IC - Árboles de decision y razonamiento basado en casos

Adam Lei Yi Chen Abolacio

January 2025

Table of Contents

Árboles de decisión binarios

Árboles de decisión con múltiples ramas

Aprendizaje en árboles de decisión

Árboles de decisión en CLIPS

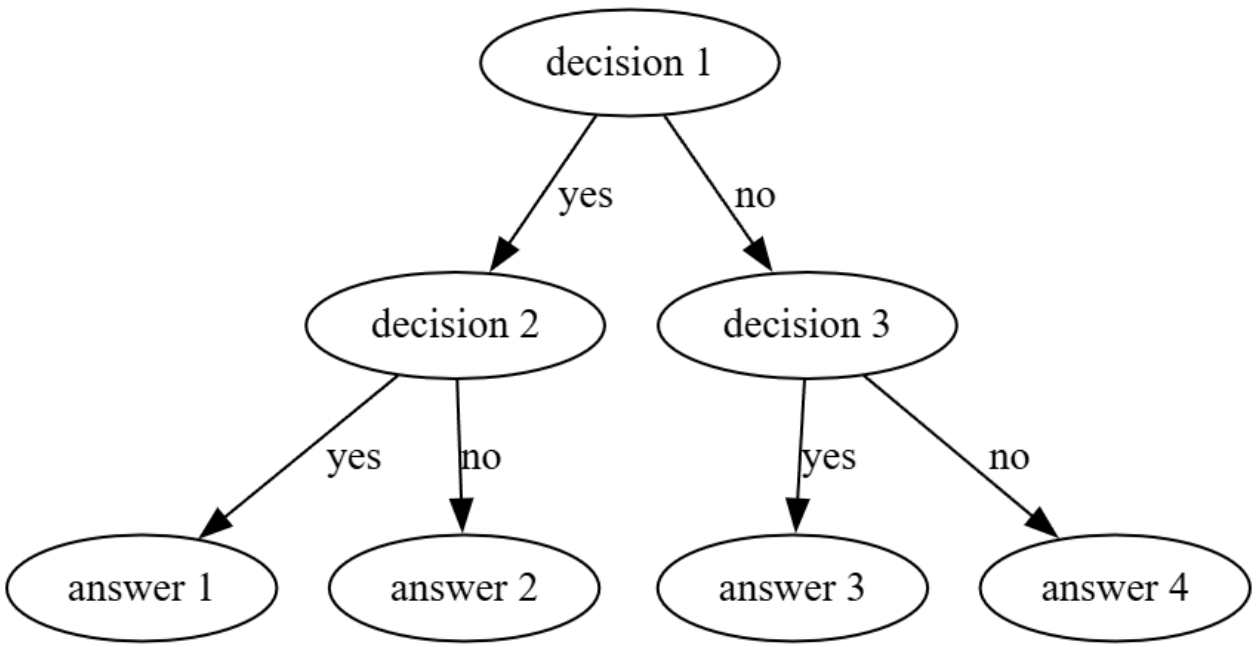
Razonamiento basado en casos

Razonamiento basado en casos vs árboles de decisión

Árboles de decisión binarios

Podemos representar un árbol de decisión binario mediante nodos de decisión y nodos de respuesta.

- Los nodos de respuesta son las hojas del árbol.
- Los nodos de decisión son el resto de nodos del árbol.
- En nuestro caso, todos los nodos de decisión siempre tienen dos hijos.



Árbol de decisión para clasificar el mejor tipo de vino para una comida

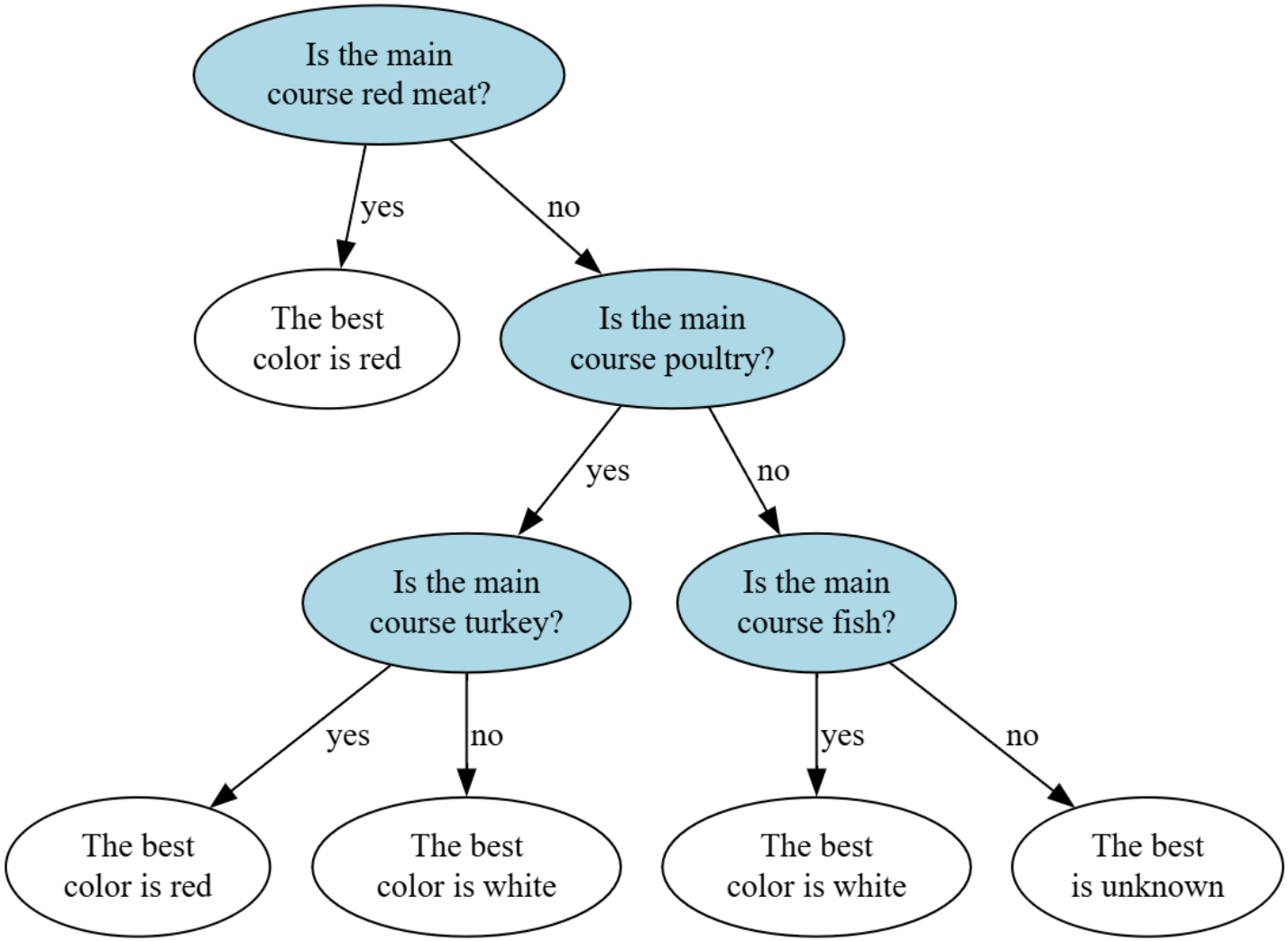
Cada línea de instrucción condicional representa un camino completo, desde el nodo root hasta un nodo answer.

Código de un árbol de decisión

```
IF the main course is red meat
THEN serve red wine

IF the main course is poultry and it is turkey
THEN serve red wine

IF the main course is poultry and it is not turkey
THEN serve white wine

IF the main course is fish
THEN serve white wine
```

Árbol de decisión para clasificar el mejor tipo de vino para una comida

Árbol de decisión equivalente



Resolución de un Árbol Binario

Algorithm 1 Solve_Binary_Tree

```
1: Set current location in the tree to the root node.
2: while current location is a decision node do
3:     Ask the question at the current node.
4:     if the reply to the question is yes then
5:         Set the current node to the yes branch.
6:     else
7:         Set the current node to the no branch.
8:     end if
9: end while
10: Return the answer at the current node.
```

Complejidad en árboles de decisión binarios

- Los árboles de decisión binarios pueden derivar en decisiones (preguntas) más complejas para representar casos donde la respuesta puede ser una serie de valores.
- Esto resulta en árboles potencialmente ineficientes debido al elevado número de preguntas.
- Objetivo: nodos de decisión más expresivos.

Table of Contents

Árboles de decisión binarios

Árboles de decisión con múltiples ramas

Aprendizaje en árboles de decisión

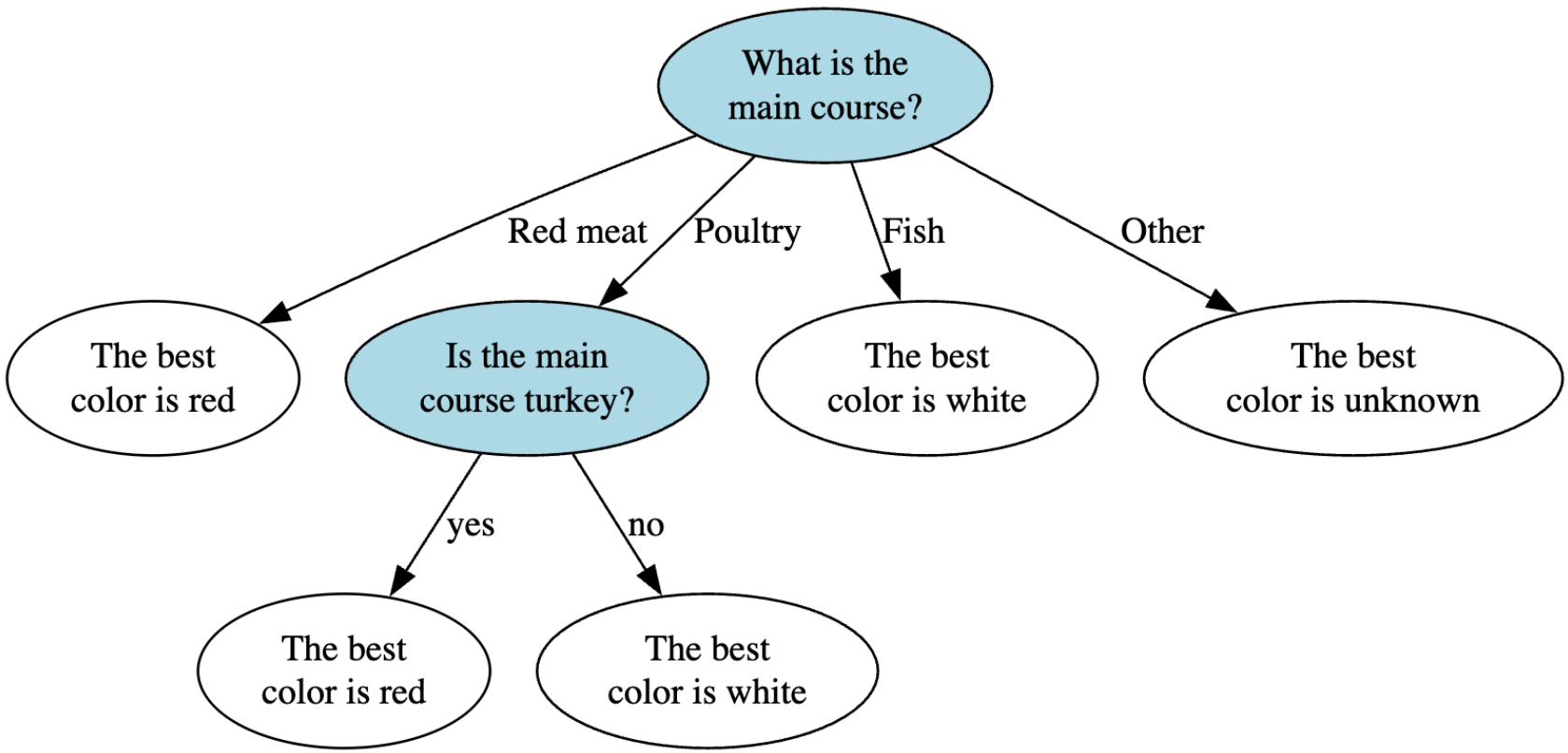
Árboles de decisión en CLIPS

Razonamiento basado en casos

Razonamiento basado en casos vs árboles de decisión

Árboles de decisión con múltiples ramas

En el ejemplo de los vinos, si cambiamos la pregunta inicial por "What is the main course?" podemos obtener un árbol de decisión que clasifica igual pero realiza como máximo 2 preguntas:



Resolución de un Árbol de Decisión

Algorithm 2 Solve_Tree

1: **Set** current tree location to the root node.
2: **while** current location is a decision node **do**
3: Ask the question at the current node until an answer in the set of valid choices for this node has been provided.
4: Set the current node to the child node of the branch associated with the choice selected.
5: **end while**
6: **Return** the answer at the current node.

Table of Contents

Árboles de decisión binarios

Árboles de decisión con múltiples ramas

Aprendizaje en árboles de decisión

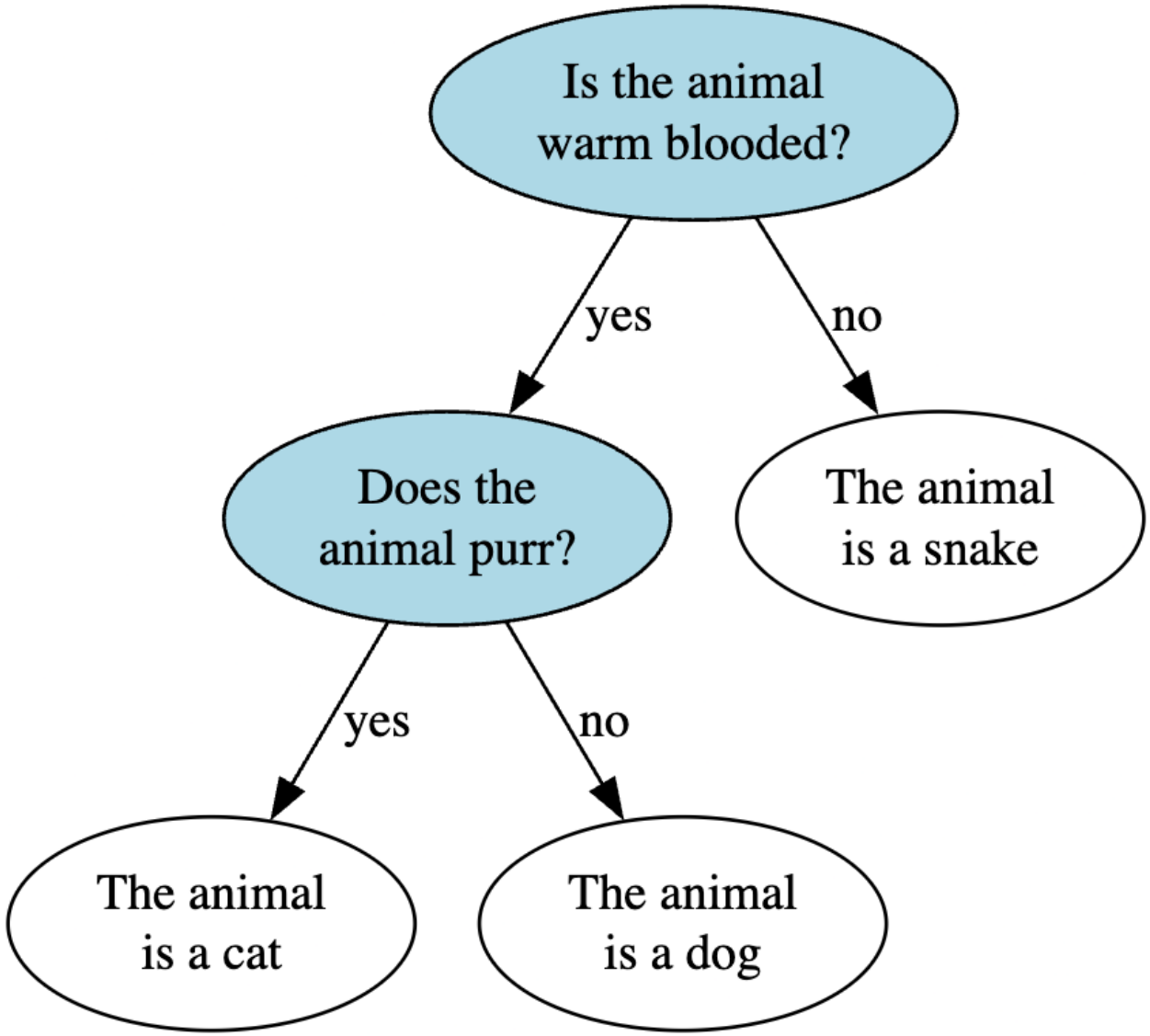
Árboles de decisión en CLIPS

Razonamiento basado en casos

Razonamiento basado en casos vs árboles de decisión

Aprendizaje en árboles de decisión

Árbol de decisión ingenuo (sólo conoce 3 animales)

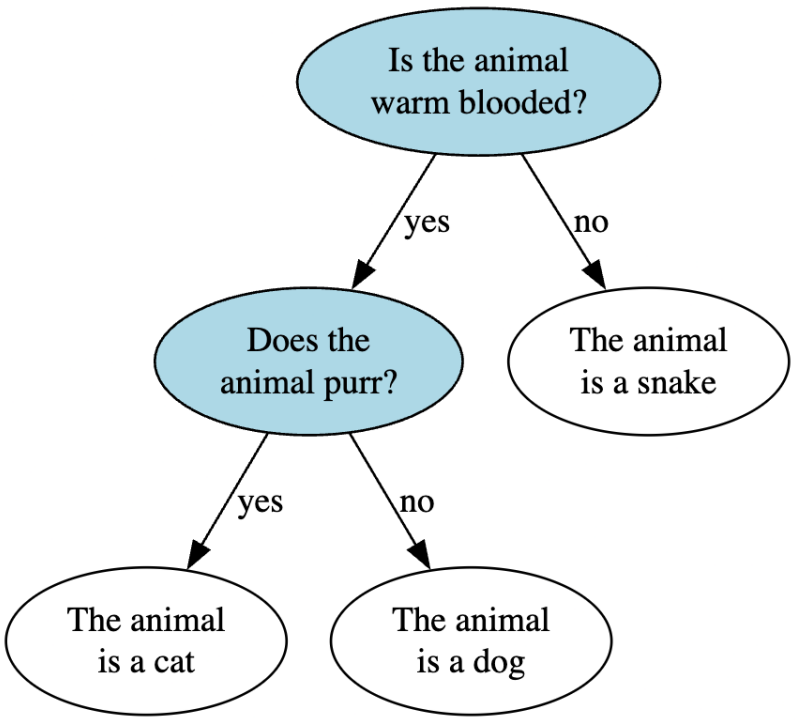


Aprendizaje en árboles de decisión

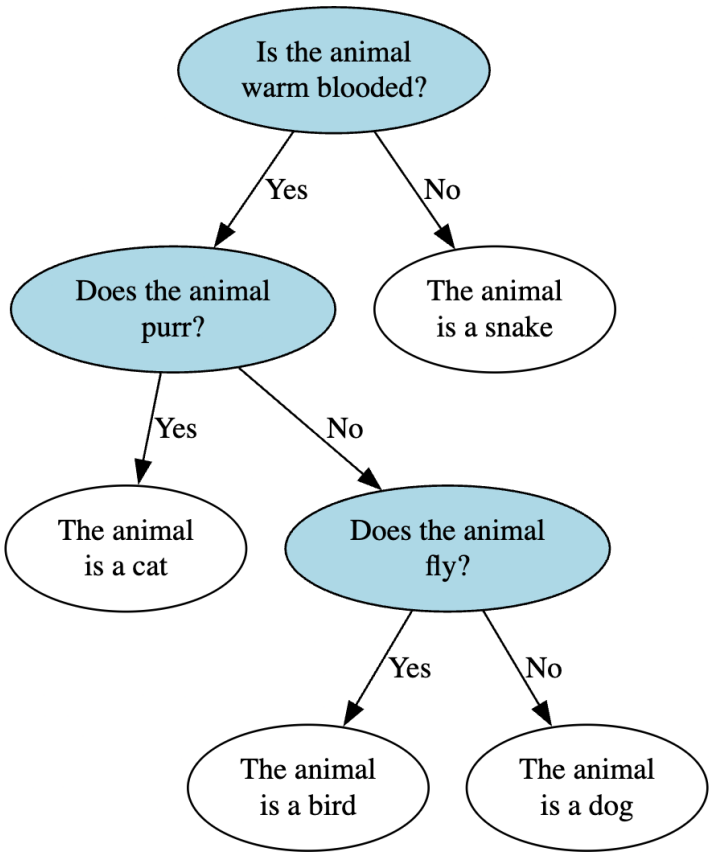
Is the animal warm-blooded? (yes or no) yes
Does the animal purr? (yes or no) no
I guess it is a dog
Am I correct? (yes or no) no
What is the animal? bird
What question when answered yes will distinguish
a bird from a dog? Does the animal fly?
Now I can guess bird
Try again? (yes or no) no

Aprendizaje en árboles de decisión

Antes



Después



Aprendizaje en árboles de decisión

Algorithm 3 Solve_Tree_and_Learn

```
1: Set the current location in the tree to the root node.
2: while the current location is a decision node do
3:     Ask the question at the current node.
4:     if the reply to the question is yes then
5:         Set the current node to the yes branch.
6:     else
7:         Set the current node to the no branch.
8:     end if
9: end while
10: Ask if the answer at the current node is correct.
11: if the answer is correct then
12:     Return the correct answer.
13: else
14:     Determine the correct answer.
15:     Determine a question which, when answered yes,
            will distinguish the current answer from the correct one.
16:     Replace the answer node with a decision node:
            - The no branch retains the current answer node.
            - The yes branch has an answer node with the correct answer.
17:     The decision node's question should be the one distinguishing both answers.
18: end if
```

Table of Contents

Árboles de decisión binarios

Árboles de decisión con múltiples ramas

Aprendizaje en árboles de decisión

Árboles de decisión en CLIPS

Razonamiento basado en casos

Razonamiento basado en casos vs árboles de decisión

Árboles de decisión en CLIPS

Objetivo: implementar el algoritmo de aprendizaje anterior en un sistema basado en reglas, en nuestro caso CLIPS.

Representaremos el árbol mediante hechos en la KB:

```
(deftemplate node
    (slot name)
    (slot type)
    (slot question)
    (slot yes-node)
    (slot no-node)
    (slot answer))
```

Árboles de decisión en CLIPS

Para guardar la información del árbol de decisión se utilizará un archivo que el sistema guardará y cargará en cada ejecución (*load-facts*, *save-facts*):

```
(node (name root) (type decision)
      (question "Is the animal warm-blooded?")
      (yes-node node1) (no-node node2))
(node (name node1) (type decision)
      (question "Does the animal purr?")
      (yes-node node3) (no-node node4))
(node (name node2) (type answer) (answer snake))
(node (name node3) (type answer) (answer cat))
(node (name node4) (type answer) (answer dog))
```

Árboles de decisión en CLIPS

- **Regla para cargar archivo:**

```
(defrule initialize
    (not (node (name root)))
    =>
    (load-facts "animal.dat")
    (assert (current-node root)))
```

- **Función para realizar preguntas al usuario:**

```
(deffunction ask-yes-or-no (?question)
    (printout t ?question " (yes or no) ")
    (bind ?answer (read))
    (while (and (neq ?answer yes) (neq ?answer no))
    (printout t ?question " (yes or no) ")
    (bind ?answer (read)))
    (return ?answer))
```

Árboles de decisión en CLIPS

- **Regla cuando el nodo actual es un nodo de decisión:**

```
(defrule ask-decision-node-question
    ?node <- (current-node ?name)
    (node (name ?name)
          (type decision)
          (question ?question))
    (not (answer ?))
    =>
    (assert (answer (ask-yes-or-no ?question))))
```

Árboles de decisión en CLIPS

- **Regla para pasar del nodo actual al nodo yes:**

```
(defrule proceed-to-yes-branch
    ?node <- (current-node ?name)
    (node (name ?name)
          (type decision)
          (yes-node ?yes-branch))
    ?answer <- (answer yes)
    =>
    (retract ?node ?answer)
    (assert (current-node ?yes-branch)))
```

- **Regla para pasar del nodo actual al nodo no:**

```
(defrule proceed-to-no-branch
    ?node <- (current-node ?name)
    (node (name ?name)
          (type decision)
          (no-node ?no-branch))
    ?answer <- (answer no)
    =>
    (retract ?node ?answer)
    (assert (current-node ?no-branch)))
```

Árboles de decisión en CLIPS

- **Regla para comprobar si el nodo respuesta es la respuesta correcta para el usuario:**

```
(defrule ask-if-answer-node-is-correct
    ?node <- (current-node ?name)

    (node (name ?name) (type answer)
          (answer ?value))
    (not (answer ?))
    =>
    (printout t "I guess it is a " ?value crlf)
    (assert
        (answer (ask-yes-or-no "Am I correct?"))))
```

Árboles de decisión en CLIPS

- **Regla para modificar el árbol con la respuesta:**

```
(defrule replace-answer-node
    ?phase <- (replace-answer-node ?name)
    ?data <- (node (name ?name)
                   (type answer)
                   (answer ?value))
    =>
    (retract ?phase)
    ; Determine what the guess should have been
    (printout t "What is the animal? ")
    (bind ?new-animal (read))
    ; Get the question for the guess
    (printout t "What question when answered yes ")
    (printout t "will distinguish " crlf " a ")
    (printout t ?new-animal " from a " ?value "? ")
    (bind ?question (readline))
    (printout t "Now I can guess " ?new-animal crlf)
    ; Create the new learned nodes
    (bind ?newnode1 (gensym*))
    (bind ?newnode2 (gensym*))
    (modify ?data (type decision)
                  (question ?question)
                  (yes-node ?newnode1)
                  (no-node ?newnode2))
    (assert (node (name ?newnode1)
                  (type answer)
                  (answer ?new-animal)))
    (assert (node (name ?newnode2)
                  (type answer)
                  (answer ?value)))
    ; Determine if the player wants to try again
    (assert (ask-try-again)))
```

Table of Contents

Árboles de decisión binarios

Árboles de decisión con múltiples ramas

Aprendizaje en árboles de decisión

Árboles de decisión en CLIPS

Razonamiento basado en casos

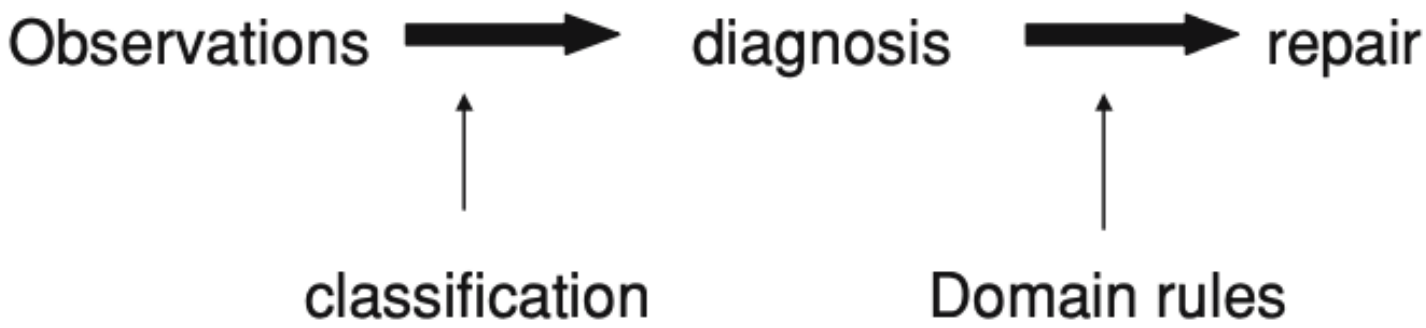
Razonamiento basado en casos vs árboles de decisión

Razonamiento basado en casos

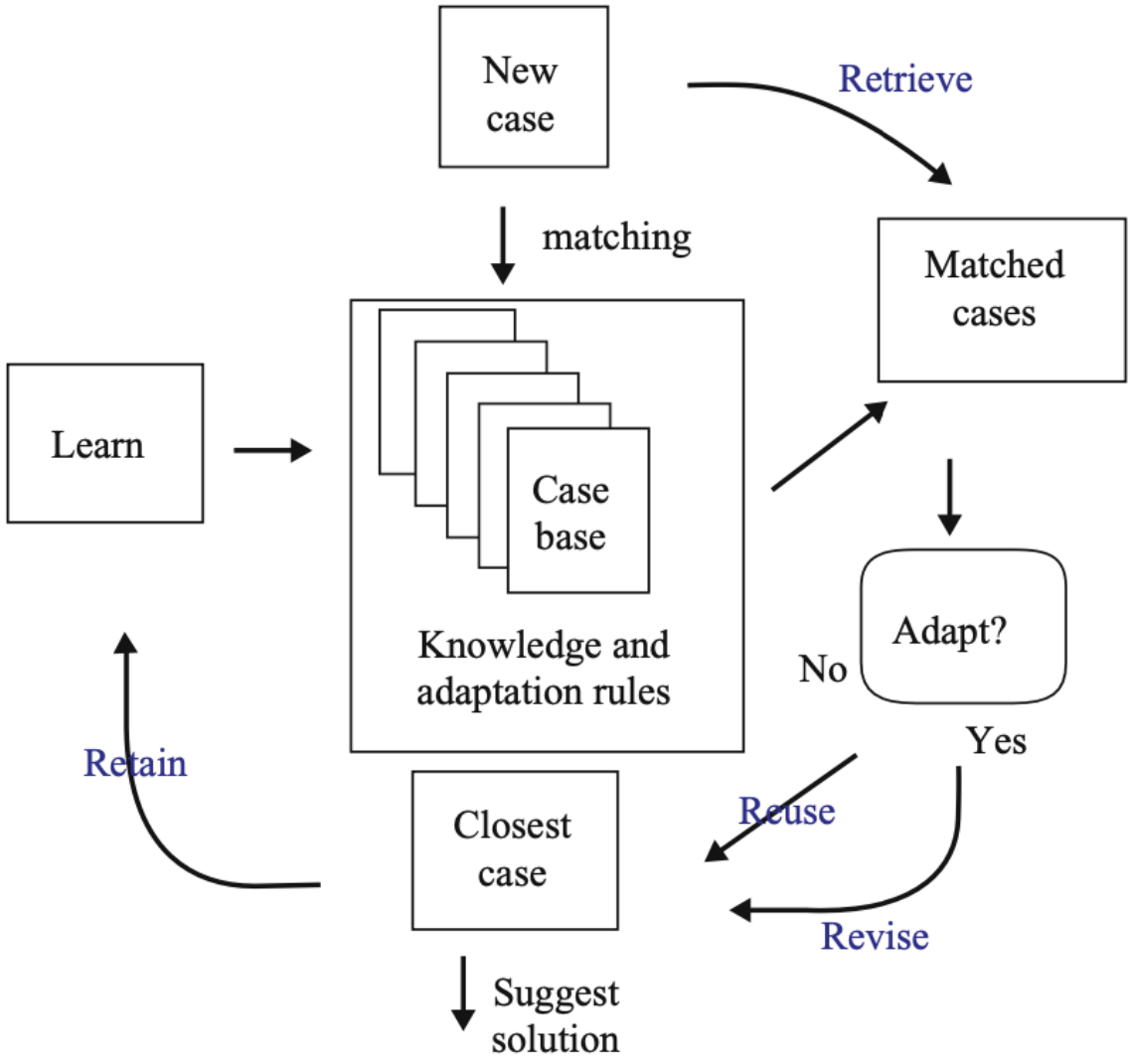
- De igual manera que los árboles de decisión, la información del sistema es almacenada en la KB mediante casos.
- El objetivo de los sistemas basados en casos es que: dado una nueva situación (nuevo caso) encontrar casos similares que sirvan para resolver el nuevo caso.
- El razonamiento basado en casos funciona de manera similar a como los humanos guían sus acciones basándose en la experiencia previa.

Razonamiento basado en casos

- Los problemas que se abordan mediante el uso de CBR tienden a ser aquellos con características de clasificación y diagnóstico

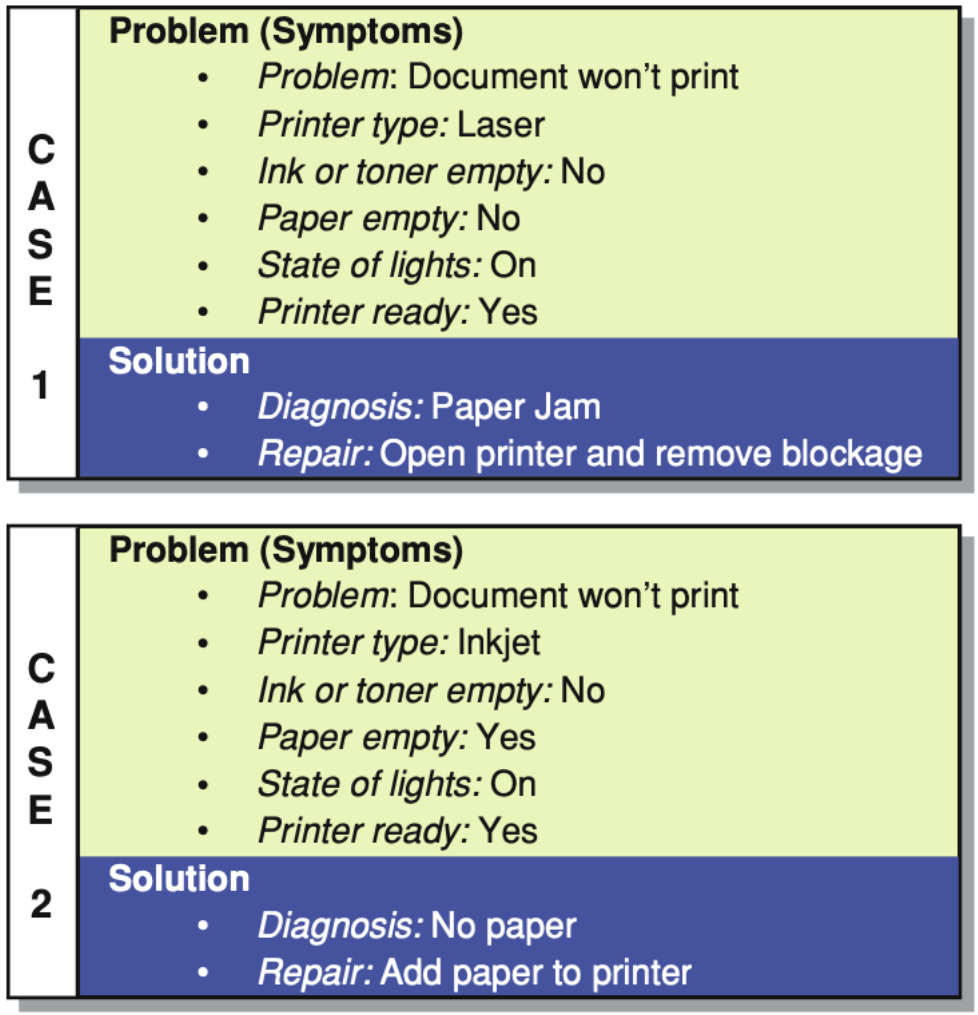


Razonamiento basado en casos

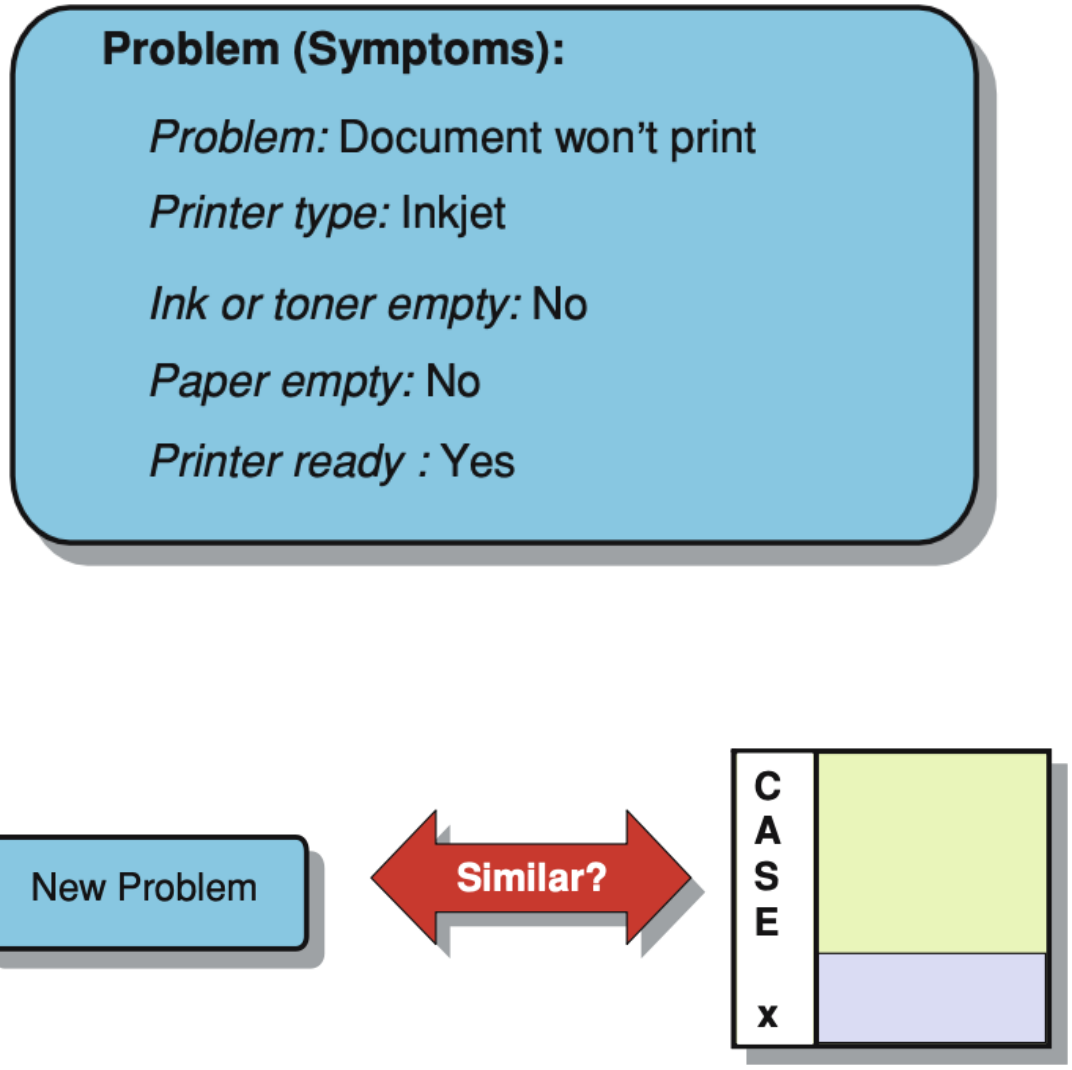


Razonamiento basado en casos

- Identificar las características importantes del problema

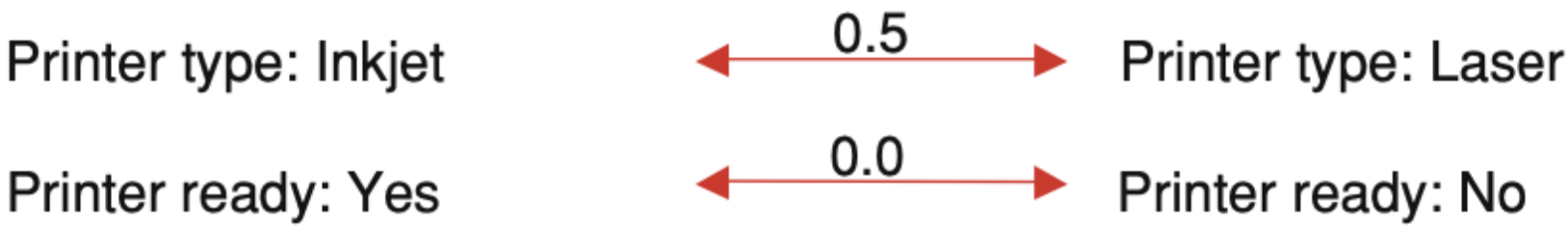


Razonamiento basado en casos

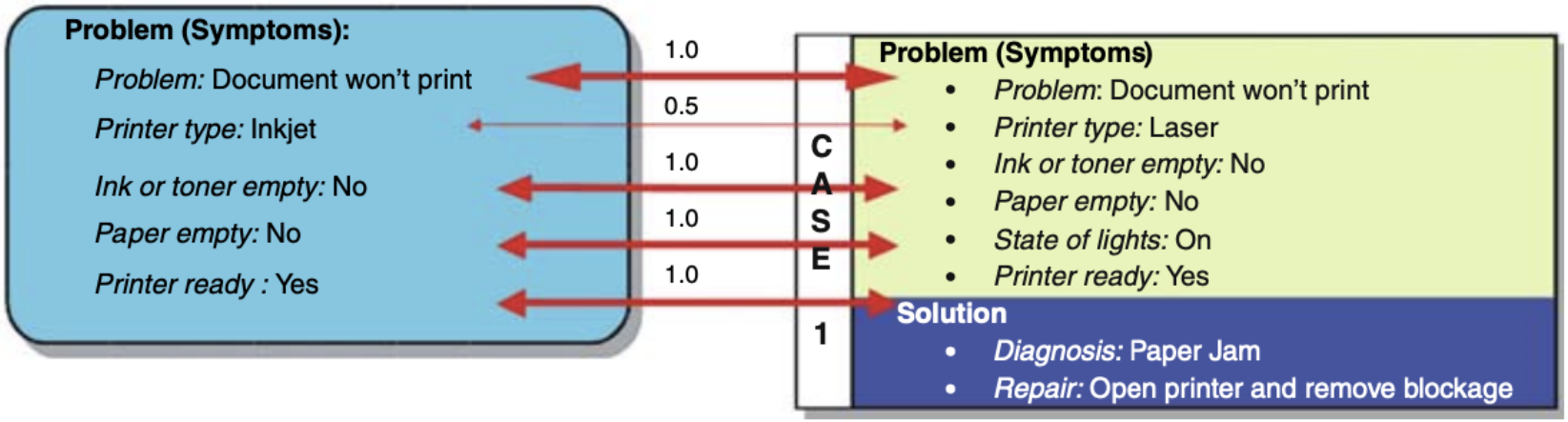


Razonamiento basado en casos

- Definir el grado de similaridad de cada componente como un valor real entre 0 y 1
- El grado de similaridad puede ser asignado en base a la experiencia
- Asignar un peso en base a la importancia de cada característica



Razonamiento basado en casos



$$Similarity(new, case_1) = \frac{1}{25}(6 \cdot 1.0 + 1 \cdot 0.5 + 6 \cdot 1.0 + 6 \cdot 1.0 + 6 \cdot 1.0)$$

$$Similarity(new, case_1) = 0.98$$

Table of Contents

Árboles de decisión binarios

Árboles de decisión con múltiples ramas

Aprendizaje en árboles de decisión

Árboles de decisión en CLIPS

Razonamiento basado en casos

Razonamiento basado en casos vs árboles de decisión

Razonamiento basado en casos vs árboles de decisión

- En los árboles de decisión las condiciones de cada problema tienen que ser exactamente iguales en cada nuevo problema, al contrario que el razonamiento basado en casos donde podemos realizar una búsqueda aproximada.
- El razonamiento basado en casos es un método más sencillo cuando el dominio del problema impide descomponer dichos problemas con facilidad.
- El razonamiento basado en casos es fácilmente adaptable a nuevas características.
- El razonamiento basado en casos puede aprovecharse de toda la información histórica.
- Dificultad en la formalización de la similitud.