

# Decisión Binaria

Pedro Cabalar

Lógica

Grado en Inteligencia Artificial  
Universidade da Coruña

May 2, 2023

# 1 Árboles de Decisión Binaria

# Decisión Binaria

- Problema de **decisión binaria** = elección entre dos alternativas
- Ejemplo: condicional `if x>10 then a else b`. En Python:

```
if x>0:  
    a  
else:  
    b
```

# Decisión Binaria

- Problema de **decisión binaria** = elección entre dos alternativas
- Ejemplo: condicional `if x>10 then a else b`. En Python:

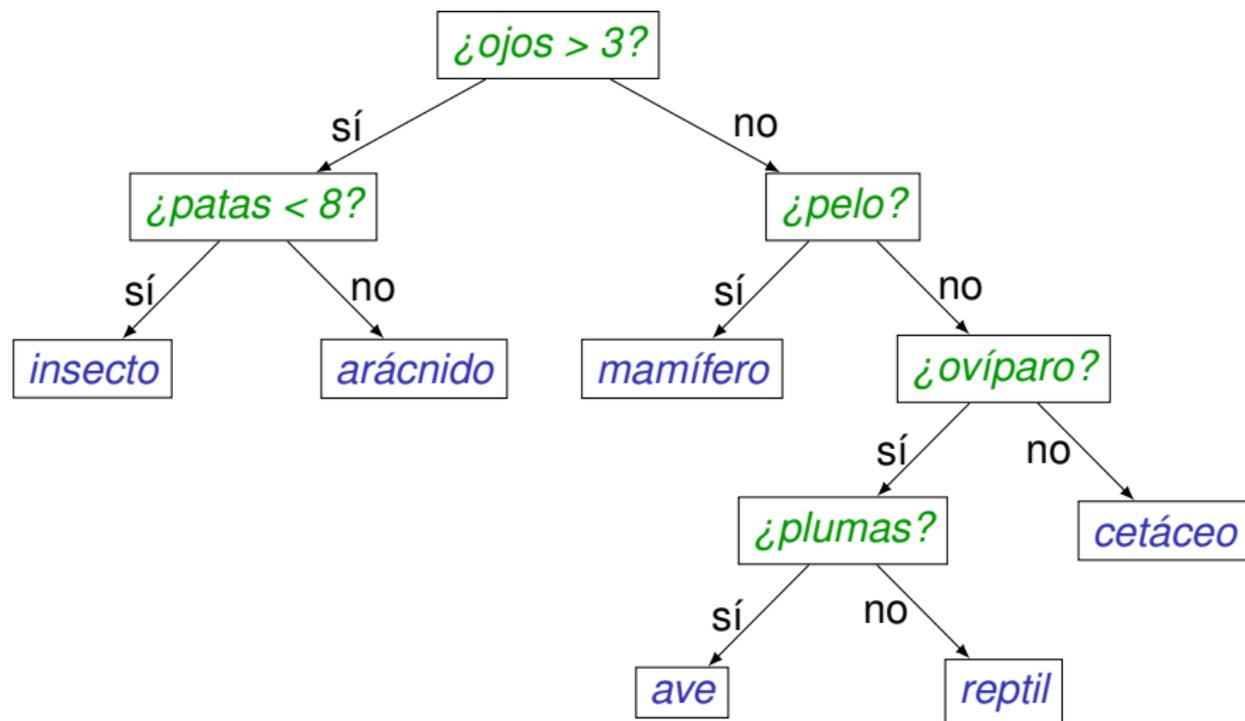
```
if x>0:  
    a  
else:  
    b
```

## Definición (Árbol de decisión binaria)

Es un **árbol binario** donde:

- Todo nodo **no hoja** tiene asociada una **condición** o **decisión binaria** sobre una variable (ej.  $x \geq 10$ ) y tiene siempre **dos hijos**: uno si la respuesta es **sí** y otro si es **no**
- Todo nodo **hoja** proporciona una **clasificación** o **decisión final**

# Ejemplo: clasificando animales



# Árboles de Decisión para aprendizaje

- Árbol de decisión = en inglés **Decision Tree (DT)**

# Árboles de Decisión para aprendizaje

- Árbol de decisión = en inglés **Decision Tree (DT)**
- Existen algoritmos de **aprendizaje automático [Quinlan 1979]** capaces de **construir un DT** a partir de un conjunto de ejemplos preclasificados

# Árboles de Decisión para aprendizaje

- Árbol de decisión = en inglés **Decision Tree (DT)**
- Existen algoritmos de **aprendizaje automático [Quinlan 1979]** capaces de **construir un DT** a partir de un conjunto de ejemplos preclasificados
- Esos algoritmos intentan detectar primero las **preguntas relevantes** = las que suponen una **ganancia de información**

# Árboles de Decisión para aprendizaje

- Árbol de decisión = en inglés **Decision Tree (DT)**
- Existen algoritmos de **aprendizaje automático [Quinlan 1979]** capaces de **construir un DT** a partir de un conjunto de ejemplos preclasificados
- Esos algoritmos intentan detectar primero las **preguntas relevantes** = las que suponen una **ganancia de información**
- Ejemplo online: juego de preguntas y respuestas **Akinator** (<https://en.akinator.com/>)



# Árboles de Decisión Booleanos

- Cualquier **función Booleana** (fórmula proposicional) se puede representar como un DT

# Árboles de Decisión Booleanos

- Cualquier **función Booleana** (fórmula proposicional) se puede **representar como un DT**
- Nodo **no hoja**: no hay variables numéricas:  
la preguntas es siempre sobre alguna **proposición**

# Árboles de Decisión Booleanos

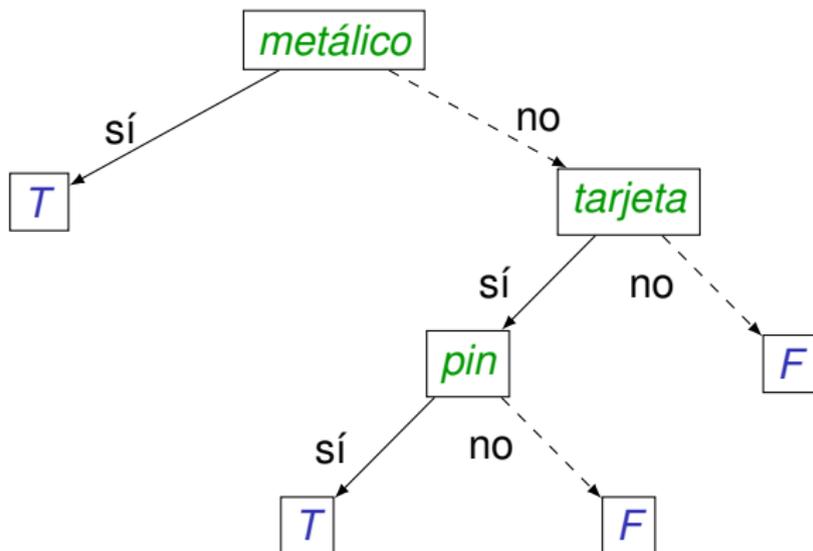
- Cualquier **función Booleana** (fórmula proposicional) se puede representar como un DT
- Nodo **no hoja**: no hay variables numéricas: la preguntas es siempre sobre alguna **proposición**
- Nodo **hoja**: contiene un valor
  - ▶  $T$  = true = cierto = el camino seguido es **modelo**
  - ▶  $F$  = false = falso = el camino seguido es **contramodelo**

# Árboles de Decisión Booleanos

- Ejemplo: el pago es válido cuando  $metalico \vee (tarjeta \wedge pin)$

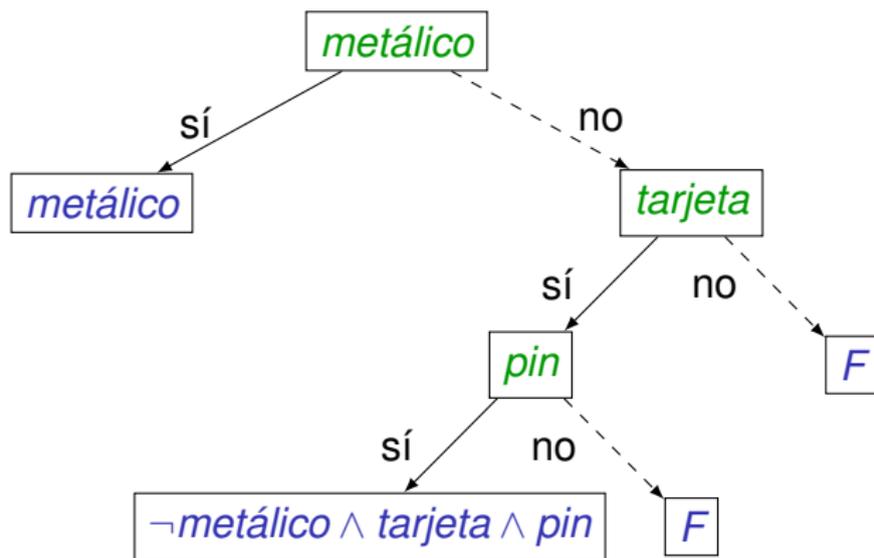
# Árboles de Decisión Booleanos

- Ejemplo: el pago es válido cuando  $metálico \vee (tarjeta \wedge pin)$



# Árboles de Decisión Booleanos

- Ejemplo: el pago es válido cuando  $metálico \vee (tarjeta \wedge pin)$

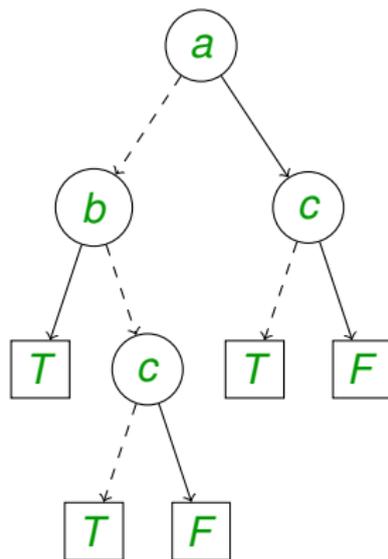


# Árboles de Decisión Booleanos

- En ocasiones, tenemos fragmentos de un árbol (subárboles) que se repiten

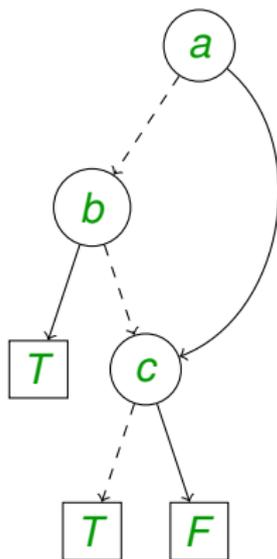
# Árboles de Decisión Booleanos

- En ocasiones, tenemos fragmentos de un árbol (subárboles) que se repiten
- Ejemplo: el subárbol de *c* se repite



# Árboles de Decisión Booleanos

- En ocasiones, tenemos fragmentos de un árbol (subárboles) que se repiten
- Ejemplo: el subárbol de *c* se repite



# Diagrama de Decisión Binario

## Definición (Diagrama de Decisión Binario)

*Es un grafo dirigido acíclico que cumple:*

- *Tiene un nodo raíz único*
- *Todo nodo **no terminal** tiene asociada una proposición  $p$  y dos arcos salientes: uno (punteado) si  $p$  es falsa y otro (continuo) si  $p$  es cierta*
- *Todo nodo **terminal** contiene  $T$  (cierto) o  $F$  (falso)*

# Diagrama de Decisión Binario

## Definición (Diagrama de Decisión Binario)

Es un grafo dirigido acíclico que cumple:

- Tiene un nodo raíz único
- Todo nodo **no terminal** tiene asociada una proposición  $p$  y dos arcos salientes: uno (punteado) si  $p$  es falsa y otro (continuo) si  $p$  es cierta
- Todo nodo **terminal** contiene  $T$  (cierto) o  $F$  (falso)

## Definición (Diagrama de Decisión Binario Reducido y Ordenado)

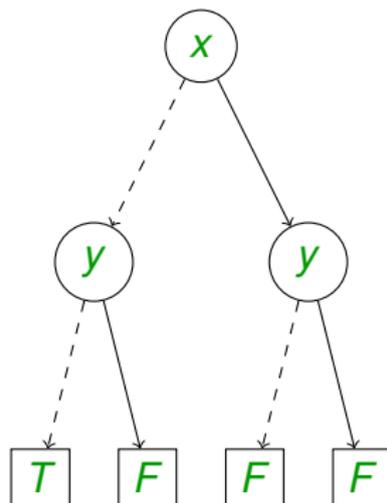
En inglés *Reduced Ordered Binary Decision Diagram (RO)BDD*.

Es un diagrama de decisión binario donde además

- Reducido = No contiene *redundancia*
- Ordenado = Las proposiciones siempre siguen el *mismo orden fijo*

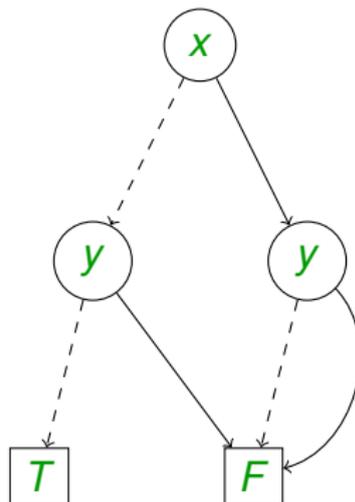
# Reducción de BDDs

Regla 1: eliminar terminales duplicados



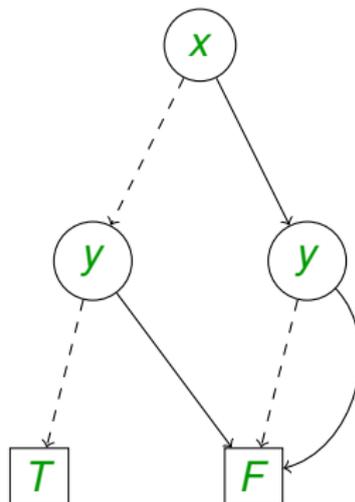
# Reducción de BDDs

Regla 1: eliminar terminales duplicados



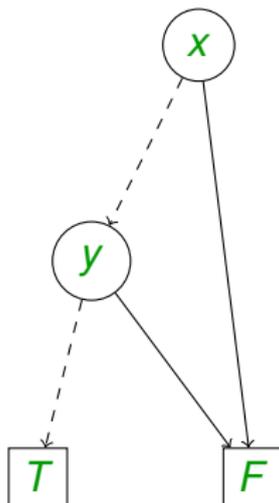
# Reducción de BDDs

Regla 2: eliminar comprobaciones redundantes



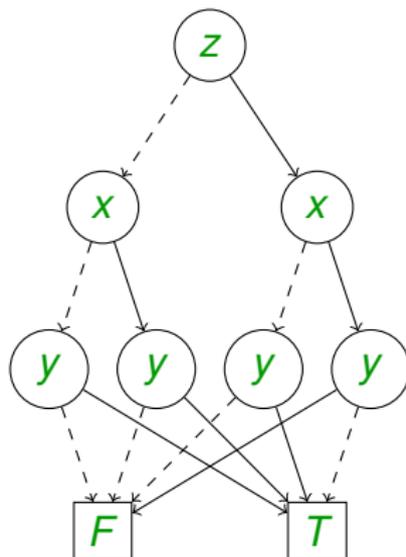
# Reducción de BDDs

Regla 2: eliminar comprobaciones redundantes



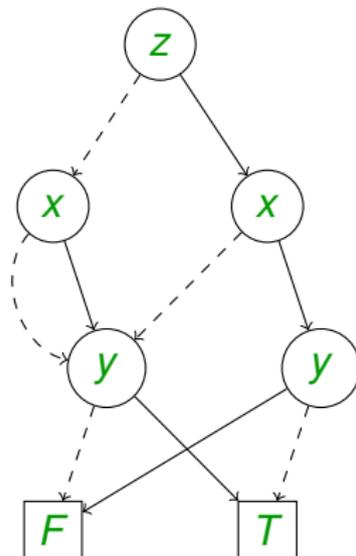
# Reducción de BDDs

Regla 3: eliminar no terminales duplicados



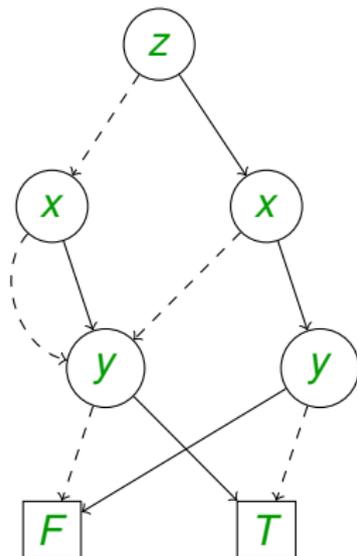
# Reducción de BDDs

Regla 3: eliminar no terminales duplicados



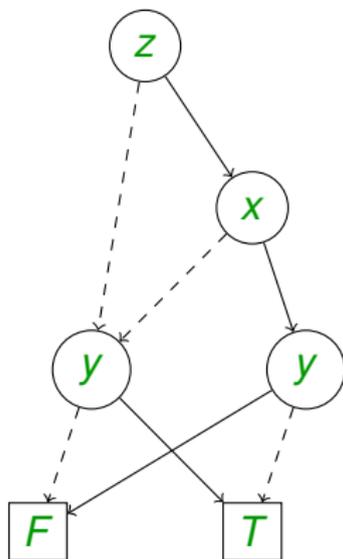
# Reducción de BDDs

Repetimos reglas 1, 2, 3 hasta que no se pueda reducir más



# Reducción de BDDs

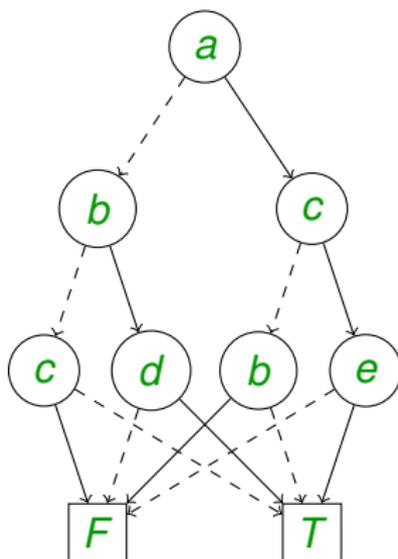
Repetimos reglas 1, 2, 3 hasta que no se pueda reducir más



- \* Ejemplo extraído del curso “Formal Verification” de Jacques Fleuriot  
University of Edinburgh, UK

# Orden de variables

- En general, el **orden en que aparecen las proposiciones** puede variar de un camino a otro



Tenemos un camino  $a \rightarrow b \rightarrow c \rightarrow F$  y otro  $a \rightarrow c \rightarrow b \rightarrow F$

# Orden de variables

- Decimos que el BDD es **ordenado** si se respeta siempre un **orden fijo** de las variables

# Orden de variables

- Decimos que el BDD es **ordenado** si se respeta siempre un **orden fijo** de las variables

## Teorema

*Dado un orden de variables, toda función Booleana tiene una representación única (o canónica) como ROBDD.*

# Orden de variables

- Decimos que el BDD es **ordenado** si se respeta siempre un **orden fijo** de las variables

## Teorema

*Dado un orden de variables, toda función Booleana tiene una representación única (o canónica) como ROBDD.*

- Como consecuencia: dos fórmulas son **equivalentes** sii sus ROBDD coinciden.

# Orden de variables

- Decimos que el BDD es **ordenado** si se respeta siempre un **orden fijo** de las variables

## Teorema

*Dado un orden de variables, toda función Booleana tiene una representación única (o canónica) como ROBDD.*

- Como consecuencia: dos fórmulas son **equivalentes** sii sus **ROBDD coinciden**.
- El orden de variables puede **impactar significativamente** en el tamaño del BDD.

# Orden de variables

- Decimos que el BDD es **ordenado** si se respeta siempre un **orden fijo** de las variables

## Teorema

*Dado un orden de variables, toda función Booleana tiene una representación única (o canónica) como ROBDD.*

- Como consecuencia: dos fórmulas son **equivalentes** sii sus **ROBDD coinciden**.
- El orden de variables puede **impactar significativamente** en el tamaño del BDD.
- Encontrar el **orden óptimo** (que minimiza el tamaño del BDD) es un problema **NP-completo**

# Orden de variables

- Decimos que el BDD es **ordenado** si se respeta siempre un **orden fijo** de las variables

## Teorema

*Dado un orden de variables, toda función Booleana tiene una representación única (o canónica) como ROBDD.*

- Como consecuencia: dos fórmulas son **equivalentes** si sus **ROBDD coinciden**.
- El orden de variables puede **impactar significativamente** en el tamaño del BDD.
- Encontrar el **orden óptimo** (que minimiza el tamaño del BDD) es un problema **NP-completo**
- Prueba a generar BDDs en esta página:

[https://eecs.ceas.uc.edu/~weaversa/BDD\\_Visualizer.html](https://eecs.ceas.uc.edu/~weaversa/BDD_Visualizer.html)