A data management approach to explainable AI

Marcelo Arenas

PUC & IMFD Chile and RelationalAI

Santiago Chile

SIGMOD PODS 2024

Explainable AI

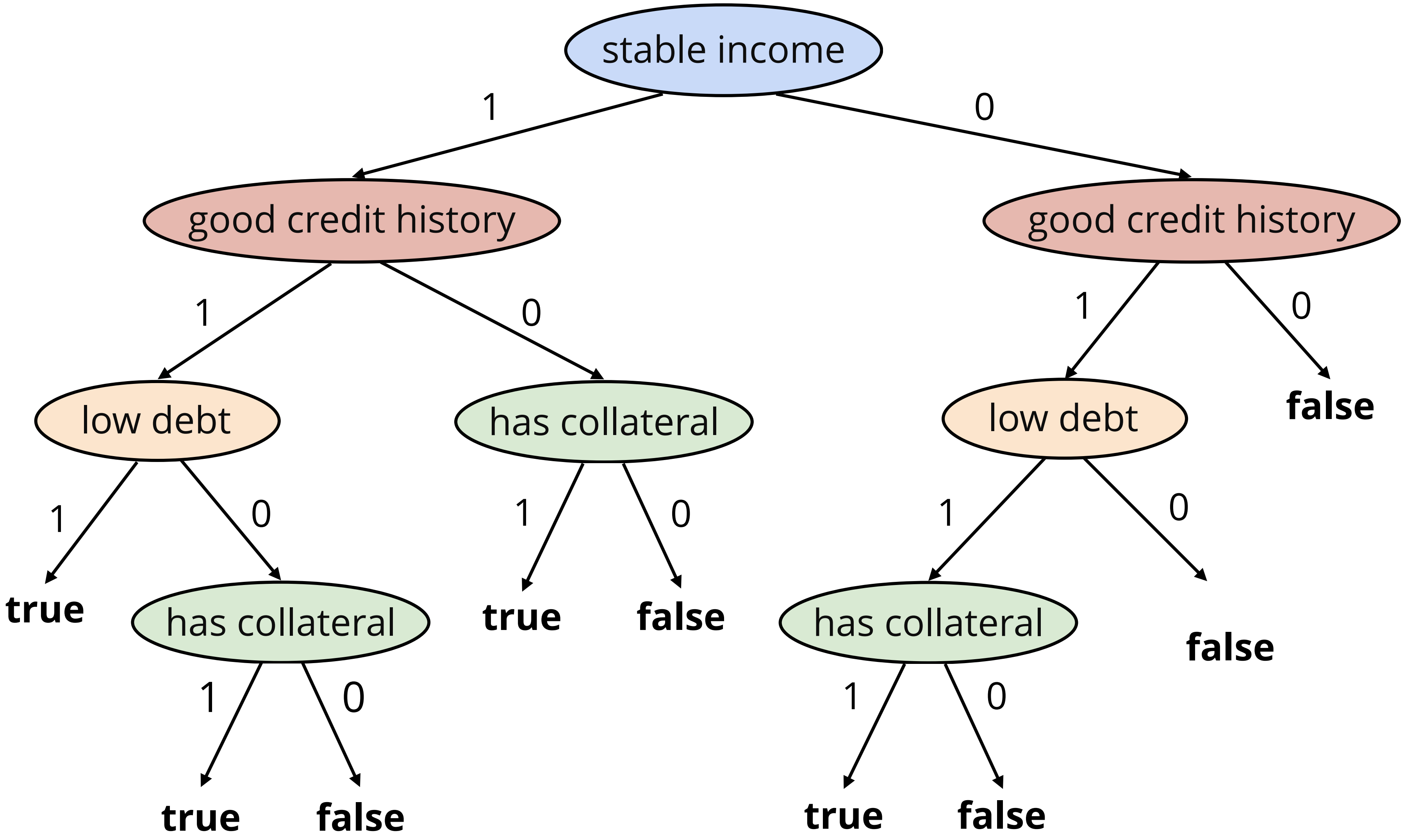
- There is a great interest in developing methods to explain predictions made by ML models
- This has led to the introduction of numerous queries and scores that aim to explain the predictions of ML models

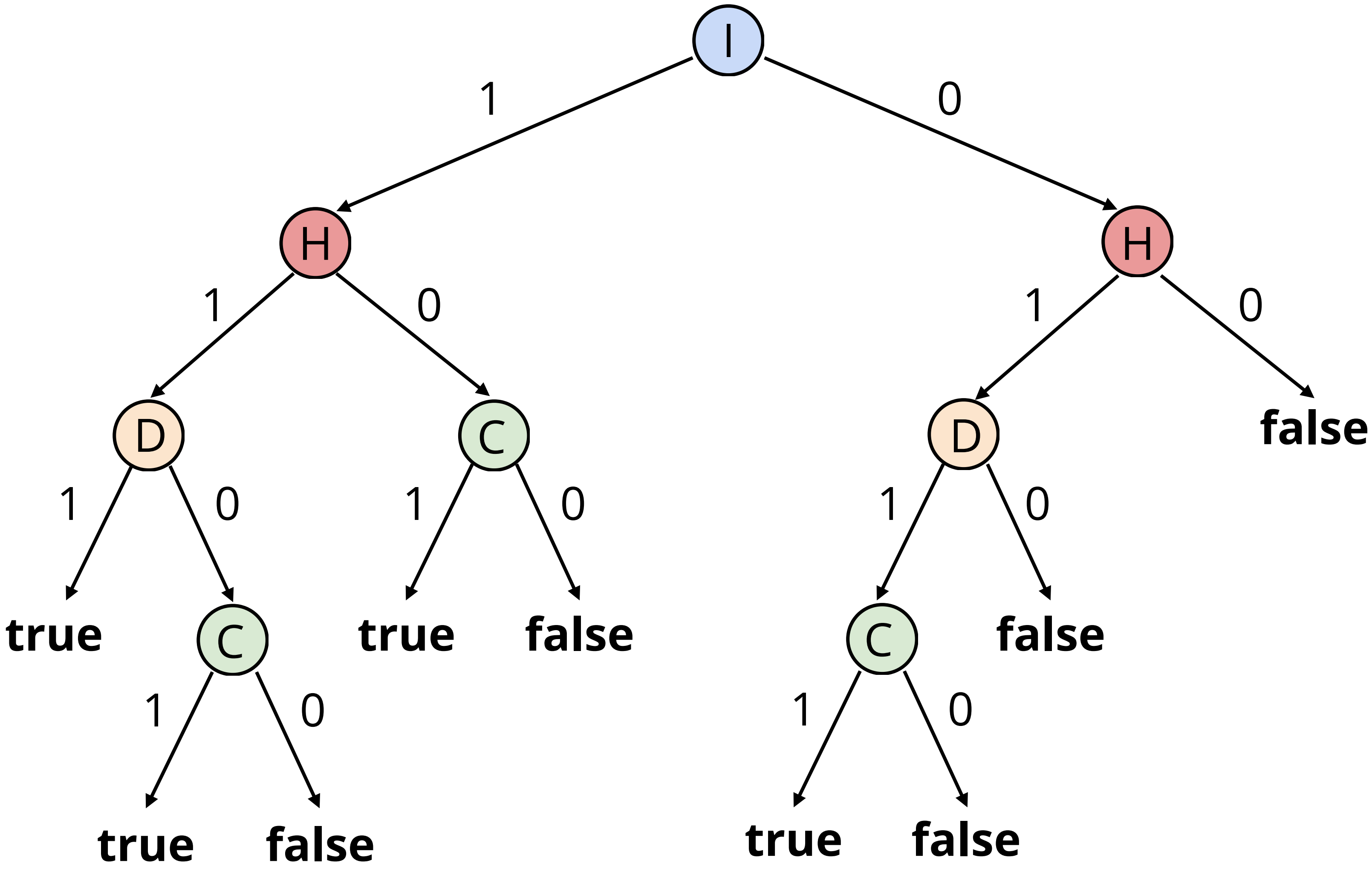
We focus here on formal explainable AI

A growing area that focuses on computing explanations with mathematical guarantees for the predictions made by ML models

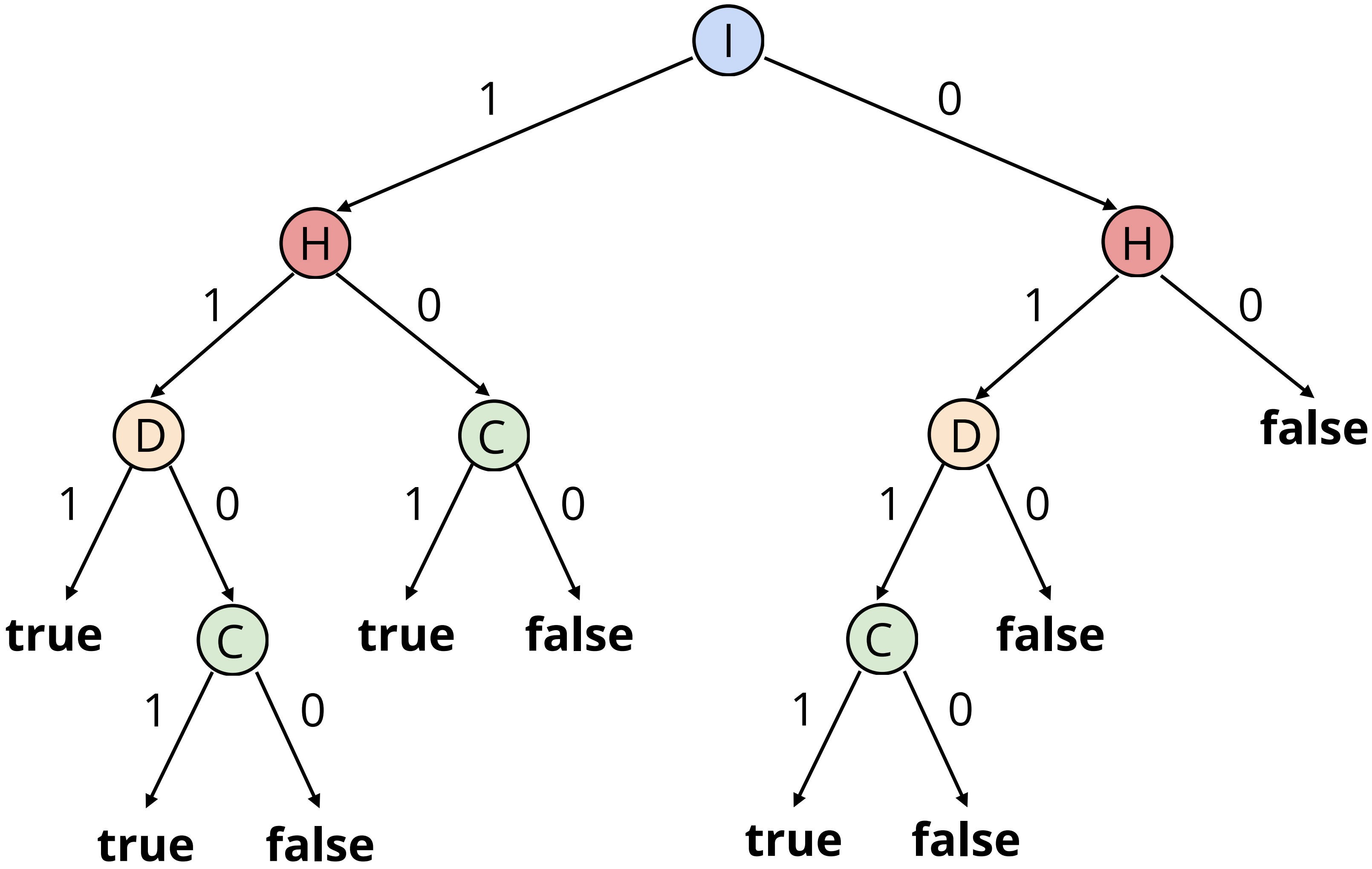
In particular, we focus on a logic-based approach to formal explainable AI

Some fundamental explainability queries

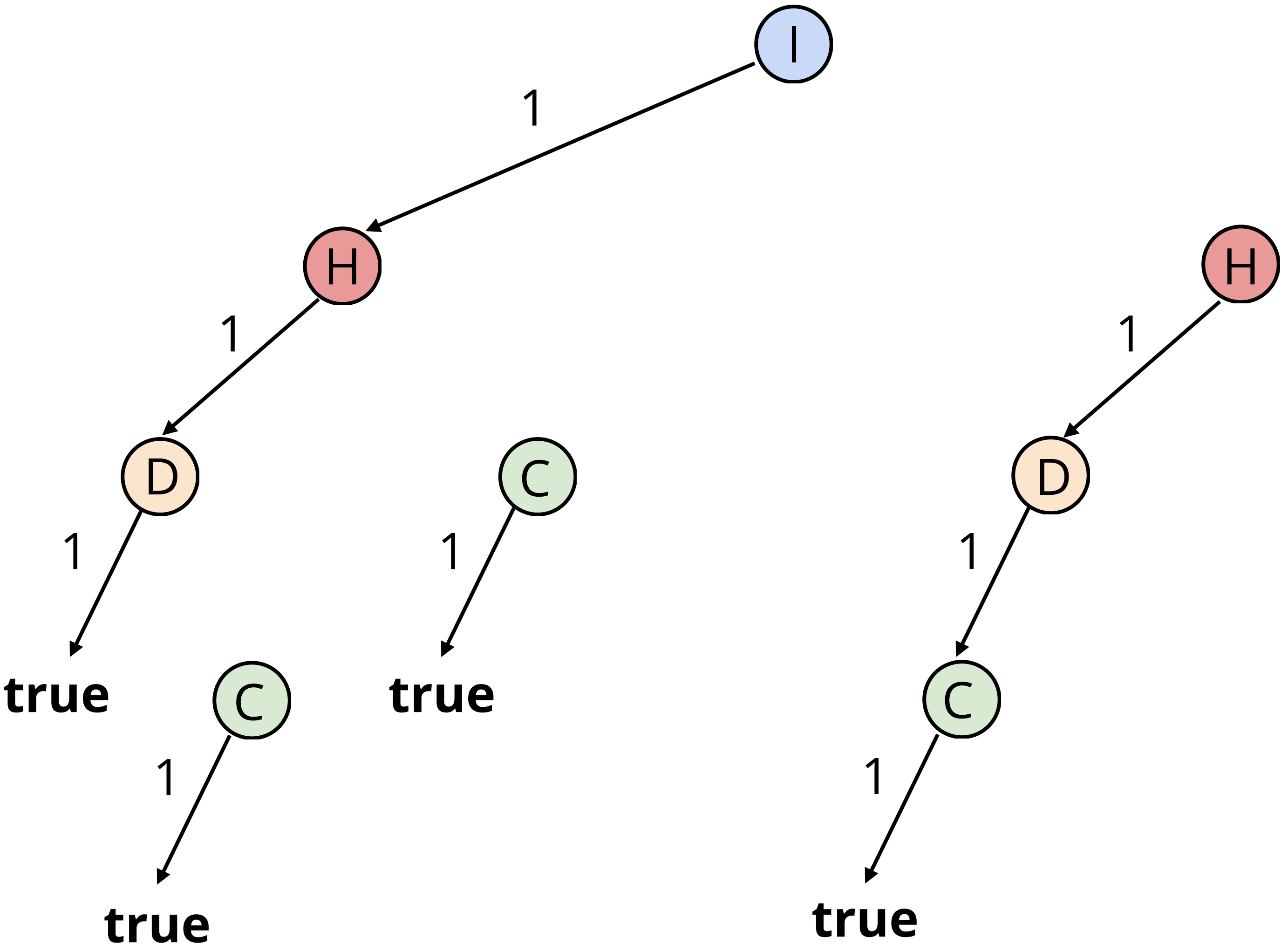

stable income
1
0
good credit history
good credit history
1
0
1
0
low debt
has collateral
low debt
false
1
0
1
0
1
0
true
has collateral
true
false
has collateral
false
1
0
1
0
true
false
true
false


I
1
0
H
H
1
0
1
0
D
C
D
false
1
0
1
0
1
0
true
C
true
false
C
false
1
0
1
0
true
false
true
false

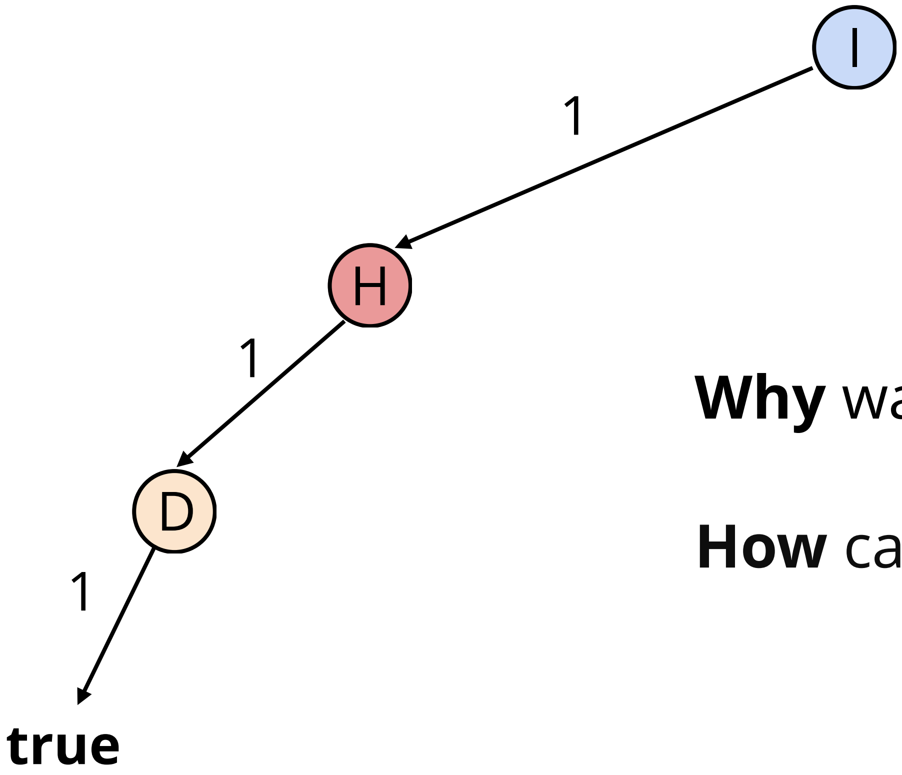
$$I \to 1 \quad H \to 1 \quad D \to 1 \quad C \to 1$$



$$I \to 1 \quad H \to 1 \quad D \to 1 \quad C \to 1$$



$I \rightarrow 1 \quad H \rightarrow 1 \quad D \rightarrow 1 \quad C \rightarrow 1$

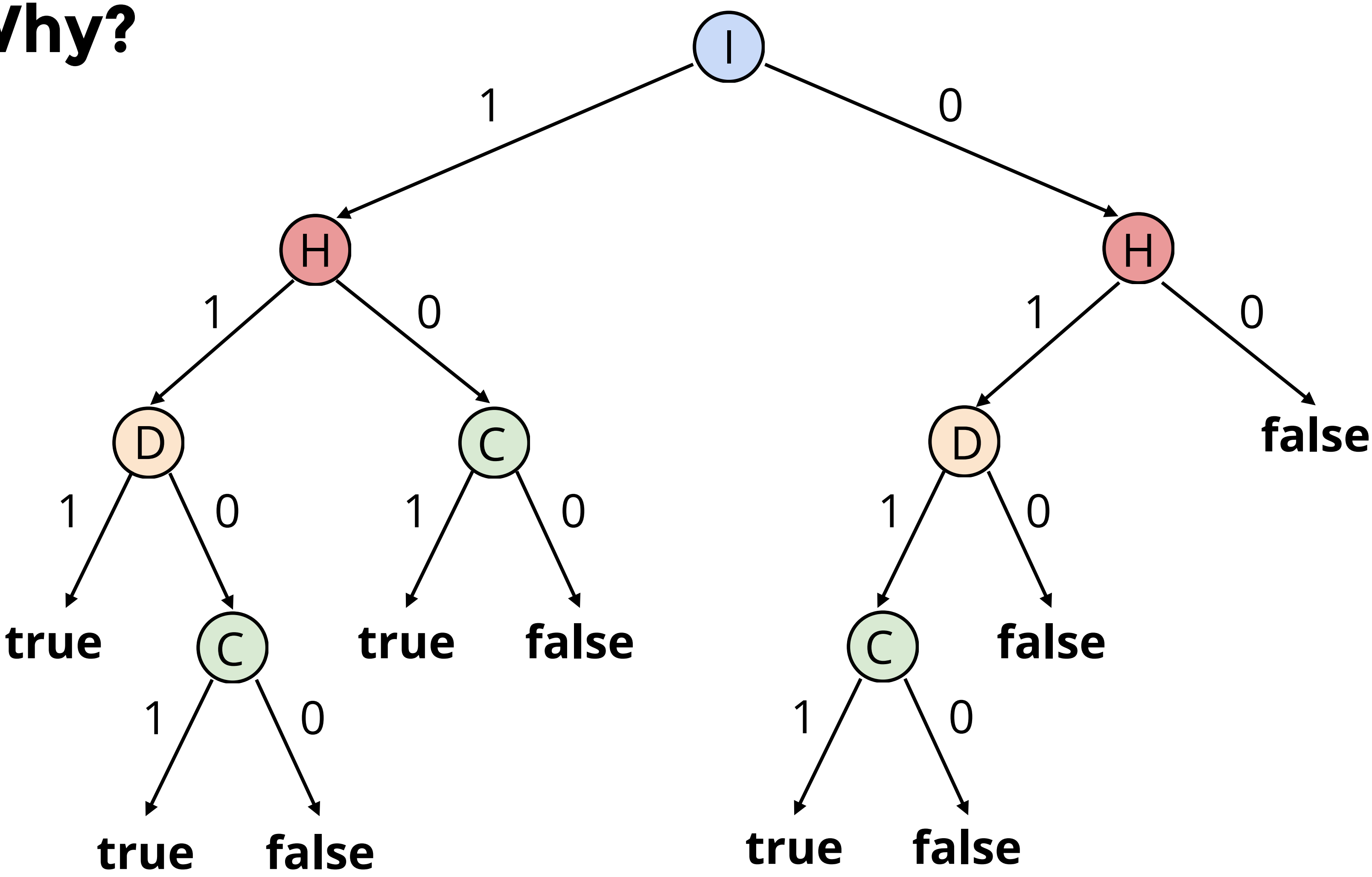


Why was the credit approved?

How can this decision be changed?

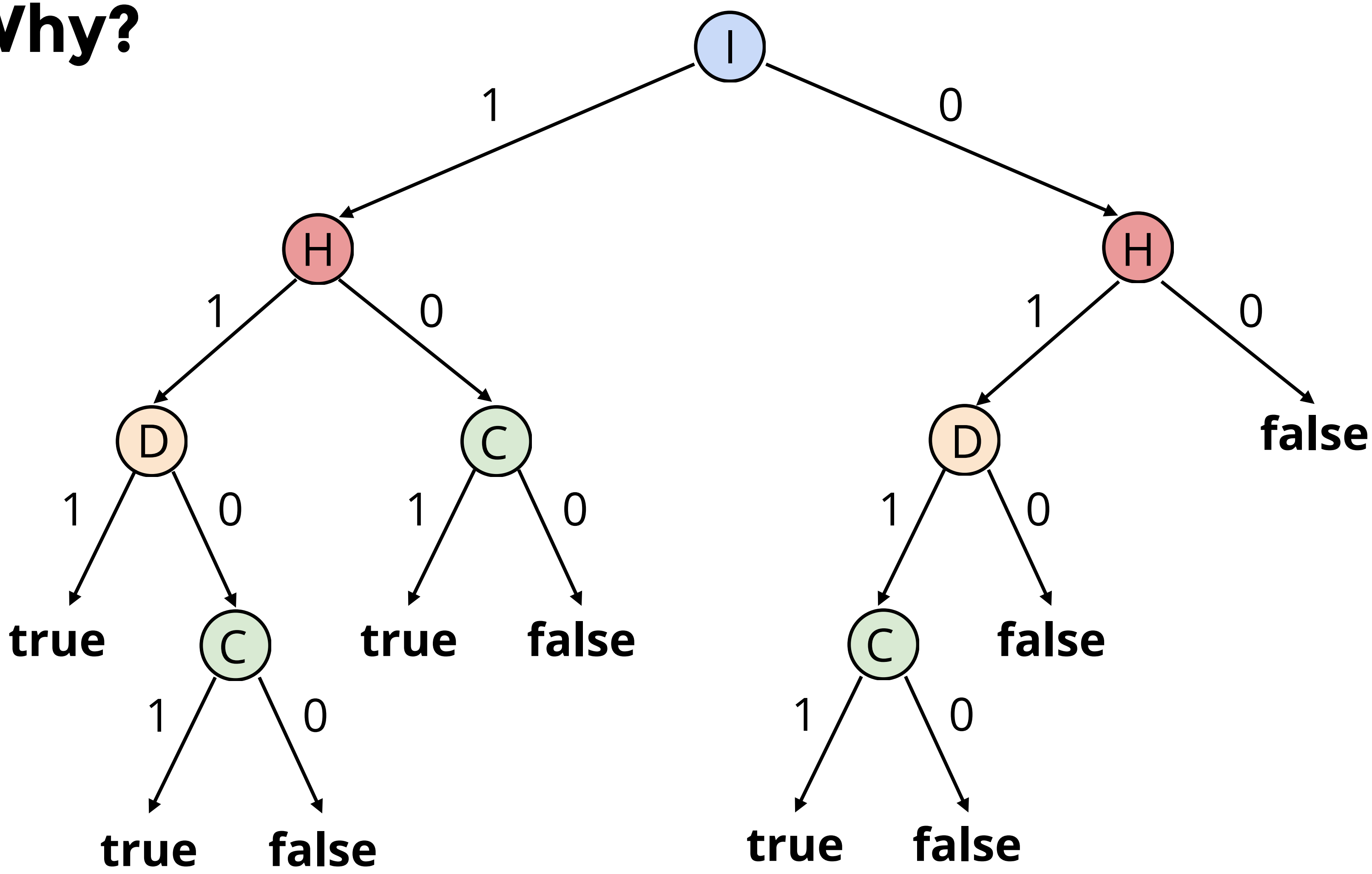
$I \rightarrow 1 \quad H \rightarrow 1 \quad D \rightarrow 1 \quad C \rightarrow 1$

Why?



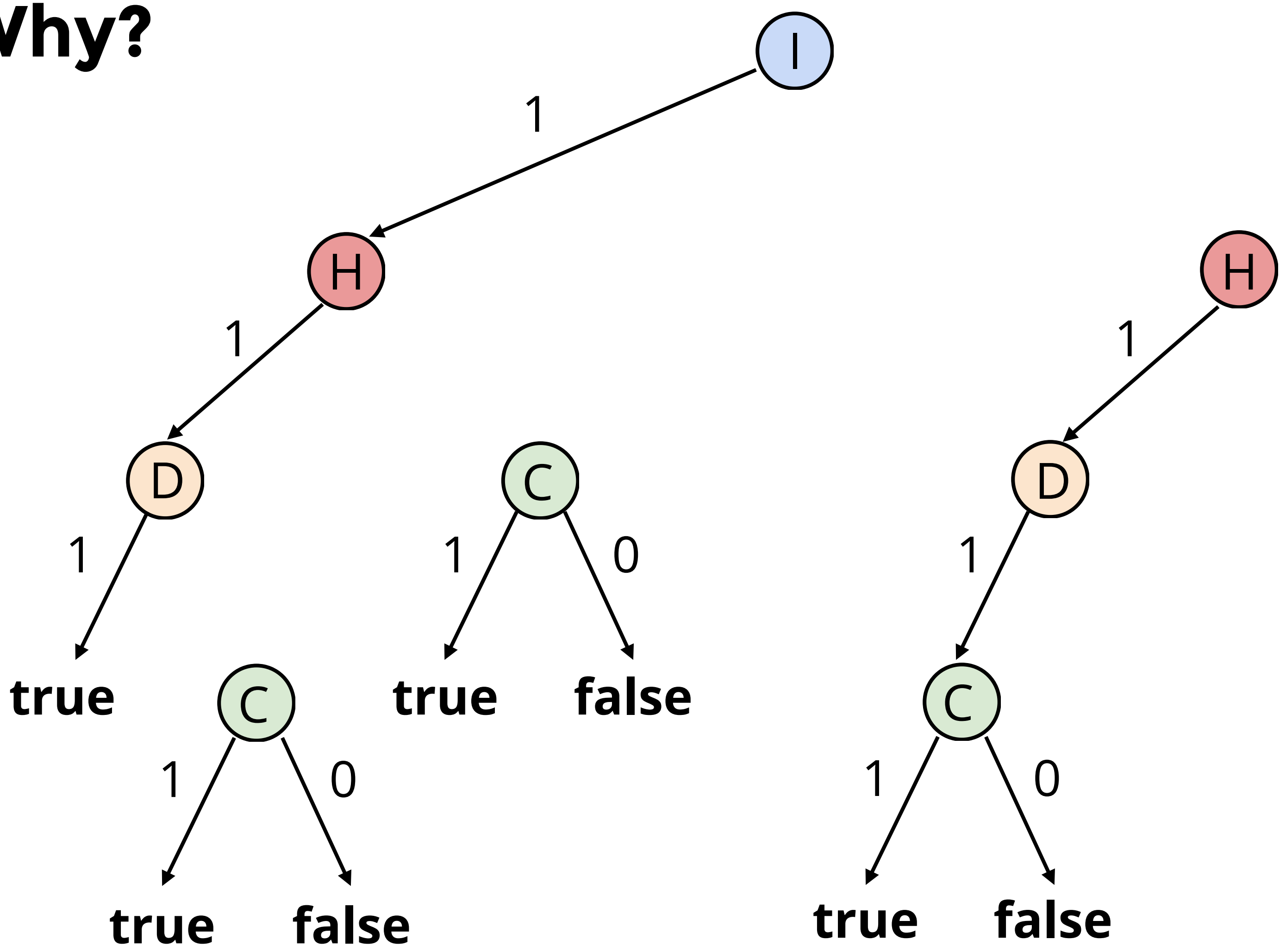
$I \rightarrow 1 \quad H \rightarrow 1 \quad D \rightarrow 1$

Why?



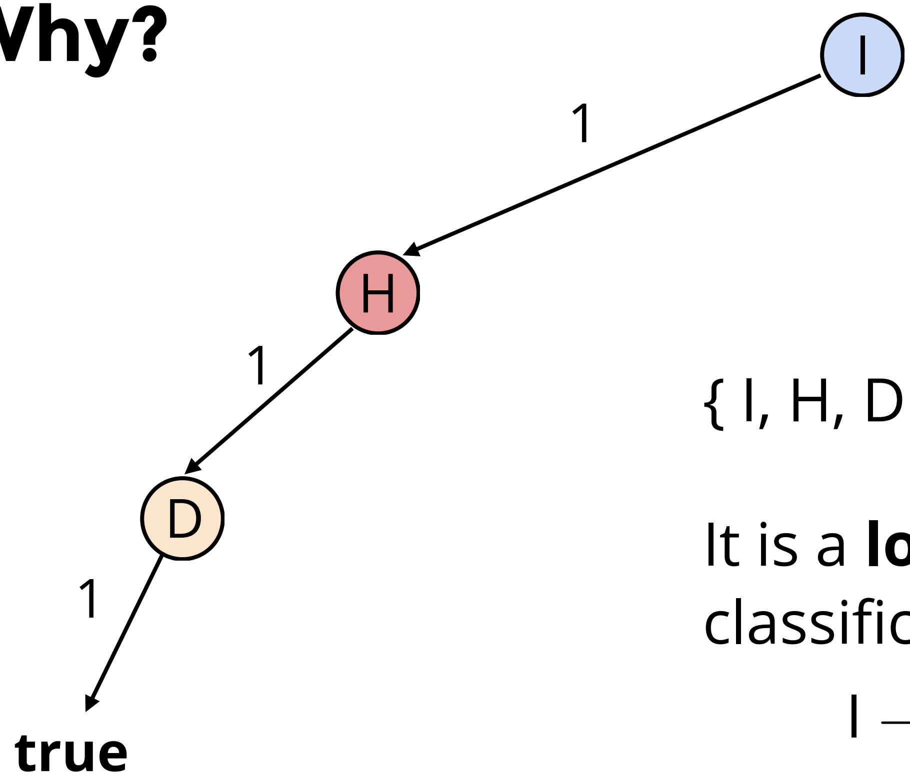
$$I \to 1 \quad H \to 1 \quad D \to 1$$

Why?



$I \rightarrow 1 \quad H \rightarrow 1 \quad D \rightarrow 1$

Why?



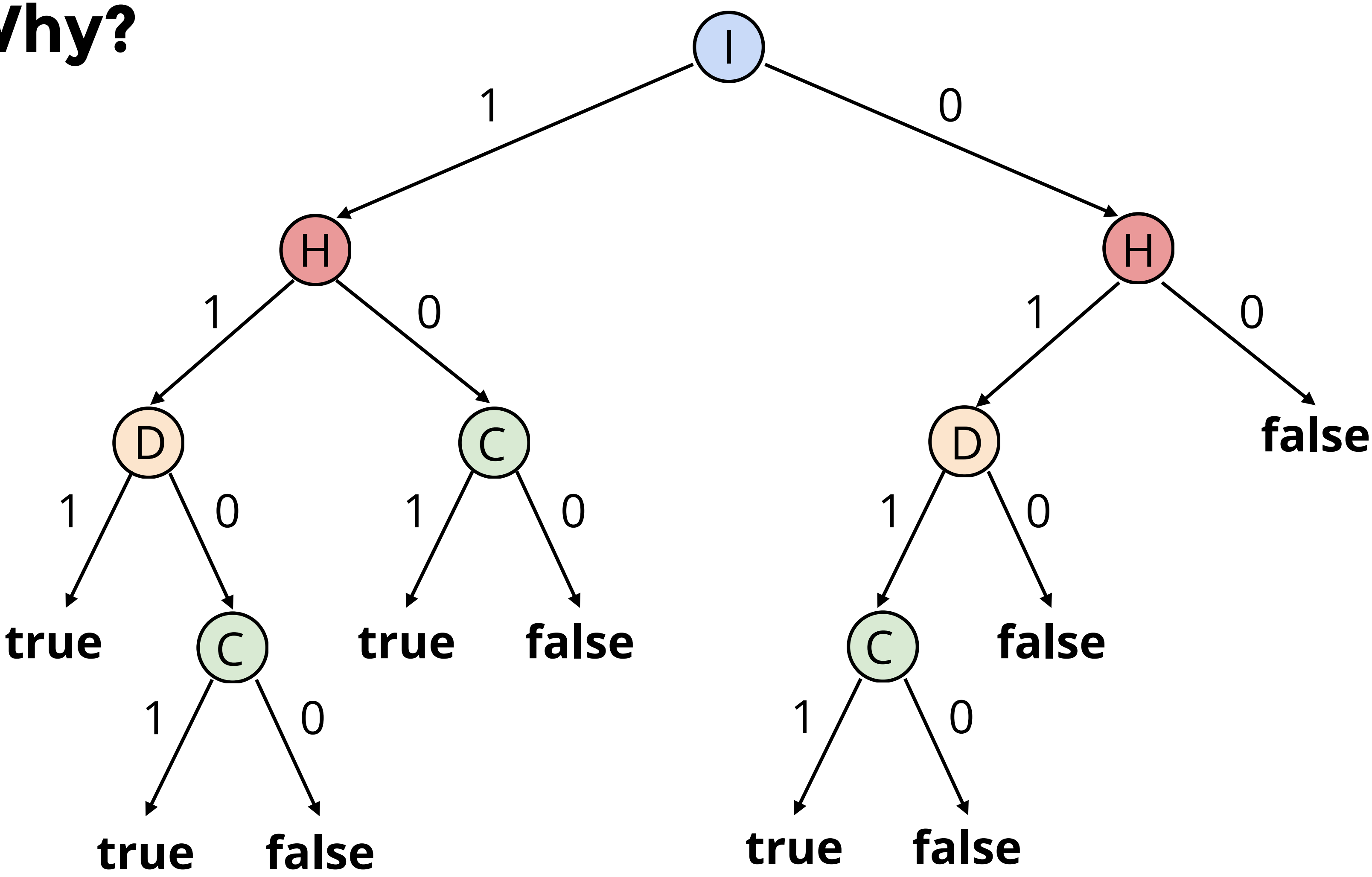
{ I, H, D } is an **abductive explanation**

It is a **local** explanation for the positive classification of the instance

$I \rightarrow 1 \quad H \rightarrow 1 \quad D \rightarrow 1 \quad C \rightarrow 1$

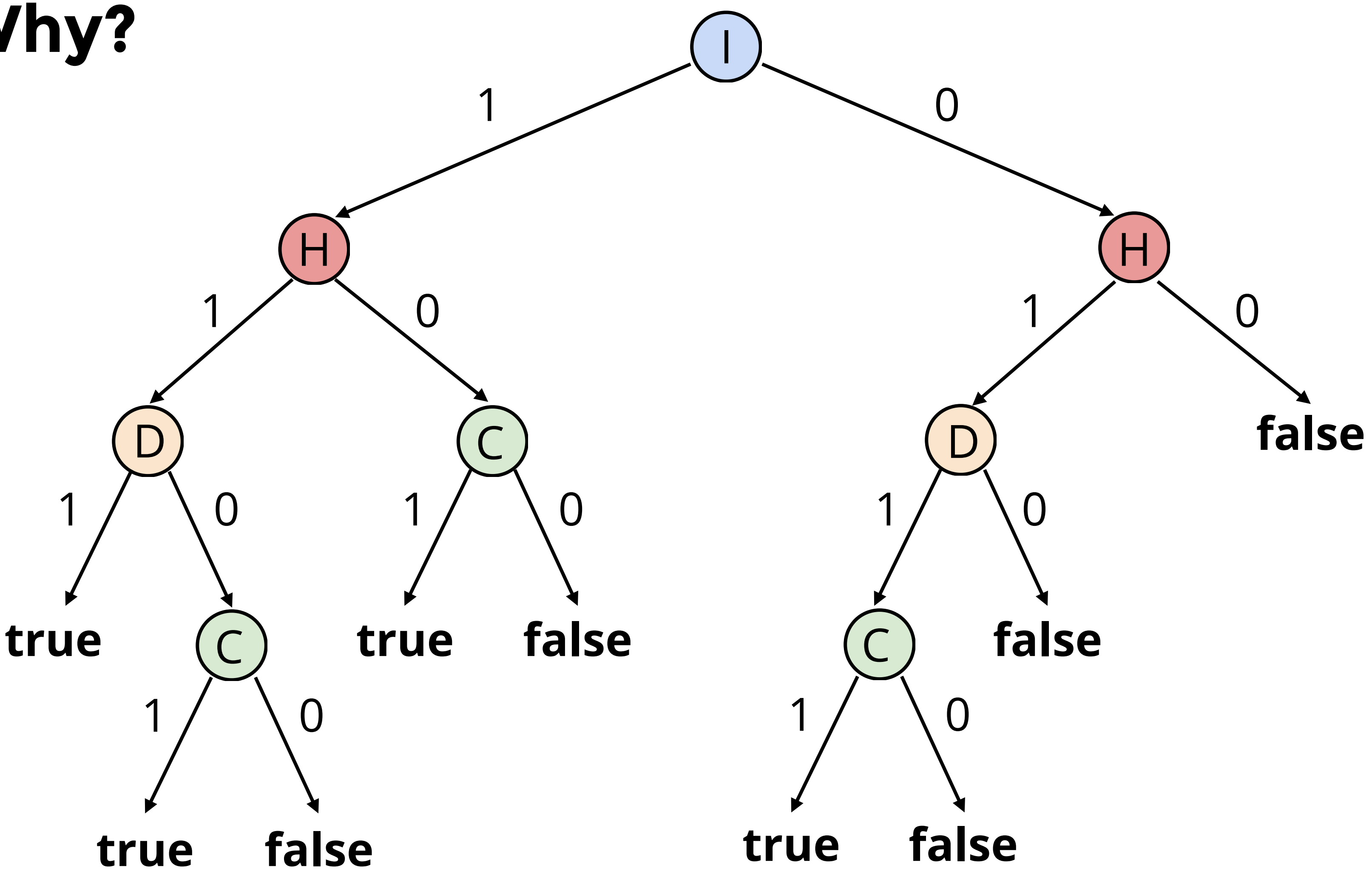
$$I \to 1 \quad H \to 1 \quad D \to 1 \quad C \to 1$$

Why?



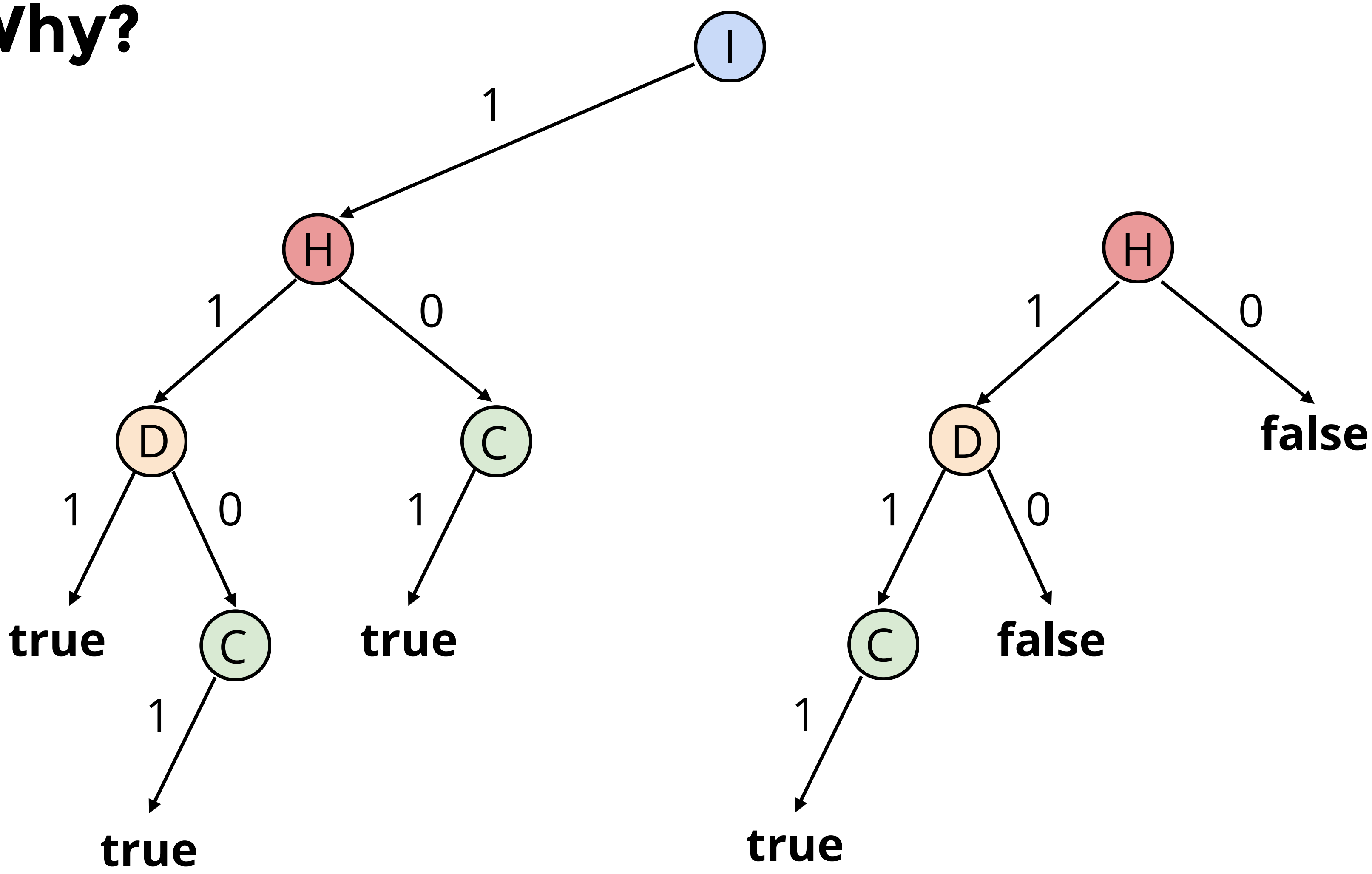
$I \to 1$ $\qquad$ $C \to 1$

Why?



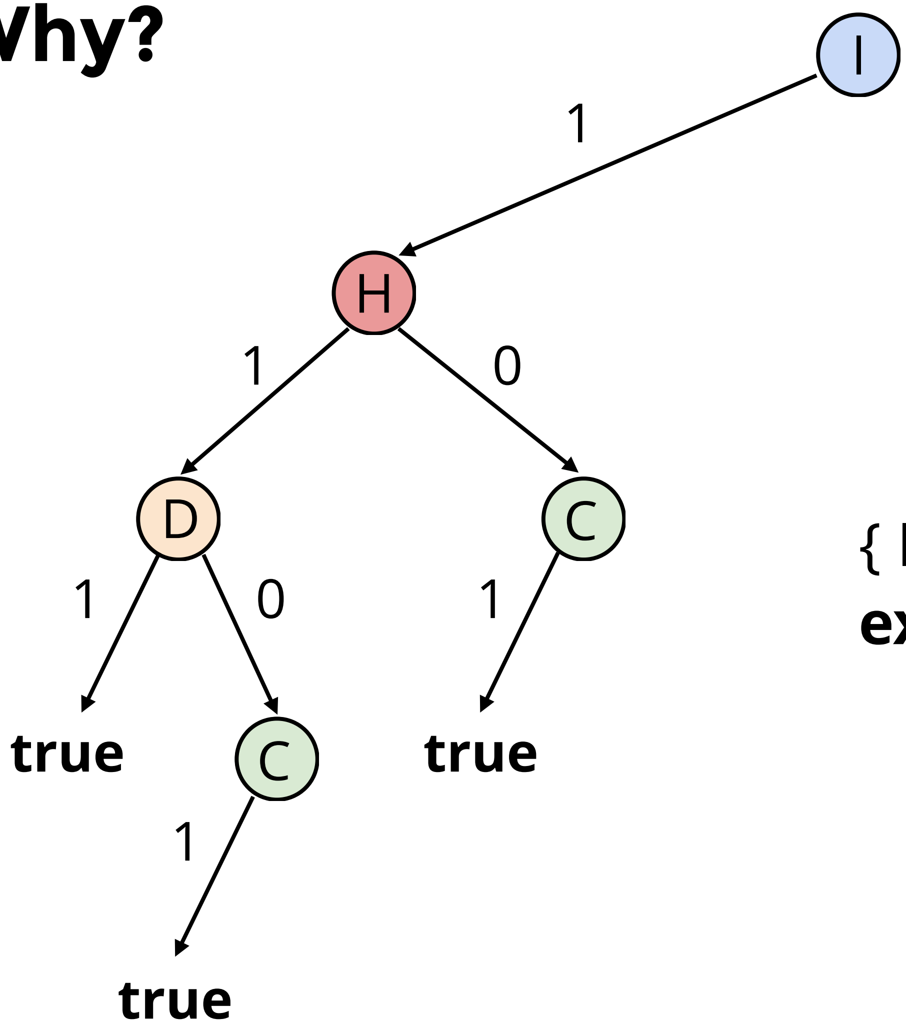
$I \to 1$ $C \to 1$

Why?



$I \rightarrow 1$ $C \rightarrow 1$

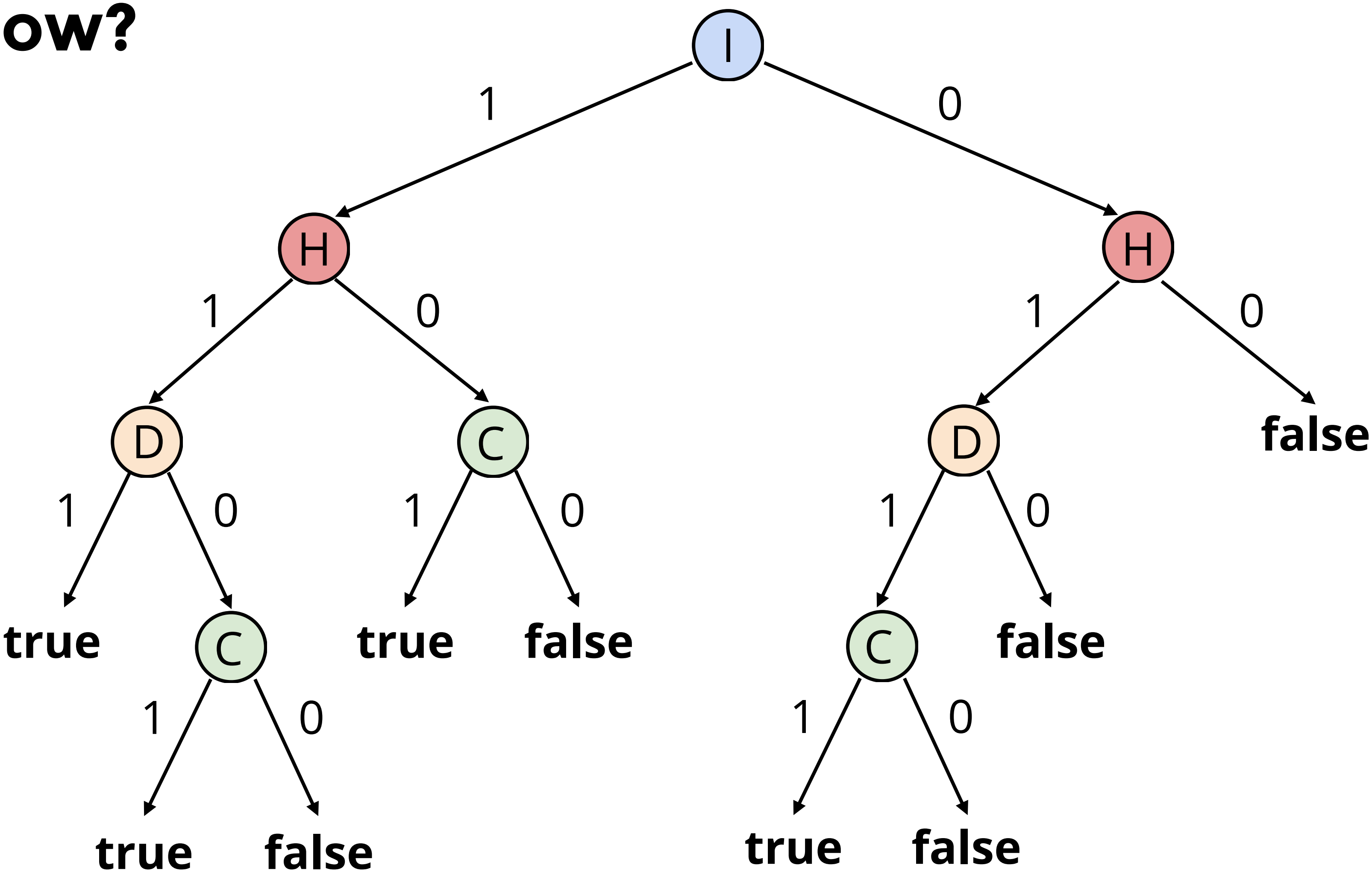
Why?



{ I, C } is also an **abductive explanation**

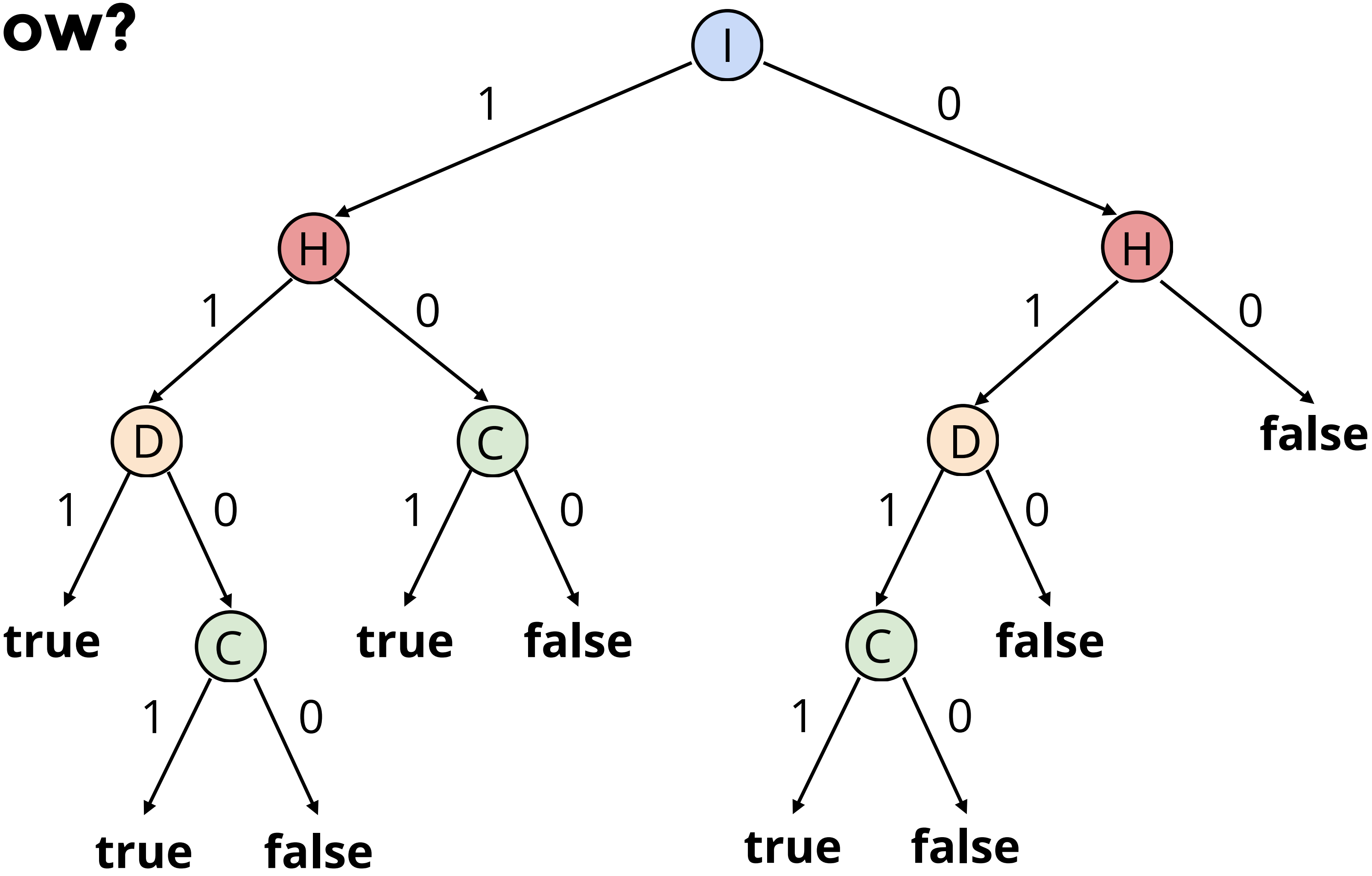
$$I \to 1 \quad H \to 1 \quad D \to 1 \quad C \to 1$$

How?



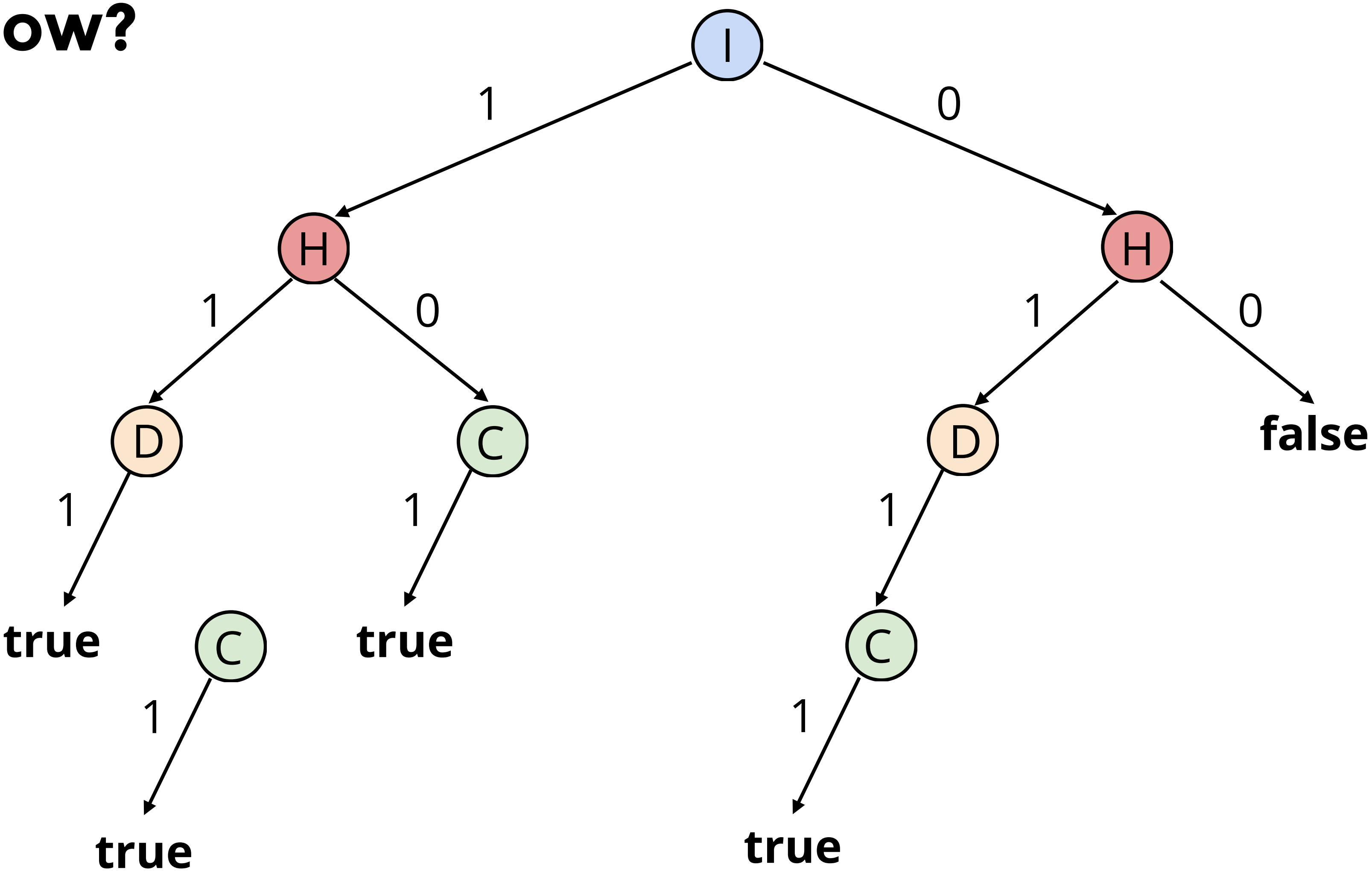
$D \to 1 \quad C \to 1$

How?



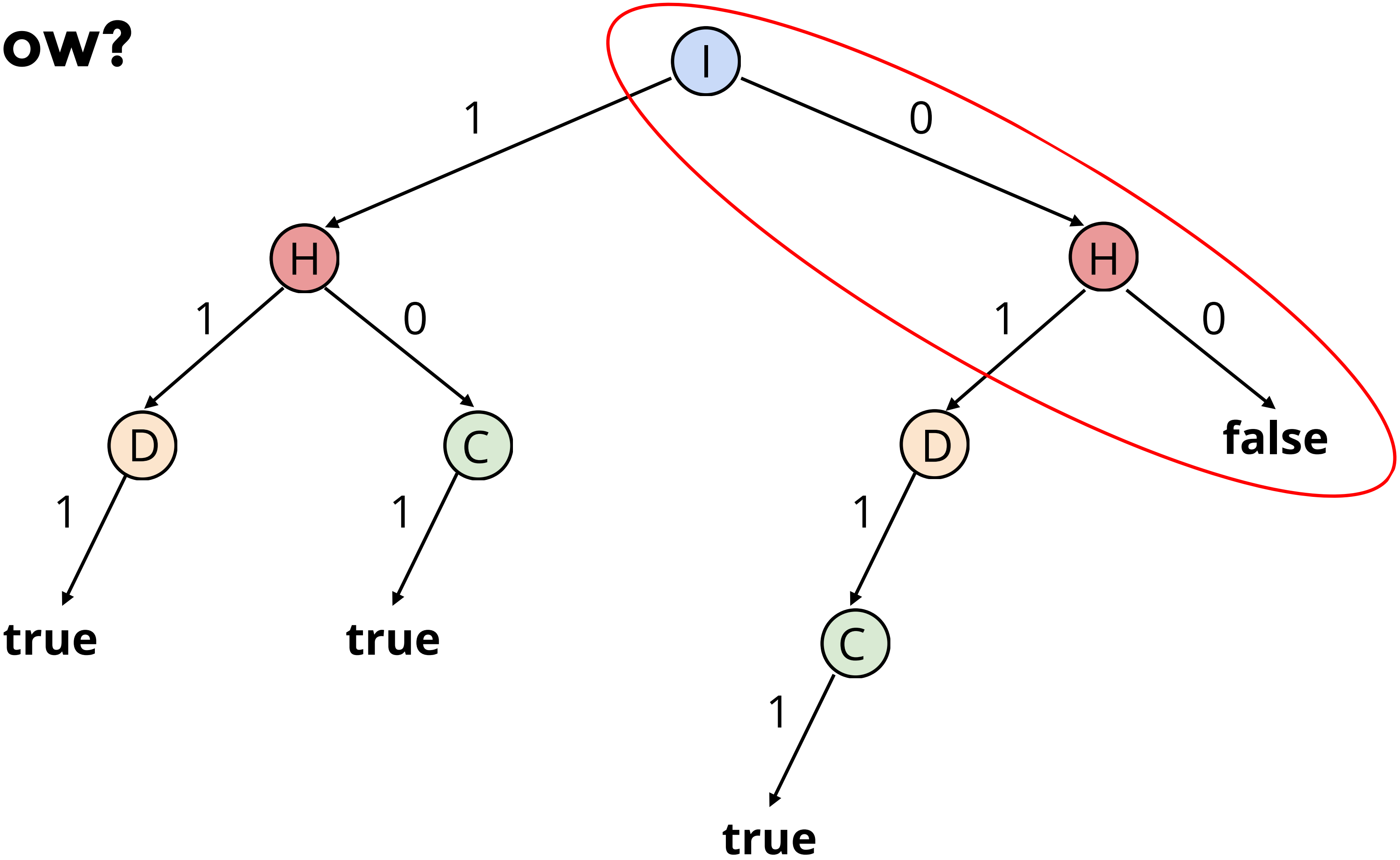
$D \rightarrow 1 \quad C \rightarrow 1$

How?



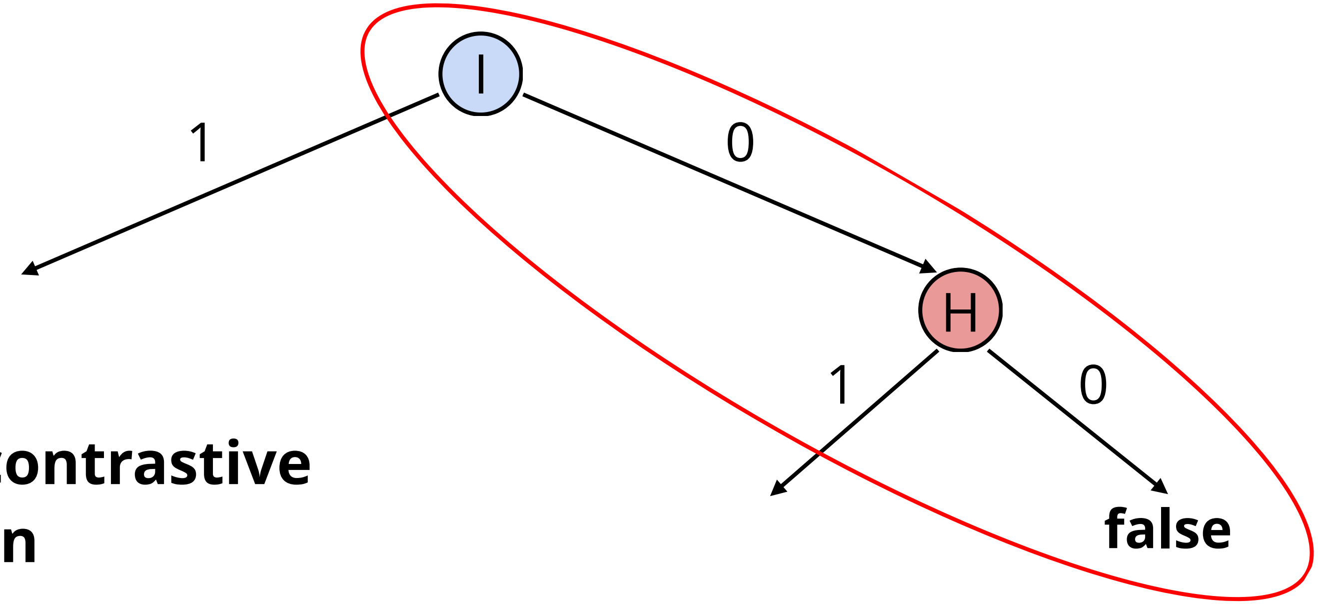
$D \to 1 \quad C \to 1$

How?



$D \to 1 \quad C \to 1$

How?

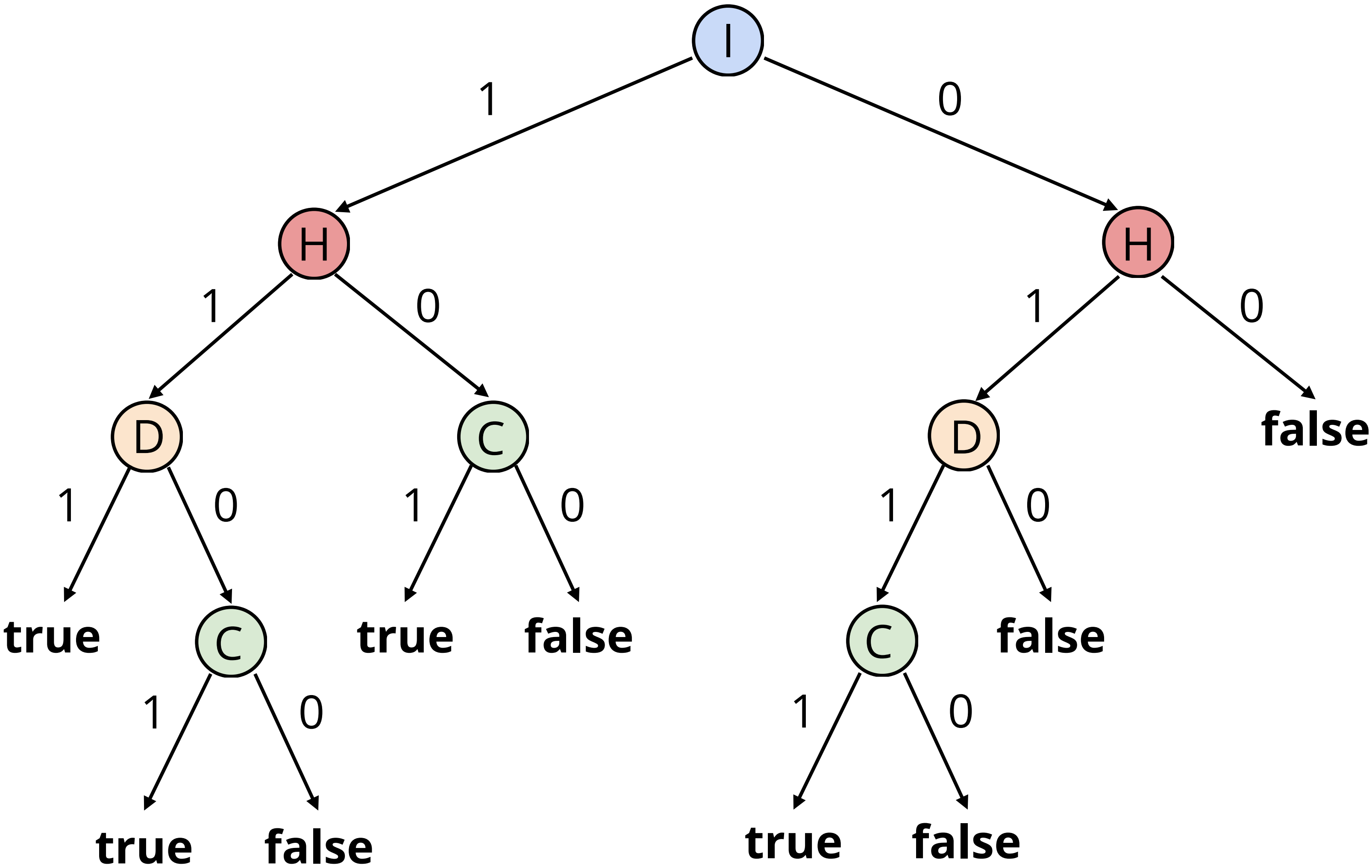


{ I, H } is a **contrastive explanation**

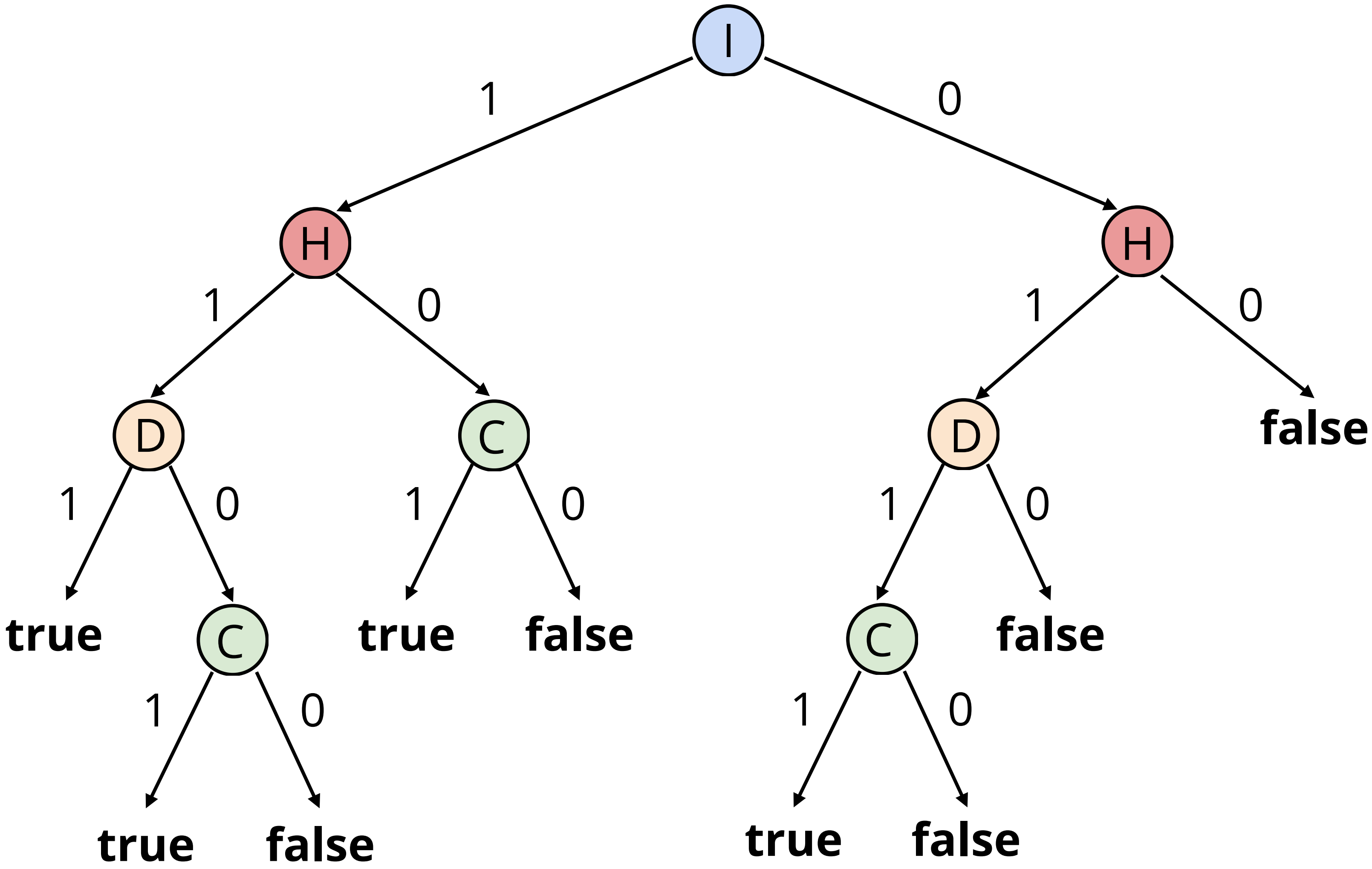
It is a local explanation for the positive classification of the instance

$I \to 1 \quad H \to 1 \quad D \to 1 \quad C \to 1$

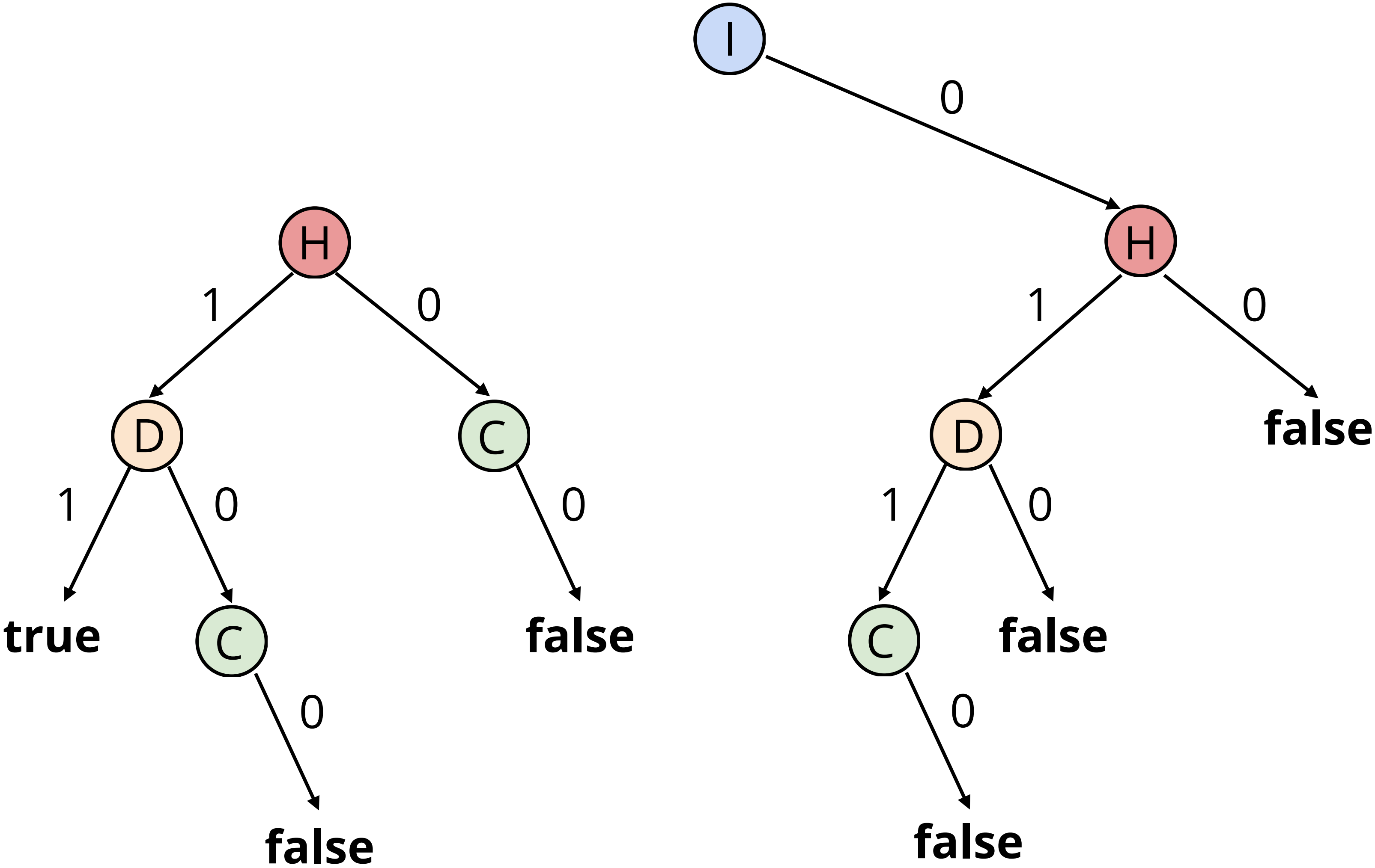
Global explanations



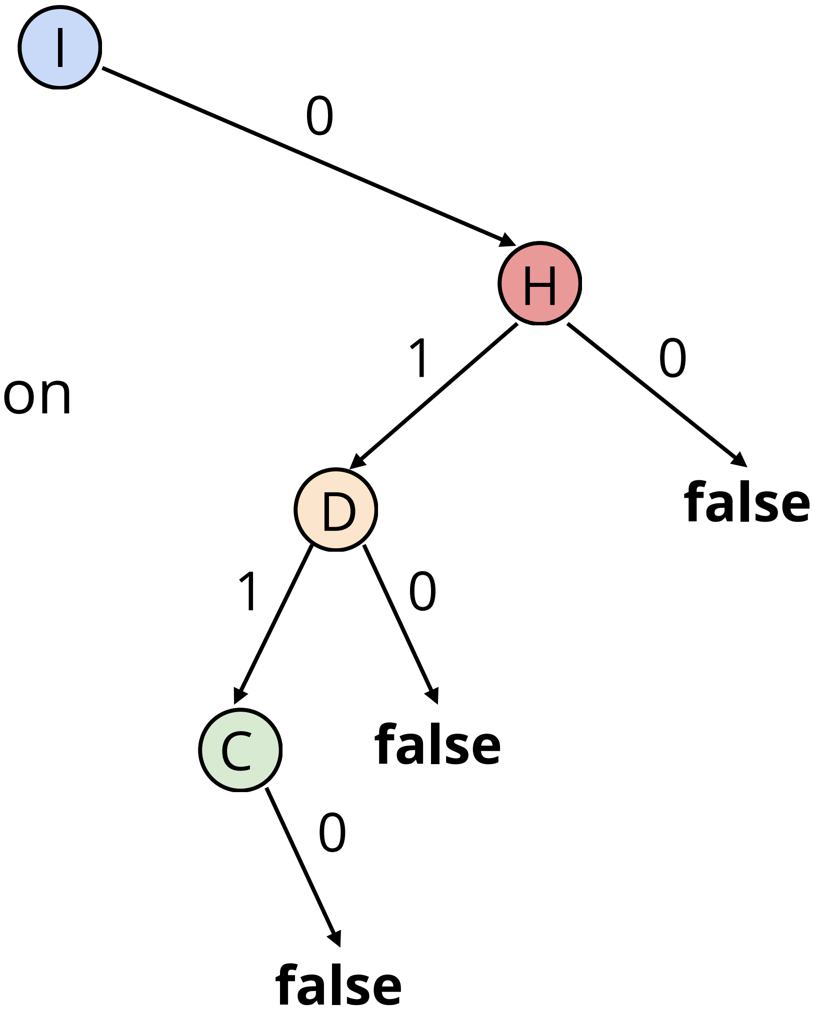
Global negative explanation : $I \rightarrow 0 \quad C \rightarrow 0$



Global negative explanation : $I \rightarrow 0 \quad C \rightarrow 0$



Global negative explanation : $I \to 0 \quad C \to 0$



{ $I \to 0$, $C \to 0$ } is a **global** abductive explanation for negative classifications

Formal explainability admits no silver bullet

¿How do we deal with an increasing number of explainability notions?

- Explainability may require combining different notions of explanation
- It is better to think of explainability as an interactive process
- One can develop a query language to express different explainability tasks
- This development would give control to the user to tailor explainability queries to their particular needs

More explainability queries

Common abductive explanation

Is there a common abductive explanation for the positive classification of

$$I \to 1 \quad H \to 1 \quad D \to 1 \quad C \to 1$$

and

$$I \to 1 \quad H \to 1 \quad D \to 0 \quad C \to 1\ ?$$

Yes, { I, C } is an answer to the query

Distinctive abductive explanation

Is there an abductive explanation for the positive classification of

$$I \to 1 \quad H \to 1 \quad D \to 1 \quad C \to 1$$

that is not an abductive explanation for the positive classification of

$$I \to 1 \quad H \to 1 \quad D \to 0 \quad C \to 1 \ ?$$

Yes, { I, H, D } is an answer to the query

Local necessity of a feature

Is there a feature assignment that is necessary for the positive classification of the input

$$I \to 1 \quad H \to 1 \quad D \to 0 \quad C \to 1 \ ?$$

Yes, $I \to 1$ is an answer to the query

Global necessity of a feature

Is there a feature assignment that is necessary to obtain a positive classification?

No, there is no such an assignment for a feature

Different orders

What is the abductive explanation with the smallest number of feature assignments for the positive classification of

$$\text{I} \to 1 \quad \text{H} \to 1 \quad \text{D} \to 1 \quad \text{C} \to 1 \text{ ?}$$

What is the global abductive explanation for positive classifications with the smallest number of feature assignments?

Different orders

What are the answers to all the previous queries if a feature is given preference over another feature?

A call for an explainability query language

Previous explainability queries clearly resemble traditional queries in databases

- In particular, some operators are needed to combine explainability queries

What are the desirable characteristics of an explainability query language?

Desiderata: simplicity

The language should be declarative, with a simple syntax and semantics

- Should be based on well-known database query languages

Desiderata: expressiveness

There should be a one-to-one correspondence between queries in the language and explanation notions

- In particular, an explanation notion should be represented by a fixed query

Desiderata: expressiveness

- Common explanation notions should be representable in the language
- Should allow the user to explore an explanation concept
- Should support the combination of different explanation approaches

Desiderata: efficiency

It should be possible to evaluate every query in the language efficiently

A desirable data complexity is P^{NP}

- Some ML models, such as decision trees, have a moderate size compared to a database
- Certain explanation tasks have an inherently high complexity
- This would enable the use of SAT solvers for query evaluation

Our goal is to develop an explainability query language that meets the previous criteria

The construction of the language

We need to specify:

- How an ML model is encoded in a database
- How an explainability task is encoded as a query over this database

A model agnostic approach

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

$\mathbf{e} \in \{0,1\}^n$ is an instance

$\mathcal{M}$ accepts $\mathbf{e}$ if $\mathcal{M}(\mathbf{e}) = 1$, otherwise $\mathcal{M}$ rejects $\mathbf{e}$

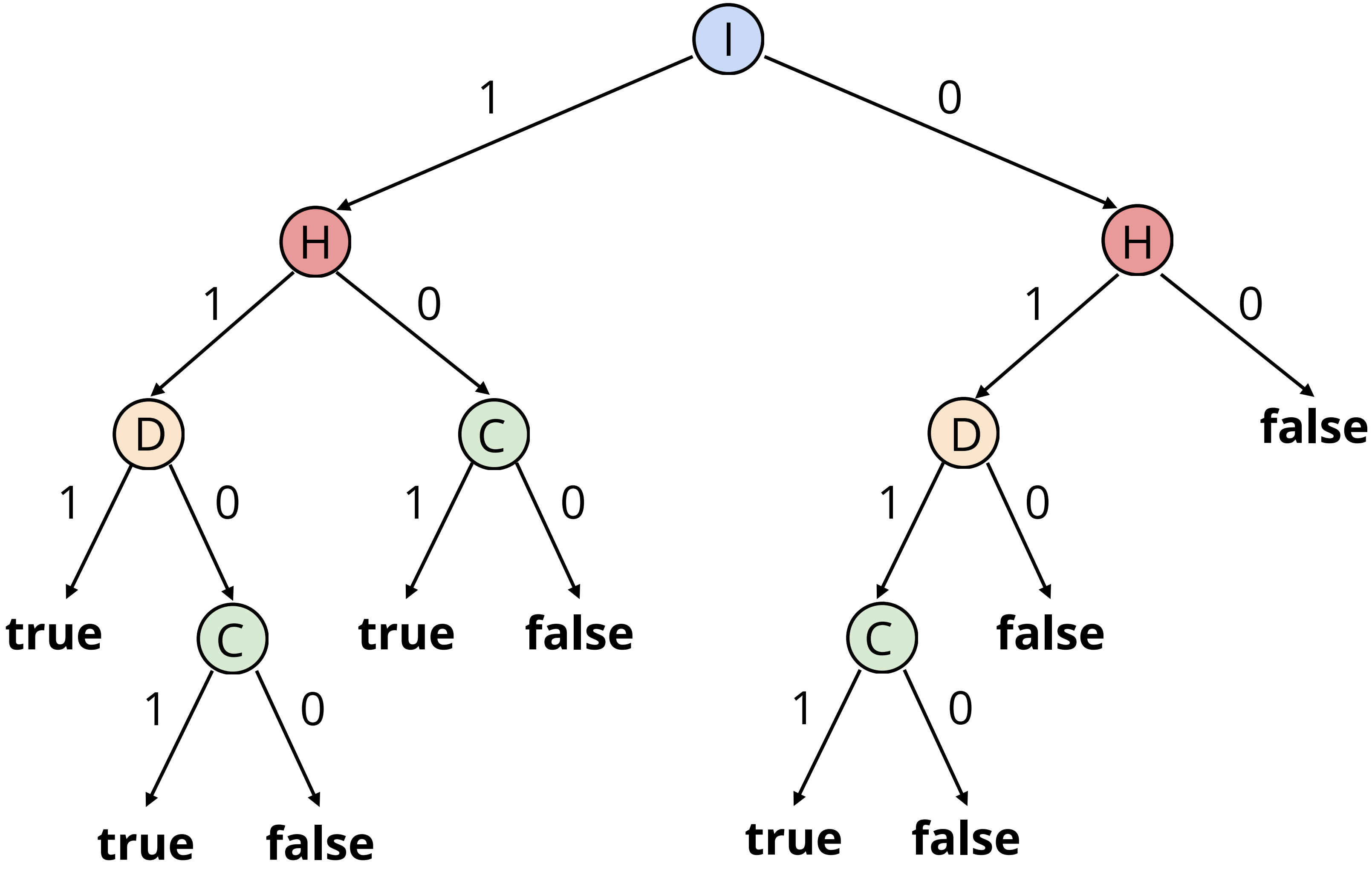
The base language: FOIL

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

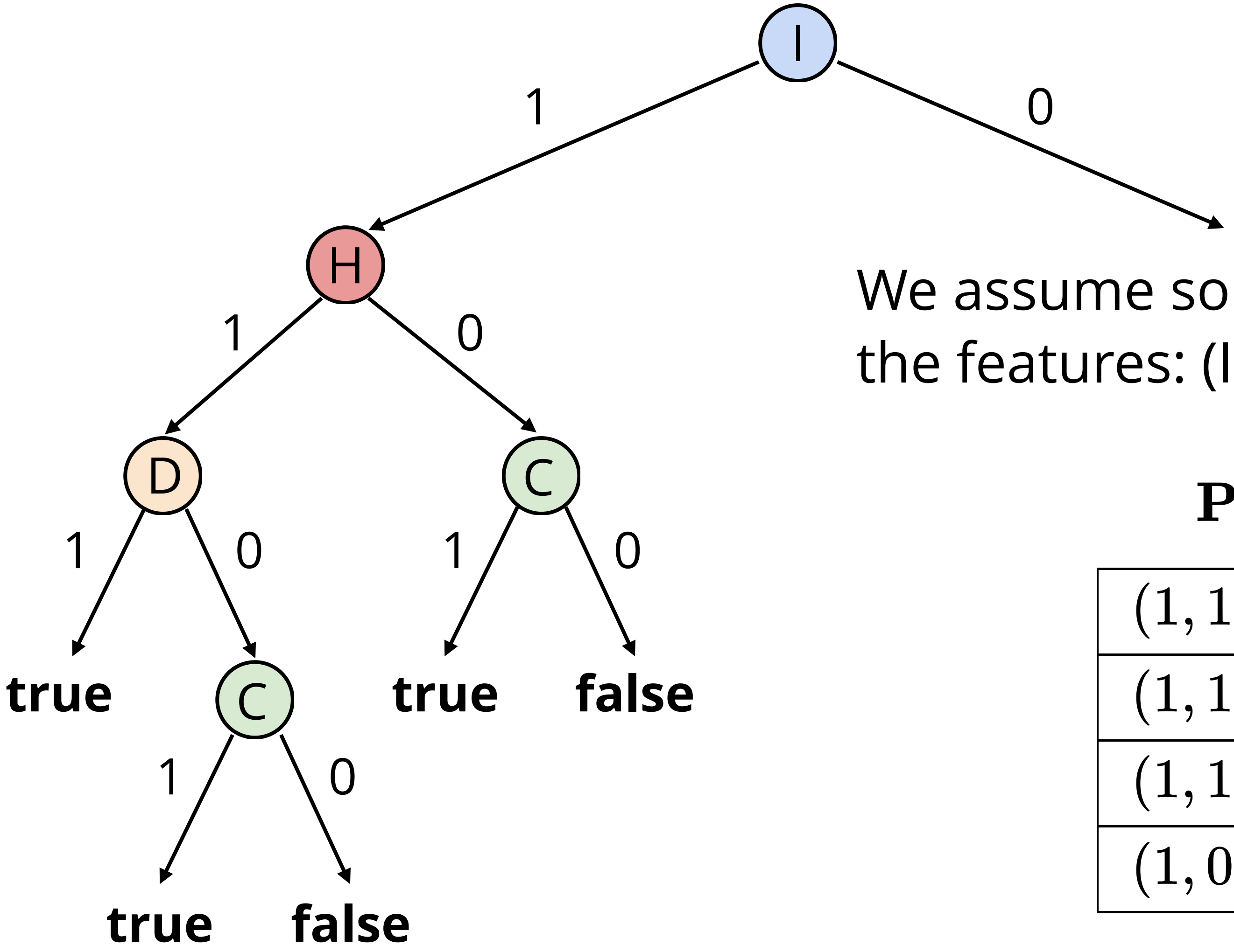
First ingredient: given a dimension n, the instances of dimension n are the objects stored in the database

A predicate $\mathbf{Pos}$ is used to store the instances that are classified positively by the ML model

The predicate **Pos**



The predicate **Pos**



We assume some order on the features: (I, H, D, C)

Pos

$(1, 1, 1, 1)$
$(1, 1, 1, 0)$
$(1, 1, 0, 1)$
$(1, 0, 0, 1)$

$\ldots$

The base language: FOIL

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

The partial instances $\mathbf{e} \in \{0, 1, \bot\}^n$ of dimension n are also objects in the database

- $\bot$ represents an unknown value

The base language: FOIL

$\mathbf{Pos}(\mathbf{e})$ is false if $\mathbf{e}$ contains an unknown value

Second ingredient: an order on partial instances based on the notion of being *more informative*

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

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

The predicate $\subseteq$

It can also be considered as a built-in predicate

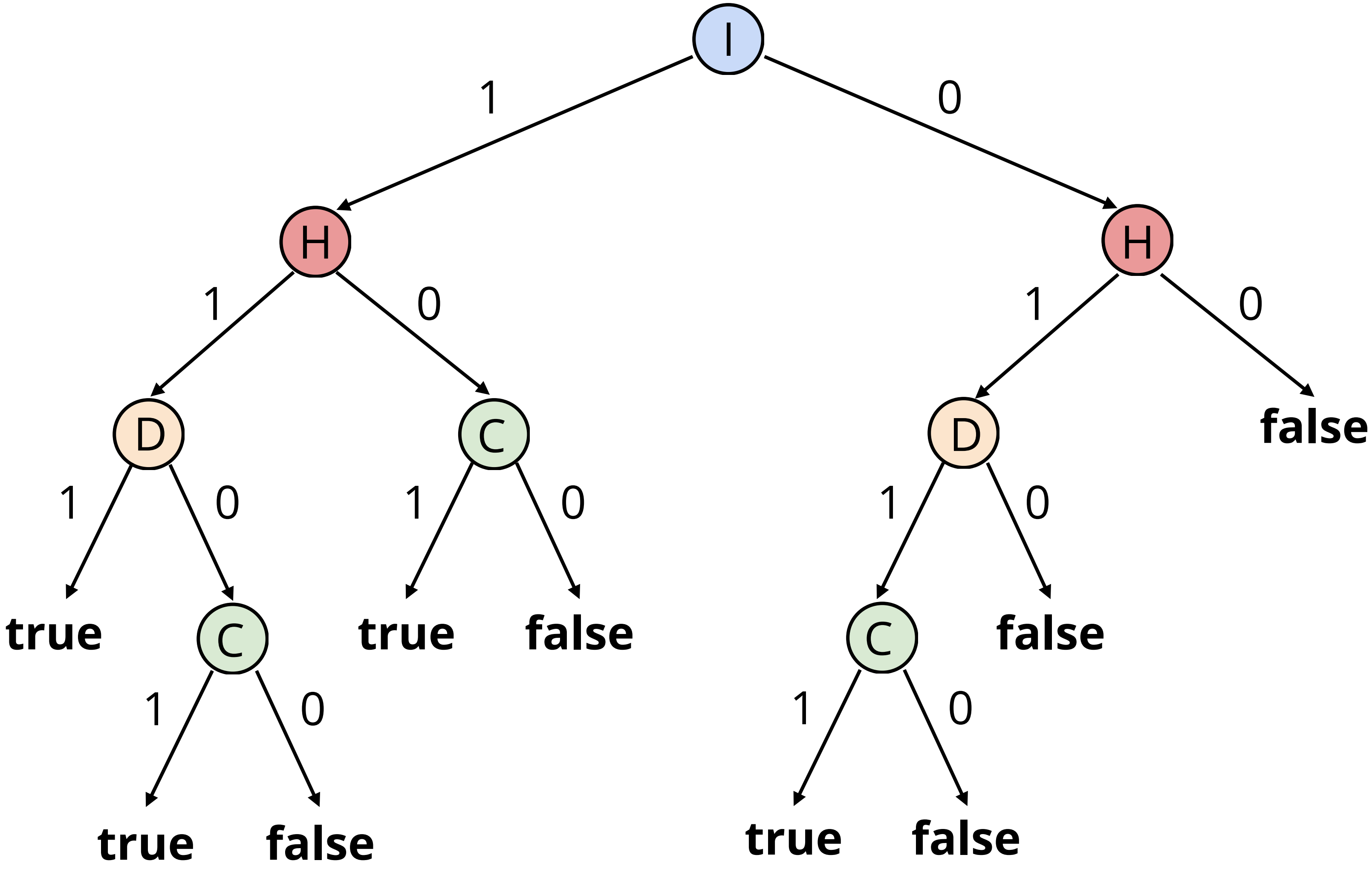
$\subseteq$

$(1,1,1,1)$	$(1,1,1,1)$
$(\bot,1,1,1)$	$(1,1,1,1)$
$(1,\bot,1,1)$	$(1,1,1,1)$

$\cdots$

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

The encoding of a model



The encoding of a model

Model $\mathcal{M}$ is represented as relational database $D_{\mathcal{M}}$:

$\mathbf{Pos}$

$(1,1,1,1)$
$(1,1,1,0)$
$(1,1,0,1)$
$(1,0,0,1)$

$\dots$

$\subseteq$

$(1,1,1,1)$	$(1,1,1,1)$
$(\bot,1,1,1)$	$(1,1,1,1)$
$(1,\bot,1,1)$	$(1,1,1,1)$

$\dots$

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

The syntax of FOIL

First-order logic defined over the vocabulary $\{\mathrm{Pos}, \subseteq\}$

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 $\mathbf{e}_1$, $\mathbf{e}_2$, ..., $\mathbf{e}_k$

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

$$\Longleftrightarrow$$

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

(in the usual sense)

Some fundamental queries in FOIL

Useful formulas

$$x \subset y \;=\; x \subseteq y \wedge \neg\, y \subseteq x$$

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

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

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

Abductive explanations

Consider the order (I, H, D, C) on the features

$\mathcal{M}(\mathbf{e}) = 1$ for $\mathbf{e} = (1, 1, 1, 1)$, and $\mathbf{e}_1 = (1, \bot, \bot, 1)$ is an abductive explanation for this

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

$$\mathrm{AE}(x, y) \ = \ \alpha(x, y) \wedge \forall z (\alpha(x, z) \rightarrow \neg\, z \subset y)$$

Abductive explanations

Consider the order (I, H, D, C) on the features

$\mathcal{M}(\mathbf{e}) = 1$ for $\mathbf{e} = (1, 1, 1, 1)$, and $\mathbf{e}_1 = (1, \perp, \perp, 1)$ is an abductive explanation for this

$$\mathcal{M} \models \mathbf{AE}(\mathbf{e}, \mathbf{e}_1)$$

Contrastive explanations

$\mathcal{M}(\mathbf{e}) = 1$ for $\mathbf{e} = (1, 1, 1, 1)$, and $\mathbf{e}_2 = (\bot, \bot, 1, 1)$ is a contrastive explanation for this

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

$$\mathrm{CE}(x, y) \;=\; \beta(x, y) \wedge \forall z (\beta(x, z) \rightarrow \neg\, z \subset x)$$

Contrastive explanations

$\mathcal{M}(\mathbf{e}) = 1$ for $\mathbf{e} = (1, 1, 1, 1)$, and $\mathbf{e}_2 = (\perp, \perp, 1, 1)$ is a contrastive explanation for this

$$\mathcal{M} \models \mathrm{CE}(\mathbf{e}, \mathbf{e}_2)$$

Common abductive explanation

Is there a common abductive explanation for the positive classification of instances $\mathbf{e} = (1, 1, 1, 1)$ and $\mathbf{e}' = (1, 1, 0, 1)$?

$$\mathrm{CAE}(x_1, x_2, y) \; = \; \mathrm{AE}(x_1, y) \wedge \mathrm{AE}(x_2, y)$$

Distinctive abductive explanation

Is there an abductive explanation for the positive classification of the instance $\mathbf{e} = (1, 1, 1, 1)$ that is not an abductive explanation for the positive classification of the instance $\mathbf{e}' = (1, 1, 0, 1)$?

$$\mathrm{DAE}(x_1, x_2, y) \ = \ \mathrm{AE}(x_1, y) \wedge \neg \mathrm{AE}(x_2, y)$$

Global necessity of a feature

Is there a feature assignment that is necessary to obtain a positive classification?

$$L_0(x) \; = \; \forall y \, (x \subseteq y) \longrightarrow (\bot, \bot, \bot, \bot)$$

Global necessity of a feature

Is there a feature assignment that is necessary to obtain a positive classification?

$$L_0(x) \ = \ \forall y \, (x \subseteq y)$$

$$L_1(x) \ = \ \exists y \, \big(L_0(y) \wedge y \subset x \wedge \neg \exists z \, (y \subset z \wedge z \subset x)\big)$$

Global necessity of a feature

Is there a feature assignment that is necessary to obtain a positive classification?

$$L_0(x) \;=\; \forall y\,(x \subseteq y)$$

$$L_1(x) \;=\; \exists y\,\big(L_0(y) \wedge y \subset x \wedge \neg\exists z\,(y \subset z \wedge z \subset x)\big)$$

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

Global necessity of a feature

Is there a feature assignment that is necessary to obtain a positive classification?

$$L_0(x) \;=\; \forall y\,(x \subseteq y)$$

$$L_1(x) \;=\; \exists y\,\big(L_0(y) \wedge y \subset x \wedge \neg\exists z\,(y \subset z \wedge z \subset x)\big)$$

$$\mathrm{GN}(x) \;= L_1(x) \wedge \forall y\,(\mathrm{Pos}(y) \rightarrow x \subseteq y)$$

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 assume $\Phi(x_1, \ldots, x_k)$ is a fixed **FOIL** formula

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

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

The evaluation problem for FOIL

How is $\mathcal{M}$ given as an input of $\mathrm{Eval}(\Phi)$?

We can assume that $\mathcal{M}$ is given as a Boolean circuit or a CNF propositional formula

In some cases, we can use simpler representations such as decision trees or OBDDs

The evaluation problem for FOIL

$\mathcal{M} \models \Phi(\mathbf{e}_1, \ldots, \mathbf{e}_k)$ if and ony if $D_{\mathcal{M}} \models \Phi(\mathbf{e}_1, \ldots, \mathbf{e}_k)$

But $D_{\mathcal{M}}$ is of exponential size in the dimension n

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

The evaluation problem for FOIL

Obviously, the complexity of $\mathrm{Eval}(\Phi)$ is high if models are given as Boolean circuits or CNF propositional formulas

- $\mathrm{Eval}(\Phi)$ is NP-complete for the FOIL formula $\Phi = \exists x \, \mathrm{Pos}(x)$
- More generally, for each level of the polynomial hierarchy, there exists an FOIL formula Φ such that $\mathrm{Eval}(\Phi)$ is hard for this level

The evaluation problem for FOIL

We focus on the case where classification models are decision trees, which are argued to be readily interpretable

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

The expressiveness of FOIL

What is the abductive explanation with the smallest number of feature assignments for the positive classification of $(1, 1, 1, 1)$?

Such an explanation is referred to as a minimum abductive explanation

The expressiveness of FOIL

Theorem:

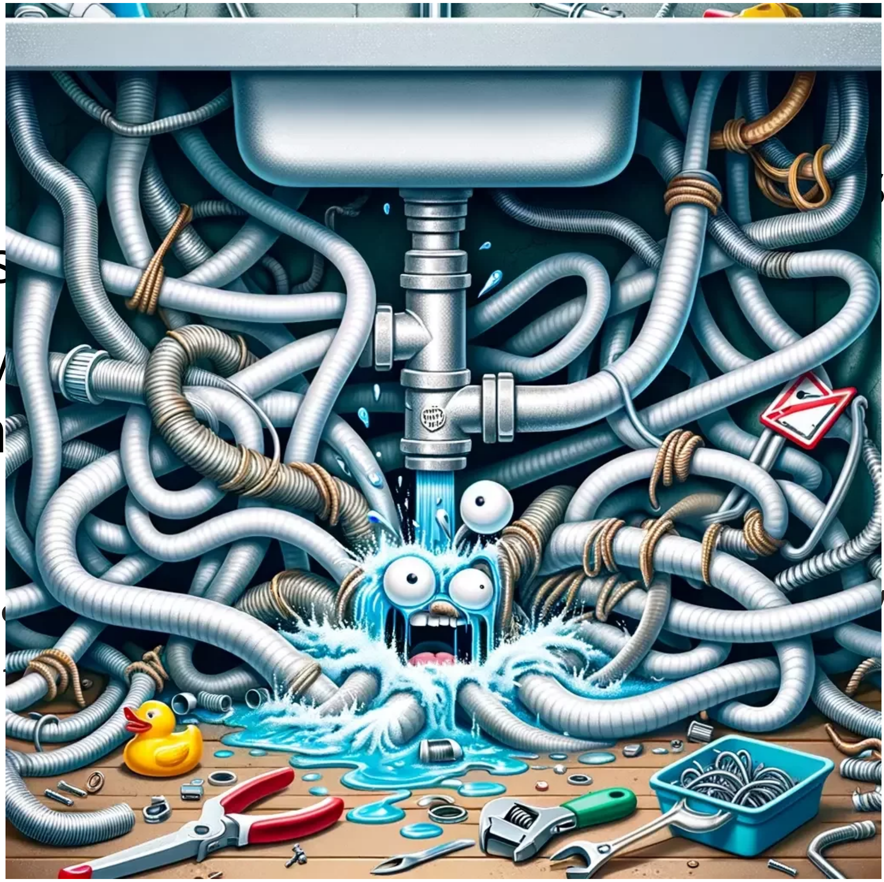
There is no **FOIL** formula $\mathrm{minAE}(x, y)$ such that, for every decision tree $\mathcal{T}$, instance $\mathbf{e}_1$ and partial instance $\mathbf{e}_2$:

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

$$\iff$$

$\mathbf{e}_2$ is a minimum abductive explanation for $\mathbf{e}_1$ over $\mathcal{T}$

How can these limitations be overcome?



- Extend … sing notions …
- Identify … evaluated efficien…
- Depart … ach to obtain … aluated more e…

Very briefly …

We present some general strategies that can be implemented in different ways to:

- Extend **FOIL** vocabulary to express missing notions of explanation
- Depart from the model-agnostic approach to obtain a query language that can be evaluated more efficiently

we follow a principled approach

Extending the vocabulary

We need a predicate to encode orders based on cardinalities

Given partial instances $\mathbf{e}_1, \mathbf{e}_2$ of dimension n:

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

$$\iff$$

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

Extending the vocabulary

$$x \prec y \;=\; x \preceq y \wedge \neg\, y \preceq x$$

$$\alpha(x,y) \;=\; \mathrm{Full}(x) \wedge y \subseteq x \wedge (\mathrm{Pos}(x) \rightarrow \mathrm{AllPos}(y)) \wedge (\neg \mathrm{Pos}(x) \rightarrow \mathrm{AllNeg}(y))$$

$$\mathrm{minAE}(x,y) \;=\; \alpha(x,y) \wedge \forall z(\alpha(x,z) \rightarrow \neg\, z \prec y)$$

But how many more predicates do we need to include?

An extended notion of atomic formula

All the *order* 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: the set of first-order formulas defined over $\{\subseteq, \preceq\}$

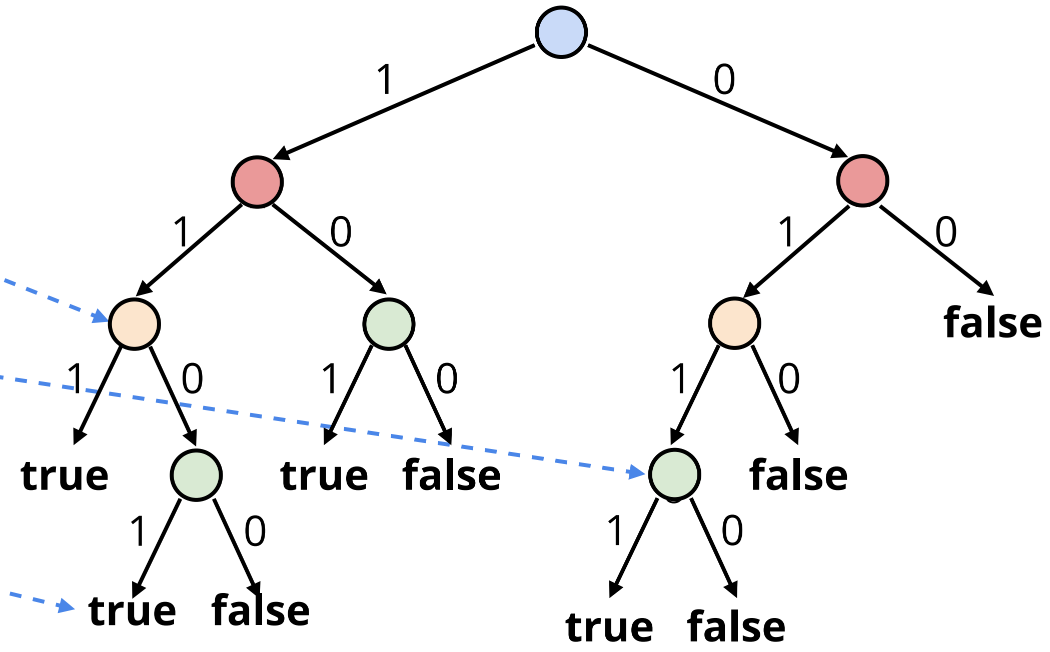
Departing from the model-agnostic approach

$\text{Node}(\mathbf{e})$:

$(1, 1, \bot, \bot)$

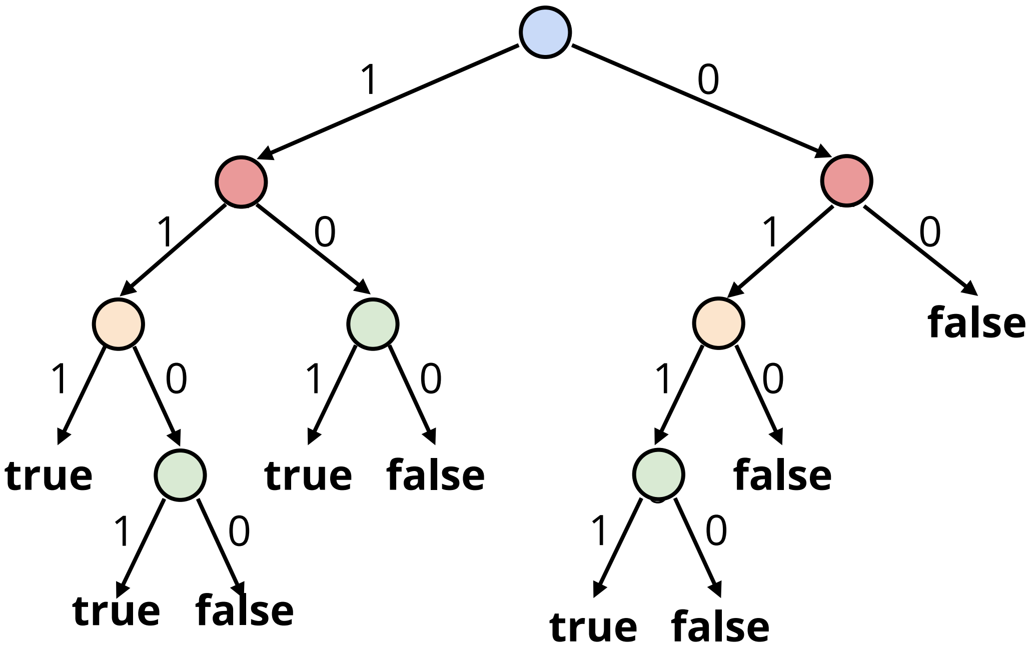
$(0, 1, 1, \bot)$

$(1, 1, 0, 1)$



Departing from the model-agnostic approach

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

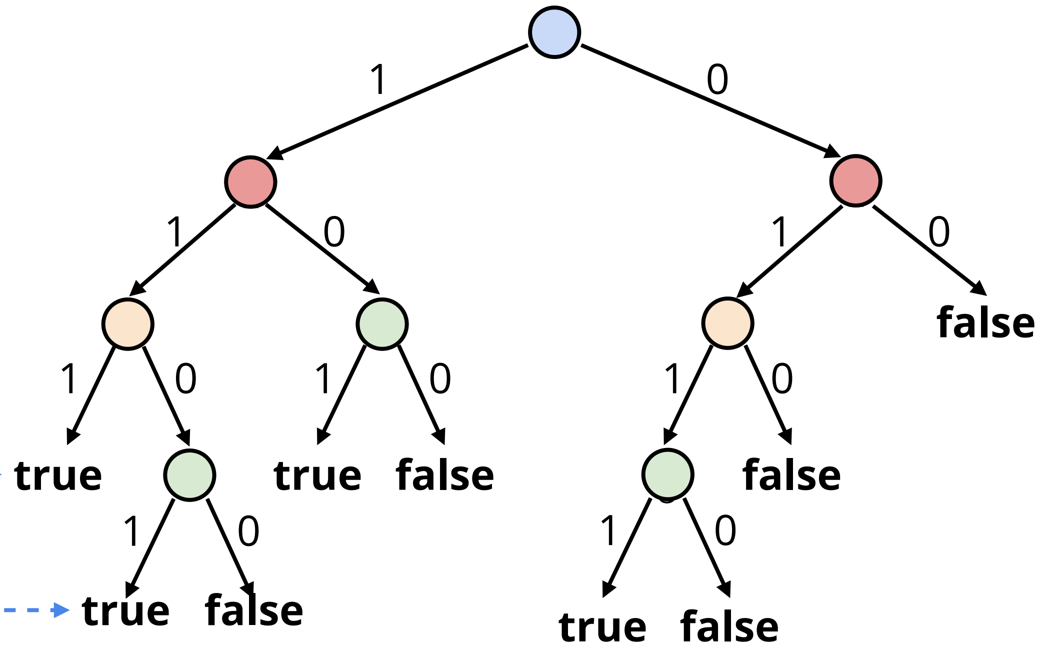


Departing from the model-agnostic approach

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

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

$(1, 1, 0, 1)$



The encoding of a model

$\mathbf{Pos}$

$(1,1,1,1)$
$(1,1,1,0)$
$(1,1,0,1)$
$(1,0,0,1)$

$\ldots$

$\subseteq$

$(1,1,1,1)$	$(1,1,1,1)$
$(\bot,1,1,1)$	$(1,1,1,1)$

$\ldots$

The encoding of a model

$\mathbf{Pos}$

$(1,1,1,1)$
$(1,1,1,0)$
$(1,1,0,1)$
$(1,0,0,1)$

$\dots$

$\subseteq$

$(1,1,1,1)$	$(1,1,1,1)$
$(\bot,1,1,1)$	$(1,1,1,1)$

$\dots$

$\preceq$

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

$\dots$

The encoding of a model

Node

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

$\ldots$

PosLeaf

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

$\ldots$

$\subseteq$

$(1, 1, 1, 1)$	$(1, 1, 1, 1)$
$(\bot, 1, 1, 1)$	$(1, 1, 1, 1)$

$\ldots$

$\preceq$

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

$\ldots$

Guarded quantification for decision trees

An efficient form of quantification is obtained by considering the notion of *guard*:

$$\exists x\,(\mathrm{Node}(x) \wedge \Phi) \qquad \forall x\,(\mathrm{Node}(x) \to \Phi)$$

$$\exists x\,(\mathrm{PosLeaf}(x) \wedge \Phi) \qquad \forall x\,(\mathrm{PosLeaf}(x) \to \Phi)$$

This a general form of quantification that can be used in other representations of ML models

Guarded quantification for decision trees

$$\alpha(x, y) \;=\; \mathrm{Full}(x) \wedge y \subseteq x \wedge (\mathrm{Pos}(x) \rightarrow \mathrm{AllPos}(y)) \wedge (\neg\mathrm{Pos}(x) \rightarrow \mathrm{AllNeg}(y))$$

define them
using guarded
quantification

Guarded quantification for decision trees

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

Guarded quantification for decision trees

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

atomic formula

Guarded quantification for decision trees

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

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

Guarded quantification for decision trees

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

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

guarded quantification

Guarded quantification for decision trees

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

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

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

guarded quantification

Guarded quantification for decision trees

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

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

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

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

Guarded quantification for decision trees

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

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

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

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

guarded quantification

Is this enough?

The combination of these strategies allow to construct query languages with the desired properties

- Including orders with feature preferences

These languages allow either:

- Some restricted form of non-guarded quantification
- An explicit minimal operator

Concluding remarks

This effort is a first step towards the definition of an explainability query language

- Definition of the basic explainability query language **FOIL**
- A principled approach that defines basic components, which can be combined to develop a robust query language

Concluding remarks

Much work remains to be done

- **Developing a user-friendly query language**
- Developing efficient query answering algorithms, and optimization techniques
- **Incorporating probabilities**
- ...

Concluding remarks

What constitutes a user-friendly explainability query language?

- How should an explanation be presented to the user?
- What is the right level of detail that has to be provided to different users? How can this level of detail be specified?
- How can it be proven that such an explanation is trustworthy?

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

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

Thanks!