A data management approach to explainable AI

Marcelo Arenas
PUC & IMFD Chile and RelationalAI

Joint work with Daniel Báez, Pablo Barceló, Diego Bustamante, José Thomas Caraball, Jorge Pérez, and Bernardo Subercaseaux

Motivation

- A growing interest in developing methods to explain predictions made by machine learning models
- This has led to the development of several notions of explanation, which have been studied independently
- Explainability admits no silver bullet; different contexts require different notions

Motivation

- Explainability may require combining different notions; it is better to think of it as an interactive process
- Instead of struggling with the increasing number of such notions, one can developed a query language for explainability task
- This gives control to the end-user to tailor explainability queries to their particular needs

A call for an explainability query language

- **Declarative:** should allow users to articulate what explanation they need without providing the computational method to achieve it
- **Simple syntax and semantics:** should be built with simple syntax and semantics, leveraging well-known database query languages
- **Specific query capability for explainability:** an explanation notion should be represented by a fixed query

A call for an explainability query language

- **Expressiveness:** must be able to represent common explanation concepts
- **Exploratory operators:** should allow the user to explore an explanation concept
- **Combination of explanations:** should support the combination of different explanation approaches

A call for an explainability query language

- **Efficient data complexity:** Although polynomial data complexity is the gold standard, P^{NP} data complexity is also desirable as it allows the use of SAT solvers
- **Verification versus computation:** should also be feasible to compute explanations efficiently

Our goal is to develop such an explainability query language

Basic ingredients: classification models are represented as **labeled graphs**, and **first-order logic** is used as query language

An explainability query language

We focus in this talk on a simple but widely used model

- **Decision trees** are widely used, in particular because they are considered *readily* interpretable models
- The main ingredients of our logical approach are already present in this case

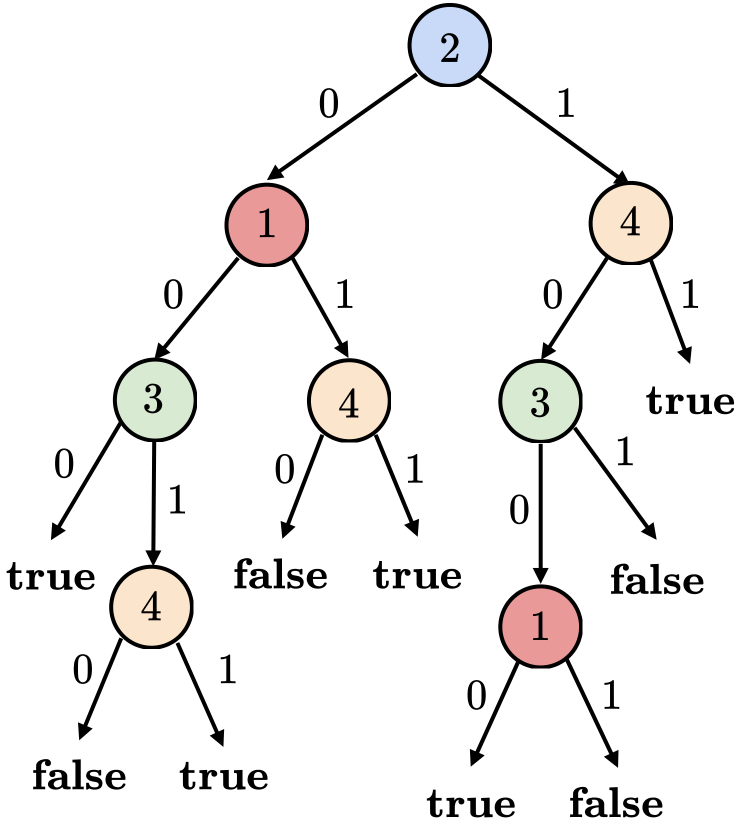
A classification model:

$$\mathcal{M} : \{0,1\}^n \to \{0,1\}$$

- The dimension of $\mathcal{M}$ is n, and each $i \in \{1, \ldots, n\}$ is called a feature
- $\mathbf{e} \in \{0,1\}^n$ is an instance
- $\mathcal{M}$ accepts $\mathbf{e}$ if $\mathcal{M}(\mathbf{e}) = 1$, otherwise $\mathcal{M}$ rejects $\mathbf{e}$

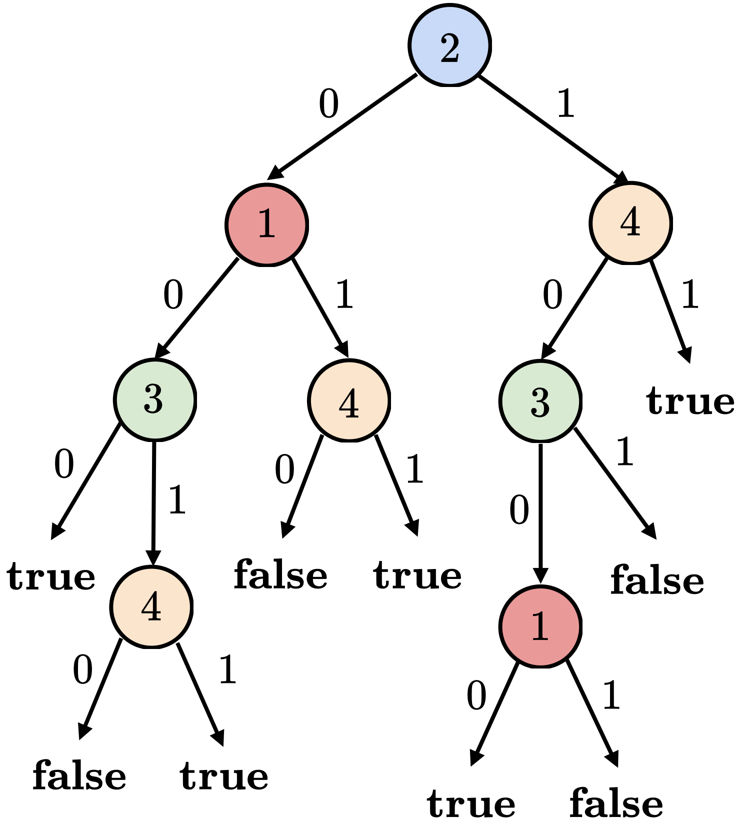
A decision tree $\mathcal{T}$ of dimension n

- Each internal node is labeled with a feature $i \in \{1, \ldots, n\}$, and has two outgoing edges labeled 0 and 1
- Each leaf is labeled **true** or **false**
- No two nodes on a path from the root to a leaf have the same label



A decision tree $\mathcal{T}$ of dimension n

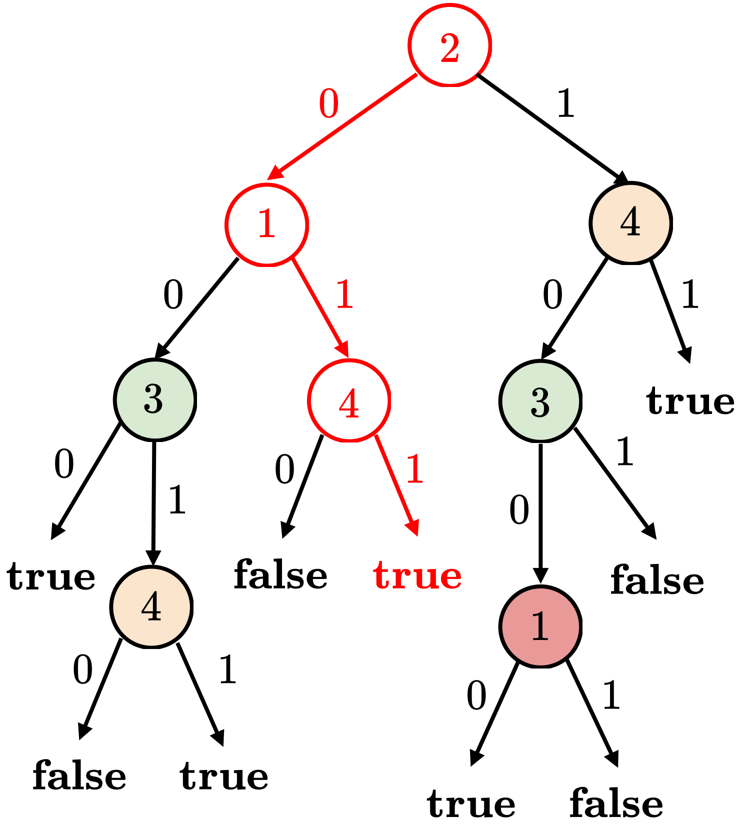
- Every instance $\mathbf{e}$ defines a unique path $n_1, e_1, n_2, \ldots, e_{k-1}, n_k$ from the root to a leaf
- $\mathcal{T}(\mathbf{e}) = 1$ if the label n_k is $\mathbf{true}$



A decision tree $\mathcal{T}$ of dimension n

- Every instance $\mathbf{e}$ defines a unique path $n_1, e_1, n_2, \ldots, e_{k-1}, n_k$ from the root to a leaf
- $\mathcal{T}(\mathbf{e}) = 1$ if the label n_k is $\mathbf{true}$

$\mathcal{T}(\mathbf{e_1}) = 1$ for instance $\mathbf{e_1} = (1, 0, 1, 1)$



Explaining the output of a model

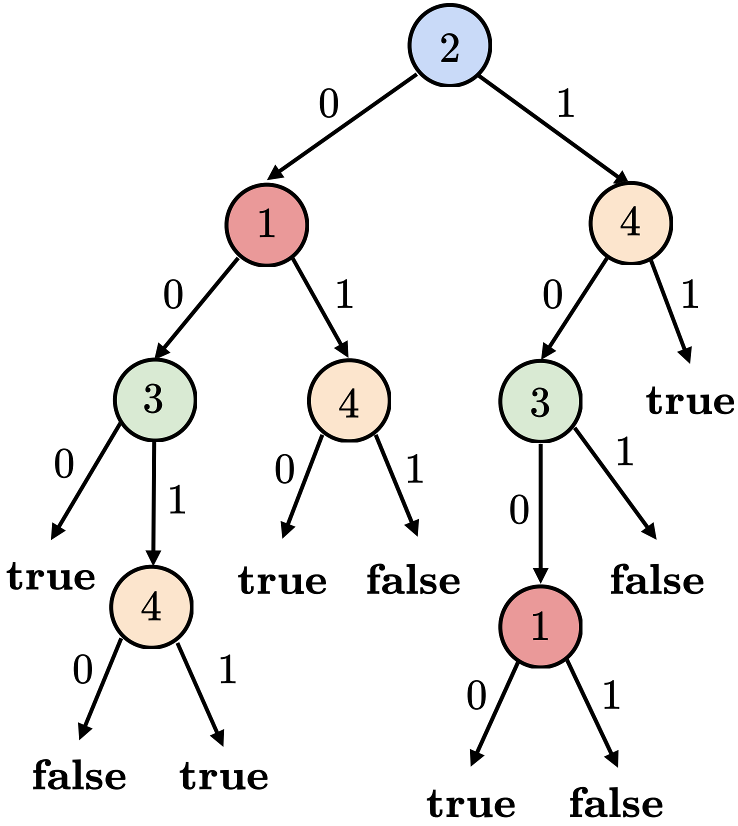
- What are interesting notions of explanation?
- What notions have been studied? What notions are used in practice?
- Can these notions be expressed as queries over decision trees?

Notions of explanation: sufficient reason

$\mathcal{T}(\mathbf{e}) = 1$ for $\mathbf{e} = (1, 1, 1, 1)$

The value of feature 3 is not needed to obtain this result

- $\{1, 2, 4\}$ is a *sufficient reason*

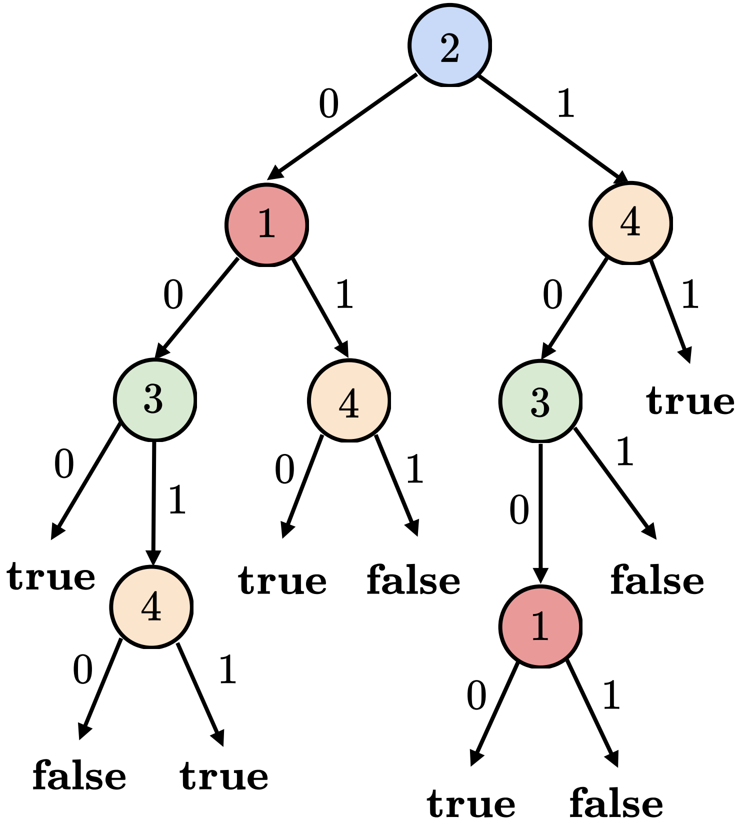


Notions of explanation: minimal sufficient reason

$\mathcal{T}(\mathbf{e}) = 1$ for $\mathbf{e} = (1, 1, 1, 1)$

The value of features 1 and 3 are not needed to obtain this result

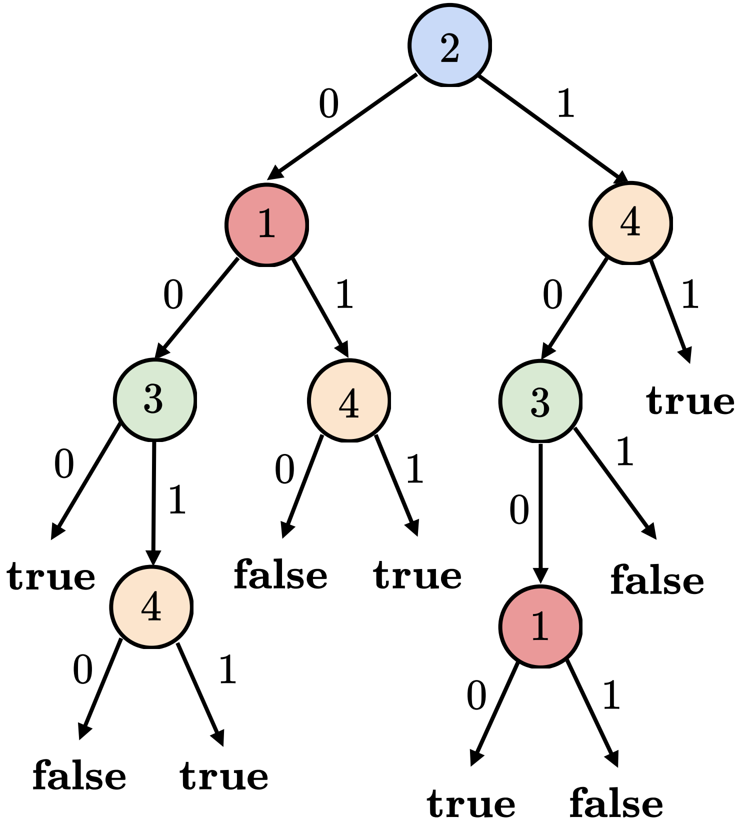
- $\{2, 4\}$ is a *minimal sufficient reason*



Notions of explanation: determinant feature set

If the values of features $\{1, 3, 4\}$ are fixed, then the output of the model is fixed

The output of the model depends only on these features

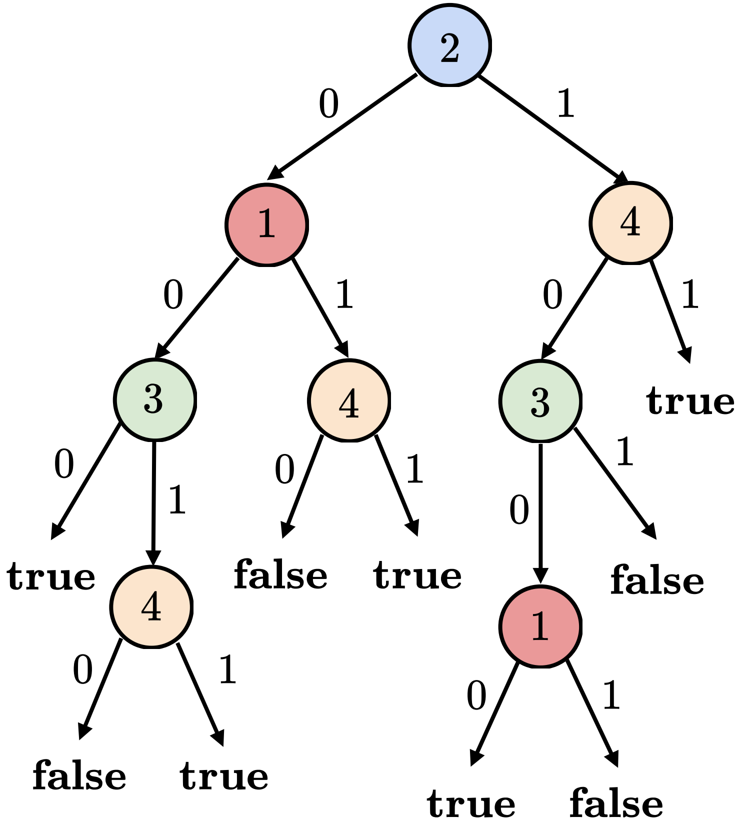


Notions of explanation: determinant feature set

If the values of features $\{1, 3, 4\}$ are fixed, then the output of the model is fixed

If $\mathbf{e}[1] = \mathbf{e}[3] = \mathbf{e}[4] = 0$:

$\mathcal{T}(\mathbf{e}) = 1$

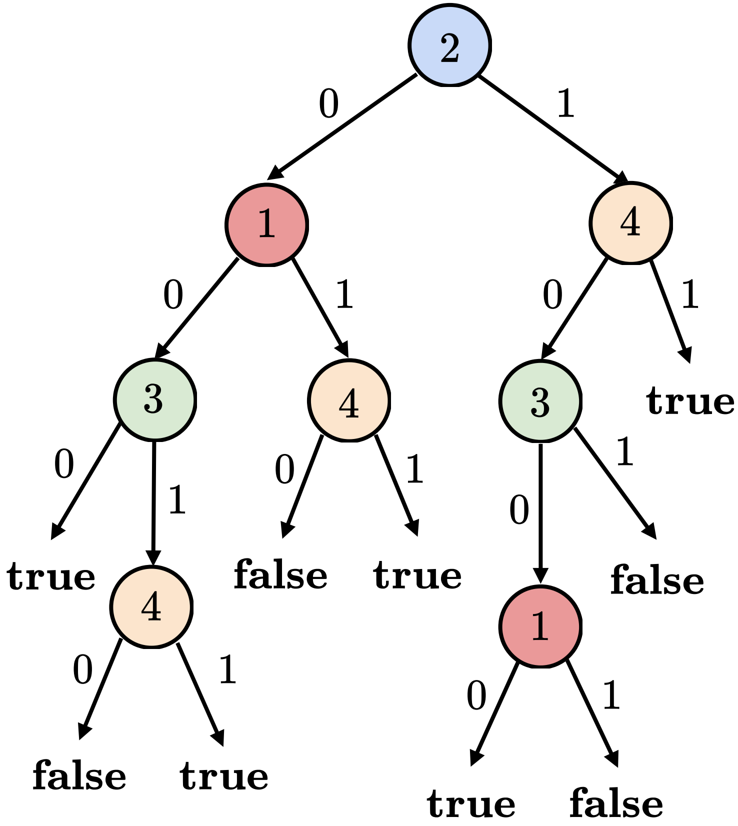


Notions of explanation: determinant feature set

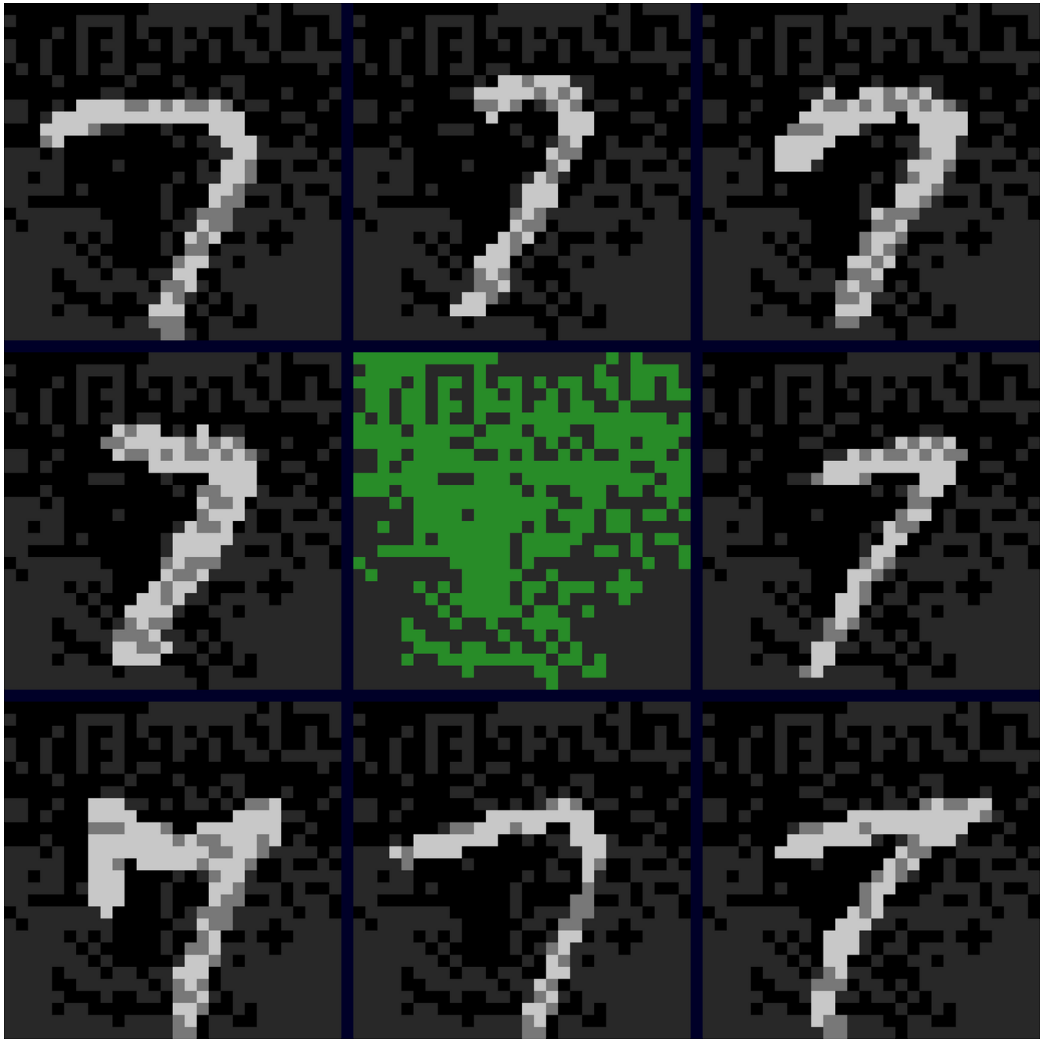
If the values of features $\{1, 3, 4\}$ are fixed, then the output of the model is fixed

If $\mathbf{e}[1] = \mathbf{e}[3] = 1$ and $\mathbf{e}[4] = 0$:

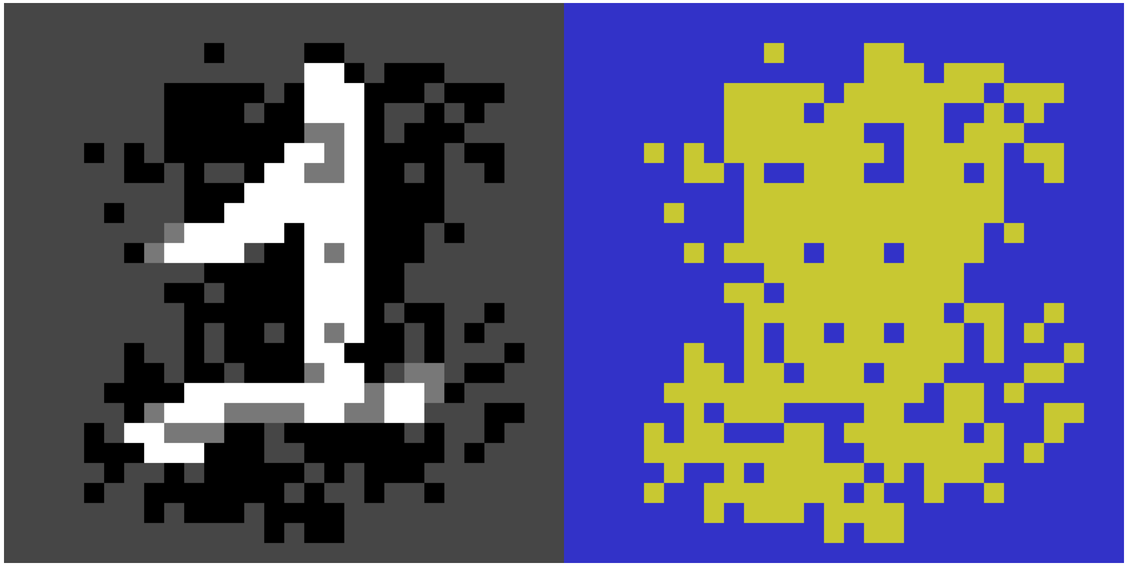
$$\mathcal{T}(\mathbf{e}) = 0$$



MNIST: determinant feature set

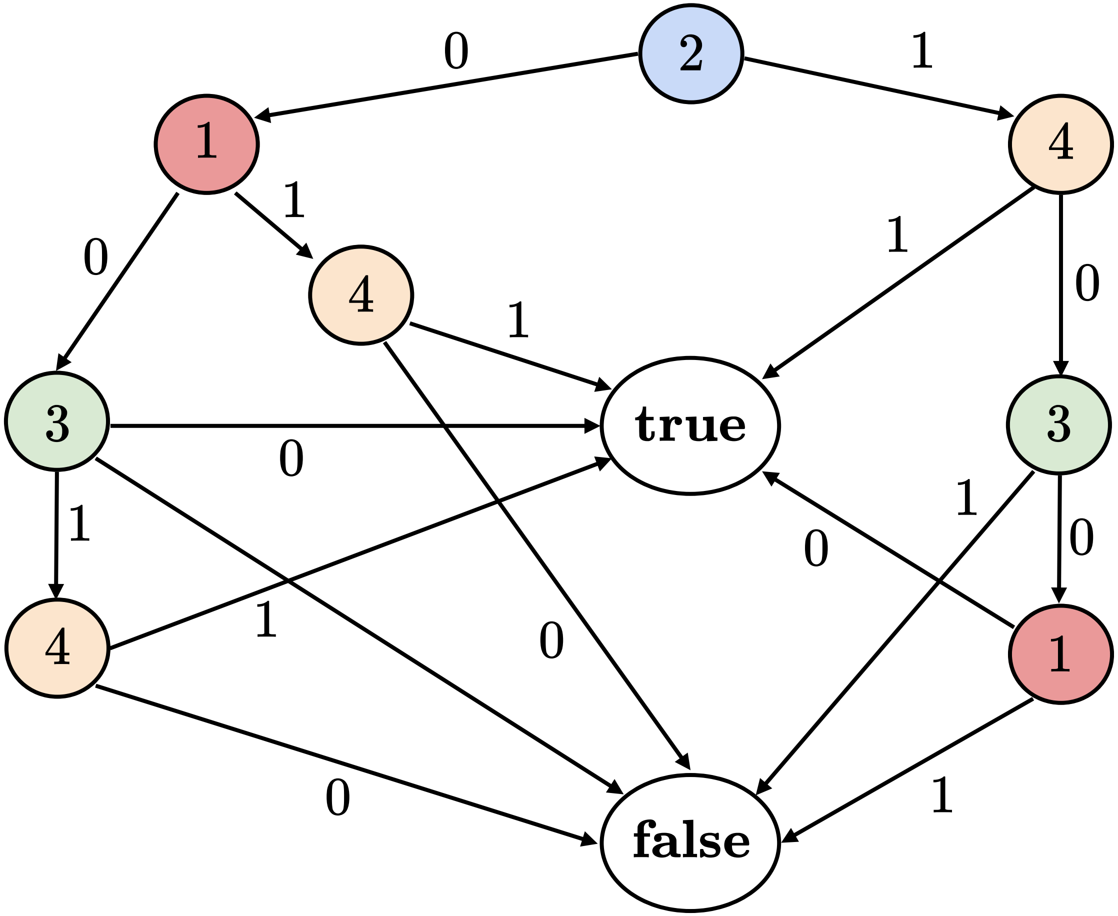


MNIST: minimal determinant feature set



How do we express the previous explainability queries?

- Is there a common framework for them?
- Is there a *natural* framework based on labeled graphs?



Binary decision diagrams (BDDs)

- OBDDs
- FBDDs

A first attempt: FOIL

First-order logic defined on a suitable vocabulary to describe classification models

Key notion: partial instance $\mathbf{e} \in \{0, 1, \bot\}^n$ of dimensión n

$\mathbf{e}_1$ is subsumed by $\mathbf{e}_2$ if $\mathbf{e}_1, \mathbf{e}_2$ are partial instances such that for every $i \in \{1, \ldots, n\}$, if $\mathbf{e}_1[i] \neq \bot$, then $\mathbf{e}_1[i] = \mathbf{e}_2[i]$

$$(1, \bot, 0, \bot) \subseteq (1, 0, 0, \bot) \subseteq (1, 0, 0, 1)$$

A first attempt: FOIL

First-order logic defined on a suitable vocabulary to describe classification models: $\{\mathrm{Pos}, \subseteq\}$

A model $\mathcal{M}$ of dimensión n is represented as a structure $\mathfrak{A}_{\mathcal{M}}$:

- The domain of $\mathfrak{A}_{\mathcal{M}}$ is $\{0, 1, \perp\}^n$
- $\mathrm{Pos}(\mathbf{e})$ holds if $\mathbf{e}$ is an instance such that $\mathcal{M}(\mathbf{e}) = 1$
- $\mathbf{e}_1 \subseteq \mathbf{e}_2$ holds if $\mathbf{e}_1, \mathbf{e}_2$ are partial instances such that $\mathbf{e}_1$ is subsumed by $\mathbf{e}_2$

The semantics of FOIL

Given a **FOIL** formula $\Phi(x_1, x_2, \ldots, x_k)$, a classification model $\mathcal{M}$ of dimensión n, and instances $\mathbf{e}_1$, $\mathbf{e}_2$, ..., $\mathbf{e}_k$

$$\mathcal{M} \models \Phi(\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k)$$

$$\iff$$

$$\mathfrak{A}_{\mathcal{M}} \models \Phi(\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k)$$

(in the usual sense)

Some examples

$$\mathrm{Full}(x) \;=\; \forall y\,(x \subseteq y \to x = y)$$

$$\mathrm{AllPos}(x) \;=\; \forall y\,\big((x \subseteq y \land \mathrm{Full}(y)) \to \mathrm{Pos}(y)\big)$$

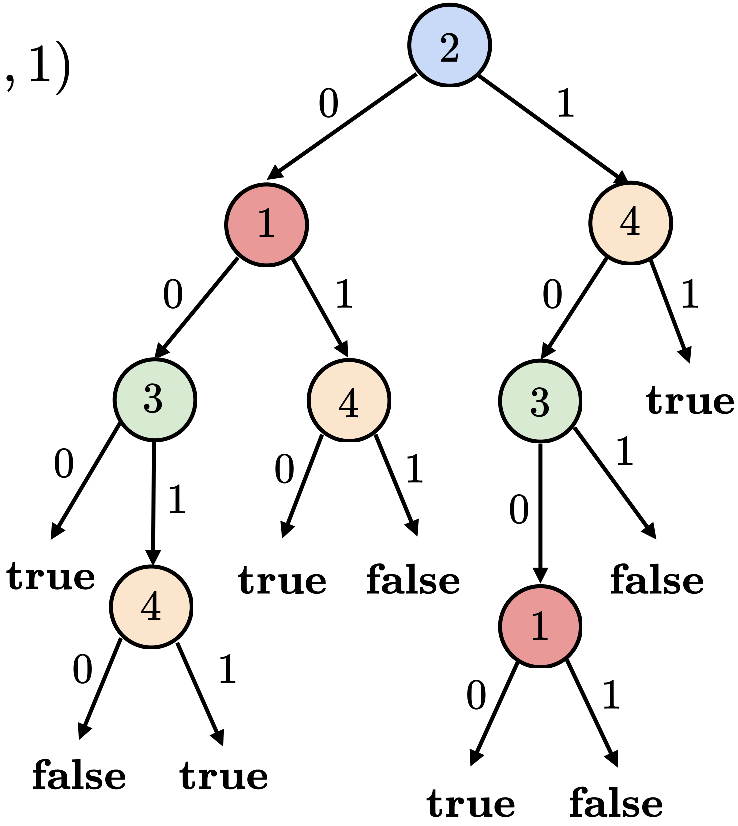
$$\mathrm{AllNeg}(x) \;=\; \forall y\,\big((x \subseteq y \land \mathrm{Full}(y)) \to \neg\mathrm{Pos}(y)\big)$$

Notions of explanation: sufficient reason

$\mathcal{T}(\mathbf{e}) = 1$ for $\mathbf{e} = (1,1,1,1)$, and $\mathbf{e}_1 = (1,1,\bot,1)$ is a sufficient reason for this

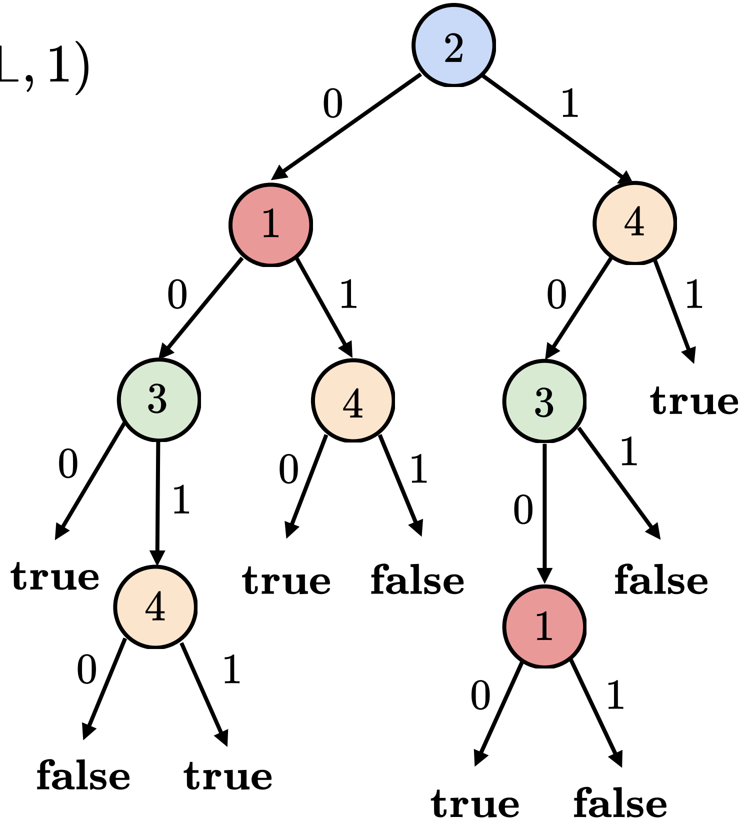
$$\mathcal{T} \models \mathrm{SR}(\mathbf{e}, \mathbf{e}_1)$$

$$\begin{aligned} \mathrm{SR}(x,y) \ = \ & \mathrm{Full}(x) \wedge y \subseteq x \ \wedge \\ & (\mathrm{Pos}(x) \rightarrow \mathrm{AllPos}(y)) \wedge \\ & (\neg\mathrm{Pos}(x) \rightarrow \mathrm{AllNeg}(y)) \end{aligned}$$



Notions of explanation: minimal sufficient reason

$\mathcal{T}(\mathbf{e}) = 1$ for $\mathbf{e} = (1,1,1,1)$, and $\mathbf{e}_2 = (\bot, 1, \bot, 1)$ is a minimal sufficient reason for this

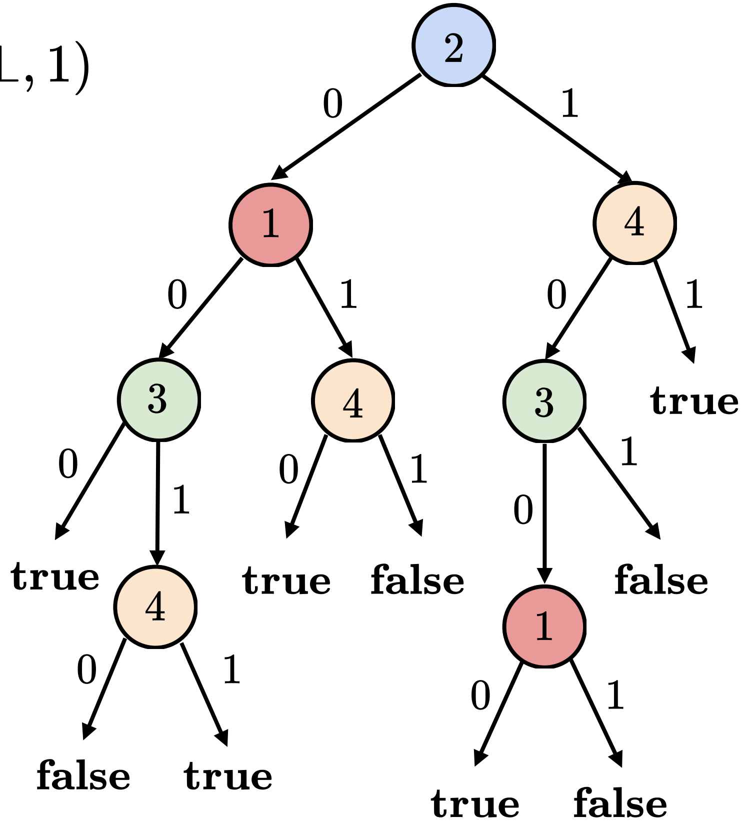


Notions of explanation: minimal sufficient reason

$\mathcal{T}(\mathbf{e}) = 1$ for $\mathbf{e} = (1,1,1,1)$, and $\mathbf{e}_2 = (\bot, 1, \bot, 1)$ is a minimal sufficient reason for this

$$\mathcal{T} \models \text{MinimalSR}(\mathbf{e}, \mathbf{e}_2)$$

$$\text{MinimalSR}(x, y) \;=\; \text{SR}(x, y) \wedge$$
$$\forall z \left((\text{SR}(x, z) \wedge z \subseteq y) \rightarrow z = y \right)$$

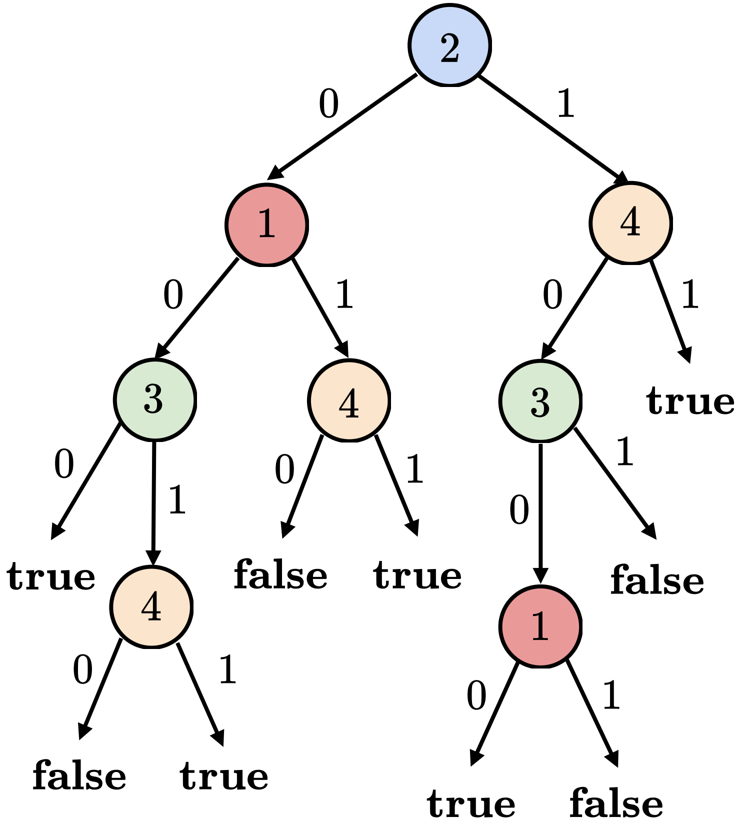


Notions of explanation: determinant feature set

If the values of features $\{1, 3, 4\}$ are fixed, then the output of the model is fixed

$\mathbf{e} = (1, \perp, 1, 1)$ is a determinant feature set

- Value 1 is not relevant in this instance



Notions of explanation: determinant feature set:

$\mathrm{SUF}(x, y)$: holds if and only if the sets of undefined features in x and y are the same

- If $\mathbf{e} = (1, \bot, 1, 1)$ and $\mathbf{e}_1 = (0, \bot, 0, 1)$, then $\mathrm{SUF}(\mathbf{e}, \mathbf{e}_1)$ holds
- If $\mathbf{e} = (1, \bot, 1, 1)$ and $\mathbf{e}_2 = (1, \bot, 1, \bot)$, then $\mathrm{SUF}(\mathbf{e}, \mathbf{e}_2)$ does not hold
- If $\mathbf{e} = (1, \bot, 1, 1)$ and $\mathbf{e}_3 = (1, 1, \bot, 1)$, then $\mathrm{SUF}(\mathbf{e}, \mathbf{e}_3)$ does not hold

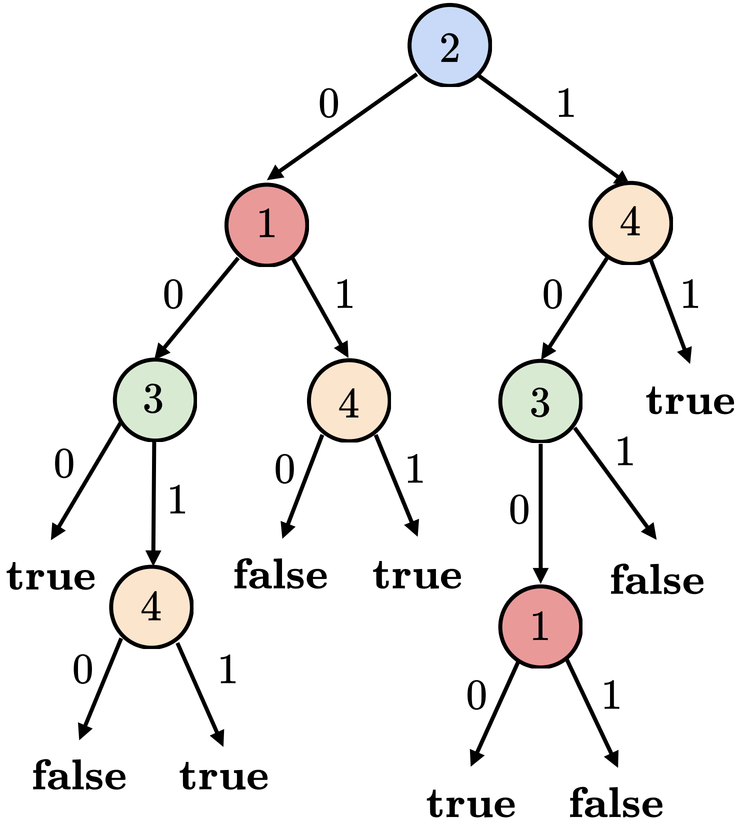
$\mathrm{SUF}(x, y)$ can be expressed in FOIL using only the predicate $\subseteq$

Notions of explanation: determinant feature set

$\mathbf{e} = (1, \bot, 1, 1)$ is a determinant feature set

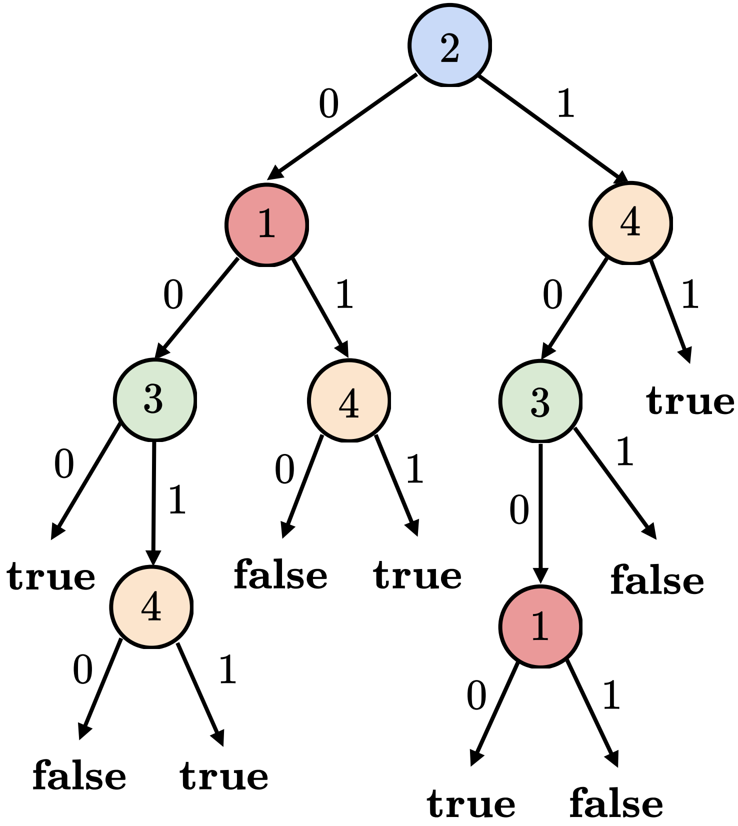
$$\mathcal{T} \models \mathrm{DFS}(\mathbf{e})$$

$$\mathrm{DFS}(x) \; = \forall y \, (\mathrm{SUF}(x, y) \rightarrow \mathrm{AllPos}(y) \vee \mathrm{AllNeg}(y))$$



Notions of explanation: minimal determinant feature set

$$\text{MinimalDFS}(x) = \text{DFS}(x) \wedge \forall y \left((\text{DFS}(y) \wedge y \subseteq x) \rightarrow y = x \right)$$



Expressiveness and complexity of FOIL

- What notions of explanation can be expressed in **FOIL**?
- What notions of explanation cannot be expressed in **FOIL**?
- What is the complexity of the evaluation problem for **FOIL**?

The evaluation problem for FOIL

We consider the data complexity of the problem, so fix a **FOIL** formula $\Phi(x_1, \ldots, x_k)$

$\mathbf{Eval(\Phi)}$**:**

- **Input:** decision tree $\mathcal{T}$ of dimension n and partial instances $\mathbf{e}_1, \ldots, \mathbf{e}_k$ of dimension n
- **Output:** yes if $\mathcal{T} \models \Phi(\mathbf{e}_1, \ldots, \mathbf{e}_k)$, and no otherwise

The evaluation problem for FOIL

$$\mathcal{T} \models \Phi(\mathbf{e}_1, \ldots, \mathbf{e}_k) \quad \text{if and ony if} \quad \mathfrak{A}_{\mathcal{T}} \models \Phi(\mathbf{e}_1, \ldots, \mathbf{e}_k)$$

But $\mathfrak{A}_{\mathcal{T}}$ could be of exponential size in the size of $\mathcal{T}$

- $\mathfrak{A}_{\mathcal{T}}$ should not be materialized to check whether $\mathcal{T} \models \Phi(\mathbf{e}_1, \ldots, \mathbf{e}_k)$
- $\mathfrak{A}_{\mathcal{T}}$ is used only to define the semantics of **FOIL**

Bad news ...

Theorem:

1. For every **FOIL** formula Φ, there exists $k \geq 0$ such that $\mathrm{Eval}(\Phi)$ is in Σ_k^{P}
2. For every $k \geq 0$, there exists a **FOIL** formula Φ such that $\mathrm{Eval}(\Phi)$ is Σ_k^{P}-hard

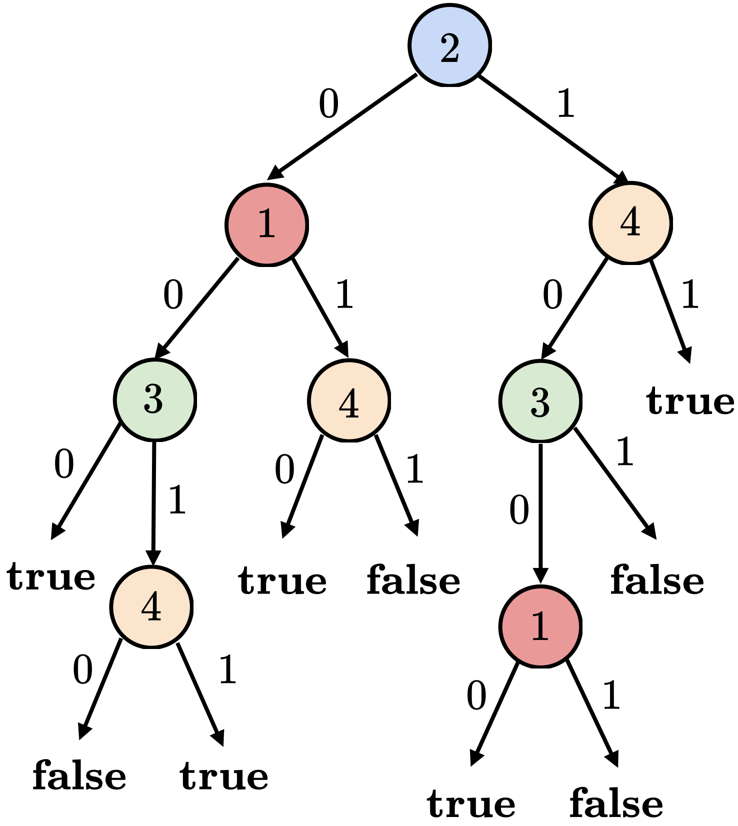
More bad news ...

$\mathcal{T}(\mathbf{e}) = 1$ for $\mathbf{e} = (1, 1, 1, 1)$

$\{2, 4\}$ is a minimum sufficient reason for $\mathbf{e}$ over $\mathcal{T}$

- There is no sufficient reason for $\mathbf{e}$ over $\mathcal{T}$ with a smaller number of features

$\mathbf{e}_2 = (\perp, 1, \perp, 1)$ is a minimum sufficient reason for $\mathbf{e}$ over $\mathcal{T}$



More bad news ...

Theorem:

There is no **FOIL** formula $\mathrm{MinimumSR}(x, y)$ such that, for every decision tree $\mathcal{T}$, instance $\mathbf{e}_1$ and partial instance $\mathbf{e}_2$:

$$\mathcal{T} \models \mathrm{MinimumSR}(\mathbf{e}_1, \mathbf{e}_2)$$

$$\iff$$

$\mathbf{e}_2$ is a minimum sufficient reason for $\mathbf{e}_1$ over $\mathcal{T}$

How do we overcome these limitations?

- We use first-order logic over a larger vocabulary
- We depart from the model-agnostic approach of FOIL and use the notion of *guarded* quantification for decision trees

The logic DT-FOIL

DT-FOIL is defined by considering two layers

1. Atomic formulas
2. Guarded formulas

The first layer

$\subseteq$ can be considered as a *syntactic* predicate, it does not refer to the models

We need another predicate like that. Given partial instances $\mathbf{e}_1, \mathbf{e}_2$ of dimension n:

$$\mathbf{e}_1 \preceq \mathbf{e}_2$$

if and only if

$$|\{i \in \{1, \ldots n\} \mid \mathbf{e}_1[i] = \bot\}| \geq |\{i \in \{1, \ldots n\} \mid \mathbf{e}_2[i] = \bot\}|$$

Why do we need another syntactic predicate?

$$\mathrm{MinimumSR}(x, y) \;=\; \mathrm{SR}(x, y) \wedge$$
$$\forall z \left((\mathrm{SR}(x, z) \wedge z \preceq y) \rightarrow y = z\right)$$

How many more predicates do we need to include?

Atomic formulas

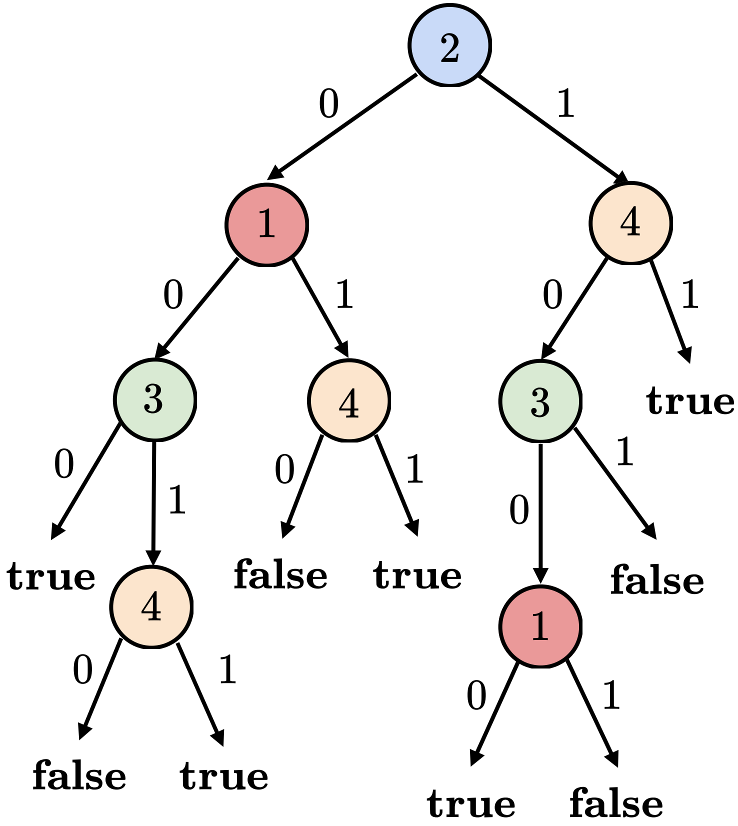
All the syntactic predicates needed in our formalism can be expressed as first-order queries over $\{\subseteq, \preceq\}$

Theorem: if Φ is a first-order formula defined over $\{\subseteq, \preceq\}$, then $\mathrm{Eval}(\Phi)$ can be solved in polynomial time

Atomic formulas of DT-FOIL: the set of first-order formulas defined over $\{\subseteq, \preceq\}$

The second layer

Node($\mathbf{e}$):



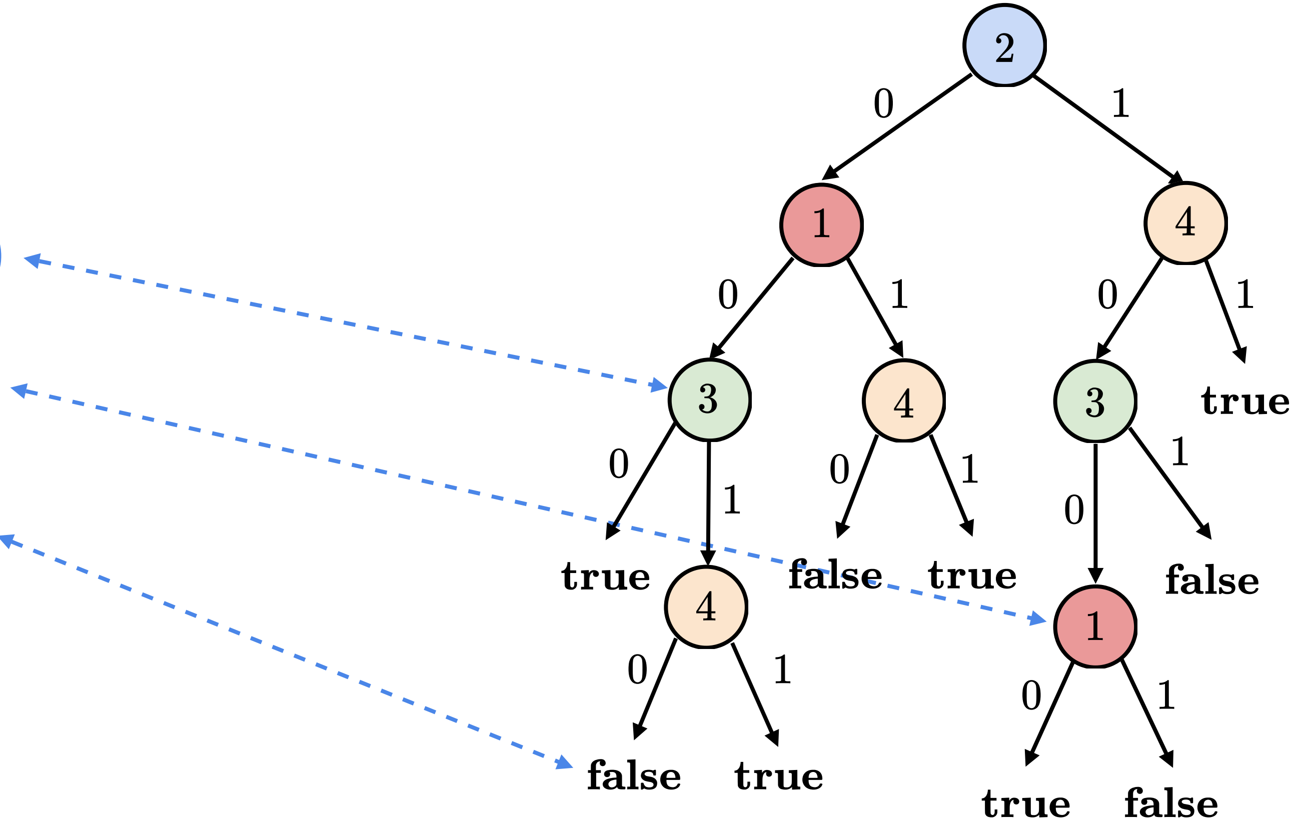
The second layer

Node($\mathbf{e}$):

$(0, 0, \perp, \perp)$

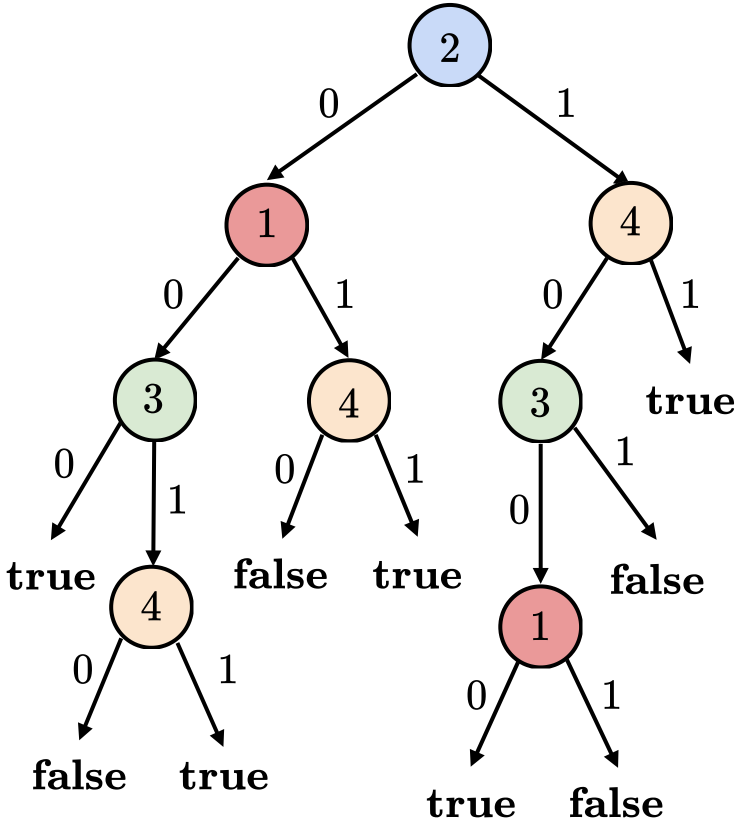
$(\perp, 1, 0, 0)$

$(0, 0, 1, 0)$



The second layer

$\mathrm{PosLeaf}(\mathbf{e})$:

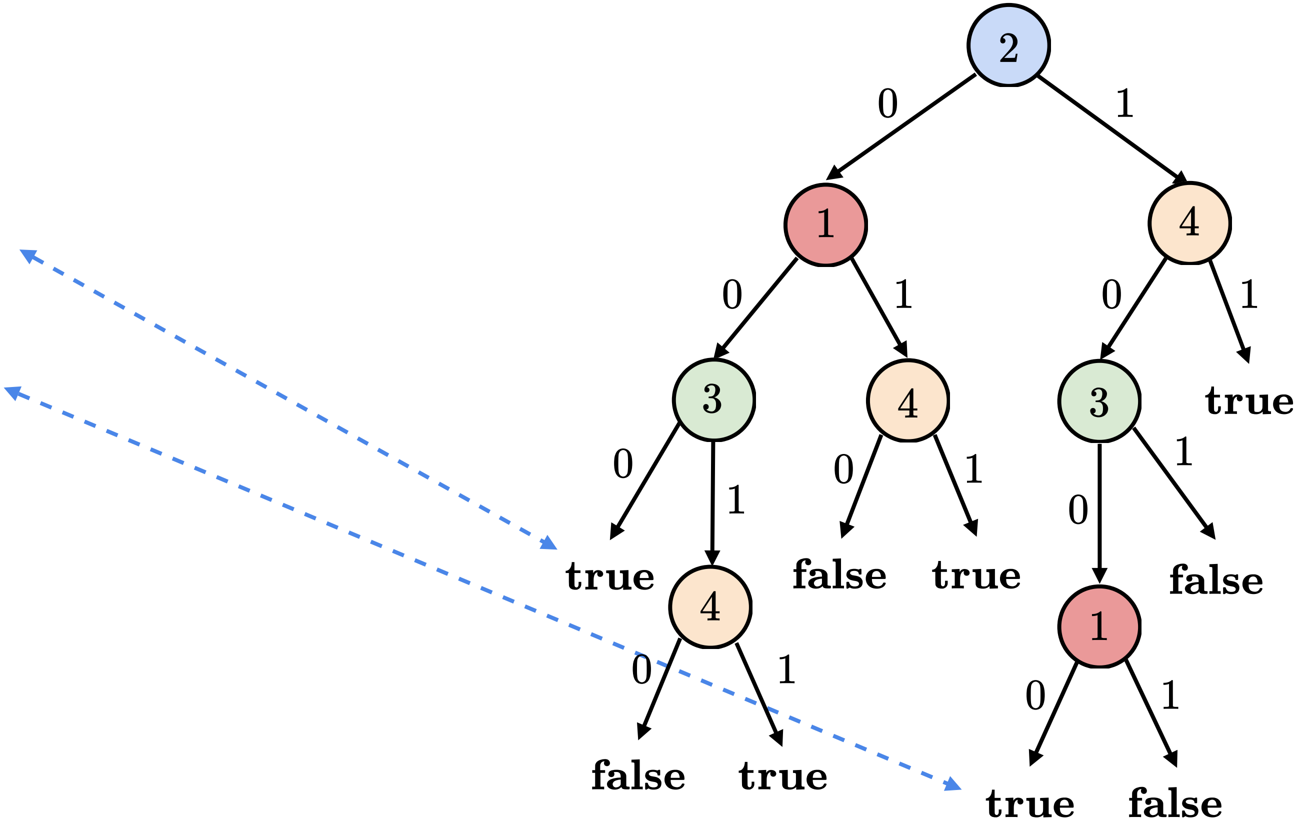


The second layer

PosLeaf($\mathbf{e}$):

$(0, 0, 0, \perp)$

$(0, 1, 0, 0)$



The definition of DT-FOIL

1. Each atomic formula is a DT-FOIL formula
2. Boolean combinations of DT-FOIL formulas are DT-FOIL formulas
3. If Φ is a DT-FOIL formula, then so are

$$\exists x \, (\text{Node}(x) \land \Phi) \qquad \forall x \, (\text{Node}(x) \rightarrow \Phi)$$

$$\exists x \, (\text{PosLeaf}(x) \land \Phi) \qquad \forall x \, (\text{PosLeaf}(x) \rightarrow \Phi)$$

An example of a DT-FOIL formula

$$
\begin{aligned}
\mathrm{SR}(x, y) \;=\; & \mathrm{Full}(x) \wedge y \subseteq x \wedge \\
& (\mathrm{Pos}(x) \rightarrow \mathrm{AllPos}(y)) \wedge \\
& (\neg \mathrm{Pos}(x) \rightarrow \mathrm{AllNeg}(y))
\end{aligned}
$$

DT-FOIL formulas

An example of a DT-FOIL formula

$$\mathrm{Cons}(x, y) = \exists z\,(x \subseteq z \wedge y \subseteq z)$$

An example of a DT-FOIL formula

$$\mathrm{Cons}(x, y) = \exists z\, (x \subseteq z \wedge y \subseteq z)$$

$$\mathrm{Pos}(x) = \mathrm{Full}(x) \wedge \exists y\, (\mathrm{PosLeaf}(y) \wedge \mathrm{Cons}(x, y))$$

An example of a DT-FOIL formula

$$\text{Cons}(x, y) = \exists z \, (x \subseteq z \wedge y \subseteq z)$$

$$\text{Pos}(x) = \text{Full}(x) \wedge \exists y \, (\text{PosLeaf}(y) \wedge \text{Cons}(x, y))$$

guarded quantification

An example of a DT-FOIL formula

$$\mathrm{Cons}(x, y) = \exists z\, (x \subseteq z \wedge y \subseteq z)$$

$$\mathrm{Pos}(x) = \mathrm{Full}(x) \wedge \exists y\, (\mathrm{PosLeaf}(y) \wedge \mathrm{Cons}(x, y))$$

$$\mathrm{Leaf}(x) = \mathrm{Node}(x) \wedge \forall y\, (\mathrm{Node}(y) \rightarrow (x \subseteq y \rightarrow x = y))$$

guarded quantification

An example of a DT-FOIL formula

$$\mathrm{Cons}(x, y) = \exists z\, (x \subseteq z \wedge y \subseteq z)$$

$$\mathrm{Pos}(x) = \mathrm{Full}(x) \wedge \exists y\, (\mathrm{PosLeaf}(y) \wedge \mathrm{Cons}(x, y))$$

$$\mathrm{Leaf}(x) = \mathrm{Node}(x) \wedge \forall y\, (\mathrm{Node}(y) \rightarrow (x \subseteq y \rightarrow x = y))$$

$$\mathrm{AllPos}(x) = \forall y\, (\mathrm{Node}(y) \rightarrow (\mathrm{Leaf}(y) \wedge \mathrm{Cons}(x, y)) \rightarrow \mathrm{PosLeaf}(y))$$

An example of a DT-FOIL formula

$$\mathrm{Cons}(x, y) = \exists z\, (x \subseteq z \wedge y \subseteq z)$$

$$\mathrm{Pos}(x) = \mathrm{Full}(x) \wedge \exists y\, (\mathrm{PosLeaf}(y) \wedge \mathrm{Cons}(x, y))$$

$$\mathrm{Leaf}(x) = \mathrm{Node}(x) \wedge \forall y\, (\mathrm{Node}(y) \rightarrow (x \subseteq y \rightarrow x = y))$$

$$\mathrm{AllPos}(x) = \forall y\, (\mathrm{Node}(y) \rightarrow (\mathrm{Leaf}(y) \wedge \mathrm{Cons}(x, y)) \rightarrow \mathrm{PosLeaf}(y))$$

guarded quantification

Is DT-FOIL enough?

- Every formula in DT-FOIL can be evaluated in polynomial time
- $\mathbb{SR}$ and $\mathbb{DFS}$ can be expressed in DT-FOIL
- But it lacks a mechanism to express optimality conditions

Two possible solutions

Q-DT-FOIL: extends DT-FOIL with non-guarded quantification, but without alternation of these quantifiers

OPT-DT-FOIL: extends DT-FOIL with a minimal operator

Two possible solutions

All the notions of explanation discussed in this talk can be expressed in **Q-DT-FOIL** and **OPT-DT-FOIL**

- $\mathrm{SR}(x,y)$, $\mathrm{MinimalSR}(x,y)$, $\mathrm{MinimumSR}(x,y)$, $\mathrm{DFS}(x)$, $\mathrm{MinimalDFS}(x)$

The evaluation problem for these logics can be solved with a polynomial number of calls to a SAT solver

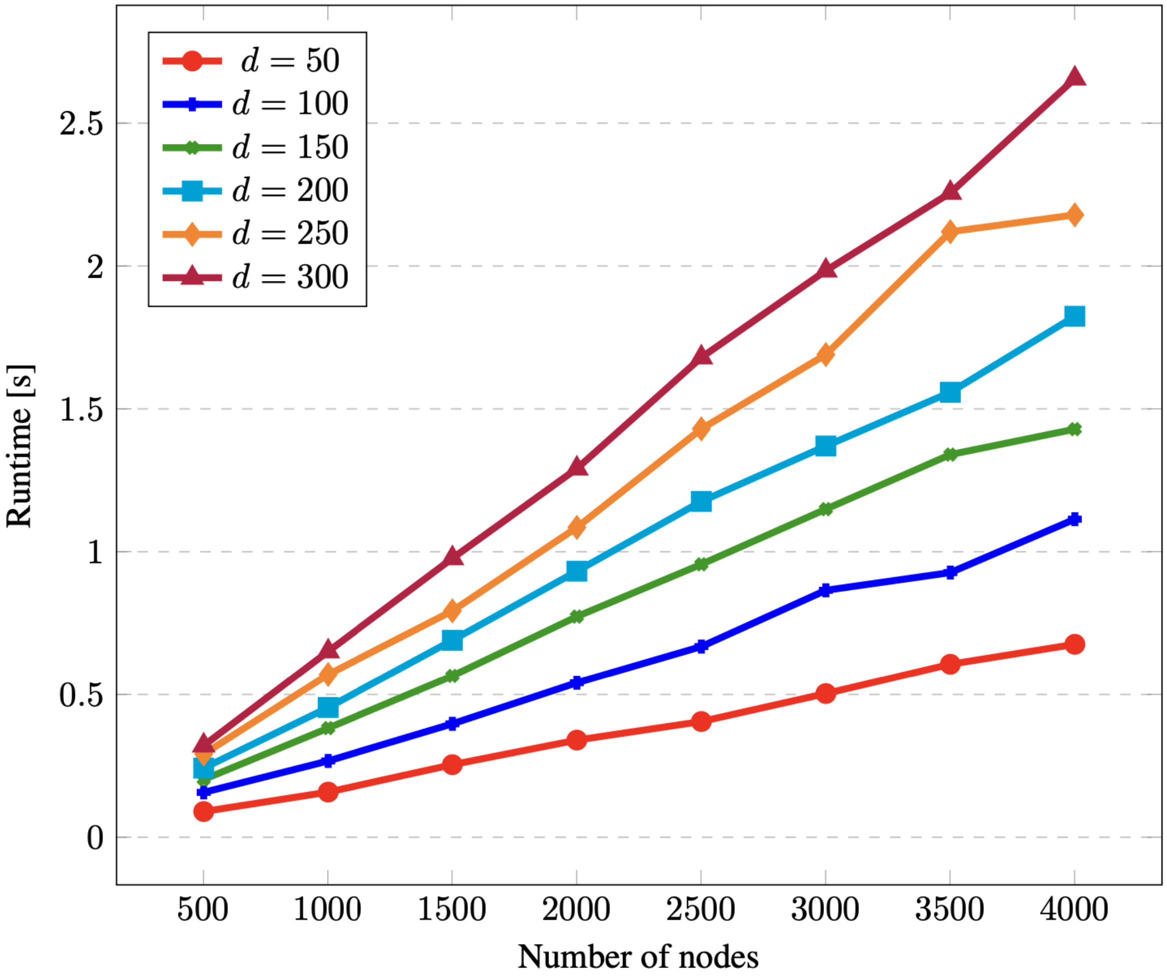
Implementation based on SAT solvers

Any SAT solver can be used

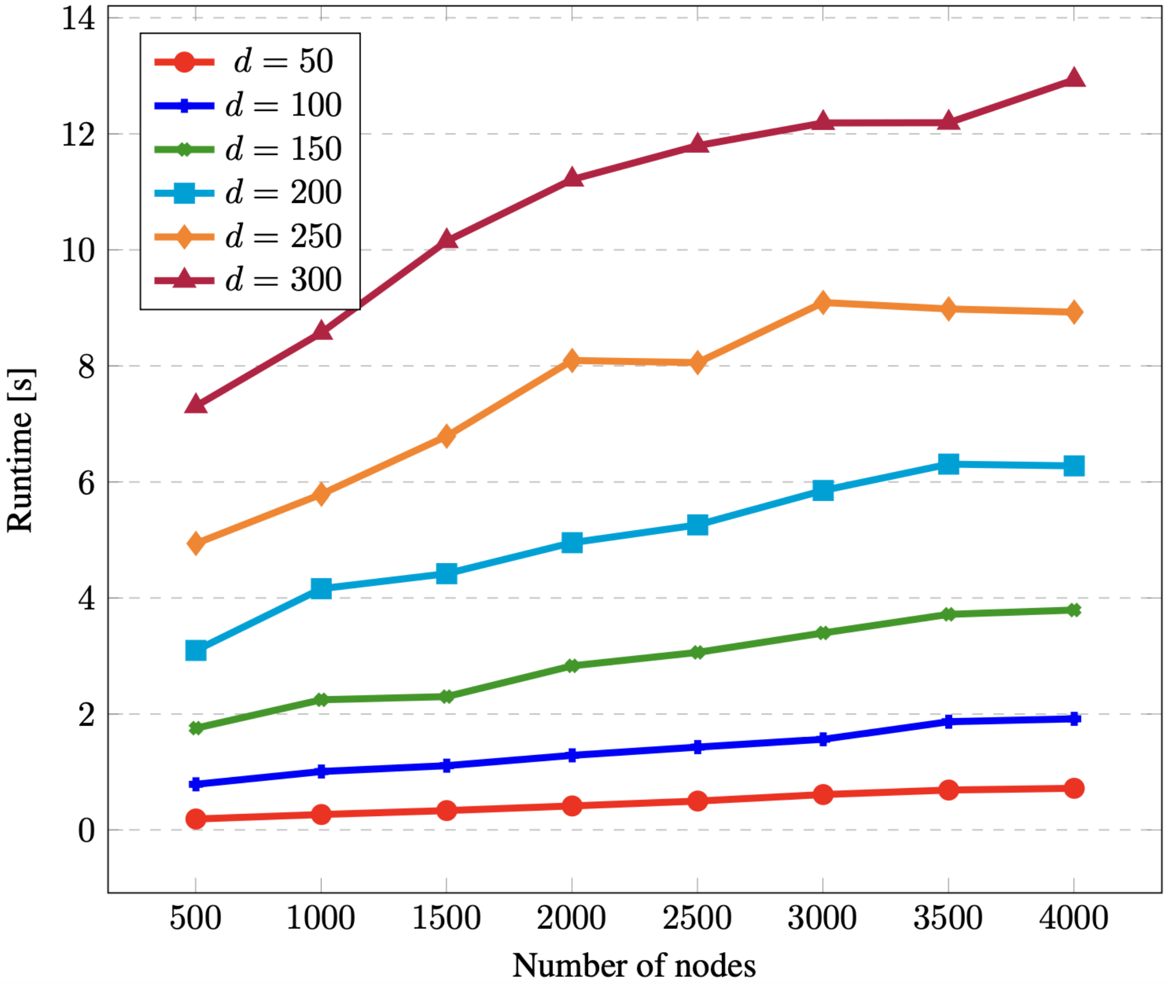
Given the complexity of the evaluation problem for **Q-DT-FOIL** and **OPT-DT-FOIL**, we use:

- **YalSAT:** to find a truth assignment that satisfies a propositional formula
- **Kissat:** to prove that a propositional formula is not satisfiable

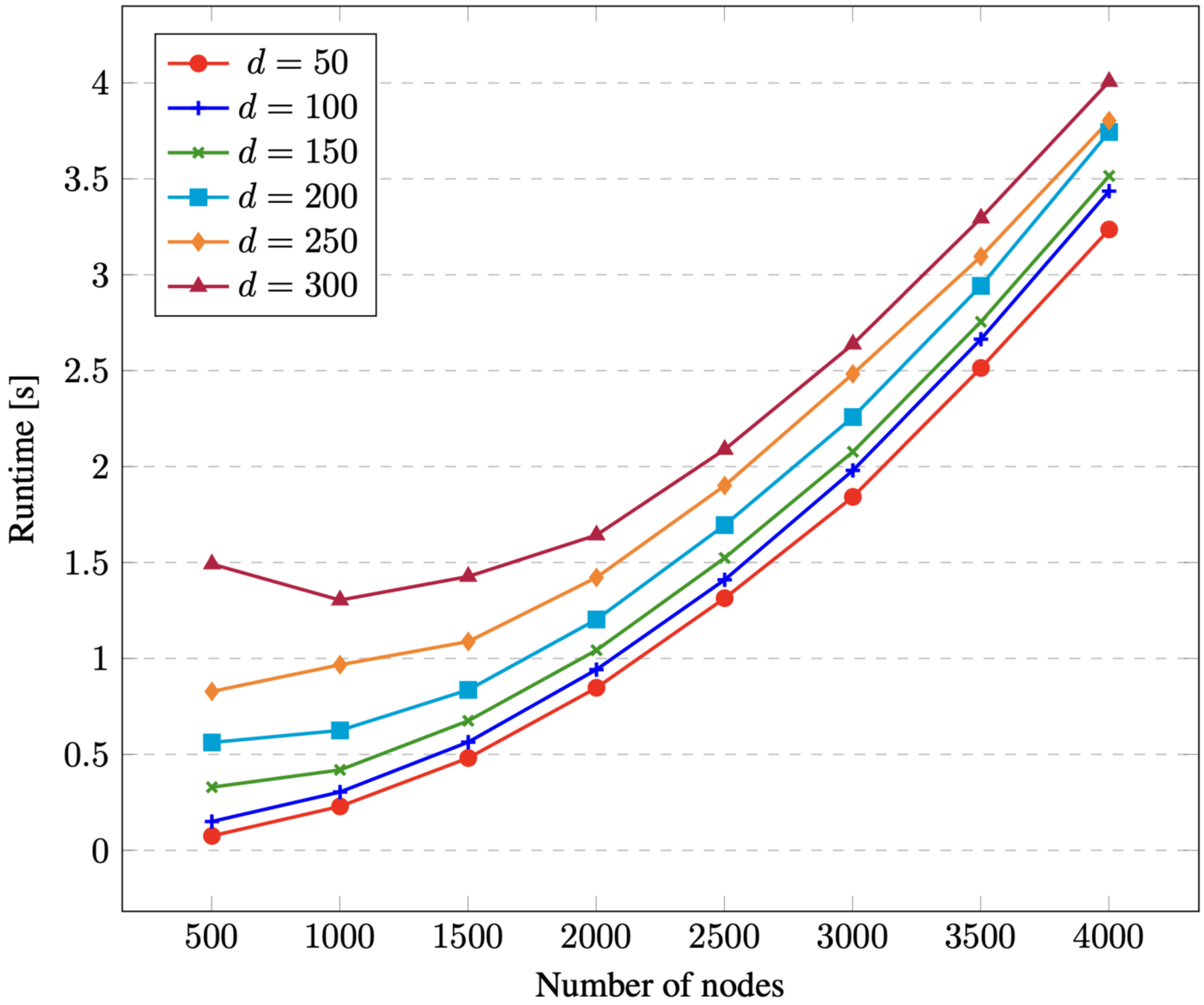
Minimal sufficient reason



Minimum sufficient reason



Minimal determinant feature set



Concluding remarks

DT-FOIL is a model-specific explainability query language

- How can the definition of **DT-FOIL** be extended to OBDDs and FBDDs?

Concluding remarks

FOIL is a model-agnostic interpretability query language

- The evaluation problem for some fragments of **FOIL** can be solved in polynomial time for decision trees and OBDDs
- What is an appropriate fragment of **FOIL** to be evaluated using SAT solvers?
- What is an appropriate explainability query language for FBDDs that is based on **FOIL**?

Concluding remarks

How can probabilities be incorporated into this framework?

- A probability distribution on the possible values of features, and a probabilistic classifier

Probabilistic circuits seem to be the right model for this

- A natural and robust generalization of Boolean circuits, with many well-understood properties

Thanks!