claim**: Each MUS of $(\mathcal{B}, \mathcal{S})$ is an AXp & each MCS of $(\mathcal{B}, \mathcal{S})$ is a CXp
  - Can use MUS/MCS algorithms for finding AXps/CXps

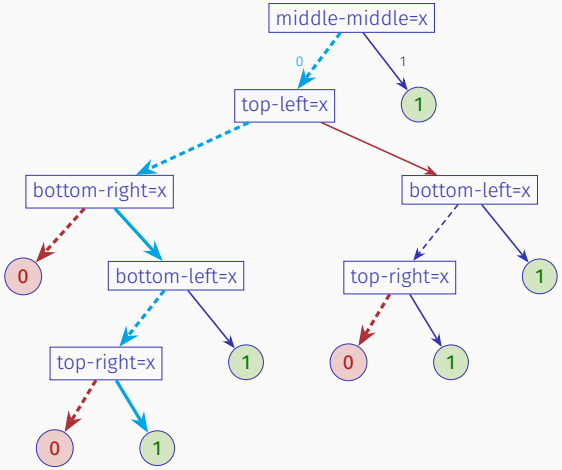
Unit #03

# Tractability in Symbolic XAI

# Outline – Unit #03

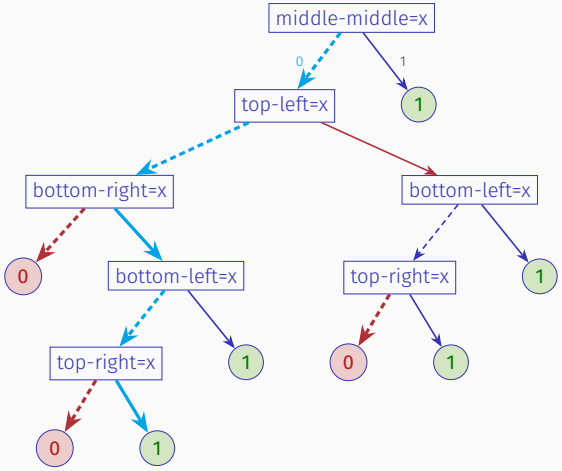
# DT explanations

[IIM20]



# DT explanations

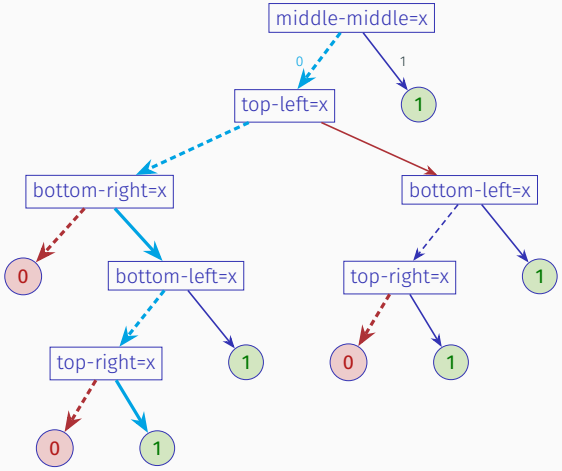
[IIM20]




- Run PI-explanation algorithm based on NP-oracles
  - Worst-case exponential time

# DT explanations

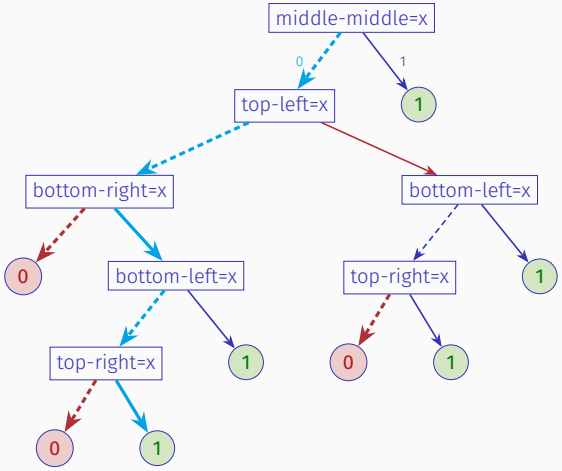
[IIM20]




- Run PI-explanation algorithm based on NP-oracles
  - Worst-case exponential time
- For prediction 1, it suffices to ensure all paths with prediction 0 remain inconsistent

# DT explanations in polynomial time

[IIM20]




- Run PI-explanation algorithm based on NP-oracles
  - Worst-case exponential time
- For prediction 1, it suffices to ensure all paths with prediction 0 remain inconsistent
  - I.e. find a subset-minimal hitting set of all 0 paths; these are the features to keep
    - E.g. BR and TR suffice for prediction
  - Well-known to be solvable in polynomial time [EG95]

# Outline – Unit #03

# Answering queries in DTs

- Finding one AXp in polynomial-time – covered

# Answering queries in DTs

- Finding one AXp in polynomial-time – covered
- Finding one CXp in polynomial-time

# Answering queries in DTs

- Finding one AXp in polynomial-time – covered
- Finding one CXp in polynomial-time
- Finding all CXps in polynomial-time

# Answering queries in DTs

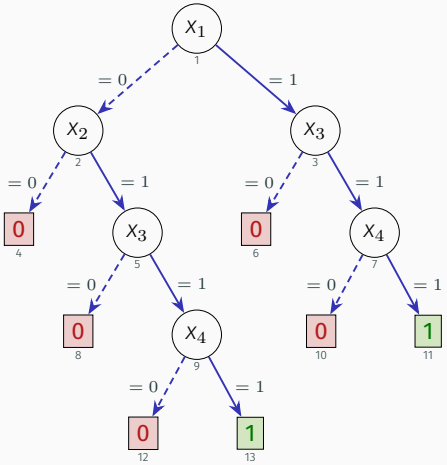
- Finding one AXp in polynomial-time – covered
- Finding one CXp in polynomial-time
- Finding all CXps in polynomial-time; hence, finding one CXp also in polynomial-time

# Answering queries in DTs

- Finding one AXp in polynomial-time – covered
- Finding one CXp in polynomial-time
- Finding all CXps in polynomial-time; hence, finding one CXp also in polynomial-time
- Practically efficient enumeration of AXps – later

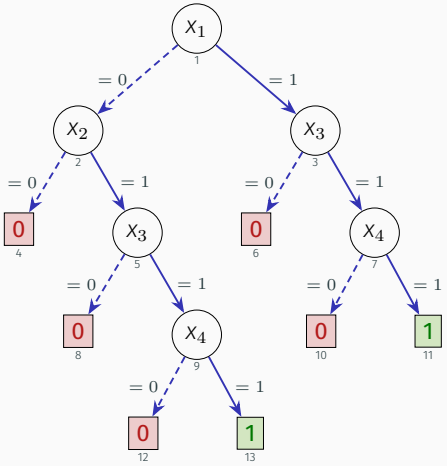
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \varnothing$



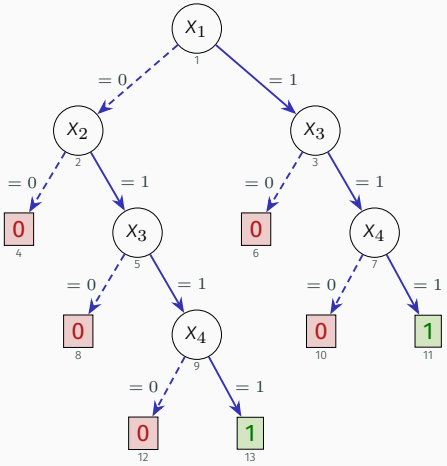
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \varnothing$
  - For each leaf node not predicting $q$:



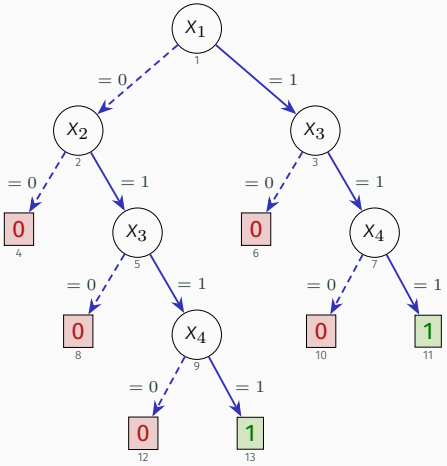
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \varnothing$
  - For each leaf node not predicting $q$:
    - $\mathcal{I}$: features with literals inconsistent with $\mathbf{v}$



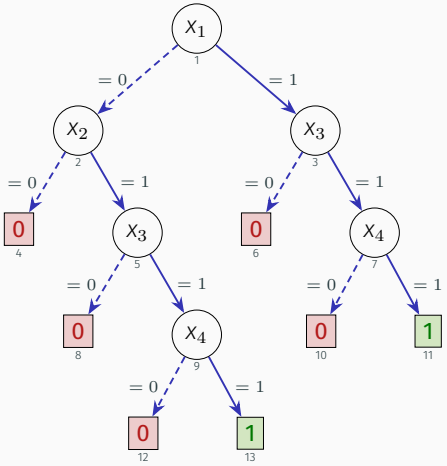
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \varnothing$
  - For each leaf node not predicting $q$:
    - $\mathcal{I}$: features with literals inconsistent with $\mathbf{v}$
    - Add $\mathcal{I}$ to $\mathcal{L}$



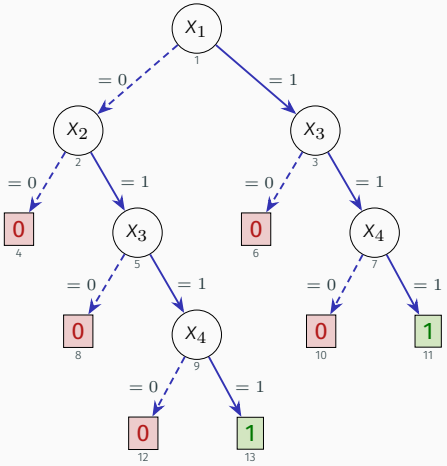
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \varnothing$
  - For each leaf node not predicting $q$:
    - $\mathcal{I}$: features with literals inconsistent with $\mathbf{v}$
    - Add $\mathcal{I}$ to $\mathcal{L}$
  - Remove from $\mathcal{L}$ non-minimal sets



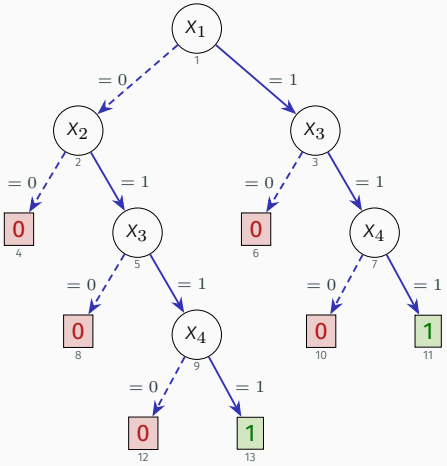
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \varnothing$
  - For each leaf node not predicting $q$:
    - $\mathcal{I}$: features with literals inconsistent with $\mathbf{v}$
    - Add $\mathcal{I}$ to $\mathcal{L}$
  - Remove from $\mathcal{L}$ non-minimal sets
  - $\mathcal{L}$ contains all the CXps of the DT



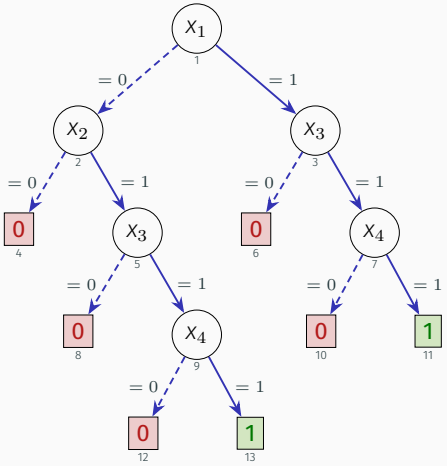
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \varnothing$
  - For each leaf node not predicting $q$:
    - $\mathcal{I}$: features with literals inconsistent with $\mathbf{v}$
    - Add $\mathcal{I}$ to $\mathcal{L}$
  - Remove from $\mathcal{L}$ non-minimal sets
  - $\mathcal{L}$ contains all the CXps of the DT
- Example: instance is $((1, 1, 1, 1), 1)$



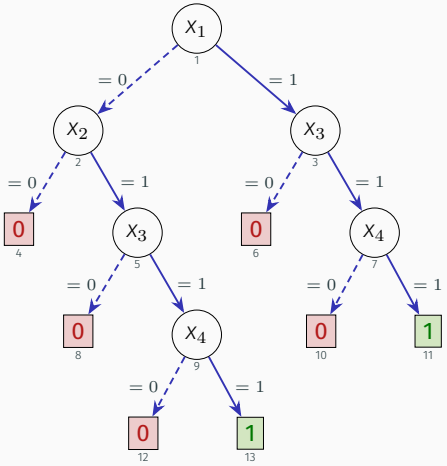
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \varnothing$
  - For each leaf node not predicting $q$:
    - $\mathcal{I}$: features with literals inconsistent with $\mathbf{v}$
    - Add $\mathcal{I}$ to $\mathcal{L}$
  - Remove from $\mathcal{L}$ non-minimal sets
  - $\mathcal{L}$ contains all the CXps of the DT
- Example: instance is $((1, 1, 1, 1), 1)$
  - Add $\{1, 2\}$ to $\mathcal{L}$



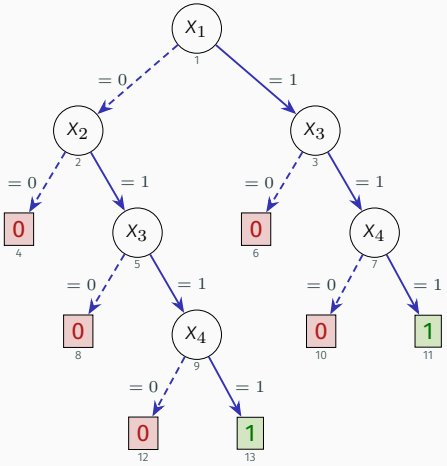
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \emptyset$
  - For each leaf node not predicting $q$:
    - $\mathcal{I}$: features with literals inconsistent with $\mathbf{v}$
    - Add $\mathcal{I}$ to $\mathcal{L}$
  - Remove from $\mathcal{L}$ non-minimal sets
  - $\mathcal{L}$ contains all the CXps of the DT
- Example: instance is $((1, 1, 1, 1), 1)$
  - Add $\{1, 2\}$ to $\mathcal{L}$
  - Add $\{1, 3\}$ to $\mathcal{L}$



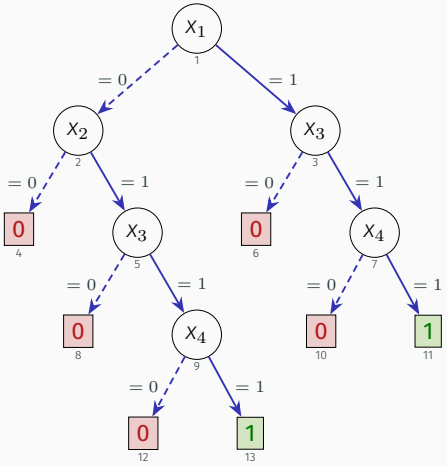
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \emptyset$
  - For each leaf node not predicting $q$:
    - $\mathcal{I}$: features with literals inconsistent with $\mathbf{v}$
    - Add $\mathcal{I}$ to $\mathcal{L}$
  - Remove from $\mathcal{L}$ non-minimal sets
  - $\mathcal{L}$ contains all the CXps of the DT
- Example: instance is $((1, 1, 1, 1), 1)$
  - Add $\{1, 2\}$ to $\mathcal{L}$
  - Add $\{1, 3\}$ to $\mathcal{L}$
  - Add $\{1, 4\}$ to $\mathcal{L}$



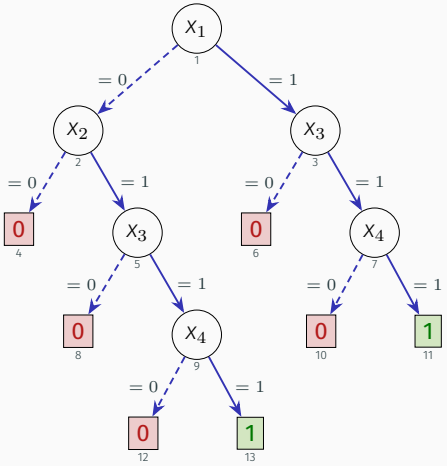
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \emptyset$
  - For each leaf node not predicting $q$:
    - $\mathcal{I}$: features with literals inconsistent with $\mathbf{v}$
    - Add $\mathcal{I}$ to $\mathcal{L}$
  - Remove from $\mathcal{L}$ non-minimal sets
  - $\mathcal{L}$ contains all the CXps of the DT
- Example: instance is $((1, 1, 1, 1), 1)$
  - Add $\{1, 2\}$ to $\mathcal{L}$
  - Add $\{1, 3\}$ to $\mathcal{L}$
  - Add $\{1, 4\}$ to $\mathcal{L}$
  - Add $\{3\}$ to $\mathcal{L}$



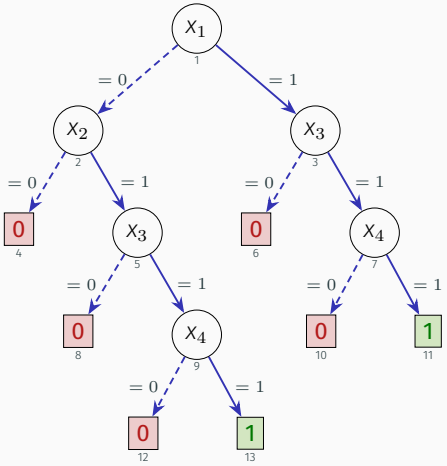
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \emptyset$
  - For each leaf node not predicting $q$:
    - $\mathcal{I}$: features with literals inconsistent with $\mathbf{v}$
    - Add $\mathcal{I}$ to $\mathcal{L}$
  - Remove from $\mathcal{L}$ non-minimal sets
  - $\mathcal{L}$ contains all the CXps of the DT
- Example: instance is $((1, 1, 1, 1), 1)$
  - Add $\{1, 2\}$ to $\mathcal{L}$
  - Add $\{1, 3\}$ to $\mathcal{L}$
  - Add $\{1, 4\}$ to $\mathcal{L}$
  - Add $\{3\}$ to $\mathcal{L}$
  - Add $\{4\}$ to $\mathcal{L}$



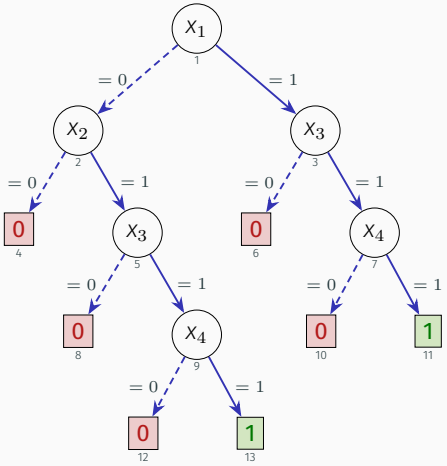
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \emptyset$
  - For each leaf node not predicting $q$:
    - $\mathcal{I}$: features with literals inconsistent with $\mathbf{v}$
    - Add $\mathcal{I}$ to $\mathcal{L}$
  - Remove from $\mathcal{L}$ non-minimal sets
  - $\mathcal{L}$ contains all the CXps of the DT
- Example: instance is $((1, 1, 1, 1), 1)$
  - Add $\{1, 2\}$ to $\mathcal{L}$
  - Add $\{1, 3\}$ to $\mathcal{L}$
  - Add $\{1, 4\}$ to $\mathcal{L}$
  - Add $\{3\}$ to $\mathcal{L}$
  - Add $\{4\}$ to $\mathcal{L}$
  - Remove from $\mathcal{L}$: $\{1, 3\}$ and $\{1, 4\}$



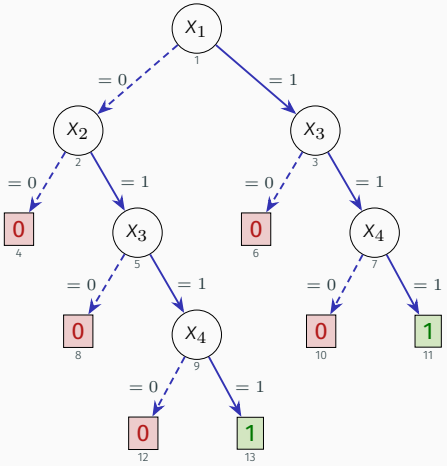
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \emptyset$
  - For each leaf node not predicting $q$:
    - $\mathcal{I}$: features with literals inconsistent with $\mathbf{v}$
    - Add $\mathcal{I}$ to $\mathcal{L}$
  - Remove from $\mathcal{L}$ non-minimal sets
  - $\mathcal{L}$ contains all the CXps of the DT
- Example: instance is $((1, 1, 1, 1), 1)$
  - Add $\{1, 2\}$ to $\mathcal{L}$
  - Add $\{1, 3\}$ to $\mathcal{L}$
  - Add $\{1, 4\}$ to $\mathcal{L}$
  - Add $\{3\}$ to $\mathcal{L}$
  - Add $\{4\}$ to $\mathcal{L}$
  - Remove from $\mathcal{L}$: $\{1, 3\}$ and $\{1, 4\}$
  - CXps: $\{\{1, 2\}, \{3\}, \{4\}\}$



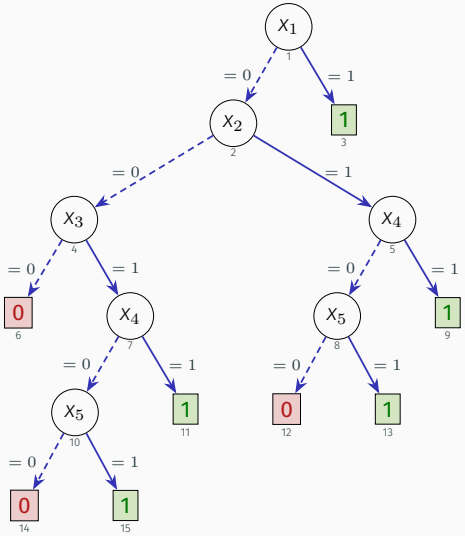
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \emptyset$
  - For each leaf node not predicting $q$:
    - $\mathcal{I}$: features with literals inconsistent with $\mathbf{v}$
    - Add $\mathcal{I}$ to $\mathcal{L}$
  - Remove from $\mathcal{L}$ non-minimal sets
  - $\mathcal{L}$ contains all the CXps of the DT
- Example: instance is $((1, 1, 1, 1), 1)$
  - Add $\{1, 2\}$ to $\mathcal{L}$
  - Add $\{1, 3\}$ to $\mathcal{L}$
  - Add $\{1, 4\}$ to $\mathcal{L}$
  - Add $\{3\}$ to $\mathcal{L}$
  - Add $\{4\}$ to $\mathcal{L}$
  - Remove from $\mathcal{L}$: $\{1, 3\}$ and $\{1, 4\}$
  - CXps: $\{\{1, 2\}, \{3\}, \{4\}\}$
  - AXps: $\{\{1, 3, 4\}, \{2, 3, 4\}\}$, by computing all MHSes



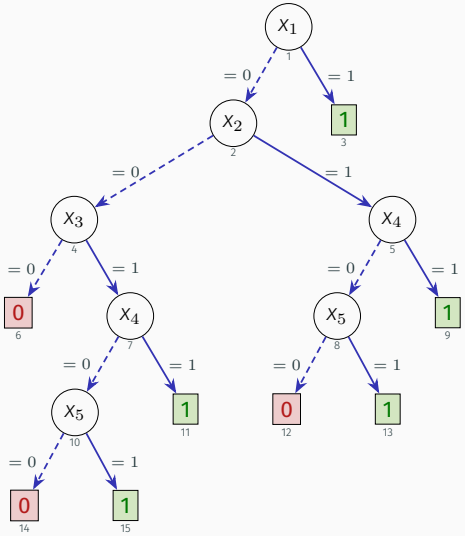
# Outline – Unit #03

# Are *interpretable* models really interpretable? – DTs



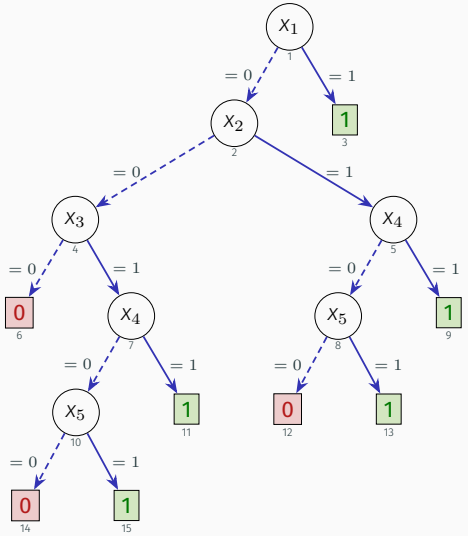

- Case of optimal decision tree (DT) [HRS19]
- Explanation for $(0, 0, 1, 0, 1)$, with prediction 1?

# Are *interpretable* models really interpretable? – DTs




- Case of optimal decision tree (DT) [HRS19]
- Explanation for $(0, 0, 1, 0, 1)$, with prediction 1?
  - Clearly, IF $\neg x_1 \land \neg x_2 \land x_3 \land \neg x_4 \land x_5$ THEN $\kappa(\mathbf{x}) = 1$

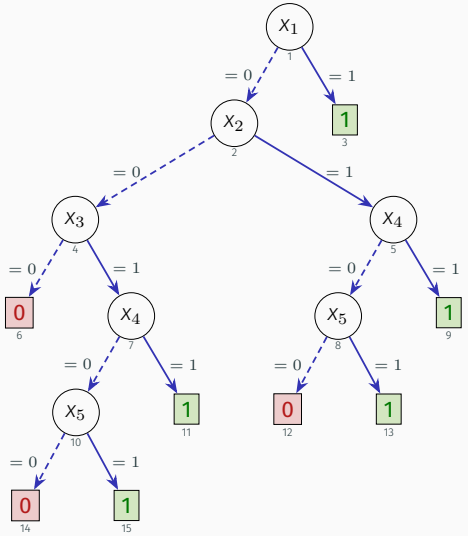
# Are *interpretable* models really interpretable? – DTs




- Case of **optimal** decision tree (DT) [HRS19]
- Explanation for $(0, 0, 1, 0, 1)$, with prediction 1?
  - Clearly, IF $\neg x_1 \land \neg x_2 \land x_3 \land \neg x_4 \land x_5$ THEN $\kappa(\mathbf{x}) = 1$
  - But, $x_1$, $x_2$, $x_4$ are irrelevant for the prediction:

| $x_3$ | $x_5$ | $x_1$ | $x_2$ | $x_4$ | $\kappa(\mathbf{x})$ |
|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

# Are *interpretable* models really interpretable? – DTs



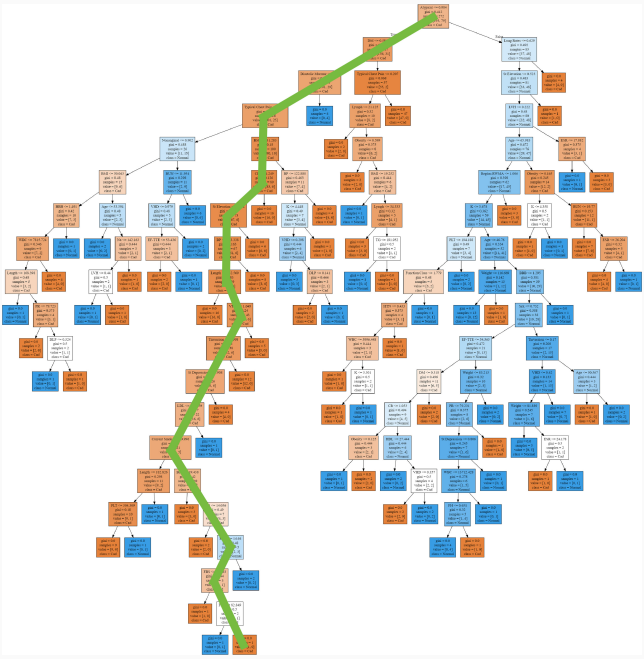

- Case of **optimal** decision tree (DT) [HRS19]
- Explanation for $(0, 0, 1, 0, 1)$, with prediction 1?
  - Clearly, IF $\neg x_1 \land \neg x_2 \land x_3 \land \neg x_4 \land x_5$ THEN $\kappa(\mathbf{x}) = 1$
  - But, $x_1$, $x_2$, $x_4$ are irrelevant for the prediction:

| $x_3$ | $x_5$ | $x_1$ | $x_2$ | $x_4$ | $\kappa(\mathbf{x})$ |
|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

$\therefore$ one AXp is $\{3, 5\}$

Compare with $\{1, 2, 3, 4, 5\}$...

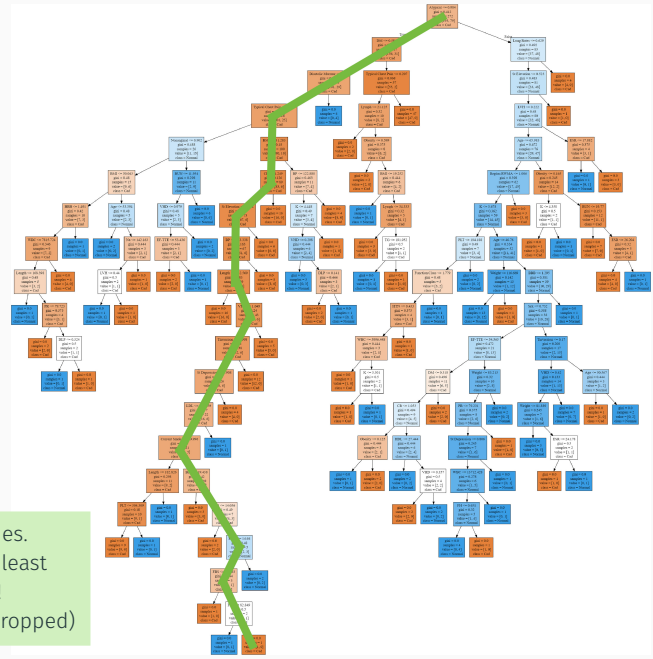
# Are *interpretable* models really interpretable? – large DTs



[GZM20]

# Are *interpretable* models really interpretable? – large DTs

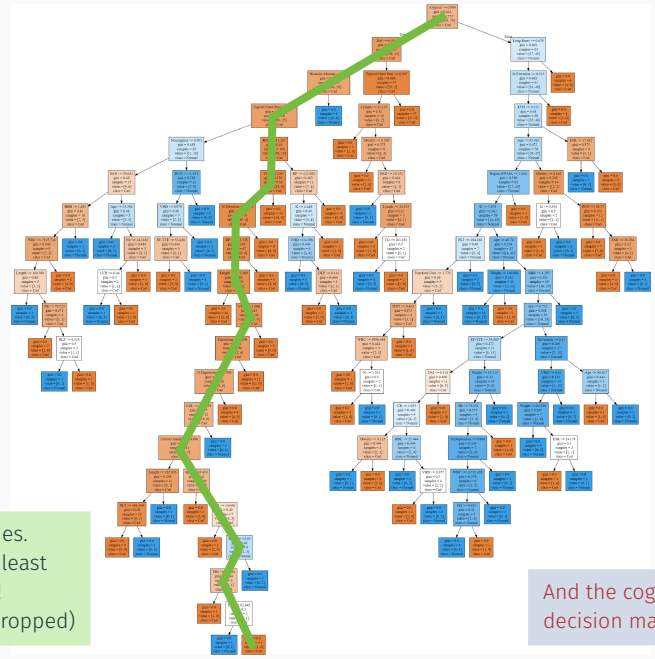
[GZM20]



Path with 19 internal nodes.
By manual inspection, at least 10 literals are redundant!
(And at least 9 features dropped)

# Are *interpretable* models really interpretable? – large DTs

[GZM20]



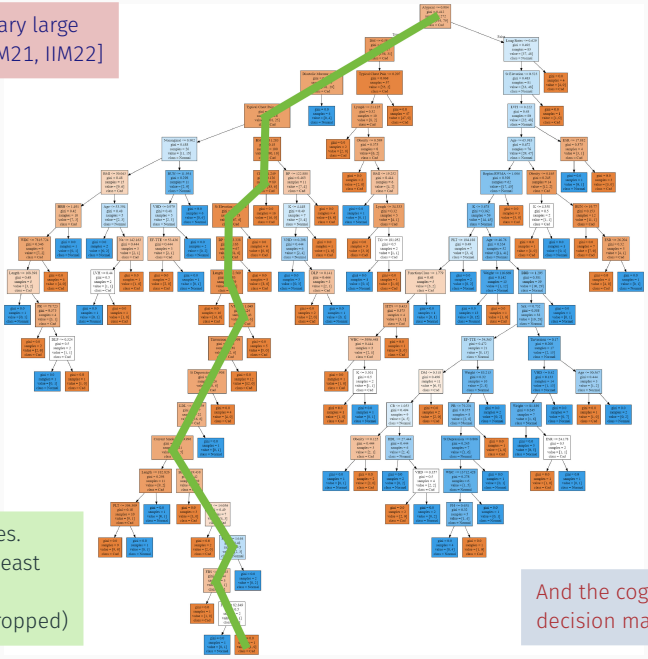
Path with 19 internal nodes.
By manual inspection, at least 10 literals are redundant!
(And at least 9 features dropped)

And the cognitive limits of human decision makers are well-known [Mil56]

# Are *interpretable* models really interpretable? – large DTs

Redundancy can be arbitrary large on path length [IIM20, HIIM21, IIM22]

[GZM20]



Path with 19 internal nodes.
By manual inspection, at least 10 literals are redundant!
(And at least 9 features dropped)

And the cognitive limits of human decision makers are well-known [Mil56]

# Are *interpretable* models really interpretable? – arbitrary redundancy [IIM20, HIIM21, IIM22]

- Classifier, with $x_1, \ldots, x_m \in \{0, 1\}$:

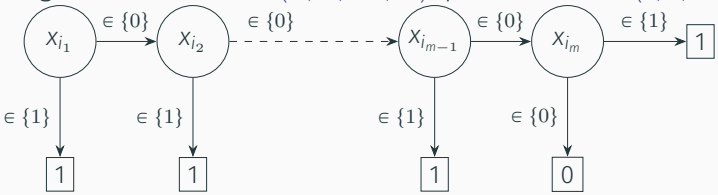
$$\kappa(x_1, x_2, \ldots, x_{m-1}, x_m) = \bigvee_{i=1}^{m} x_i$$

# Are *interpretable* models really interpretable? – arbitrary redundancy [IIM20, HIIM21, IIM22]

- Classifier, with $x_1, \ldots, x_m \in \{0, 1\}$:

$$\kappa(x_1, x_2, \ldots, x_{m-1}, x_m) = \bigvee_{i=1}^{m} x_i$$

- Build DT, by picking variables in order $\langle i_1, i_2, \ldots, i_m \rangle$, permutation of $\langle 1, 2, \ldots, m \rangle$:

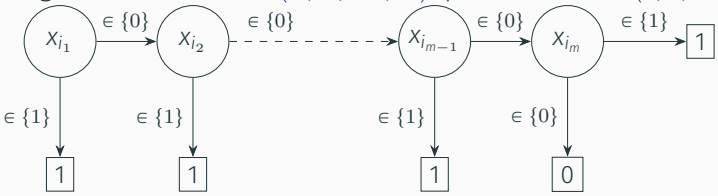


# Are *interpretable* models really interpretable? – arbitrary redundancy [IIM20, HIIM21, IIM22]

- Classifier, with $x_1, \ldots, x_m \in \{0, 1\}$:

$$\kappa(x_1, x_2, \ldots, x_{m-1}, x_m) = \bigvee_{i=1}^{m} x_i$$

- Build DT, by picking variables in order $\langle i_1, i_2, \ldots, i_m \rangle$, permutation of $\langle 1, 2, \ldots, m \rangle$:



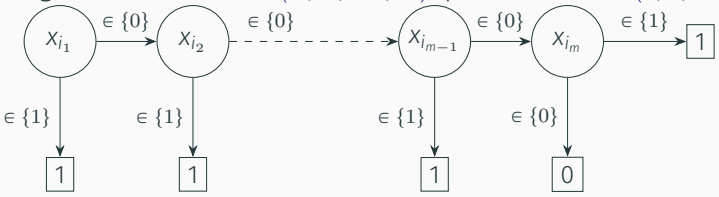

- Point: $(x_{i_1}, x_{i_2}, \ldots, x_{i_{m-1}}, x_{i_m}) = (0, 0, \ldots, 0, 1)$, and prediction 1

# Are *interpretable* models really interpretable? – arbitrary redundancy [IIM20, HIIM21, IIM22]

- Classifier, with $x_1, \ldots, x_m \in \{0, 1\}$:

$$\kappa(x_1, x_2, \ldots, x_{m-1}, x_m) = \bigvee_{i=1}^{m} x_i$$

- Build DT, by picking variables in order $\langle i_1, i_2, \ldots, i_m \rangle$, permutation of $\langle 1, 2, \ldots, m \rangle$:




- Point: $(x_{i_1}, x_{i_2}, \ldots, x_{i_{m-1}}, x_{i_m}) = (0, 0, \ldots, 0, 1)$, and prediction 1
- Explanation using path in DT: $\{i_1, i_2, \ldots, i_m\}$, i.e.

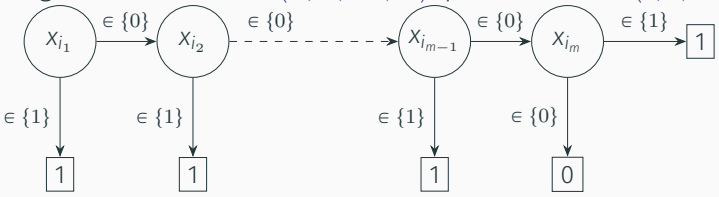
$$(x_{i_1} = 0) \wedge (x_{i_2} = 0) \wedge \ldots \wedge (x_{i_{m-1}} = 0) \wedge (x_{i_m} = 1) \rightarrow \kappa(x_1, \ldots, x_m)$$

# Are *interpretable* models really interpretable? – arbitrary redundancy [IIM20, HIIM21, IIM22]

- Classifier, with $x_1, \ldots, x_m \in \{0, 1\}$:

$$\kappa(x_1, x_2, \ldots, x_{m-1}, x_m) = \bigvee_{i=1}^{m} x_i$$

- Build DT, by picking variables in order $\langle i_1, i_2, \ldots, i_m \rangle$, permutation of $\langle 1, 2, \ldots, m \rangle$:




- Point: $(x_{i_1}, x_{i_2}, \ldots, x_{i_{m-1}}, x_{i_m}) = (0, 0, \ldots, 0, 1)$, and prediction 1
- Explanation using path in DT: $\{i_1, i_2, \ldots, i_m\}$, i.e.

$$(x_{i_1} = 0) \wedge (x_{i_2} = 0) \wedge \ldots \wedge (x_{i_{m-1}} = 0) \wedge (x_{i_m} = 1) \rightarrow \kappa(x_1, \ldots, x_m)$$

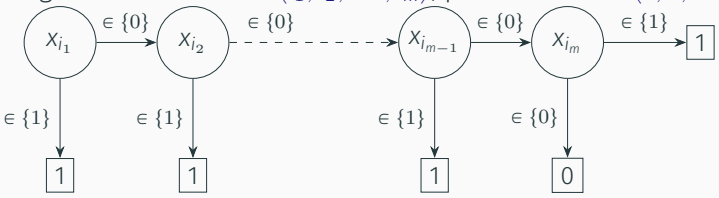
- But $\{i_m\}$ suffices for prediction, i.e. $\forall(\mathbf{x} \in \{0, 1\}^m).(x_{i_m}) \rightarrow \kappa(\mathbf{x})$

# Are *interpretable* models really interpretable? – arbitrary redundancy [IIM20, HIIM21, IIM22]

- Classifier, with $x_1, \ldots, x_m \in \{0, 1\}$:

$$\kappa(x_1, x_2, \ldots, x_{m-1}, x_m) = \bigvee_{i=1}^{m} x_i$$

- Build DT, by picking variables in order $\langle i_1, i_2, \ldots, i_m \rangle$, permutation of $\langle 1, 2, \ldots, m \rangle$:




- Point: $(x_{i_1}, x_{i_2}, \ldots, x_{i_{m-1}}, x_{i_m}) = (0, 0, \ldots, 0, 1)$, and prediction 1
- Explanation using path in DT: $\{i_1, i_2, \ldots, i_m\}$, i.e.

$$(x_{i_1} = 0) \land (x_{i_2} = 0) \land \ldots \land (x_{i_{m-1}} = 0) \land (x_{i_m} = 1) \rightarrow \kappa(x_1, \ldots, x_m)$$

- But $\{i_m\}$ suffices for prediction, i.e. $\forall(\mathbf{x} \in \{0, 1\}^m).(x_{i_m}) \rightarrow \kappa(\mathbf{x})$
- AXp's can be arbitrarily smaller than paths in (optimal) DTs! [IIM20, IIM22]

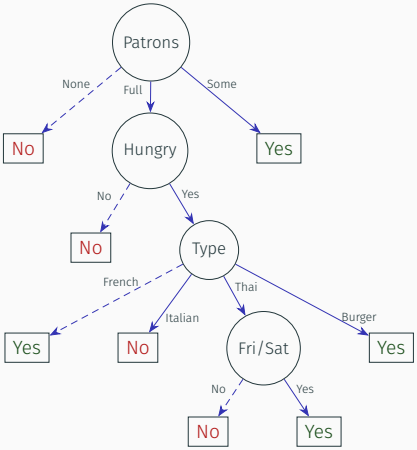
# Explanation redundancy in DTs is ubiquitous – published DT examples

[IIM22]

| DT Ref | D | #N | #P | %R | %C | %m | %M | %avg |
|---|---|---|---|---|---|---|---|---|
| [Alp14, Ch. 09, Fig. 9.1] | 2 | 5 | 3 | 33 | 25 | 50 | 50 | 50 |
| [Alp16, Ch. 03, Fig. 3.2] | 2 | 5 | 3 | 33 | 25 | 50 | 50 | 50 |
| [Bra20, Ch. 01, Fig. 1.3] | 4 | 9 | 5 | 60 | 25 | 25 | 50 | 36 |
| [BA97, Figure 1] | 3 | 12 | 7 | 14 | 8 | 33 | 33 | 33 |
| [BBHK10, Ch. 08, Fig. 8.2] | 3 | 7 | 4 | 25 | 12 | 50 | 50 | 50 |
| [BFOS84, Ch. 01, Fig. 1.1] | 3 | 7 | 4 | 50 | 25 | 33 | 33 | 33 |
| [DL01, Ch. 01, Fig. 1.2a] | 2 | 5 | 3 | 33 | 25 | 33 | 33 | 33 |
| [DL01, Ch. 01, Fig. 1.2b] | 2 | 5 | 3 | 33 | 25 | 33 | 33 | 33 |
| [KMND20, Ch. 04, Fig. 4.14] | 3 | 7 | 4 | 25 | 12 | 50 | 50 | 50 |
| [KMND20, Sec. 4.7, Ex. 4] | 2 | 5 | 3 | 33 | 25 | 50 | 50 | 50 |
| [Qui93, Ch. 01, Fig. 1.3] | 3 | 12 | 7 | 28 | 17 | 33 | 50 | 41 |
| [RM08, Ch. 01, Fig. 1.5] | 3 | 9 | 5 | 20 | 12 | 33 | 33 | 33 |
| [RM08, Ch. 01, Fig. 1.4] | 3 | 7 | 4 | 50 | 25 | 33 | 33 | 33 |
| [WFHP17, Ch. 01, Fig. 1.2] | 3 | 7 | 4 | 25 | 12 | 50 | 50 | 50 |
| [VLE+16, Figure 4] | 6 | 39 | 20 | 65 | 63 | 20 | 40 | 33 |
| [Fla12, Ch. 02, Fig. 2.1(right)] | 2 | 5 | 3 | 33 | 25 | 50 | 50 | 50 |
| [Kot13, Figure 1] | 3 | 10 | 6 | 33 | 11 | 33 | 33 | 33 |
| [Mor82, Figure 1] | 3 | 9 | 5 | 80 | 75 | 33 | 50 | 41 |
| [PM17, Ch. 07, Fig. 7.4] | 3 | 7 | 4 | 50 | 25 | 33 | 33 | 33 |
| [RN10, Ch. 18, Fig. 18.6] | 4 | 12 | 8 | 25 | 6 | 25 | 33 | 29 |
| [SB14, Ch. 18, Page 212] | 2 | 5 | 3 | 33 | 25 | 50 | 50 | 50 |
| [Zho12, Ch. 01, Fig. 1.3] | 2 | 5 | 3 | 33 | 25 | 33 | 33 | 33 |
| [BHO09, Figure 1b] | 4 | 13 | 7 | 71 | 50 | 33 | 50 | 36 |
| [Zho21, Ch. 04, Fig. 4.3] | 4 | 14 | 9 | 11 | 2 | 25 | 25 | 25 |

# Many DTs have paths that are not minimal XPs – Russell&Norvig's book

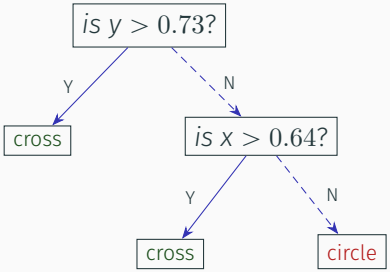
[RN10]




- Explanation for $(P, H, T, W) = (\text{Full}, \text{Yes}, \text{Thai}, \text{No})$?

# Many DTs have paths that are not minimal XPs – Zhou's book

[Zho12]

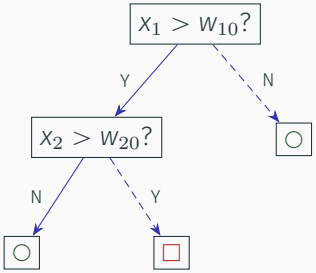



- Explanation for $(x, y) = (1.25, -1.13)$?

Obs: True explanations can be computed for categorical, integer or real-valued features !

# Many DTs have paths that are not minimal XPs – Alpaydin's book

[Alp14]

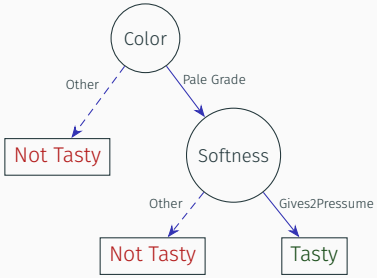



- Explanation for $(x_1, x_2) = (\alpha, \beta)$, with $\alpha > w_{10}$ and $\beta \leqslant w_{20}$?

Obs: True explanations can be computed for categorical, integer or real-valued features !

# Many DTs have paths that are not minimal XPs – S.-S.&B.-D.'s book

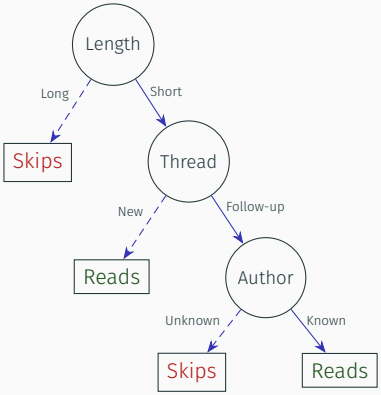
[SB14]




- Explanation for $(\text{color}, \text{softness}) = (\text{Pale Grade}, \text{Other})$?

# Many DTs have paths that are not minimal XPs – Poole&Mackworth's book

[PM17]




- Explanation for $(L, T, A) = (\text{Short}, \text{Follow-Up}, \text{Unknown})$?
- Explanation for $(L, T, A) = (\text{Short}, \text{Follow-Up}, \text{Known})$?

# Explanation redundancy in DTs is ubiquitous – DTs from datasets

[IIM20, HIIM21, IIM22]

| Dataset | (#F | #S) | IAI | | | | | | | | | ITI | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | D | #N | %A | #P | %R | %C | %m | %M | %avg | D | #N | %A | #P | %R | %C | %m | %M | %avg |
| adult | ( 12 | 6061) | 6 | 83 | 78 | 42 | 33 | 25 | 20 | 40 | 25 | 17 | 509 | 73 | 255 | 75 | 91 | 10 | 66 | 22 |
| anneal | ( 38 | 886) | 6 | 29 | 99 | 15 | 26 | 16 | 16 | 33 | 21 | 9 | 31 | 100 | 16 | 25 | 4 | 12 | 20 | 16 |
| backache | ( 32 | 180) | 4 | 17 | 72 | 9 | 33 | 39 | 25 | 33 | 30 | 3 | 9 | 91 | 5 | 80 | 87 | 50 | 66 | 54 |
| bank | ( 19 | 36293) | 6 | 113 | 88 | 57 | 5 | 12 | 16 | 20 | 18 | 19 | 1467 | 86 | 734 | 69 | 64 | 7 | 63 | 27 |
| biodegradation | ( 41 | 1052) | 5 | 19 | 65 | 10 | 30 | 1 | 25 | 50 | 33 | 8 | 71 | 76 | 36 | 50 | 8 | 14 | 40 | 21 |
| cancer | ( 9 | 449) | 6 | 37 | 87 | 19 | 36 | 9 | 20 | 25 | 21 | 5 | 21 | 84 | 11 | 54 | 10 | 25 | 50 | 37 |
| car | ( 6 | 1728) | 6 | 43 | 96 | 22 | 86 | 89 | 20 | 80 | 45 | 11 | 57 | 98 | 29 | 65 | 41 | 16 | 50 | 30 |
| colic | ( 22 | 357) | 6 | 55 | 81 | 28 | 46 | 6 | 16 | 33 | 20 | 4 | 17 | 80 | 9 | 33 | 27 | 25 | 25 | 25 |
| compas | ( 11 | 1155) | 6 | 77 | 34 | 39 | 17 | 8 | 16 | 20 | 17 | 15 | 183 | 37 | 92 | 66 | 43 | 12 | 60 | 27 |
| contraceptive | ( 9 | 1425) | 6 | 99 | 49 | 50 | 8 | 2 | 20 | 60 | 37 | 17 | 385 | 48 | 193 | 27 | 32 | 12 | 66 | 21 |
| dermatology | ( 34 | 366) | 6 | 33 | 90 | 17 | 23 | 3 | 16 | 33 | 21 | 7 | 17 | 95 | 9 | 22 | 0 | 14 | 20 | 17 |
| divorce | ( 54 | 150) | 5 | 15 | 90 | 8 | 50 | 19 | 20 | 33 | 24 | 2 | 5 | 96 | 3 | 33 | 16 | 50 | 50 | 50 |
| german | ( 21 | 1000) | 6 | 25 | 61 | 13 | 38 | 10 | 20 | 40 | 29 | 10 | 99 | 72 | 50 | 46 | 13 | 12 | 40 | 22 |
| heart-c | ( 13 | 302) | 6 | 43 | 65 | 22 | 36 | 18 | 20 | 33 | 22 | 4 | 15 | 75 | 8 | 87 | 81 | 25 | 50 | 34 |
| heart-h | ( 13 | 293) | 6 | 37 | 59 | 19 | 31 | 4 | 20 | 40 | 24 | 8 | 25 | 77 | 13 | 61 | 60 | 20 | 50 | 32 |
| kr-vs-kp | ( 36 | 3196) | 6 | 49 | 96 | 25 | 80 | 75 | 16 | 60 | 33 | 13 | 67 | 99 | 34 | 79 | 43 | 7 | 70 | 35 |
| lending | ( 9 | 5082) | 6 | 45 | 73 | 23 | 73 | 80 | 16 | 50 | 25 | 14 | 507 | 65 | 254 | 69 | 80 | 12 | 75 | 25 |
| letter | ( 16 | 18668) | 6 | 127 | 58 | 64 | 1 | 0 | 20 | 20 | 20 | 46 | 4857 | 68 | 2429 | 6 | 7 | 6 | 25 | 9 |
| lymphography | ( 18 | 148) | 6 | 61 | 76 | 31 | 35 | 25 | 16 | 33 | 21 | 6 | 21 | 86 | 11 | 9 | 0 | 16 | 16 | 16 |
| mortality | (118 | 13442) | 6 | 111 | 74 | 56 | 8 | 14 | 16 | 20 | 17 | 26 | 865 | 76 | 433 | 61 | 61 | 7 | 54 | 19 |
| mushroom | ( 22 | 8124) | 6 | 39 | 100 | 20 | 80 | 44 | 16 | 33 | 24 | 5 | 23 | 100 | 12 | 50 | 31 | 20 | 40 | 25 |
| pendigits | ( 16 | 10992) | 6 | 121 | 88 | 61 | 0 | 0 | — | — | — | 38 | 937 | 85 | 469 | 25 | 86 | 6 | 25 | 11 |
| promoters | ( 58 | 106) | 1 | 3 | 90 | 2 | 0 | 0 | — | — | — | 3 | 9 | 81 | 5 | 20 | 14 | 33 | 33 | 33 |
| recidivism | ( 15 | 3998) | 6 | 105 | 61 | 53 | 28 | 22 | 16 | 33 | 18 | 15 | 611 | 51 | 306 | 53 | 38 | 9 | 44 | 16 |
| seismic_bumps | ( 18 | 2578) | 6 | 37 | 89 | 19 | 42 | 19 | 20 | 33 | 24 | 8 | 39 | 93 | 20 | 60 | 79 | 20 | 60 | 42 |
| shuttle | ( 9 | 58000) | 6 | 63 | 99 | 32 | 28 | 7 | 20 | 33 | 23 | 23 | 159 | 99 | 80 | 33 | 9 | 14 | 50 | 30 |
| soybean | ( 35 | 623) | 6 | 63 | 88 | 32 | 9 | 5 | 25 | 25 | 25 | 16 | 71 | 89 | 36 | 22 | 1 | 9 | 12 | 10 |
| spambase | ( 57 | 4210) | 6 | 63 | 75 | 32 | 37 | 12 | 16 | 33 | 19 | 15 | 143 | 91 | 72 | 76 | 98 | 7 | 58 | 25 |
| spect | ( 22 | 228) | 6 | 45 | 82 | 23 | 60 | 51 | 20 | 50 | 35 | 6 | 15 | 86 | 8 | 87 | 98 | 50 | 83 | 65 |
| splice | ( 2 | 3178) | 3 | 7 | 50 | 4 | 0 | 0 | — | — | — | 88 | 177 | 55 | 89 | 0 | 0 | — | — | — |

$$
\begin{array}{lllll}
\mathsf{R}_1: & \text{IF} & (x_1 \wedge x_3) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_2: & \text{ELSE IF} & (x_2 \wedge x_4 \wedge x_6) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_3: & \text{ELSE IF} & (\neg x_1 \wedge x_3) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_4: & \text{ELSE IF} & (x_4 \wedge x_6) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_5: & \text{ELSE IF} & (\neg x_1 \wedge \neg x_3) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_6: & \text{ELSE IF} & (x_6) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_{\text{DEF}}: & \text{ELSE} & & & \kappa(\mathbf{x}) = 1
\end{array}
$$

- Instance: $((0, 1, 0, 1, 0, 1), 0)$, i.e. rule $\mathsf{R}_2$ fires

$$
\begin{array}{lllll}
\mathsf{R}_1: & \text{IF} & (x_1 \wedge x_3) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_2: & \text{ELSE IF} & (x_2 \wedge x_4 \wedge x_6) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_3: & \text{ELSE IF} & (\neg x_1 \wedge x_3) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_4: & \text{ELSE IF} & (x_4 \wedge x_6) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_5: & \text{ELSE IF} & (\neg x_1 \wedge \neg x_3) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_6: & \text{ELSE IF} & (x_6) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_{\text{DEF}}: & \text{ELSE} & & & \kappa(\mathbf{x}) = 1
\end{array}
$$

- Instance: $((0, 1, 0, 1, 0, 1), 0)$, i.e. rule $\mathsf{R}_2$ fires
- What is the abductive explanation?

$$
\begin{array}{lllll}
\mathsf{R}_1: & \text{IF} & (x_1 \land x_3) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_2: & \text{ELSE IF} & (x_2 \land x_4 \land x_6) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_3: & \text{ELSE IF} & (\neg x_1 \land x_3) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_4: & \text{ELSE IF} & (x_4 \land x_6) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_5: & \text{ELSE IF} & (\neg x_1 \land \neg x_3) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_6: & \text{ELSE IF} & (x_6) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_{\mathsf{DEF}}: & \text{ELSE} & & & \kappa(\mathbf{x}) = 1
\end{array}
$$

- Instance: $((0, 1, 0, 1, 0, 1), 0)$, i.e. rule $\mathsf{R}_2$ fires
- What is the abductive explanation?
- Recall: one AXp is $\{3, 4, 6\}$

$$
\begin{array}{lllll}
\mathsf{R}_1: & \text{IF} & (x_1 \land x_3) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_2: & \text{ELSE IF} & (x_2 \land x_4 \land x_6) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_3: & \text{ELSE IF} & (\neg x_1 \land x_3) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_4: & \text{ELSE IF} & (x_4 \land x_6) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_5: & \text{ELSE IF} & (\neg x_1 \land \neg x_3) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_6: & \text{ELSE IF} & (x_6) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_{\mathsf{DEF}}: & \text{ELSE} & & & \kappa(\mathbf{x}) = 1
\end{array}
$$

- Instance: $((0, 1, 0, 1, 0, 1), 0)$, i.e. rule $\mathsf{R}_2$ fires
- What is the abductive explanation?
- Recall: one AXp is $\{3, 4, 6\}$
  - Why?
    - We need 3 (or 1) so that $\mathsf{R}_1$ cannot fire
    - With 3, we do not need 2, since with 4 and 6 fixed, then $\mathsf{R}_4$ is guaranteed to fire
  - Some questions:
    - Would average human decision maker be able to understand the AXp?
    - Would he/she be able to compute one AXp, by manual inspection?

# Are *interpretable* models really interpretable? – DLs

[MSI23]

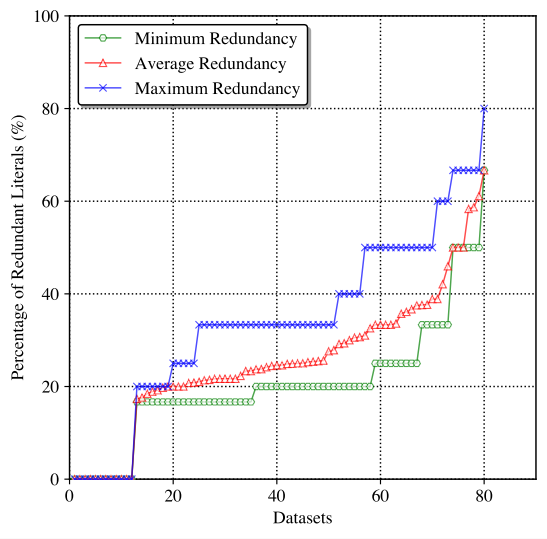
$$
\begin{array}{lllll}
\mathsf{R}_1: & \text{IF} & (x_1 \land x_3) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_2: & \text{ELSE IF} & (x_2 \land x_4 \land x_6) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_3: & \text{ELSE IF} & (\neg x_1 \land x_3) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_4: & \text{ELSE IF} & (x_4 \land x_6) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_5: & \text{ELSE IF} & (\neg x_1 \land \neg x_3) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_6: & \text{ELSE IF} & (x_6) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_{\mathsf{DEF}}: & \text{ELSE} & & & \kappa(\mathbf{x}) = 1
\end{array}
$$

- Instance: $((0, 1, 0, 1, 0, 1), 0)$, i.e. rule $\mathsf{R}_2$ fires
- What is the abductive explanation?
- Recall: one AXp is $\{3, 4, 6\}$
  - Why?
    - We need 3 (or 1) so that $\mathsf{R}_1$ cannot fire
    - With 3, we do not need 2, since with 4 and 6 fixed, then $\mathsf{R}_4$ is guaranteed to fire
  - Some questions:
    - Would average human decision maker be able to understand the AXp?
    - Would he/she be able to compute one AXp, by manual inspection?
      (BTW, **we have proved that computing one AXp for DLs is computationally hard...**)

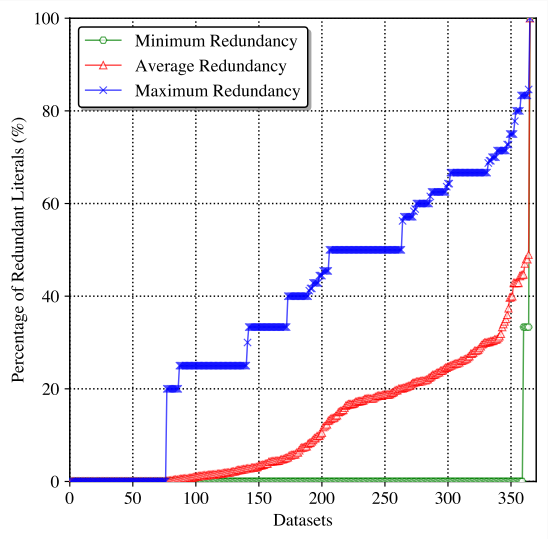
[IM21, MSI23]

# Are *interpretable* models really interpretable? – DTs/DLs in practice

[MSI23]




DTs learned with Interpretable AI, max depth 6




DLs learned with CN2

# Outline – Unit #03

# From DTs to explained DSs

[HM23]

- Decision sets raise a number of issues:
  - Overlap: Two rules with different predictions can fire on the same input
  - Incomplete coverage: For some inputs, no rule may fire
    - A default rule defeats the purpose of unordered rules

# From DTs to explained DSs

[HM23]

- Decision sets raise a number of issues:
  - Overlap: Two rules with different predictions can fire on the same input
  - Incomplete coverage: For some inputs, no rule may fire
    - A default rule defeats the purpose of unordered rules
  - A DS without overlap and complete coverage computes a classification function

# From DTs to explained DSs

[HM23]

- Decision sets raise a number of issues:
  - Overlap: Two rules with different predictions can fire on the same input
  - Incomplete coverage: For some inputs, no rule may fire
    - A default rule defeats the purpose of unordered rules
  - A DS without overlap and complete coverage computes a classification function
- And explaining DSs is computationally hard...

# From DTs to explained DSs

[HM23]

- Decision sets raise a number of issues:
  - Overlap: Two rules with different predictions can fire on the same input
  - Incomplete coverage: For some inputs, no rule may fire
    - A default rule defeats the purpose of unordered rules
  - A DS without overlap and complete coverage computes a classification function
- And explaining DSs is computationally hard...

- One can extract explained DSs from DTs

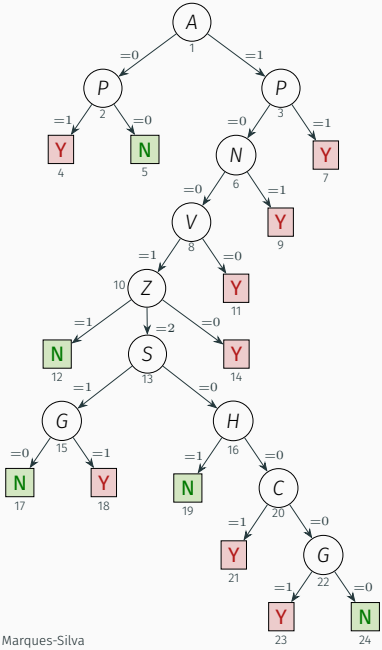
# From DTs to explained DSs

[HM23]

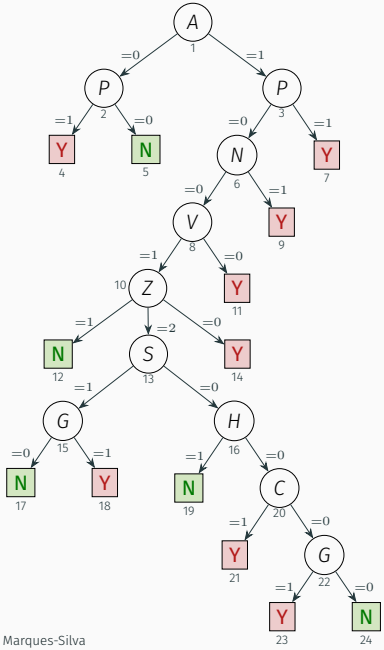
- Decision sets raise a number of issues:
  - Overlap: Two rules with different predictions can fire on the same input
  - Incomplete coverage: For some inputs, no rule may fire
    - A default rule defeats the purpose of unordered rules
  - A DS without overlap and complete coverage computes a classification function
- And explaining DSs is computationally hard...

- One can extract explained DSs from DTs
  - Extract one AXp (viewed as a logic rule) from each path in DT
  - Resulting rules are non-overlapping, and cover feature space

# Example


A
1
=0
=1
P
2
P
3
=1
=0
=0
=1
Y
4
N
5
N
6
Y
7
=0
=1
V
8
Y
9
=1
=0
10
Z
Y
11
=1
=2
=0
N
12
S
13
Y
14
=1
=0
G
15
H
16
=0
=1
=1
=0
N
17
Y
18
N
19
C
20
=1
=0
Y
21
G
22
=1
=0
Y
23
N
24

# Example




$R_{01}$: IF $[P]$ THEN $\kappa(\cdot) = \mathbf{Y}$

$R_{02}$: IF $[\overline{A} \wedge \overline{P}]$THEN $\kappa(\cdot) = \mathbf{N}$

$R_{03}$: IF $[\overline{P} \wedge \overline{N} \wedge V \wedge Z = 1]$ THEN $\kappa(\cdot) = \mathbf{N}$

$R_{04}$: IF $[\overline{P} \wedge \overline{N} \wedge V \wedge Z = 2 \wedge S \wedge \overline{G}]$ THEN $\kappa(\cdot) = \mathbf{N}$

$R_{05}$: IF $[A \wedge Z = 2 \wedge S \wedge G]$ THEN $\kappa(\cdot) = \mathbf{Y}$

$R_{06}$: IF $[\overline{P} \wedge \overline{N} \wedge V \wedge Z = 2 \wedge \overline{S} \wedge H]$ THEN $\kappa(\cdot) = \mathbf{N}$

$R_{07}$: IF $[A \wedge Z = 2 \wedge \overline{S} \wedge \overline{H} \wedge C]$ THEN $\kappa(\cdot) = \mathbf{Y}$

$R_{08}$: IF $[A \wedge Z = 2 \wedge \overline{H} \wedge G]$ THEN $\kappa(\cdot) = \mathbf{Y}$

$R_{09}$: IF $[\overline{P} \wedge \overline{N} \wedge V \wedge Z = 2 \wedge \overline{C} \wedge \overline{G}]$ THEN $\kappa(\cdot) = \mathbf{N}$

$R_{10}$: IF $[A \wedge Z = 0]$ THEN $\kappa(\cdot) = \mathbf{Y}$

$R_{11}$: IF $[A \wedge \overline{V}]$ THEN $\kappa(\cdot) = \mathbf{Y}$

$R_{12}$: IF $[A \wedge N]$ THEN $\kappa(\cdot) = \mathbf{Y}$

# Outline – Unit #03

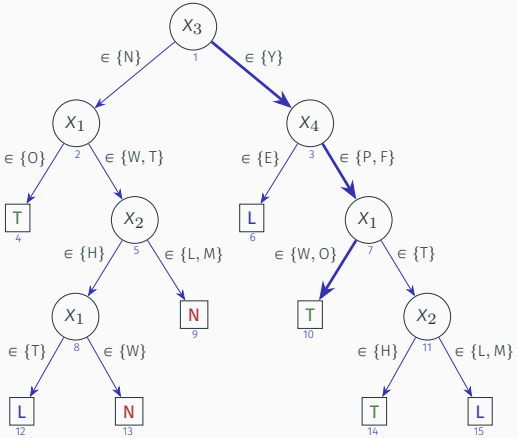
# Explanation graphs – overview of results

[HIIM21]

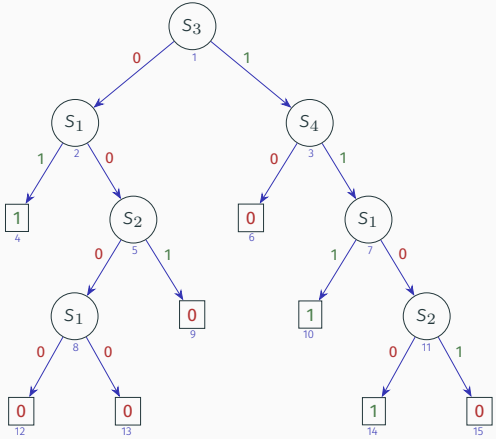
- Concept of explanation graph (XpG)
- Explanations of decision trees reducible to XpG's
- Explanations of decision graphs reducible to XpG's
- Explanations of OBDDs reducible to XpG's
- Explanations of OMDDs reducible to XpG's
- Explanations (AXp's and CXp's) of XpG's computed in polynomial time

# Example of XpG – DTs

- DT; point: $(O, L, Y, P)$; prediction $T$:

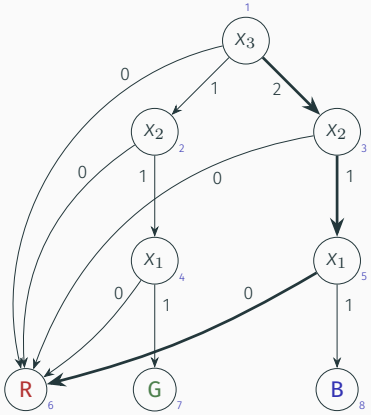



- XpG:

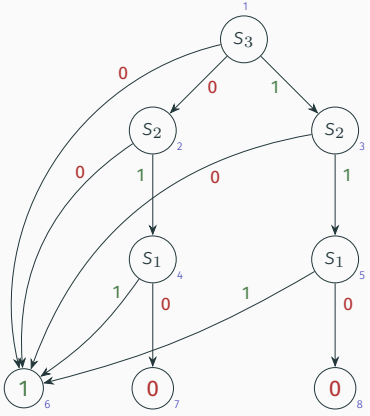


# Example of XpG – OMDDs

- OMBBD; point: $(0, 1, 2)$; prediction *R*:




- XpG:



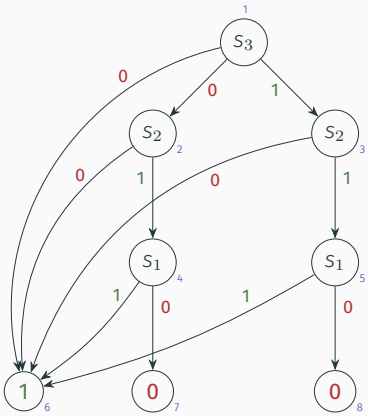
# Finding one AXp for XpGs – polynomial time

- Algorithm (with no inconsistent paths):

  $\mathcal{S} \leftarrow \mathcal{F}$

  For each feature $i$ in $\mathcal{F}$

- XpG:



# Finding one AXp for XpGs – polynomial time

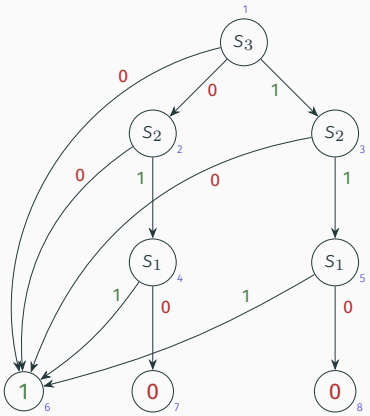
- Algorithm (with no inconsistent paths):

  $\mathcal{S} \leftarrow \mathcal{F}$

  For each feature $i$ in $\mathcal{F}$

  Drop feature $i$ from $\mathcal{S}$, i.e. $i$ is free

- XpG:



# Finding one AXp for XpGs – polynomial time

- Algorithm (with no inconsistent paths):

  $\mathcal{S} \leftarrow \mathcal{F}$

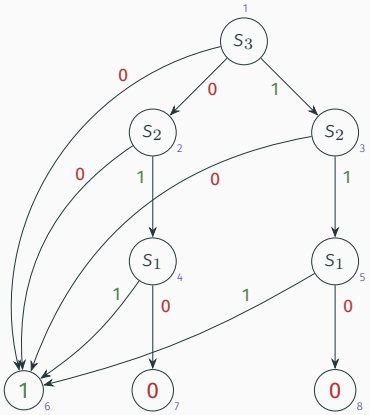
  For each feature $i$ in $\mathcal{F}$

  Drop feature $i$ from $\mathcal{S}$, i.e. $i$ is free

  If path to some 0 not blocked by 0-valued literals, then

  Add feature $i$ back to $\mathcal{S}$

- XpG:

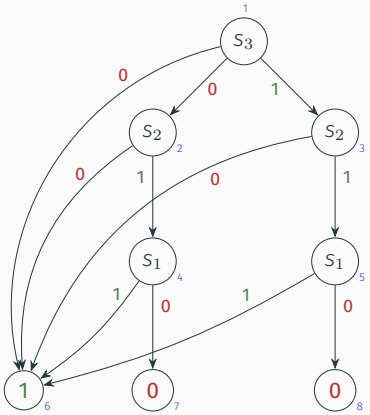


# Finding one AXp for XpGs – polynomial time

- Algorithm (with no inconsistent paths):

  $\mathcal{S} \leftarrow \mathcal{F}$
  For each feature $i$ in $\mathcal{F}$
  Drop feature $i$ from $\mathcal{S}$, i.e. $i$ is free
  If path to some 0 not blocked by
  0-valued literals, then
  Add feature $i$ back to $\mathcal{S}$
  Return $\mathcal{S}$

- XpG:



# Finding one AXp for XpGs – polynomial time

- Algorithm (with no inconsistent paths):

  $\mathcal{S} \leftarrow \mathcal{F}$

  For each feature *i* in $\mathcal{F}$

  Drop feature *i* from $\mathcal{S}$, i.e. *i* is free

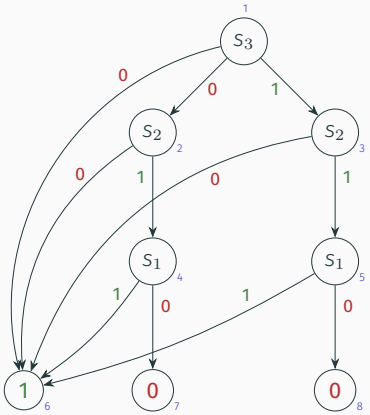
  If path to some 0 not blocked by 0-valued literals, then

  Add feature *i* back to $\mathcal{S}$

  Return $\mathcal{S}$

- Example:
  - $\mathcal{S} = \{1, 2, 3\}$

- XpG:



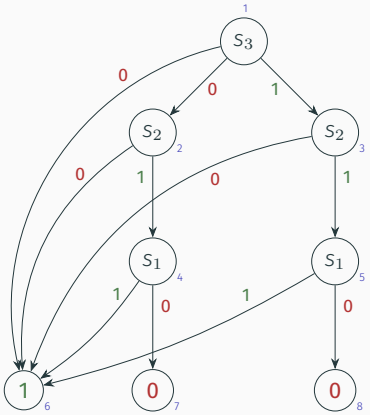
# Finding one AXp for XpGs – polynomial time

- Algorithm (with no inconsistent paths):

  $\mathcal{S} \leftarrow \mathcal{F}$
  For each feature $i$ in $\mathcal{F}$
  Drop feature $i$ from $\mathcal{S}$, i.e. $i$ is free
  If path to some 0 not blocked by
  0-valued literals, then
  Add feature $i$ back to $\mathcal{S}$
  Return $\mathcal{S}$

- Example:
  - $\mathcal{S} = \{1, 2, 3\}$
  - Feature 1 cannot be dropped, e.g.
    $s_3 \rightarrow s_2 \rightarrow s_1 \rightarrow 0$

- XpG:



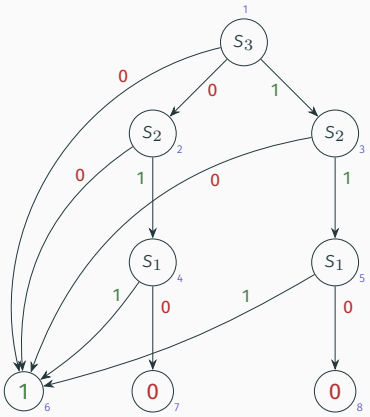
# Finding one AXp for XpGs – polynomial time

- Algorithm (with no inconsistent paths):

  $\mathcal{S} \leftarrow \mathcal{F}$
  For each feature $i$ in $\mathcal{F}$
  Drop feature $i$ from $\mathcal{S}$, i.e. $i$ is free
  If path to some 0 not blocked by
  0-valued literals, then
  Add feature $i$ back to $\mathcal{S}$
  Return $\mathcal{S}$

- Example:
  - $\mathcal{S} = \{1, 2, 3\}$
  - Feature 1 cannot be dropped, e.g.
    $s_3 \rightarrow s_2 \rightarrow s_1 \rightarrow 0$
  - Both features 2 and 3 dropped from $\mathcal{S}$

- XpG:



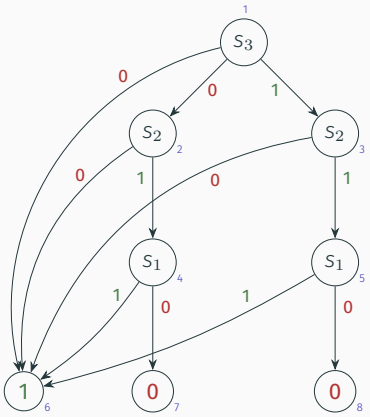
# Finding one AXp for XpGs – polynomial time

- Algorithm (with no inconsistent paths):

  $\mathcal{S} \leftarrow \mathcal{F}$
  For each feature $i$ in $\mathcal{F}$
  Drop feature $i$ from $\mathcal{S}$, i.e. $i$ is free
  If path to some **0** not blocked by
  0-valued literals, then
  Add feature $i$ back to $\mathcal{S}$
  Return $\mathcal{S}$

- Example:
  - $\mathcal{S} = \{1, 2, 3\}$
  - Feature 1 cannot be dropped, e.g.
    $s_3 \rightarrow s_2 \rightarrow s_1 \rightarrow 0$
  - Both features 2 and 3 dropped from $\mathcal{S}$
  - Return $\mathcal{S} = \{1\}$

- XpG:



# Outline – Unit #03

# Example monotonic classifier – $(\mathbf{v}, c) = ((10, 10, 5, 0), A)$

[MGC+21]

| Variable | Meaning | Range |
|---|---|---|
| $\kappa(\cdot) \triangleq M$ | Student grade | $\in \{A, B, C, D, E, F\}$ |
| $S$ | Final score | $\in \{0, \ldots, 10\}$ |

| Feat. id | Feat. var. | Feat. name | Domain |
|---|---|---|---|
| 1 | $Q$ | Quiz | $\{0, \ldots, 10\}$ |
| 2 | $X$ | Exam | $\{0, \ldots, 10\}$ |
| 3 | $H$ | Homework | $\{0, \ldots, 10\}$ |
| 4 | $R$ | Project | $\{0, \ldots, 10\}$ |

$$M = \text{ITE}(S \geqslant 9, A, \text{ITE}(S \geqslant 7, B, \text{ITE}(S \geqslant 5, C, \text{ITE}(S \geqslant 4, D, \text{ite}(S \geqslant 2, E, F)))))$$

$$S = \max\left[0.3 \times Q + 0.6 \times X + 0.1 \times H, R\right]$$

Also, $F \preccurlyeq E \preccurlyeq D \preccurlyeq C \preccurlyeq B \preccurlyeq A$

And, $\kappa(\mathbf{x_1}) \preccurlyeq \kappa(\mathbf{x}_2)$ if $\mathbf{x}_1 \leqslant \mathbf{x}_2$

# Explaining monotonic classifiers

- Instance $(\mathbf{v}, c)$
- Domain for $i \in \mathcal{F}$: $\lambda(i) \leqslant x_i \leqslant \mu(i)$
- Idea: refine lower and upper bounds on the prediction
  - $\mathbf{v}_{\mathsf{L}}$ and $\mathbf{v}_{\mathsf{U}}$
- Utilities:
  - FixAttr($i$):

    $\mathbf{v}_{\mathsf{L}} \leftarrow (v_{\mathsf{L}_1}, \ldots, v_i, \ldots, v_{\mathsf{L}_N})$
    $\mathbf{v}_{\mathsf{U}} \leftarrow (v_{\mathsf{U}_1}, \ldots, v_i, \ldots, v_{\mathsf{U}_N})$
    $(\mathcal{A}, \mathcal{B}) \leftarrow (\mathcal{A} \backslash \{i\}, \mathcal{B} \cup \{i\})$
    **return** $(\mathbf{v}_{\mathsf{L}}, \mathbf{v}_{\mathsf{U}}, \mathcal{A}, \mathcal{B})$
  - FreeAttr($i$):

    $\mathbf{v}_{\mathsf{L}} \leftarrow (v_{\mathsf{L}_1}, \ldots, \lambda(i), \ldots, v_{\mathsf{L}_N})$
    $\mathbf{v}_{\mathsf{U}} \leftarrow (v_{\mathsf{U}_1}, \ldots, \mu(i), \ldots, v_{\mathsf{U}_N})$
    $(\mathcal{A}, \mathcal{B}) \leftarrow (\mathcal{A} \backslash \{i\}, \mathcal{B} \cup \{i\})$
    **return** $(\mathbf{v}_{\mathsf{L}}, \mathbf{v}_{\mathsf{U}}, \mathcal{A}, \mathcal{B})$

# Computing one AXp

1: $\mathbf{v}_{\mathsf{L}} \leftarrow (v_1, \ldots, v_N)$
2: $\mathbf{v}_{\mathsf{U}} \leftarrow (v_1, \ldots, v_N)$ ▷ Ensures: $\kappa(\mathbf{v}_{\mathsf{L}}) = \kappa(\mathbf{v}_{\mathsf{U}})$
3: $(\mathcal{C}, \mathcal{D}, \mathcal{P}) \leftarrow (\mathcal{F}, \varnothing, \varnothing)$ ▷ $\mathcal{S}$: Some possible seed
4: **for all** $i \in \mathcal{S}$ **do**
5: $\quad (\mathbf{v}_{\mathsf{L}}, \mathbf{v}_{\mathsf{U}}, \mathcal{C}, \mathcal{D}) \leftarrow \text{FreeAttr}(i, \mathbf{v}, \mathbf{v}_{\mathsf{L}}, \mathbf{v}_{\mathsf{U}}, \mathcal{C}, \mathcal{D})$ ▷ Require: $\kappa(\mathbf{v}_{\mathsf{L}}) = \kappa(\mathbf{v}_{\mathsf{U}})$, given $\mathcal{S}$
6: **for all** $i \in \mathcal{F} \backslash \mathcal{S}$ **do** ▷ Loop inv.: $\kappa(\mathbf{v}_{\mathsf{L}}) = \kappa(\mathbf{v}_{\mathsf{U}})$
7: $\quad (\mathbf{v}_{\mathsf{L}}, \mathbf{v}_{\mathsf{U}}, \mathcal{C}, \mathcal{D}) \leftarrow \text{FreeAttr}(i, \mathbf{v}, \mathbf{v}_{\mathsf{L}}, \mathbf{v}_{\mathsf{U}}, \mathcal{C}, \mathcal{D})$
8: $\quad$ **if** $\kappa(\mathbf{v}_{\mathsf{L}}) \neq \kappa(\mathbf{v}_{\mathsf{U}})$ **then** ▷ If invariant broken, fix it
9: $\quad\quad (\mathbf{v}_{\mathsf{L}}, \mathbf{v}_{\mathsf{U}}, \mathcal{D}, \mathcal{P}) \leftarrow \text{FixAttr}(i, \mathbf{v}, \mathbf{v}_{\mathsf{L}}, \mathbf{v}_{\mathsf{U}}, \mathcal{D}, \mathcal{P})$
10: **return** $\mathcal{P}$

- **Obs:** $\mathcal{S} = \varnothing$ for computing a single AXp/CXp

# Computing one AXp – example

- $\lambda(i) = 0$ and $\mu(i) = 10$
- $\mathbf{v} = (10, 10, 5, 0)$, with $\kappa(\mathbf{v}) = A$
- Q: find one AXp (CXp is similar)

| Feat. | Initial values | | Changed values | | Predictions | | Dec. | Resulting values | |
|---|---|---|---|---|---|---|---|---|---|
| | $\mathbf{v}_L$ | $\mathbf{v}_U$ | $\mathbf{v}_L$ | $\mathbf{v}_U$ | $\kappa(\mathbf{v}_L)$ | $\kappa(\mathbf{v}_U)$ | | $\mathbf{v}_L$ | $\mathbf{v}_U$ |
| 1 | (10,10,5,0) | (10,10,5,0) | (0,10,5,0) | (10,10,5,0) | $C$ | $A$ | ✓ | (10,10,5,0) | (10,10,5,0) |
| 2 | (10,10,5,0) | (10,10,5,0) | (10,0,5,0) | (10,10,5,0) | $E$ | $A$ | ✓ | (10,10,5,0) | (10,10,5,0) |
| 3 | (10,10,5,0) | (10,10,5,0) | (10,10,0,0) | (10,10,10,0) | $A$ | $A$ | ✗ | (10,10,0,0) | (10,10,10,0) |
| 4 | (10,10,0,0) | (10,10,10,0) | (10,10,0,0) | (10,10,10,10) | $A$ | $A$ | ✗ | (10,10,0,0) | (10,10,10,10) |

# Outline – Unit #03

# Recap computation of (W)AXps/(W)CXps

$$\begin{aligned}
\mathsf{WAXp}(\mathcal{X}) &:= \forall(\mathbf{x} \in \mathbb{F}).\ \bigwedge\nolimits_{j \in \mathcal{X}}(x_j = v_j) \rightarrow (\sigma(\mathbf{x})) \\
\mathsf{WCXp}(\mathcal{Y}) &:= \exists(\mathbf{x} \in \mathbb{F}).\ \bigwedge\nolimits_{j \notin \mathcal{Y}}(x_j = v_j) \land (\neg\sigma(\mathbf{x}))
\end{aligned}$$

# Recap computation of (W)AXps/(W)CXps

$$\text{WAXp}(\mathcal{X}) \quad := \quad \forall(\mathbf{x} \in \mathbb{F}). \bigwedge\nolimits_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$$

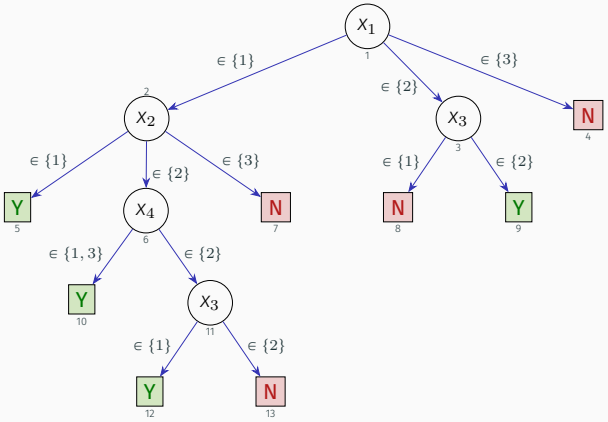
$$\text{WCXp}(\mathcal{Y}) \quad := \quad \exists(\mathbf{x} \in \mathbb{F}). \bigwedge\nolimits_{j \notin \mathcal{Y}} (x_j = v_j) \land (\neg\sigma(\mathbf{x}))$$

**Input**: Predicate $\mathbb{P}$, parameterized by $\mathcal{T}$, $\mathcal{M}$
**Output**: One XP $\mathcal{S}$

1: **procedure** oneXP($\mathbb{P}$)
2: $\quad \mathcal{S} \leftarrow \mathcal{F}$ $\quad \triangleright$ Initialization: $\mathbb{P}(\mathcal{S})$ holds
3: $\quad$ **for** $i \in \mathcal{F}$ **do** $\quad \triangleright$ Loop invariant: $\mathbb{P}(\mathcal{S})$ holds
4: $\quad\quad$ **if** $\mathbb{P}(\mathcal{S} \backslash \{i\})$ **then**
5: $\quad\quad\quad \mathcal{S} \leftarrow \mathcal{S} \backslash \{i\}$ $\quad \triangleright$ Update $\mathcal{S}$ only if $\mathbb{P}(\mathcal{S} \backslash \{i\})$ holds
6: $\quad$ **return** $\mathcal{S}$ $\quad \triangleright$ Returned set $\mathcal{S}$: $\mathbb{P}(\mathcal{S})$ holds

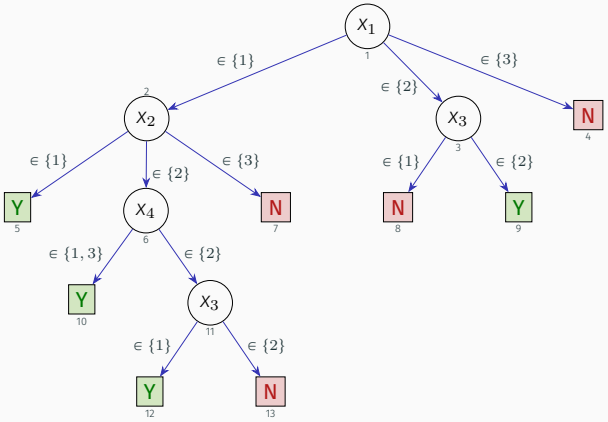
# Review exercise – one AXp for example DT

- Instance: $(\mathbf{v}, c) = ((1, 2, 1, 2), \mathsf{Y})$



# Review exercise – one AXp for example DT

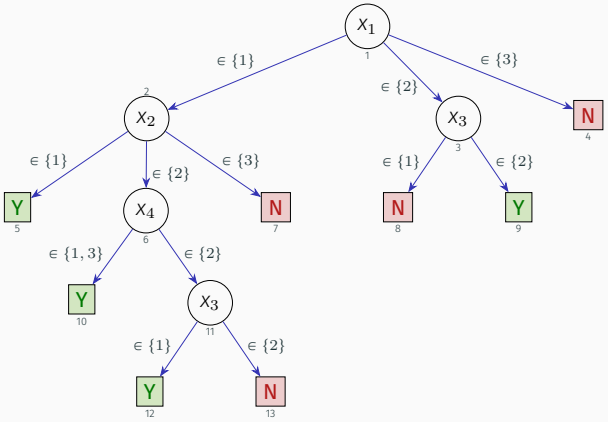
- Instance: $(\mathbf{v}, c) = ((1, 2, 1, 2), \mathsf{Y})$




- Finding on AXp:

# Review exercise – one AXp for example DT

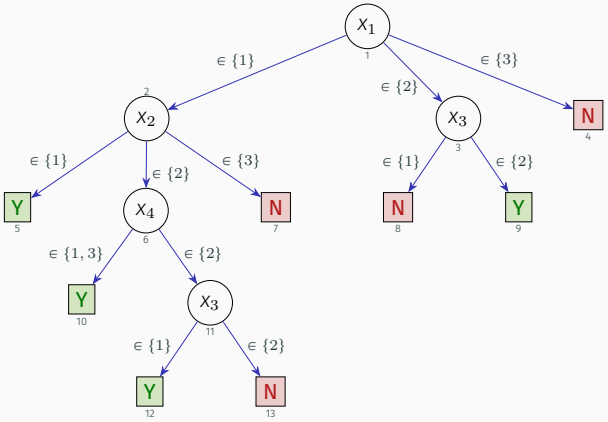
- Instance: $(\mathbf{v}, c) = ((1, 2, 1, 2), \mathsf{Y})$




- Finding on AXp:
  - 1st path inconsistent: $H_1 = \{3\}$

# Review exercise – one AXp for example DT

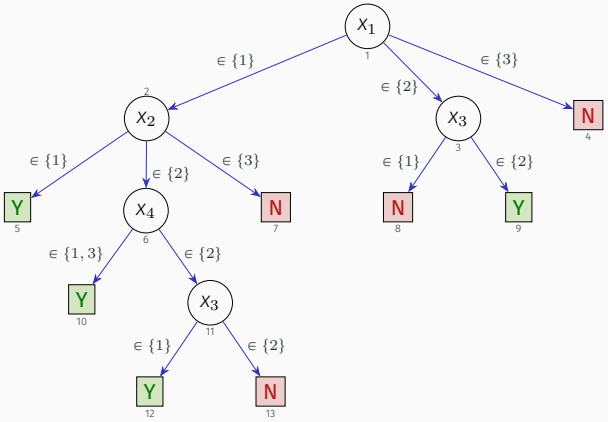
- Instance: $(\mathbf{v}, c) = ((1, 2, 1, 2), \mathsf{Y})$




- Finding on AXp:
  - 1st path inconsistent: $H_1 = \{3\}$
  - 2nd path inconsistent: $H_2 = \{2\}$

# Review exercise – one AXp for example DT

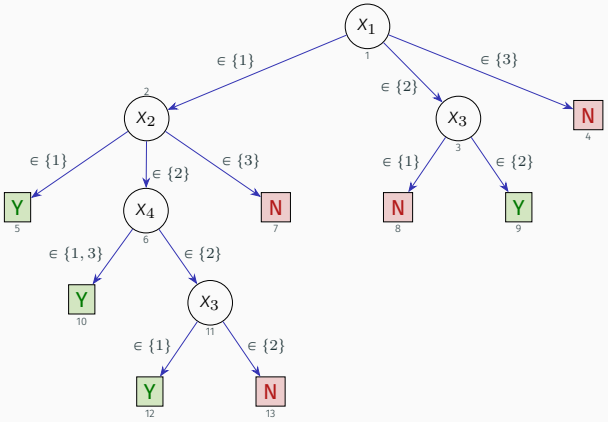
- Instance: $(\mathbf{v}, c) = ((1, 2, 1, 2), \mathsf{Y})$




- Finding on AXp:
  - 1st path inconsistent: $H_1 = \{3\}$
  - 2nd path inconsistent: $H_2 = \{2\}$
  - 3rd path inconsistent: $H_3 = \{1\}$

# Review exercise – one AXp for example DT

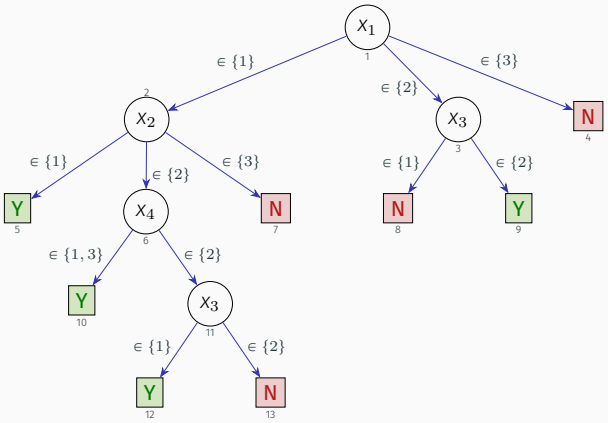
- Instance: $(\mathbf{v}, c) = ((1, 2, 1, 2), \mathsf{Y})$




- Finding on AXp:
  - 1st path inconsistent: $H_1 = \{3\}$
  - 2nd path inconsistent: $H_2 = \{2\}$
  - 3rd path inconsistent: $H_3 = \{1\}$
  - 4th path inconsistent: $H_4 = \{1\}$

# Review exercise – one AXp for example DT

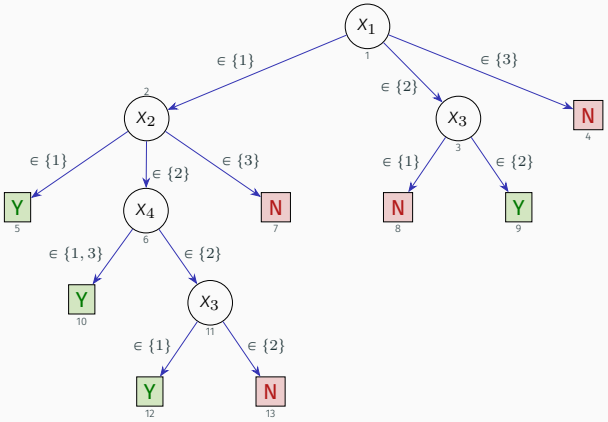
- Instance: $(\mathbf{v}, c) = ((1, 2, 1, 2), \mathsf{Y})$




- Finding on AXp:
  - 1st path inconsistent: $H_1 = \{3\}$
  - 2nd path inconsistent: $H_2 = \{2\}$
  - 3rd path inconsistent: $H_3 = \{1\}$
  - 4th path inconsistent: $H_4 = \{1\}$
- AXp is MHS of $H_j$ sets: $\{1, 2, 3\}$

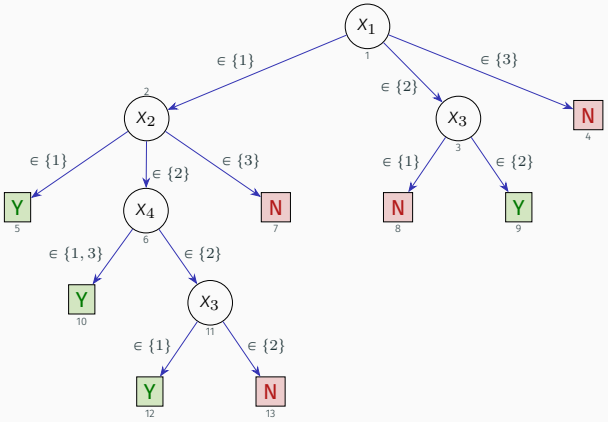
# Review exercise – all CXps & AXps for example DT

- Instance: $(\mathbf{v}, c) = ((1, 2, 1, 2), \mathsf{Y})$



# Review exercise – all CXps & AXps for example DT

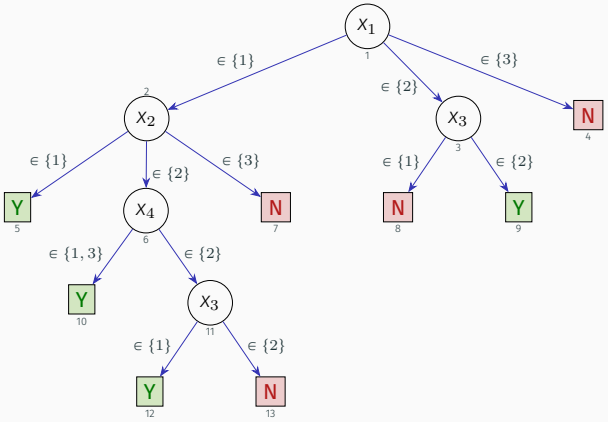
- Instance: $(\mathbf{v}, c) = ((1, 2, 1, 2), \mathsf{Y})$




- Finding CXps:

# Review exercise – all CXps & AXps for example DT

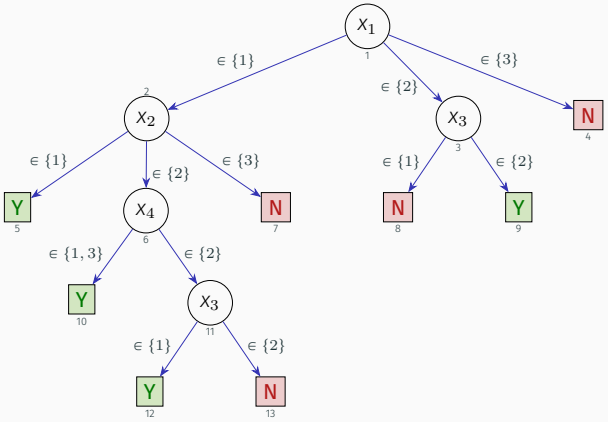
- Instance: $(\mathbf{v}, c) = ((1, 2, 1, 2), \mathsf{Y})$




- Finding CXps:
  - 1st path: $I_1 = \{3\}$

# Review exercise – all CXps & AXps for example DT

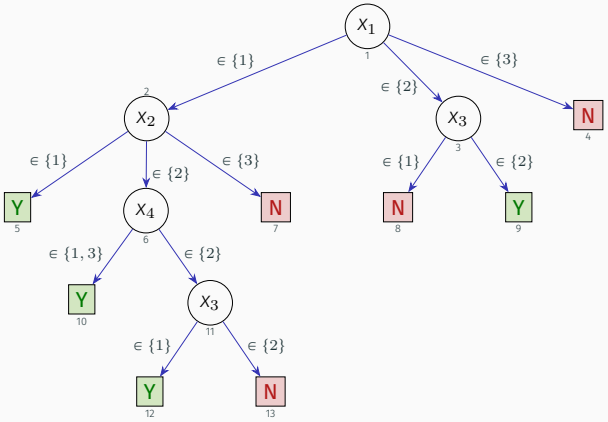
- Instance: $(\mathbf{v}, c) = ((1, 2, 1, 2), \mathsf{Y})$




- Finding CXps:
  - 1st path: $l_1 = \{3\}$
  - 2nd path: $l_2 = \{2\}$

# Review exercise – all CXps & AXps for example DT

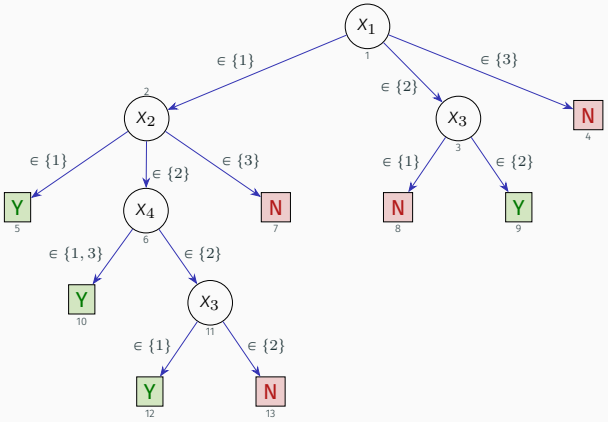
- Instance: $(\mathbf{v}, c) = ((1, 2, 1, 2), \mathsf{Y})$




- Finding CXps:
  - 1st path: $l_1 = \{3\}$
  - 2nd path: $l_2 = \{2\}$
  - 3rd path: $l_3 = \{1\}$

# Review exercise – all CXps & AXps for example DT

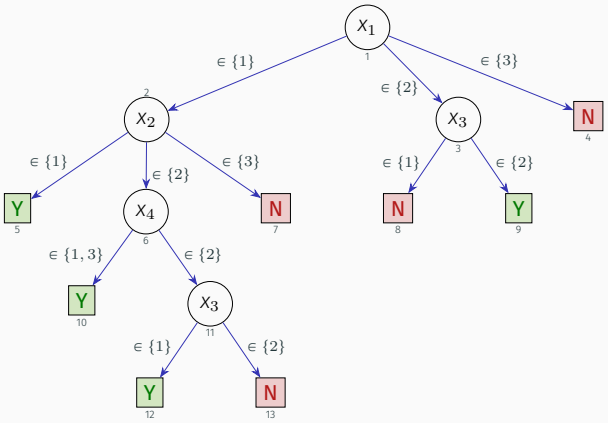
- Instance: $(\mathbf{v}, c) = ((1, 2, 1, 2), \mathsf{Y})$




- Finding CXps:
  - 1st path: $l_1 = \{3\}$
  - 2nd path: $l_2 = \{2\}$
  - 3rd path: $l_3 = \{1\}$
  - 4th path: $l_4 = \{1\}$

# Review exercise – all CXps & AXps for example DT

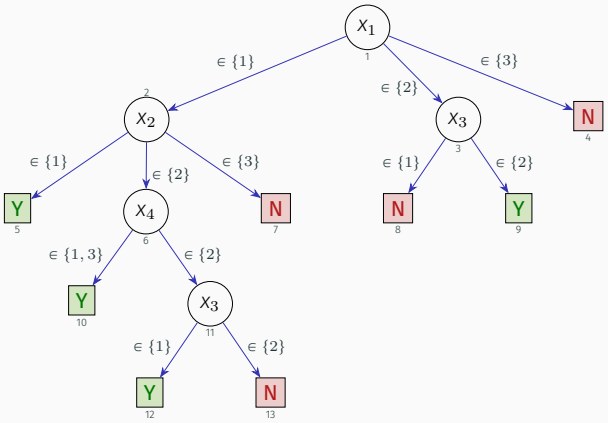
- Instance: $(\mathbf{v}, c) = ((1, 2, 1, 2), \mathsf{Y})$




- Finding CXps:
  - 1st path: $l_1 = \{3\}$
  - 2nd path: $l_2 = \{2\}$
  - 3rd path: $l_3 = \{1\}$
  - 4th path: $l_4 = \{1\}$
  - $\mathcal{L} = \{\{1\}, \{2\}, \{3\}\} = \mathbb{C}$

# Review exercise – all CXps & AXps for example DT

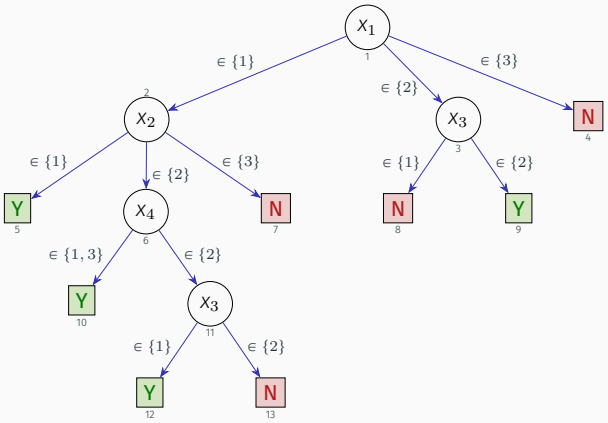
- Instance: $(\mathbf{v}, c) = ((1, 2, 1, 2), \mathsf{Y})$




- Finding CXps:
  - 1st path: $l_1 = \{3\}$
  - 2nd path: $l_2 = \{2\}$
  - 3rd path: $l_3 = \{1\}$
  - 4th path: $l_4 = \{1\}$
  - $\mathcal{L} = \{\{1\}, \{2\}, \{3\}\} = \mathbb{C}$
- Finding AXps:
  (i.e. all MHSes of sets in $\mathbb{C}$

# Review exercise – all CXps & AXps for example DT

- Instance: $(\mathbf{v}, c) = ((1, 2, 1, 2), \mathsf{Y})$




- Finding CXps:
  - 1st path: $l_1 = \{3\}$
  - 2nd path: $l_2 = \{2\}$
  - 3rd path: $l_3 = \{1\}$
  - 4th path: $l_4 = \{1\}$
  - $\mathcal{L} = \{\{1\}, \{2\}, \{3\}\} = \mathbb{C}$
- Finding AXps:
  (i.e. all MHSes of sets in $\mathbb{C}$
  - $\mathbb{A} = \{\{1, 2, 3\}\}$

# Another review exercise – one AXp for example DL

- DL:

$$
\begin{array}{lllll}
\mathsf{R}_1: & \text{IF} & (x_1 \land x_3) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_2: & \text{ELSE IF} & (x_1 \land x_5) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_3: & \text{ELSE IF} & (x_2 \land x_4) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_4: & \text{ELSE IF} & (x_1 \land x_7) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_5: & \text{ELSE IF} & (\neg x_4 \land x_6) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_6: & \text{ELSE IF} & (\neg x_4 \land \neg x_6) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_7: & \text{ELSE IF} & (\neg x_2 \land x_6) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_{\mathsf{DEF}}: & \text{ELSE} & & & \kappa(\mathbf{x}) = 0
\end{array}
$$

# Another review exercise – one AXp for example DL

- DL:

$$
\begin{array}{lllll}
\mathsf{R}_1: & \text{IF} & (x_1 \land x_3) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_2: & \text{ELSE IF} & (x_1 \land x_5) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_3: & \text{ELSE IF} & (x_2 \land x_4) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_4: & \text{ELSE IF} & (x_1 \land x_7) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_5: & \text{ELSE IF} & (\neg x_4 \land x_6) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_6: & \text{ELSE IF} & (\neg x_4 \land \neg x_6) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_7: & \text{ELSE IF} & (\neg x_2 \land x_6) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_{\mathsf{DEF}}: & \text{ELSE} & & & \kappa(\mathbf{x}) = 0
\end{array}
$$

- Instance: $\mathbf{v} = (0, 1, 0, 1, 0, 1, 0)$
  - The prediction is 1, due to $\mathsf{R}_3$

# Another review exercise – one AXp for example DL

- DL:

$$
\begin{array}{lllll}
\mathsf{R}_1: & \text{IF} & (x_1 \land x_3) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_2: & \text{ELSE IF} & (x_1 \land x_5) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_3: & \text{ELSE IF} & (x_2 \land x_4) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_4: & \text{ELSE IF} & (x_1 \land x_7) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_5: & \text{ELSE IF} & (\neg x_4 \land x_6) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_6: & \text{ELSE IF} & (\neg x_4 \land \neg x_6) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_7: & \text{ELSE IF} & (\neg x_2 \land x_6) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_{\mathsf{DEF}}: & \text{ELSE} & & & \kappa(\mathbf{x}) = 0
\end{array}
$$

- Instance: $\mathbf{v} = (0, 1, 0, 1, 0, 1, 0)$
  - The prediction is 1, due to $\mathsf{R}_3$
- AXp:

# Another review exercise – one AXp for example DL

- DL:

$$
\begin{array}{lllll}
\mathsf{R}_1: & \text{IF} & (x_1 \land x_3) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_2: & \text{ELSE IF} & (x_1 \land x_5) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_3: & \text{ELSE IF} & (x_2 \land x_4) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_4: & \text{ELSE IF} & (x_1 \land x_7) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_5: & \text{ELSE IF} & (\neg x_4 \land x_6) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_6: & \text{ELSE IF} & (\neg x_4 \land \neg x_6) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_7: & \text{ELSE IF} & (\neg x_2 \land x_6) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_{\mathsf{DEF}}: & \text{ELSE} & & & \kappa(\mathbf{x}) = 0
\end{array}
$$

- Instance: $\mathbf{v} = (0, 1, 0, 1, 0, 1, 0)$
  - The prediction is 1, due to $\mathsf{R}_3$
- AXp: $\{1, 2\}$

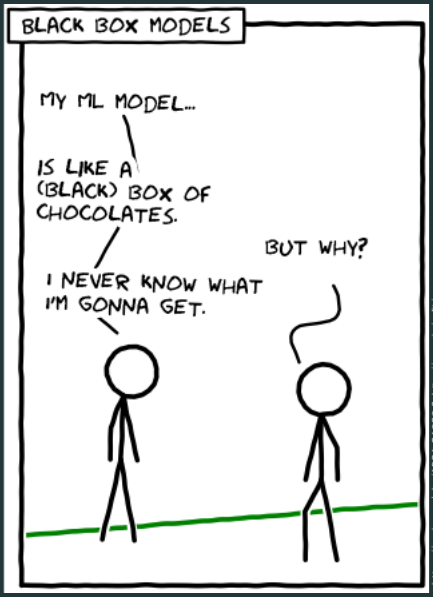
# Another review exercise – one AXp for example DL

- DL:

$$
\begin{array}{lllll}
\mathsf{R}_1: & \text{IF} & (x_1 \land x_3) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_2: & \text{ELSE IF} & (x_1 \land x_5) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_3: & \text{ELSE IF} & (x_2 \land x_4) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_4: & \text{ELSE IF} & (x_1 \land x_7) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\mathsf{R}_5: & \text{ELSE IF} & (\neg x_4 \land x_6) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_6: & \text{ELSE IF} & (\neg x_4 \land \neg x_6) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_7: & \text{ELSE IF} & (\neg x_2 \land x_6) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\mathsf{R}_{\mathsf{DEF}}: & \text{ELSE} & & & \kappa(\mathbf{x}) = 0
\end{array}
$$

- Instance: $\mathbf{v} = (0, 1, 0, 1, 0, 1, 0)$
  - The prediction is 1, due to $\mathsf{R}_3$
- AXp: $\{1, 2\}$
- Quiz: write down the constraints and confirm AXp with SAT solver

Questions?


BLACK BOX MODELS
MY ML MODEL...
IS LIKE A (BLACK) BOX OF CHOCOLATES.
I NEVER KNOW WHAT I'M GONNA GET.
BUT WHY?
https://arxiv.org/abs/1901.01686 & http://cmx.io/edit/

# References i


[ABOS22] Marcelo Arenas, Pablo Barceló, Miguel A. Romero Orth, and Bernardo Subercaseaux.
**On computing probabilistic explanations for decision trees.**
In *NeurIPS*, 2022.

[Alp14] Ethem Alpaydin.
***Introduction to machine learning.***
MIT press, 2014.

[Alp16] Ethem Alpaydin.
***Machine Learning: The New AI.***
MIT Press, 2016.

[BA97] Leonard A. Breslow and David W. Aha.
**Simplifying decision trees: A survey.**
*Knowledge Eng. Review*, 12(1):1–40, 1997.

[BBHK10] Michael R. Berthold, Christian Borgelt, Frank Höppner, and Frank Klawonn.
***Guide to Intelligent Data Analysis - How to Intelligently Make Sense of Real Data*, volume 42 of *Texts in Computer Science*.**
Springer, 2010.

# References ii


[BFOS84] Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone.
*Classification and Regression Trees.*
Wadsworth, 1984.

[BHO09] Christian Bessiere, Emmanuel Hebrard, and Barry O'Sullivan.
Minimising decision tree size as combinatorial optimisation.
In *CP*, pages 173–187, 2009.

[Bra20] Max Bramer.
*Principles of Data Mining, 4th Edition.*
Undergraduate Topics in Computer Science. Springer, 2020.

[DL01] Sašo Džeroski and Nada Lavrač, editors.
*Relational data mining.*
Springer, 2001.

[EG95] Thomas Eiter and Georg Gottlob.
Identifying the minimal transversals of a hypergraph and related problems.
*SIAM J. Comput.*, 24(6):1278–1304, 1995.

[Fla12] Peter A. Flach.
*Machine Learning - The Art and Science of Algorithms that Make Sense of Data.*
Cambridge University Press, 2012.

# References iii


[GZM20] Mohammad M. Ghiasi, Sohrab Zendehboudi, and Ali Asghar Mohsenipour.
**Decision tree-based diagnosis of coronary artery disease: CART model.**
*Comput. Methods Programs Biomed.*, 192:105400, 2020.

[HIIM21] Xuanxiang Huang, Yacine Izza, Alexey Ignatiev, and Joao Marques-Silva.
**On efficiently explaining graph-based classifiers.**
In *KR*, November 2021.
Preprint available from `https://arxiv.org/abs/2106.01350`.

[HM23] Xuanxiang Huang and João Marques-Silva.
**From decision trees to explained decision sets.**
In *ECAI*, pages 1100–1108, 2023.

[HRS19] Xiyang Hu, Cynthia Rudin, and Margo Seltzer.
**Optimal sparse decision trees.**
In *NeurIPS*, pages 7265–7273, 2019.

[IHI+22] Yacine Izza, Xuanxiang Huang, Alexey Ignatiev, Nina Narodytska, Martin C. Cooper, and João Marques-Silva.
**On computing probabilistic abductive explanations.**
*CoRR*, abs/2212.05990, 2022.

# References iv


[IHI+23] Yacine Izza, Xuanxiang Huang, Alexey Ignatiev, Nina Narodytska, Martin C. Cooper, and João Marques-Silva.
**On computing probabilistic abductive explanations.**
*Int. J. Approx. Reason.*, 159:108939, 2023.

[IIM20] Yacine Izza, Alexey Ignatiev, and Joao Marques-Silva.
**On explaining decision trees.**
*CoRR*, abs/2010.11034, 2020.

[IIM22] Yacine Izza, Alexey Ignatiev, and João Marques-Silva.
**On tackling explanation redundancy in decision trees.**
*J. Artif. Intell. Res.*, 75:261–321, 2022.

[IM21] Alexey Ignatiev and Joao Marques-Silva.
**SAT-based rigorous explanations for decision lists.**
In *SAT*, pages 251–269, July 2021.

[INAM20] Alexey Ignatiev, Nina Narodytska, Nicholas Asher, and João Marques-Silva.
**From contrastive to abductive explanations and back again.**
In *AIxIA*, pages 335–355, 2020.

[INM19a] Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva.
**Abduction-based explanations for machine learning models.**
In *AAAI*, pages 1511–1519, 2019.

# References v


[INM19b] Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva.
**On relating explanations and adversarial examples.**
In *NeurIPS*, pages 15857–15867, 2019.

[KMND20] John D Kelleher, Brian Mac Namee, and Aoife D'arcy.
***Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies.***
MIT Press, 2020.

[Kot13] Sotiris B. Kotsiantis.
**Decision trees: a recent overview.**
*Artif. Intell. Rev.*, 39(4):261–283, 2013.

[LL17] Scott M. Lundberg and Su-In Lee.
**A unified approach to interpreting model predictions.**
In *NIPS*, pages 4765–4774, 2017.

[Mar22] João Marques-Silva.
**Logic-based explainability in machine learning.**
In *Reasoning Web*, pages 24–104, 2022.

# References vi


[Mar24] Joao Marques-Silva.
**Logic-based explainability: Past, present & future.**
*CoRR*, abs/2406.11873, 2024.

[MGC+21] Joao Marques-Silva, Thomas Gerspacher, Martinc C. Cooper, Alexey Ignatiev, and Nina Narodytska.
**Explanations for monotonic classifiers.**
In *ICML*, pages 7469–7479, July 2021.

[Mil56] George A Miller.
**The magical number seven, plus or minus two: Some limits on our capacity for processing information.**
*Psychological review*, 63(2):81–97, 1956.

[Mil19] Tim Miller.
**Explanation in artificial intelligence: Insights from the social sciences.**
*Artif. Intell.*, 267:1–38, 2019.

[MM20] João Marques-Silva and Carlos Mencía.
**Reasoning about inconsistent formulas.**
In *IJCAI*, pages 4899–4906, 2020.

[Mor82] Bernard M. E. Moret.
**Decision trees and diagrams.**
*ACM Comput. Surv.*, 14(4):593–623, 1982.

# References vii


[MSI23] Joao Marques-Silva and Alexey Ignatiev.
**No silver bullet: interpretable ml models must be explained.**
*Frontiers in Artificial Intelligence*, 6, 2023.

[PM17] David Poole and Alan K. Mackworth.
***Artificial Intelligence - Foundations of Computational Agents.***
CUP, 2017.

[Qui93] J Ross Quinlan.
***C4.5: programs for machine learning.***
Morgan-Kaufmann, 1993.

[Rei87] Raymond Reiter.
**A theory of diagnosis from first principles.**
*Artif. Intell.*, 32(1):57–95, 1987.

[RM08] Lior Rokach and Oded Z Maimon.
***Data mining with decision trees: theory and applications.***
World scientific, 2008.

[RN10] Stuart J. Russell and Peter Norvig.
***Artificial Intelligence - A Modern Approach.***
Pearson Education, 2010.

# References viii


[RSG16] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin.
**"why should I trust you?": Explaining the predictions of any classifier.**
In *KDD*, pages 1135–1144, 2016.

[RSG18] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin.
**Anchors: High-precision model-agnostic explanations.**
In *AAAI*, pages 1527–1535. AAAI Press, 2018.

[SB14] Shai Shalev-Shwartz and Shai Ben-David.
***Understanding Machine Learning - From Theory to Algorithms.***
Cambridge University Press, 2014.

[SCD18] Andy Shih, Arthur Choi, and Adnan Darwiche.
**A symbolic approach to explaining bayesian network classifiers.**
In *IJCAI*, pages 5103–5111, 2018.

[VLE+16] Gilmer Valdes, José Marcio Luna, Eric Eaton, Charles B Simone, Lyle H Ungar, and Timothy D Solberg.
**MediBoost: a patient stratification tool for interpretable decision making in the era of precision medicine.**
*Scientific reports*, 6(1):1–8, 2016.

# References ix


[WFHP17] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal.
*Data Mining*.
Morgan Kaufmann, 2017.

[WMHK21] Stephan Wäldchen, Jan MacDonald, Sascha Hauch, and Gitta Kutyniok.
The computational complexity of understanding binary classifier decisions.
*J. Artif. Intell. Res.*, 70:351–387, 2021.

[Zho12] Zhi-Hua Zhou.
*Ensemble methods: foundations and algorithms*.
CRC press, 2012.

[Zho21] Zhi-Hua Zhou.
*Machine Learning*.
Springer, 2021.