# Can algorithmics be useful in machine learning? Application of Banzhaf values to explain tree models

Piotr Sankowski

# Main Result

Adam Karczmarz

Anish Mukherjee

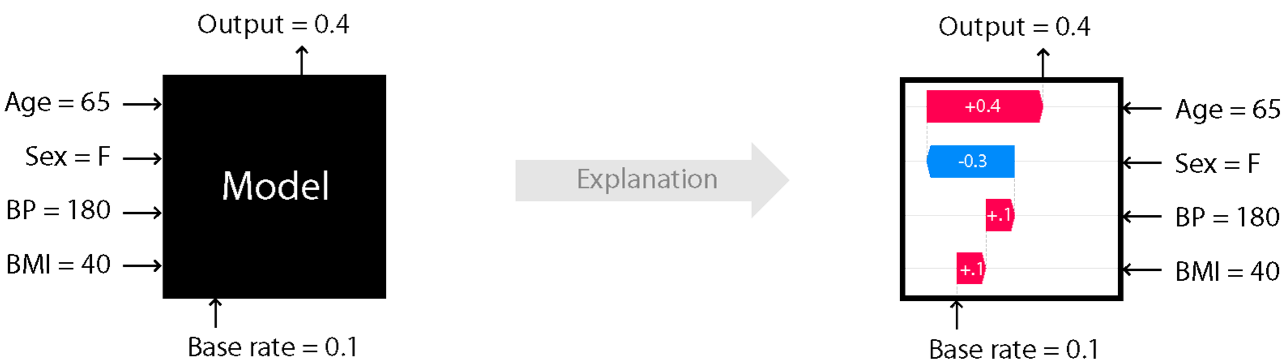
Piotr Sankowski

Piotr Wygocki

Tomasz Michalak

**Karczmarz, Mukherjee, Wygocki, Michalak, Sankowski, "*Improved Feature Importance Computation for Tree Models Based on the Banzhaf Value*" The Conference on Uncertainty in Artificial Intelligence (UAI), Eindhoven, Netherlands, August 1-5, 2022.**

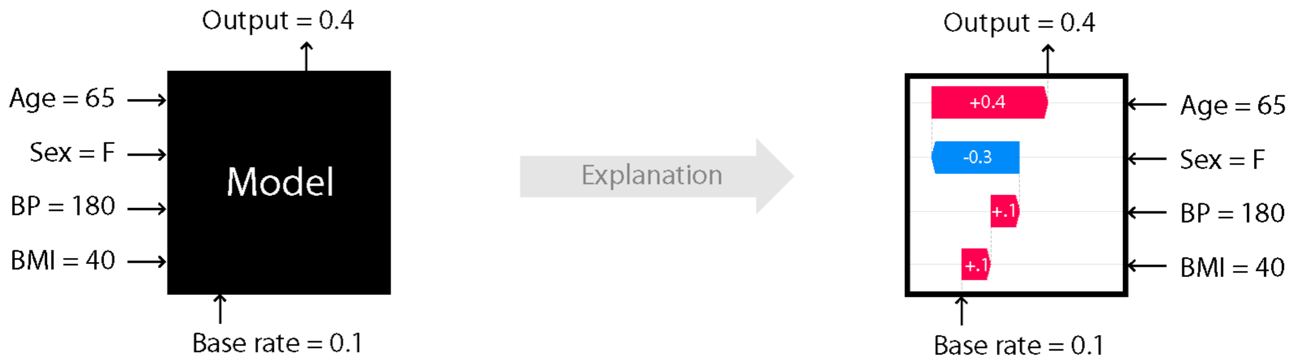
# SHAP and TreeSHAP




Source: https://shap.readthedocs.io/en/latest/index.html

# SHAP and TreeSHAP

SHAP




Source: https://shap.readthedocs.io/en/latest/index.html

**Lundberg et al. (2018, 2020) proposed TreeSHAP - an exact algorithm to compute the Shapley value-based explanations for tree models in $O(TLD^2 + n)$. Here:**

- $n$ – is the number of features
- $T$ – the number of trees
- $L$ – the number of leaves
- $D$ – the maximum depth of a tree

# Take away message

**Technical contribution:**

We advocate the Banzhaf value for tree models:

1. It can be computed noticeably faster
2. It seems to be more numerically stable
3. Our experimental comparison shows:
    - essentially the same global impacts
    - close explanations of individual predictions

**Meta level:**

- Game theory and algorithmic view
- Interplay of the above areas with AI is growing
- Many more interesting problems to come

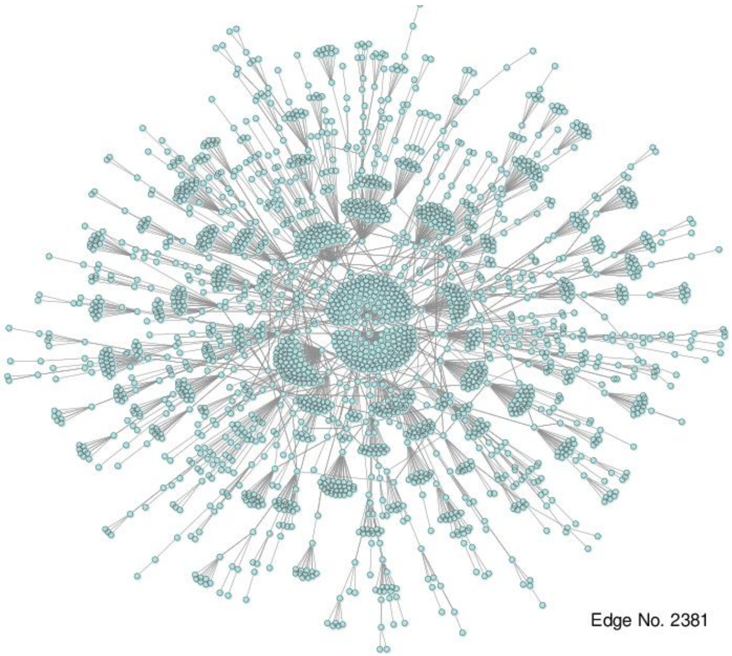
# Scale-free Networks

***Definition:*** **An undirected graph G is called a power-law graph with parameter $\alpha > 1$ if the fraction of vertices of degree $k$ is proportional to $k^{-\alpha}$.**

***Theorem:*** **If G is „power-law graph” then th**

**heuristic finds maximum clique**

- **in polynomial time for for $\alpha > 3$,**
- **subexponential time for $2 < \alpha < 3$.**




**Pawel Brach, Marek Cygan, Jakub Lacki, Piotr Sankowski:**

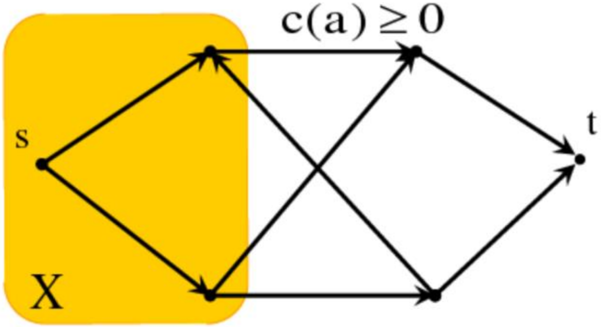
***Algorithmic Complexity of Power Law Networks*. SODA 2016: 1306-1325**

# Scale-free Networks

*Definition:* If $\Omega$ is a finte set, a function $f: 2^{\Omega} \to R$ is submodular when

- For every $S, T \subseteq \Omega$ we have that $f(S) + f(T) \geq f(S \cup T) + f(S \cap T)$.

*Theorem:* When a submodular function is decomposable into sum of simple submodular functions then its minimum can be found in time needed to solve the maximum flow problem.



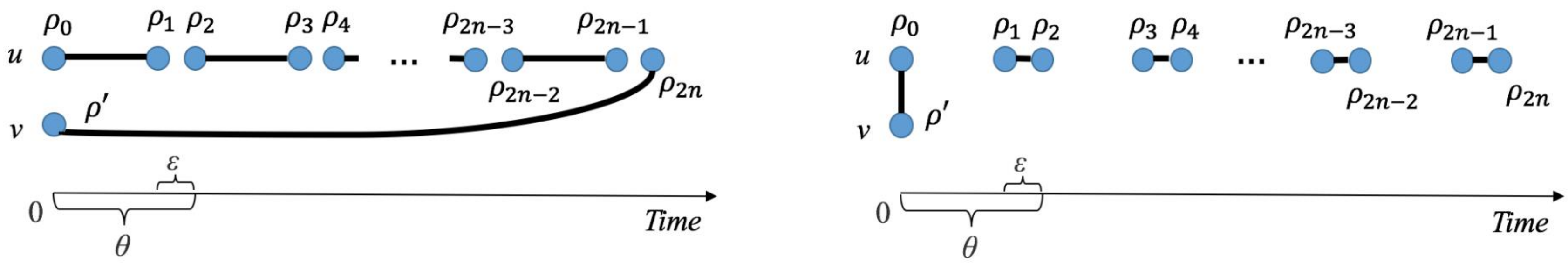

Kyriakos Axiotis, Adam Karczmarz, Anish Mukherjee, Piotr Sankowski, Adrian Vladu: *Decomposable Submodular Function Minimization via Maximum Flow*. ICML 2021: 446-456.

Li Chen, Rasmus Kyng, Yang P. Liu, Richard Peng, Maximilian Probst Gutenberg, Sushant Sachdeva: *Maximum Flow and Minimum-Cost Flow in Almost-Linear Time*. FOCS 2022: 612-623

# Stochastic Arrivials

***Definition: In the Min-cost Perfect Matching with Delays (MPMD) problem we need to match online requests by paying:***

- **the connection cost,**
- **the waiting time cost.**




***Theorem: For stochastic arrivals the greedy heuristic is constant competetive in expectation.***

**Mathieu Mari, Michał Pawłowski, Runtian Ren and Piotr Sankowski:** ***Online matching with delays and stochastic arrival times, AAMAS 2023.***

# Plan of the Talk

1. Values in Cooperative Game Theory

# Plan of the Talk

1. Values in Cooperative Game Theory
2. Our algorithm for the Banzhaf value vs. TreeSHAP

# Plan of the Talk

1. Values in Cooperative Game Theory
2. Our algorithm for the Banzhaf value vs. TreeSHAP
3. Advantages of the Banzhaf value for tree models - experimental analysis

# Plan of the Talk

1. Values in Cooperative Game Theory
2. Our algorithm for the Banzhaf value vs. TreeSHAP
3. Advantages of the Banzhaf value for tree models - experimental analysis

# Coalitional Games

Given the **set of agents**:

A = $\{a_1, a_2, a_3\}$

# Coalitional Games

Given the **set of agents**:

A = $\{a_1, a_2, a_3\}$

The possible **coalitions** are:

| $a_1$ | $a_2$ | $a_3$ | $a_1 a_2$ | $a_1 a_3$ | $a_2 a_3$ | $a_1 a_2 a_3$ |
|---|---|---|---|---|---|---|
| 4 | 0 | 3 | 10 | 7 | 12 | 19 |

$$v: 2^A \rightarrow \mathbb{R}$$

$$v(\emptyset) = 0$$

# Coalitional Games

Given the **set of agents**:

A = $\{a_1, a_2, a_3\}$

The possible **coalitions** are:

| $a_1$ | $a_2$ | $a_3$ | $a_1 a_2$ | $a_1 a_3$ | $a_2 a_3$ | $a_1 a_2 a_3$ |
|---|---|---|---|---|---|---|
| 4 | 0 | 3 | 10 | 7 | 12 | 19 |

$$v: 2^A \rightarrow \mathbb{R}$$

$$v(\emptyset) = 0$$

There is an ongoing debate how to define this function in the context of explainability.

We will divide the payoff of the grand coalition in a way that corresponds to players' contribution to the game.

# Axioms: some basic assumptions

- **We will divide the payoff of the grand coalition in a way that corresponds to players' contribution to the game.**

# Axioms: some basic assumptions

- We will divide the payoff of the grand coalition in a way that corresponds to players' contribution to the game.
- But how to measure this contribution to the game?

# Axioms: some basic assumptions

- We will divide the payoff of the grand coalition in a way that corresponds to players’ contribution to the game.
- But how to measure this contribution to the game?
- We will measure this contribution using the economic concept of a marginal contribution.

# Axioms: some basic assumptions

- We will divide the payoff of the grand coalition in a way that corresponds to players' contribution to the game.
- But how to measure this contribution to the game?
- We will measure this contribution using the economic concept of a marginal contribution.

**Marginal contribution:**

Let $C \subseteq A \setminus \{a_i\}$. Then:

$$MC(a_i, C) = v(C \cup \{a_i\}) - v(C).$$

# Axioms: some basic assumptions

- We will divide the payoff of the grand coalition in a way that corresponds to players' contribution to the game.
- But how to measure this contribution to the game?
- We will measure this contribution using the economic concept of a marginal contribution.

**Marginal contribution:**

Let $C \subseteq A \setminus \{a_i\}$. Then:

$$MC(a_i, C) = v(C \cup \{a_i\}) - v(C).$$

- We are interested in a method that considers the marginal contributions of a player to all the coalitions in the game.

# Shapley value

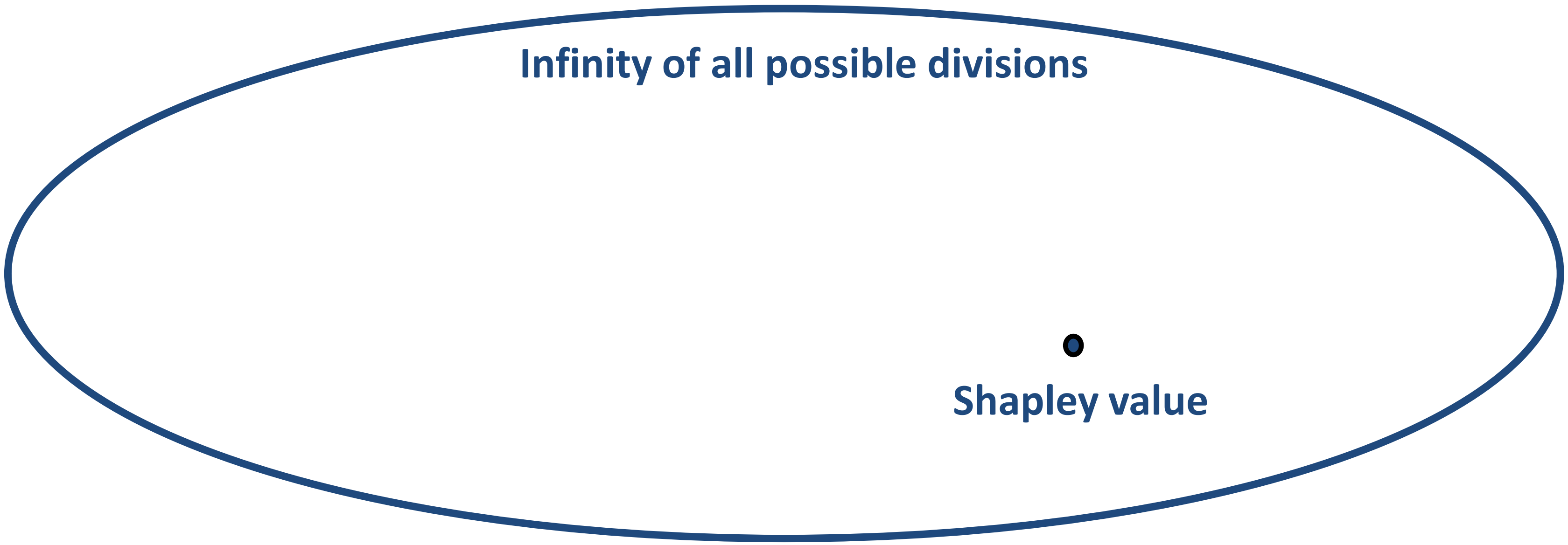
- **Symmetry** – any two players who always contribute the same to all coalitions should get the same payoff (i.e. the same share of the grand coalition)
- **Null player** – a player who does not contribute anything to any coalition should get nothing
- **Additivity** – for additive games, the payoffs should be also additive
- **Efficiency** – the total value of the grand coalition should be distributed among the players – there should be no leftovers and we should not be able to distribute more than we have

**Shapley value**

There exists unique value that satisfies Symmetry, Null player, Additivity and Efficiency. It is defined as follows:

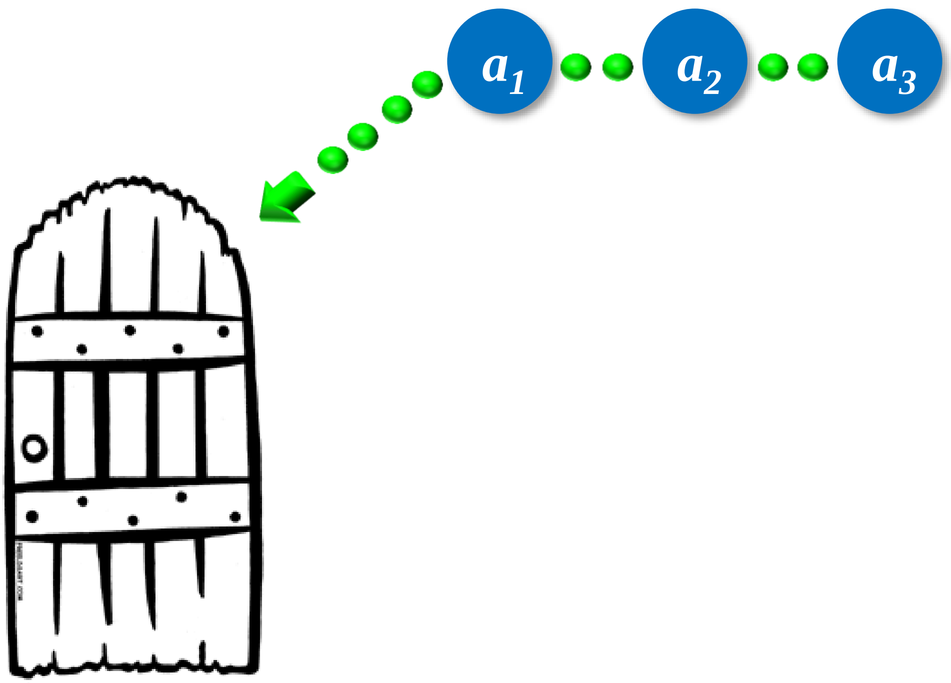
$$Sh_i(v) = \sum_{C \subseteq A \setminus \{a_i\}} \frac{|C|!\,(|A| - |C| - 1)!}{|A|!} [v(C \cup \{a_i\}) - v(C)]$$

# Taxonomy of Solutions



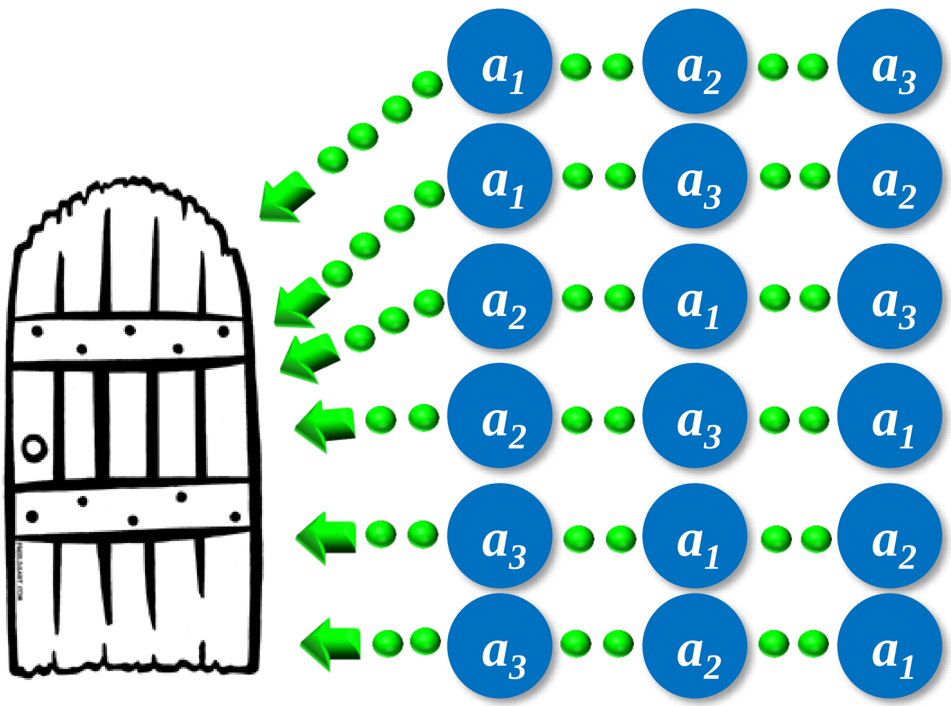
# Shapley Value – Intuition

*by Shapley*



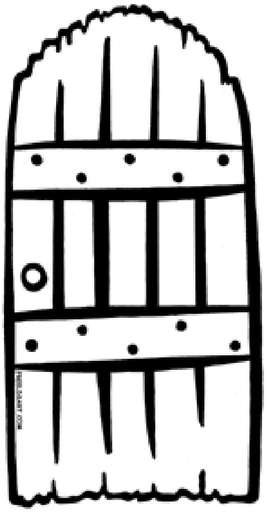

| $a_1$ | $a_2$ | $a_3$ | $a_1 a_2$ | $a_1 a_3$ | $a_2 a_3$ | $a_1 a_2 a_3$ |
|---|---|---|---|---|---|---|
| 5 | 5 | 5 | 12 | 12 | 12 | 24 |

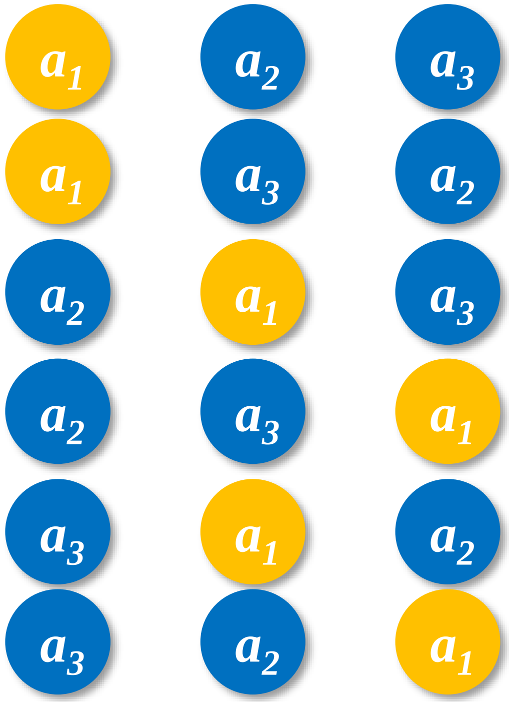
# Shapley Value – Intuition

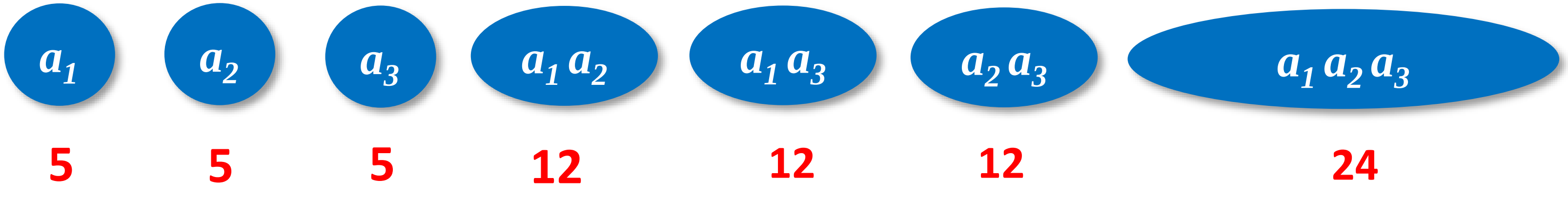



| $a_1$ | $a_2$ | $a_3$ | $a_1 a_2$ | $a_1 a_3$ | $a_2 a_3$ | $a_1 a_2 a_3$ |
|---|---|---|---|---|---|---|
| 5 | 5 | 5 | 12 | 12 | 12 | 24 |

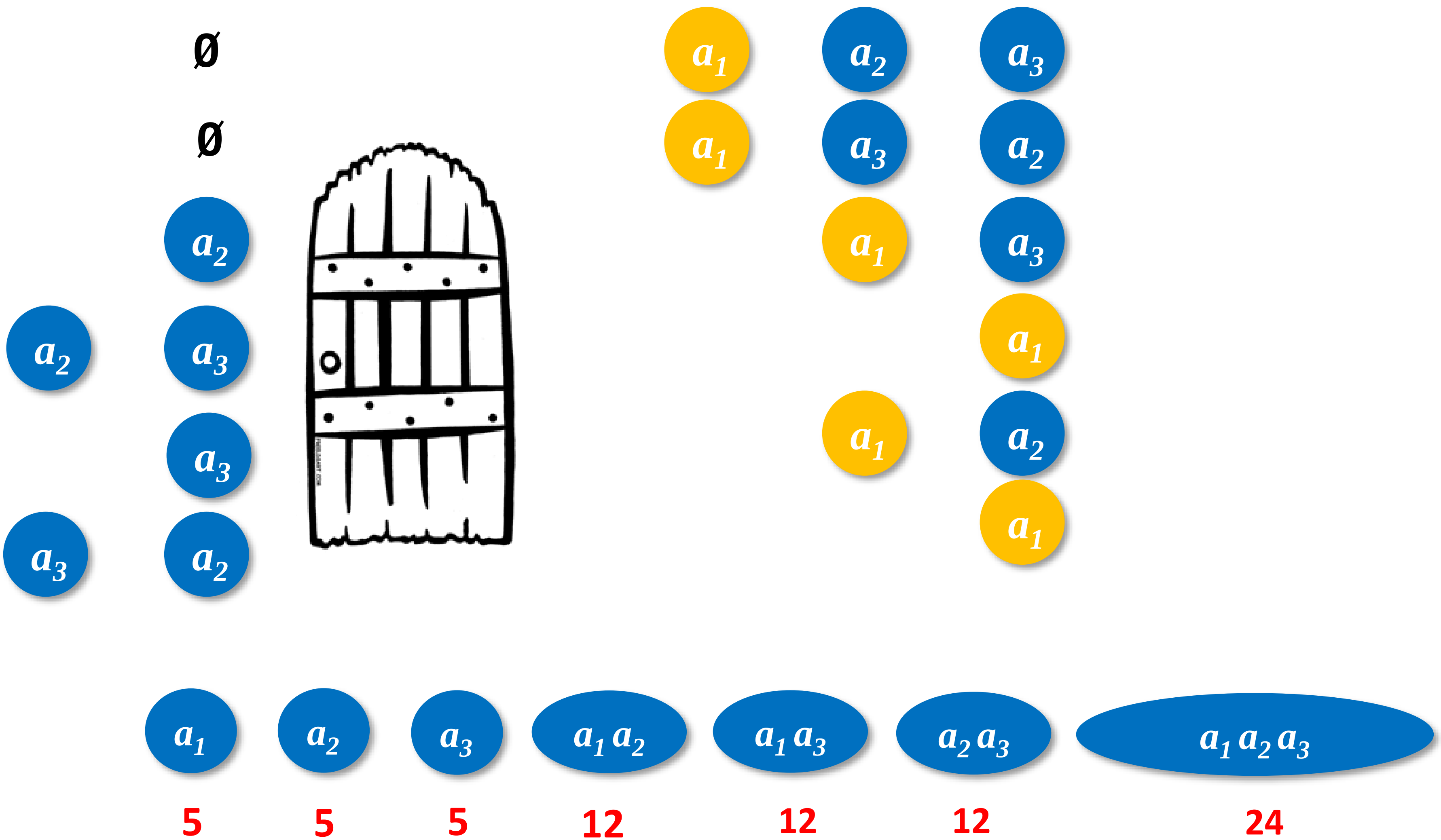
# Shapley Value – Intuition



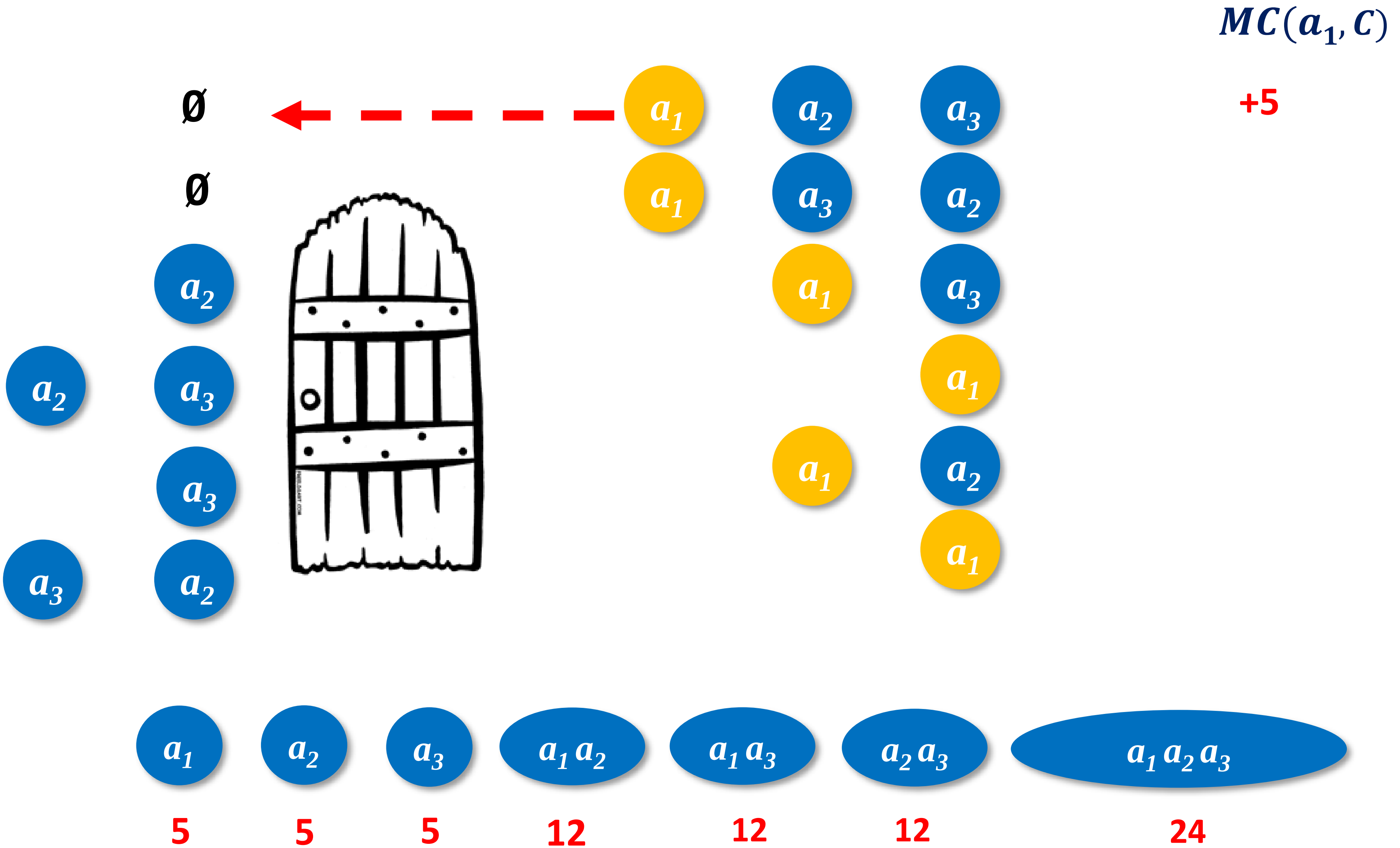


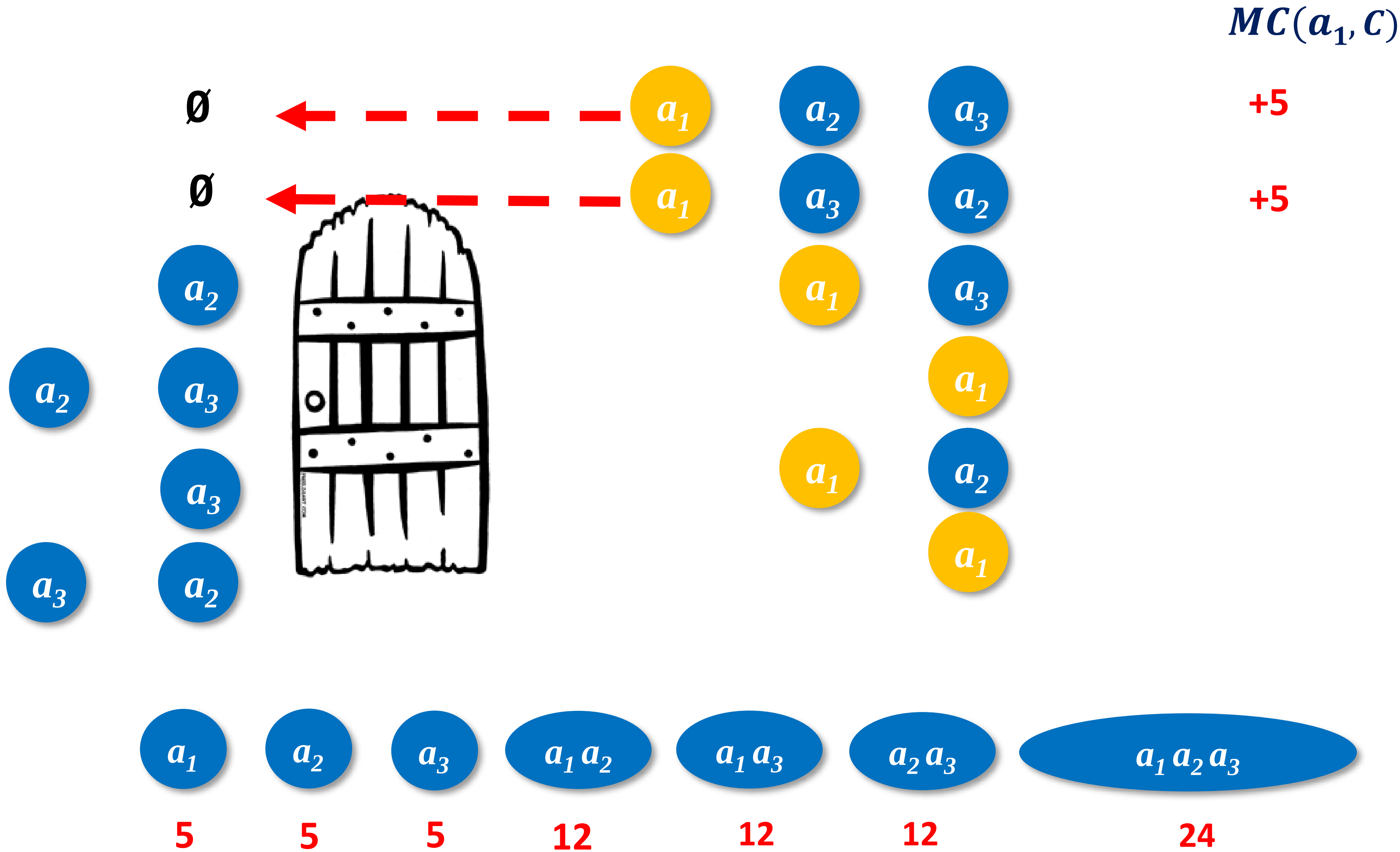
# Shapley Value – Intuition



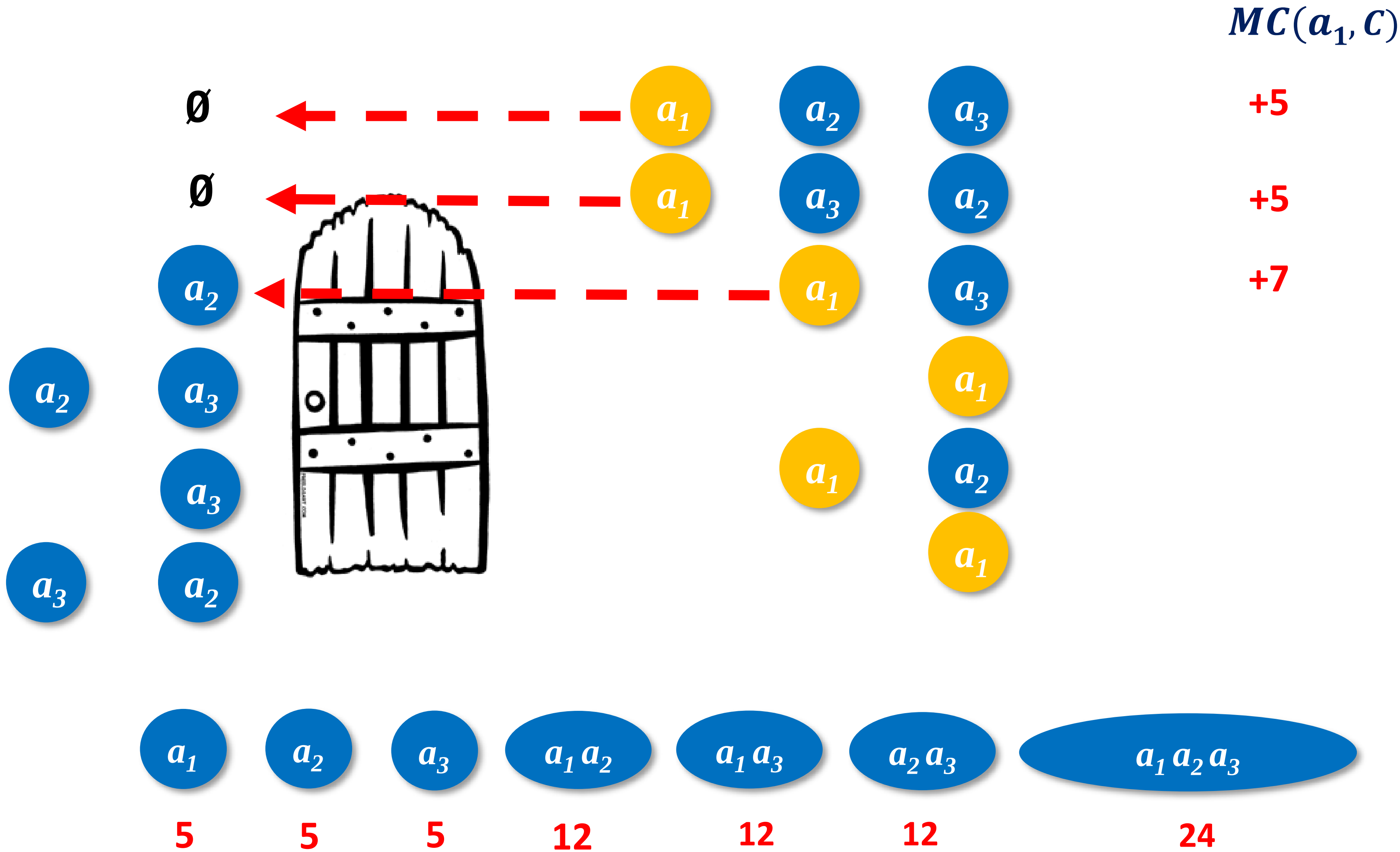
# Shapley Value – Intuition



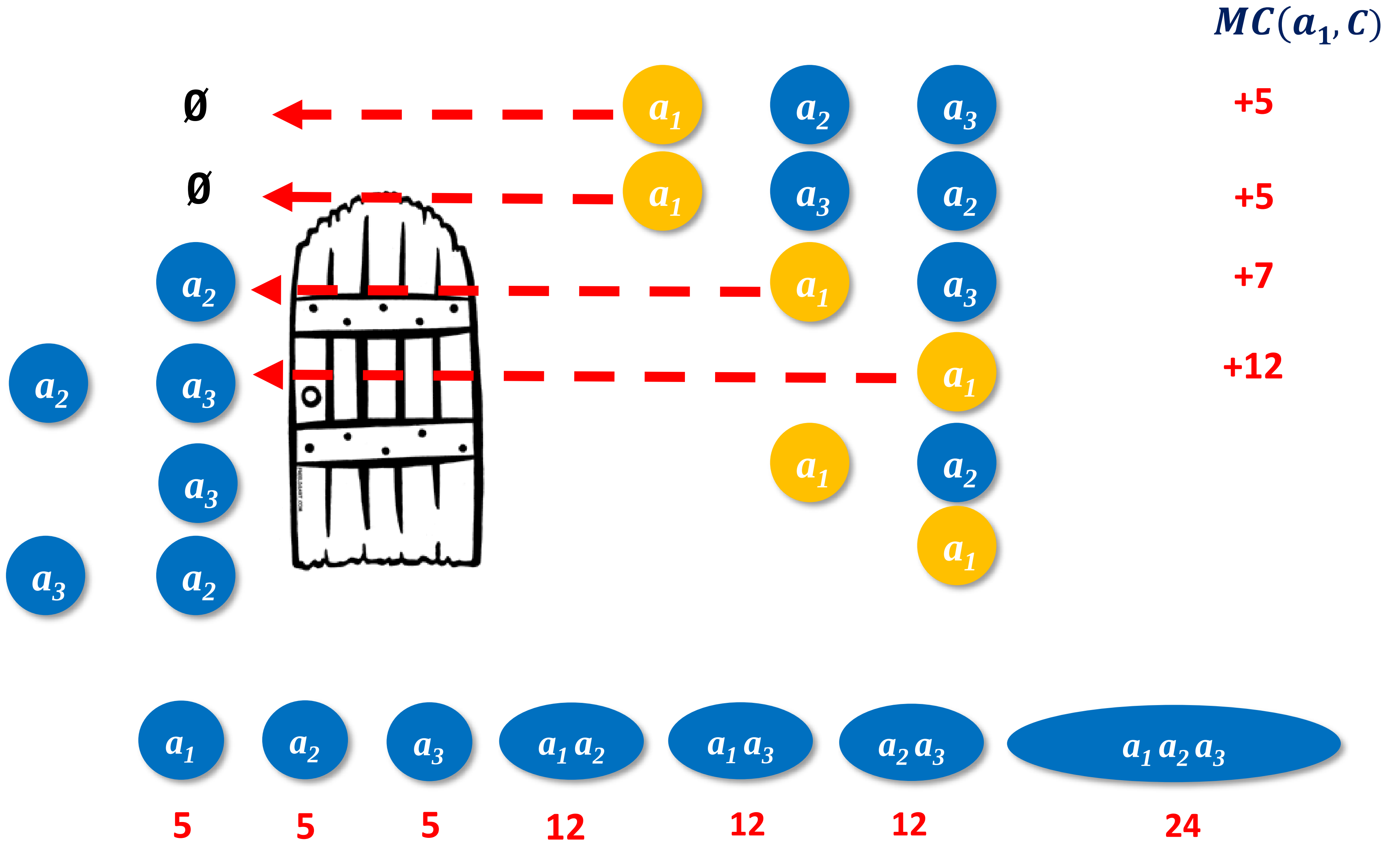
# Shapley Value – Intuition



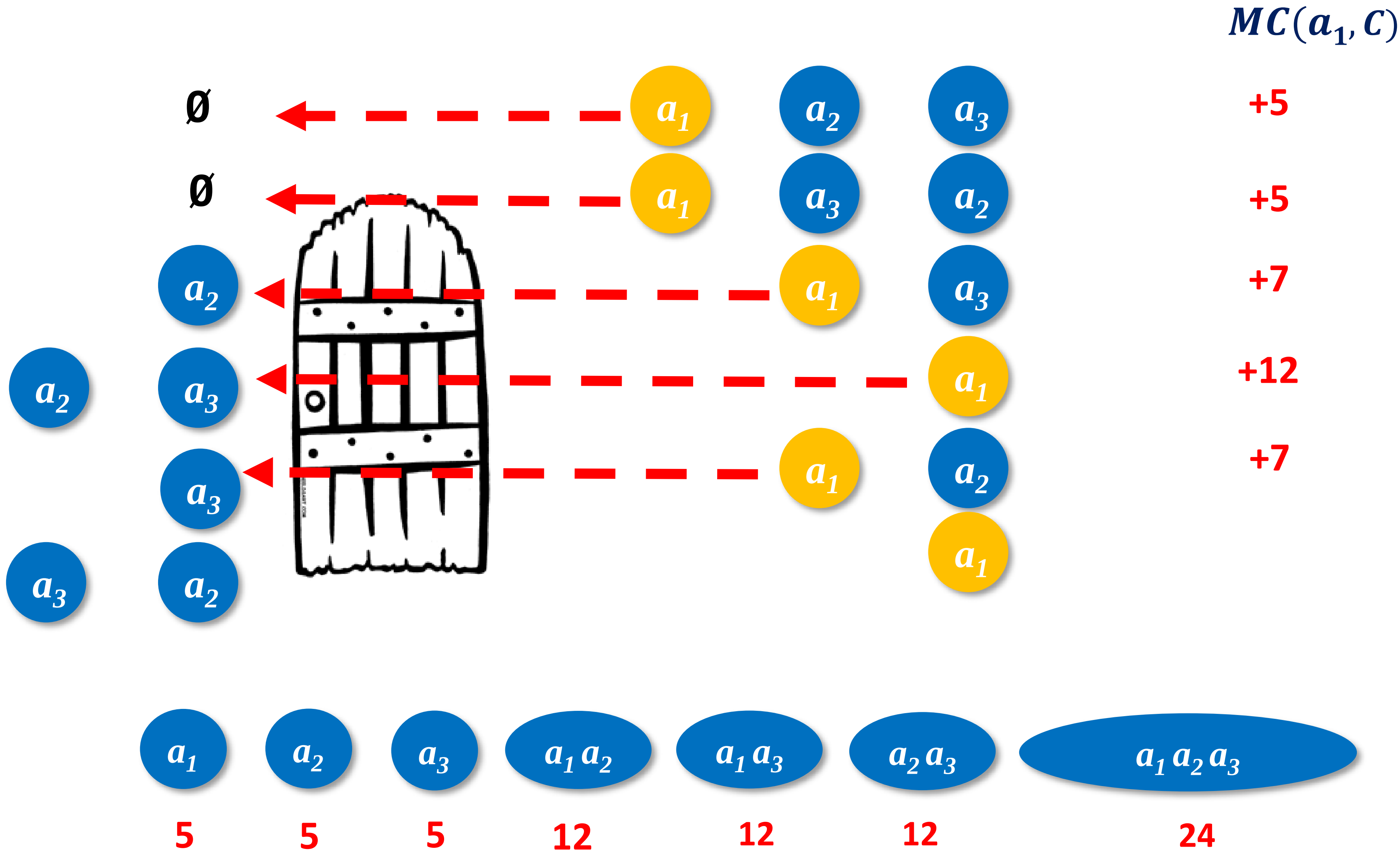
# Shapley Value – Intuition



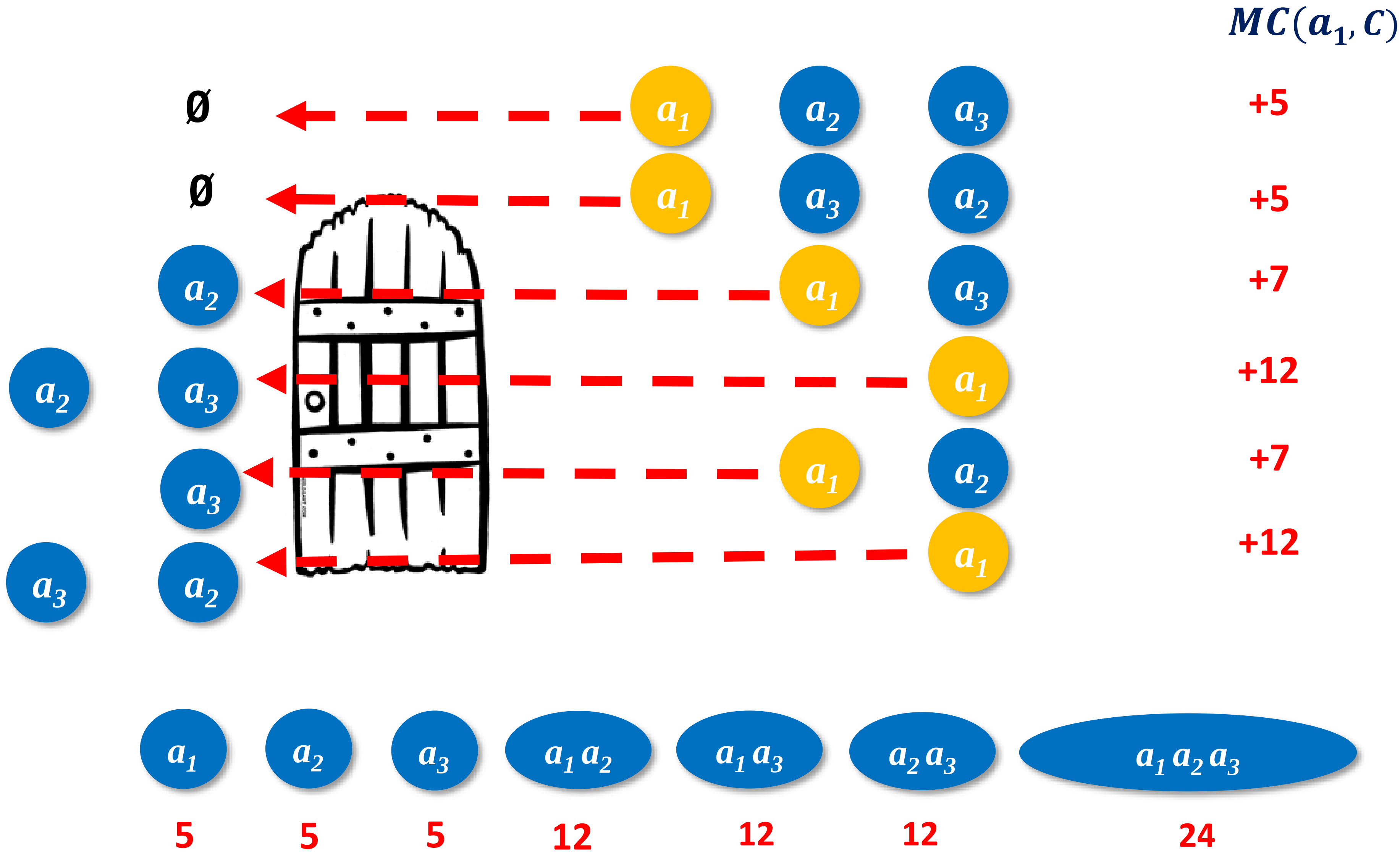
# Shapley Value – Intuition



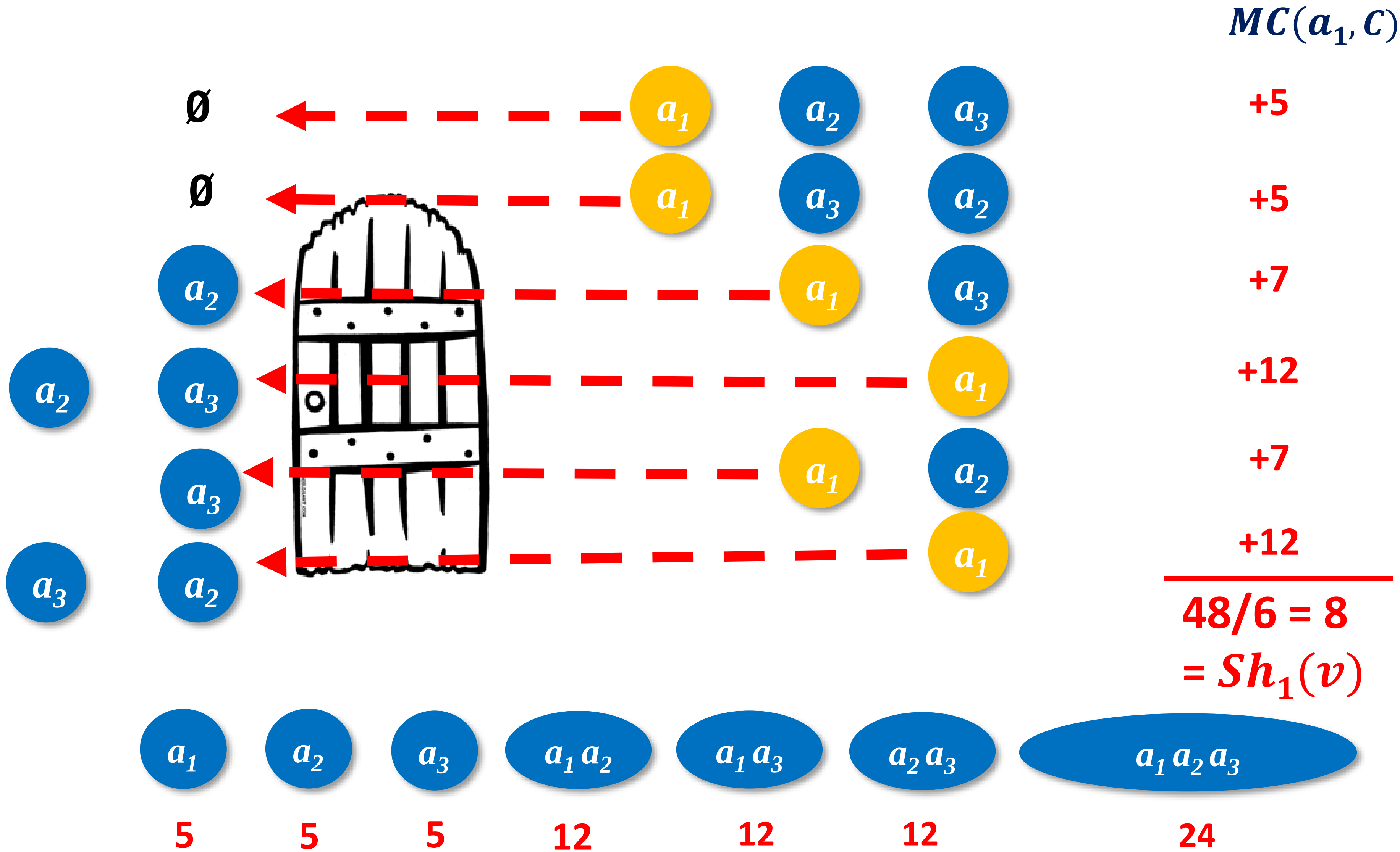
# Shapley Value – Intuition



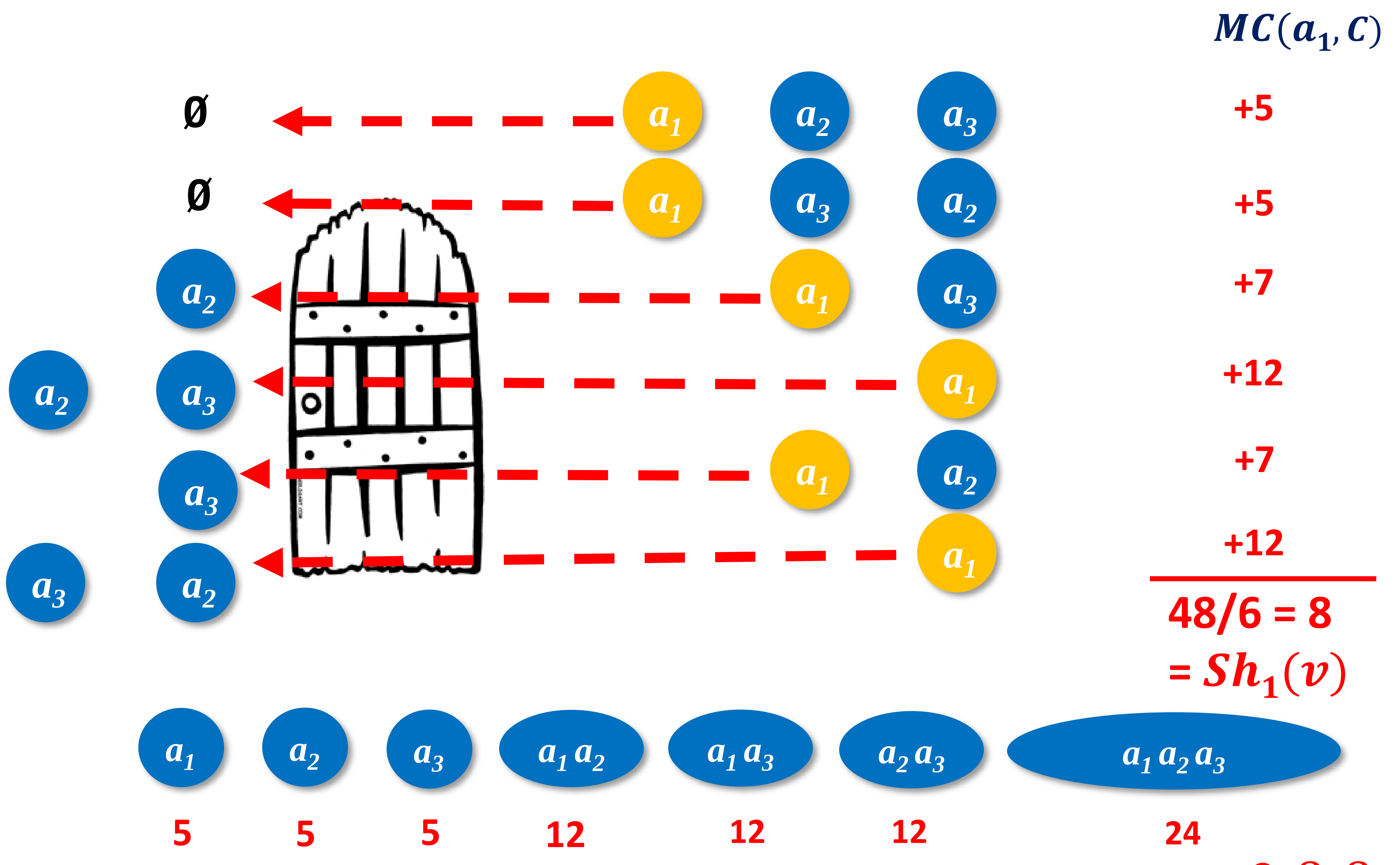
# Shapley Value – Intuition



# Shapley Value – Intuition



# Shapley Value – Intuition



# Comparison

**Shapley value – weighted average marginal contribution of a player to all coalitions**

$2^{|A|}$

$$Sh_i(v) = \sum_{C \subseteq A \setminus \{a_i\}} \frac{|C|!\,(|A| - |C| - 1)!}{|A|!} [v(C \cup \{a_i\}) - v(C)]$$

**Symmetry, null player, additivity, efficiency**

**simple average marginal contribution of a player to all coalitions**

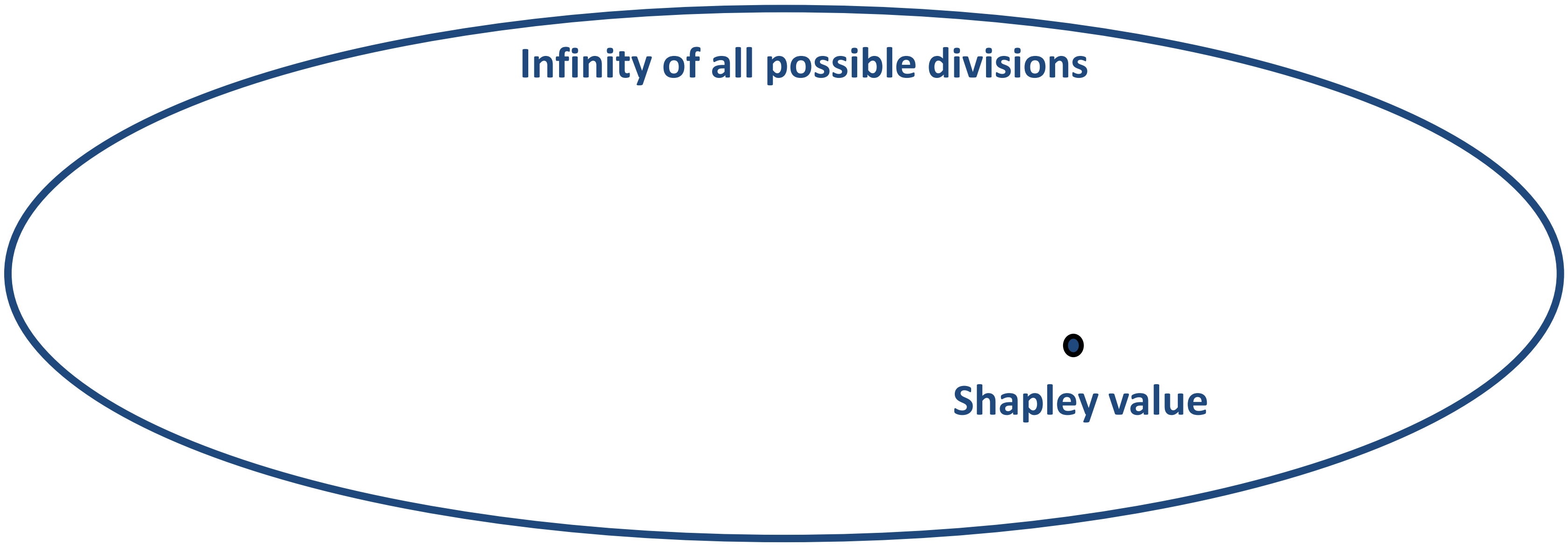
$2^{|A|}$

$$Bh_i(v) = \frac{1}{2^{|A|-1}} \sum_{C \subseteq A \setminus \{a_i\}} (v(C \cup \{a_i\}) - v(C))$$

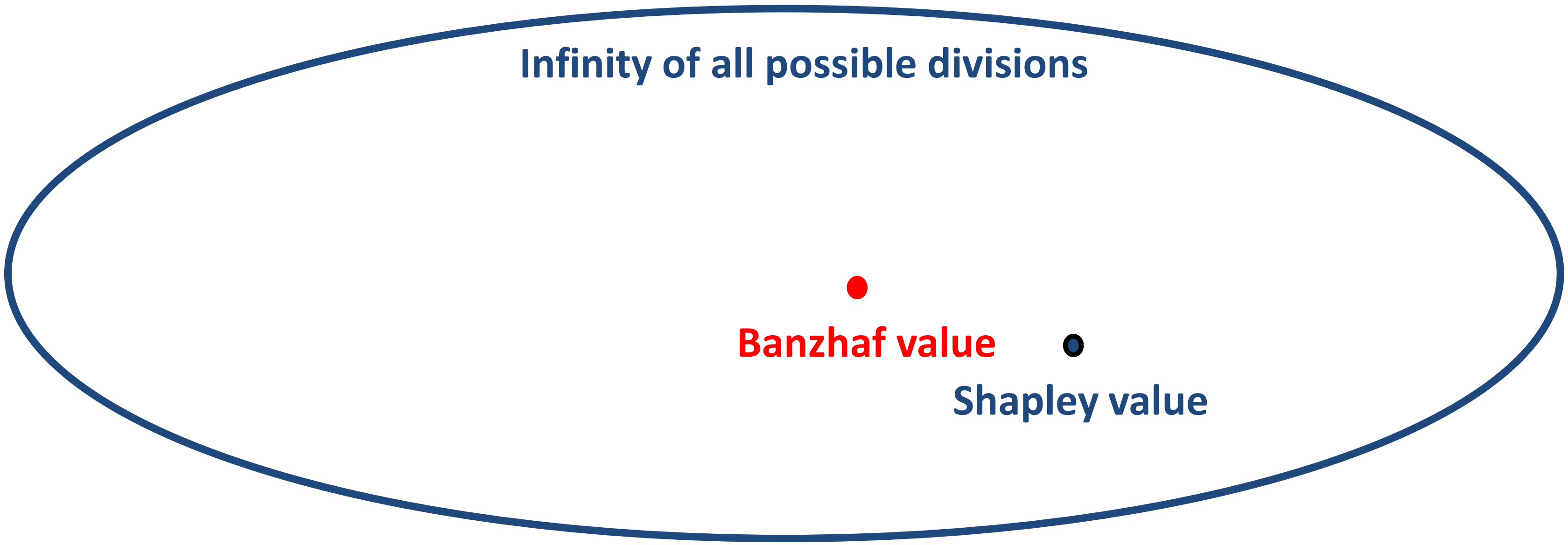
**Banzhaf value**

**Symmetry, null player, additivity**

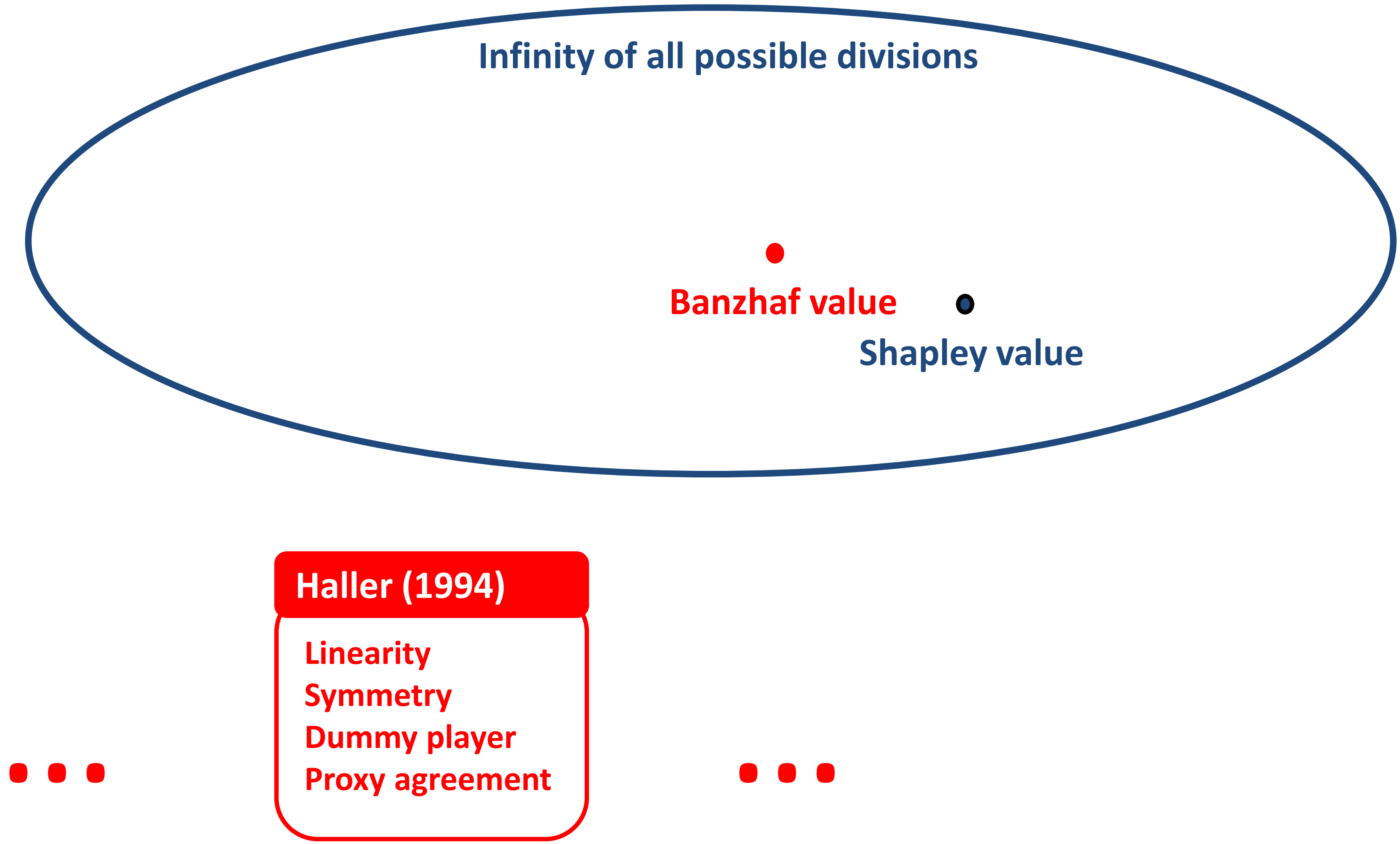
# Taxonomy of Solutions



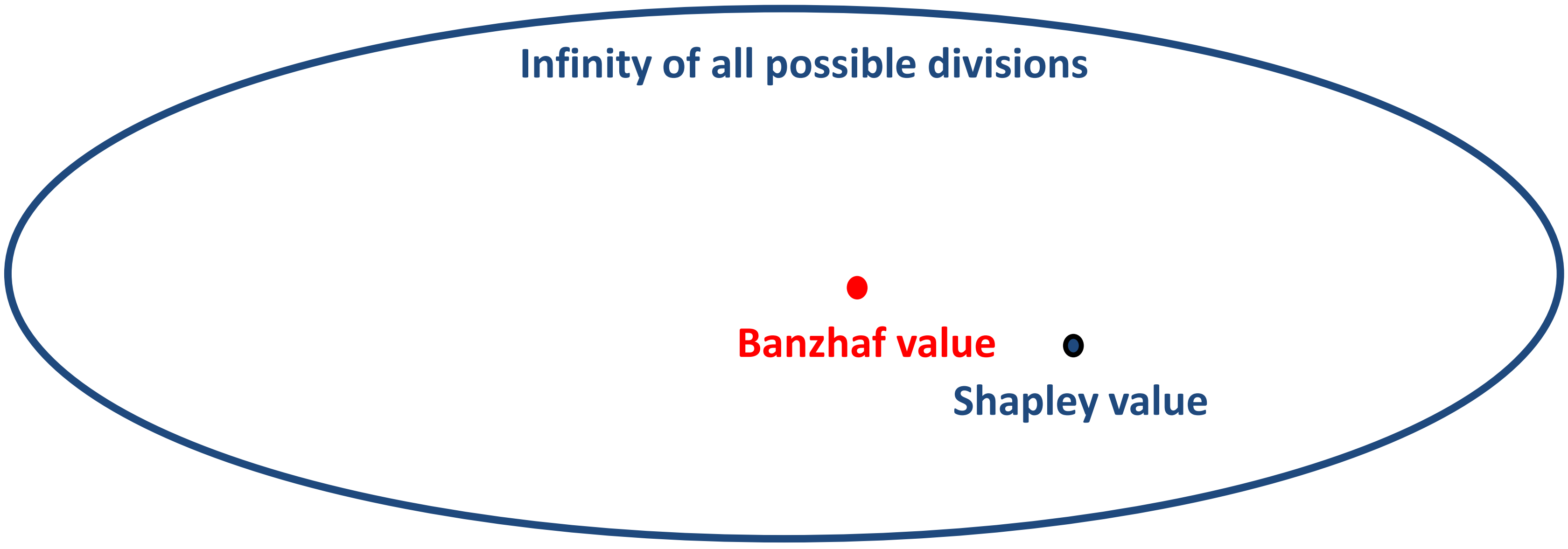
# Taxonomy of Solutions



# Taxonomy of Solutions

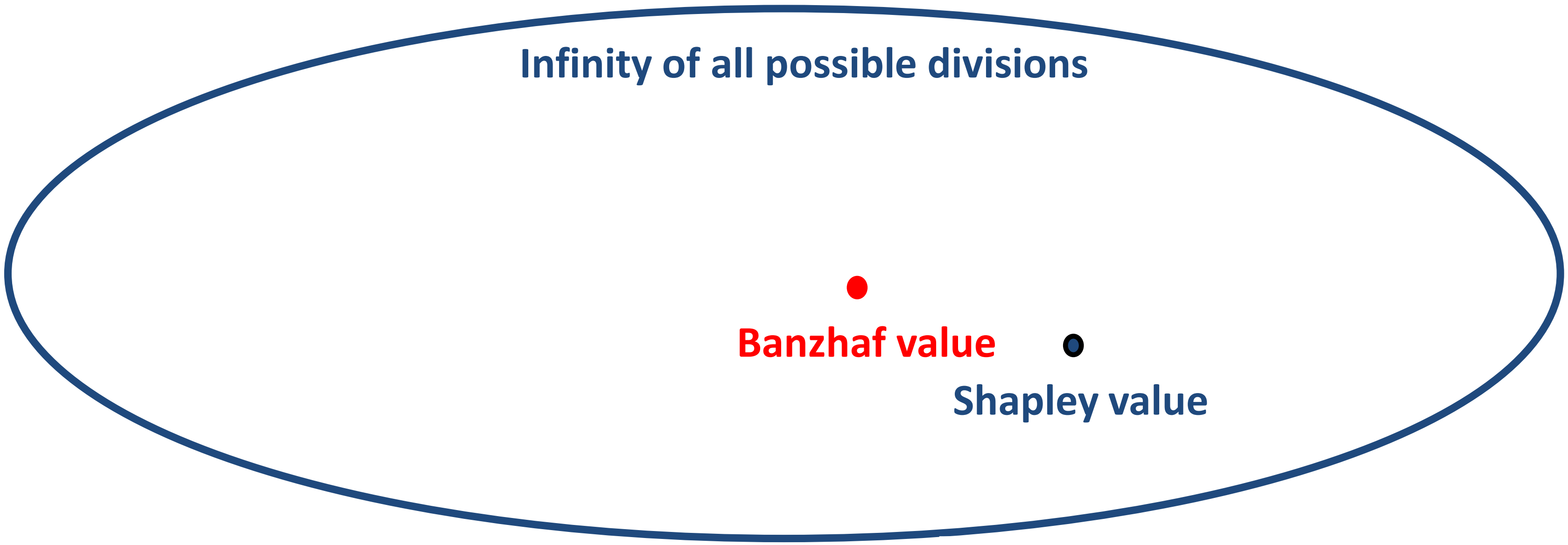


# Taxonomy of Solutions

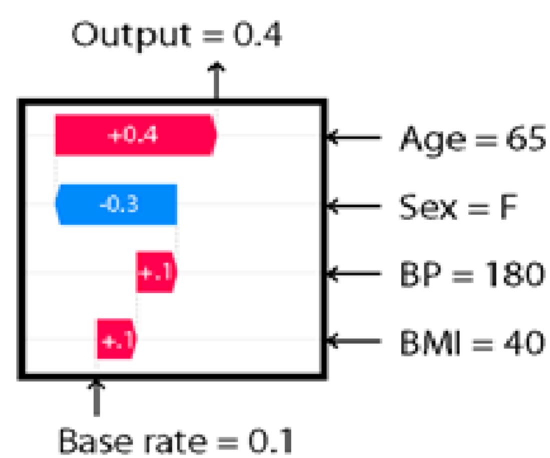



But what about **efficiency**?

# Taxonomy of Solutions




But what about **efficiency**?




Source: https://shap.readthedocs.io/en/latest/index.html

# The Normalized Banzhaf value

**Normalized Banzhaf value**

$2^{|A|}$

$$Bh_i^N(v) = \frac{Bh_i}{\sum_{j \in A} Bh_j} v(A)$$

# The Normalized Banzhaf value

**Normalized Banzhaf value**

$2^{|A|}$

$$Bh_i^N(v) = \frac{Bh_i}{\sum_{j \in A} Bh_j} v(A)$$

**van den Brink & van der Laan (1998)**

**efficiency**
**null player out**
**additive game property**
**independence of irrelevant permutations**
**proportional proxy agreement**

# The Normalized Banzhaf value

**Normalized Banzhaf value**

$2^{|A|}$

$$Bh_i^N(v) = \frac{Bh_i}{\sum_{j \in A} Bh_j} v(A)$$

**van den Brink & van der Laan (1998)**

efficiency
null player out
additive game property
independence of irrelevant permutations
proportional proxy agreement

Do these axioms sound strange?

# The Normalized Banzhaf value

**Normalized Banzhaf value**

$2^{|A|}$

$$Bh_i^N(v) = \frac{Bh_i}{\sum_{j \in A} Bh_j} v(A)$$

**van den Brink & van der Laan (1998)**

- efficiency
- null player out
- additive game property
- independence of irrelevant permutations
- proportional proxy agreement

**Normalized Banzhaf value**

**van den Brink & van der Laan (1998)**

- efficiency
- null player out
- additive game property
- independence of irrelevant permutations
- coalitional unanimity proxy agreement

**Shapley value**

# The Normalized Banzhaf value

**Normalized Banzhaf value**

$2^{|A|}$

$$Bh_i^N(v) = \frac{Bh_i}{\sum_{j \in A} Bh_j} v(A)$$

**van den Brink & van der Laan (1998)**

- efficiency
- null player out
- additive game property
- independence of irrelevant permutations
- proportional proxy agreement

**van den Brink & van der Laan (1998)**

- efficiency
- null player out
- additive game property
- independence of irrelevant permutations
- coalitional unanimity proxy agreement

**Normalized Banzhaf value**

**Shapley value**

# Banzhaf vs. Shapley: Another conceptual issue

**Why are supposed to weight the contribution of each feature with the total number of orderings of the present as well as the absent features?**

# Banzhaf vs. Shapley: Another conceptual issue

**Why are supposed to weight the contribution of each feature with the total number of orderings of the present as well as the absent features?**

**It is not obvious why feature vector (man; 40) is different from (40;man), and why this should matter for feature importance.**

# Banzhaf vs. Shapley: Another conceptual issue

Why are supposed to weight the contribution of each feature with the total number of orderings of the present as well as the absent features?

It is not obvious why feature vector (man; 40) is different from (40;man), and why this should matter for feature importance.

From the literature on weighted voting games we learn that:

- Even axioms that seem to be the most basic ones can lead to paradoxes

# Banzhaf vs. Shapley: Another conceptual issue

Why are supposed to weight the contribution of each feature with the total number of orderings of the present as well as the absent features?

It is not obvious why feature vector (man; 40) is different from (40;man), and why this should matter for feature importance.

From the literature on weighted voting games we learn that:

- Even axioms that seem to be the most basic ones can lead to paradoxes
- Thus, some authors recommend to use probabilistic approach when choosing between Shapley and Banzhaf value:
  - use Shapley when the order matters
  - use Banzhaf when it does not

# Shapley & Banzhaf Values – Computational Challenge

|A|!

$$Sh_i(v) = \frac{1}{|A|!} \sum_{\text{all } \pi} [v(C_\pi(i) \cup \{a_i\}) - v(C_\pi(i))]$$

$2^{|A|}$

$$Sh_i(v) = \sum_{C \subseteq A \setminus \{a_i\}} \frac{|C|!\,(|A| - |C| - 1)!}{|A|!} [v(C \cup \{a_i\}) - v(C)]$$

$2^{|A|}$

$$Bh_i(v) = \frac{1}{2^{|A|-1}} \sum_{C \subseteq A \setminus \{a_i\}} (v(C \cup \{a_i\}) - v(C))$$

➔ **Computational Challenge** 🡐 **?**

# Plan of the Talk

1. Values in Cooperative Game Theory
2. Our algorithm for the Banzhaf value vs. TreeSHAP
3. Advantages of the Banzhaf value for tree models - experimental analysis

# Our Key Algorithmic Result

| | Shapley value | Banzhaf value |
|---|---|---|
| **TreeSHAP**<br>**Lundberg et al.** | $O(TLD^2 + n)$ | ✗ |
| **Our algorithms** | | |

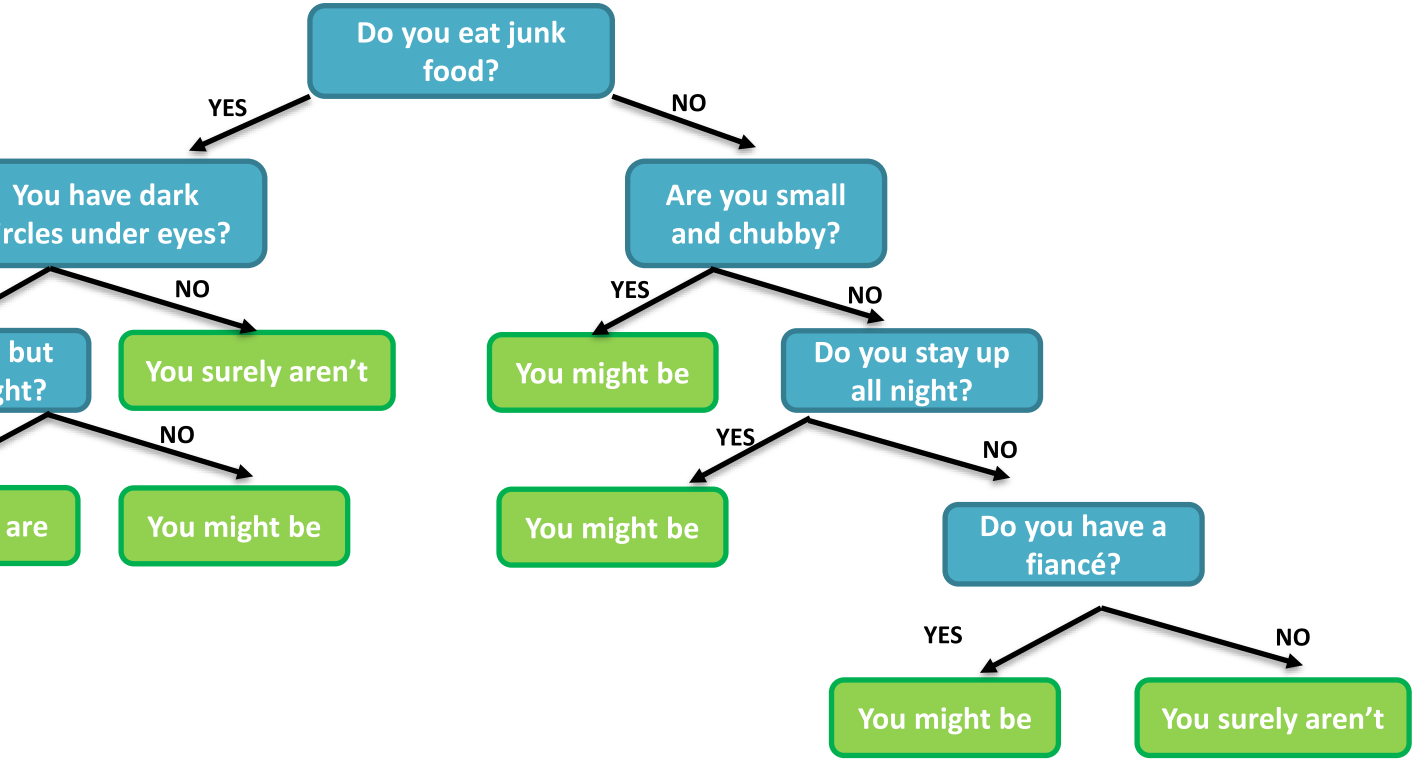
# Our Key Algorithmic Result

| | Shapley value | Banzhaf value |
|---|---|---|
| TreeSHAP<br>Lundberg et al. | $O(TLD^2 + n)$ | ✗ |
| Our algorithms | | $O(TL + n)$ |

# Our Key Algorithmic Result

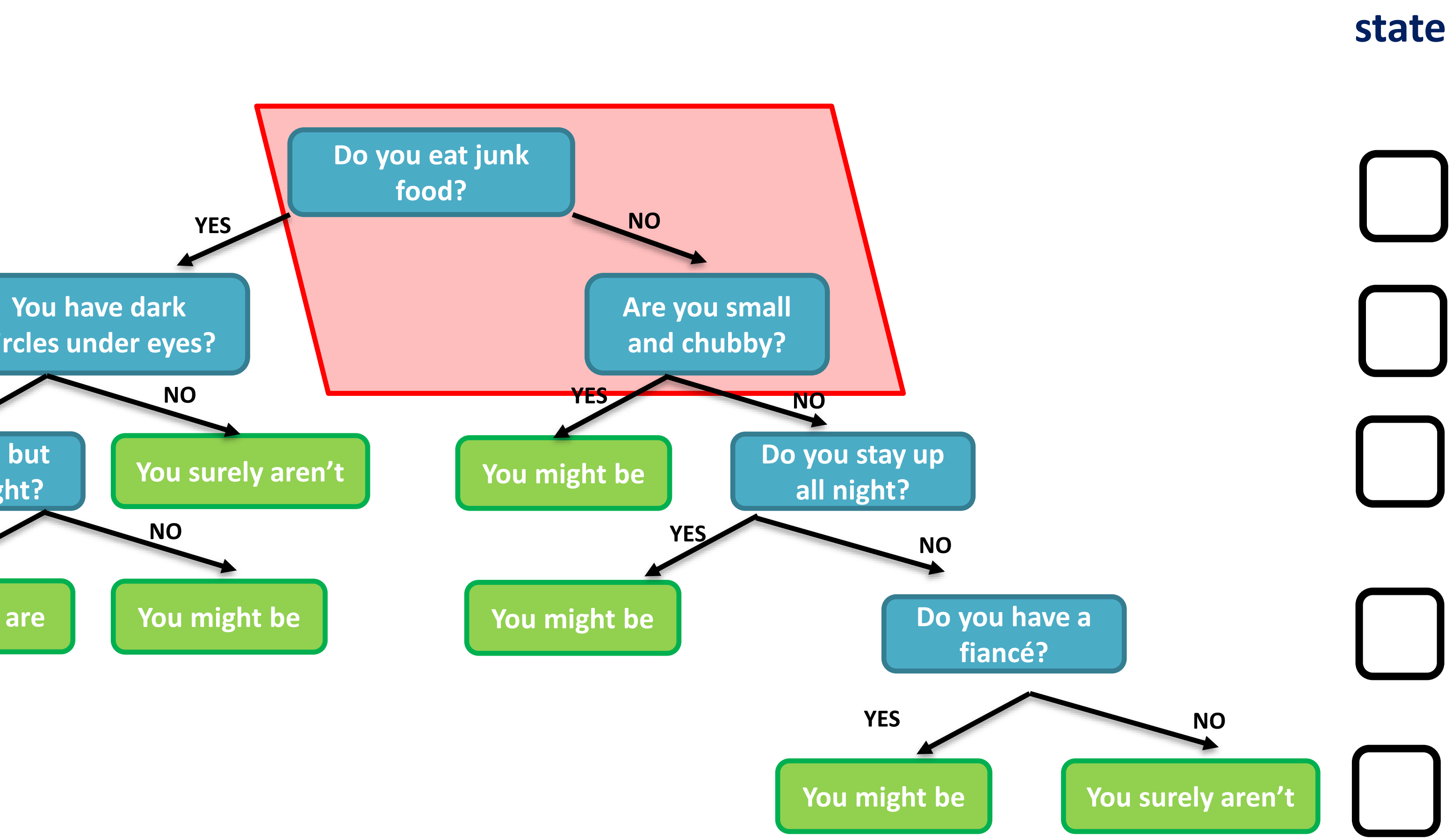
| | Shapley value | Banzhaf value |
|---|---|---|
| TreeSHAP<br>Lundberg et al. | $O(TLD^2 + n)$ | ✗ |
| Our algorithms | $O(TLD + n)$ | $O(TL + n)$ |

# Intuition behind TreeSHAP



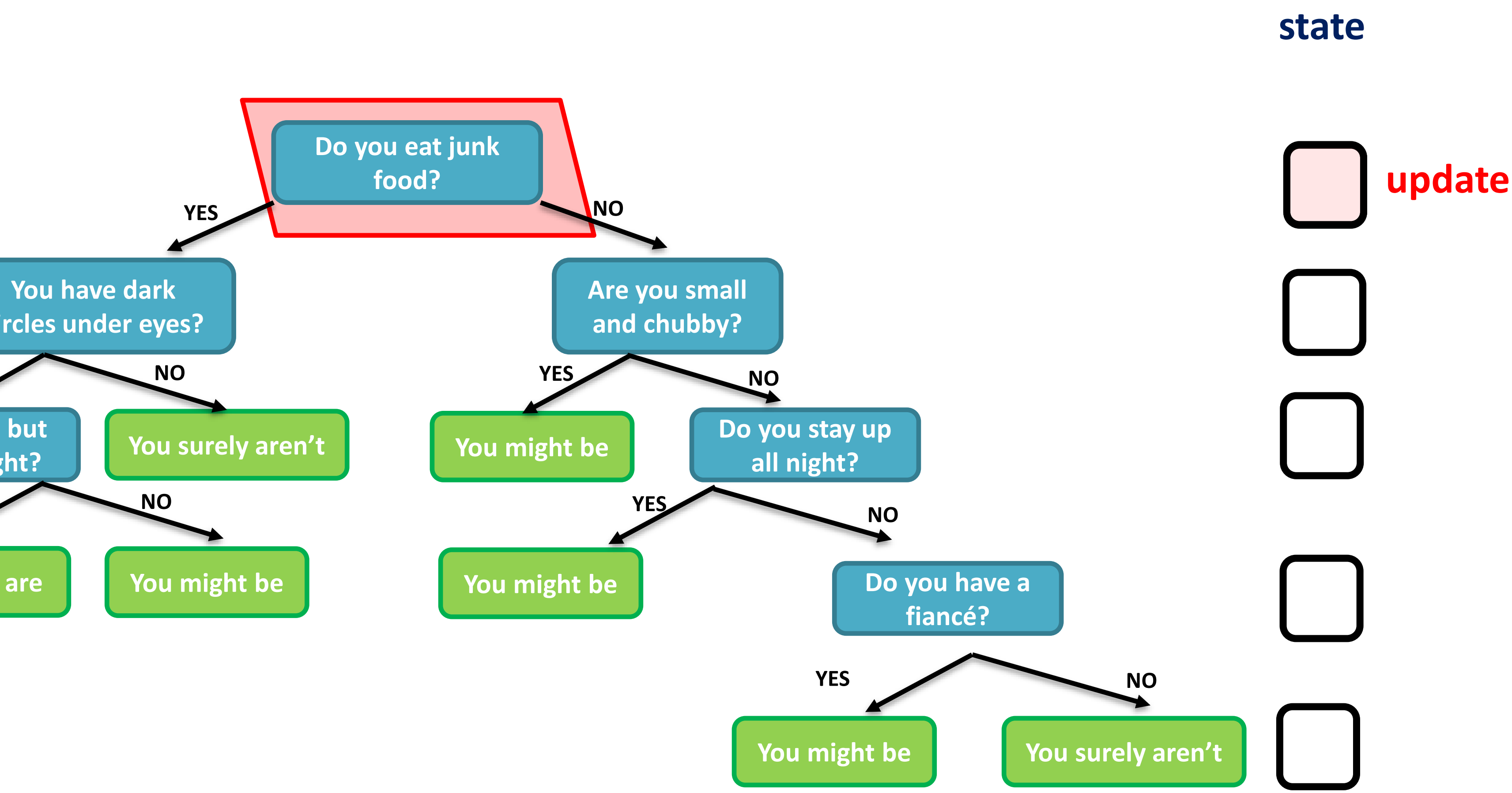

**TreeSHAP is a dynamic programming algorithm.**

# Intuition behind TreeSHAP



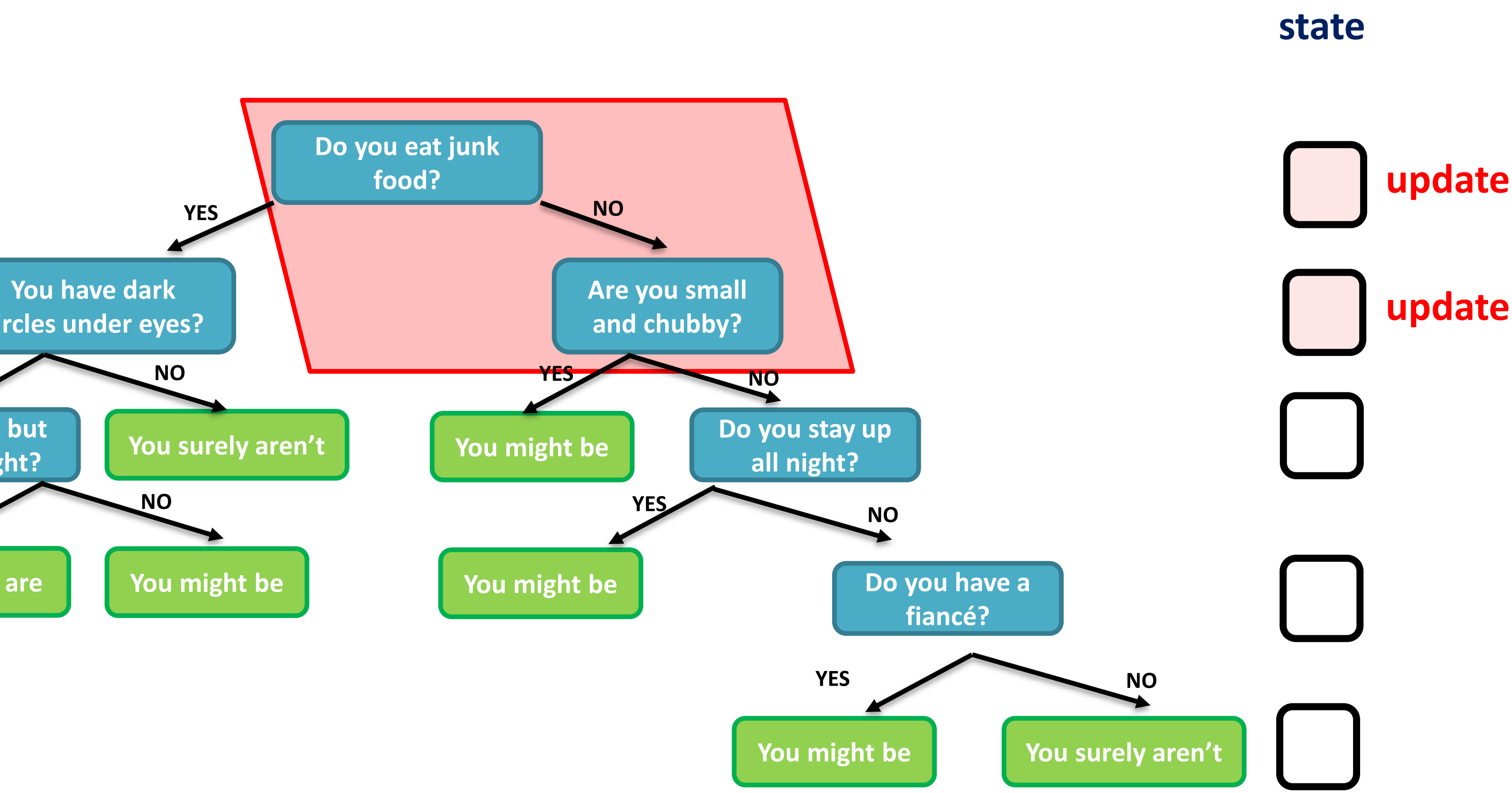

It starts from the root and goes down to the leaves extending the size of the subproblems. With each move it updates some state of up to $D$ values. The update is related to the coefficient $\frac{|C|!(|A|-|C|-1)!}{|A|!}$ in the Shapley value and to sizes of coalitions, in particular.

# Intuition behind TreeSHAP



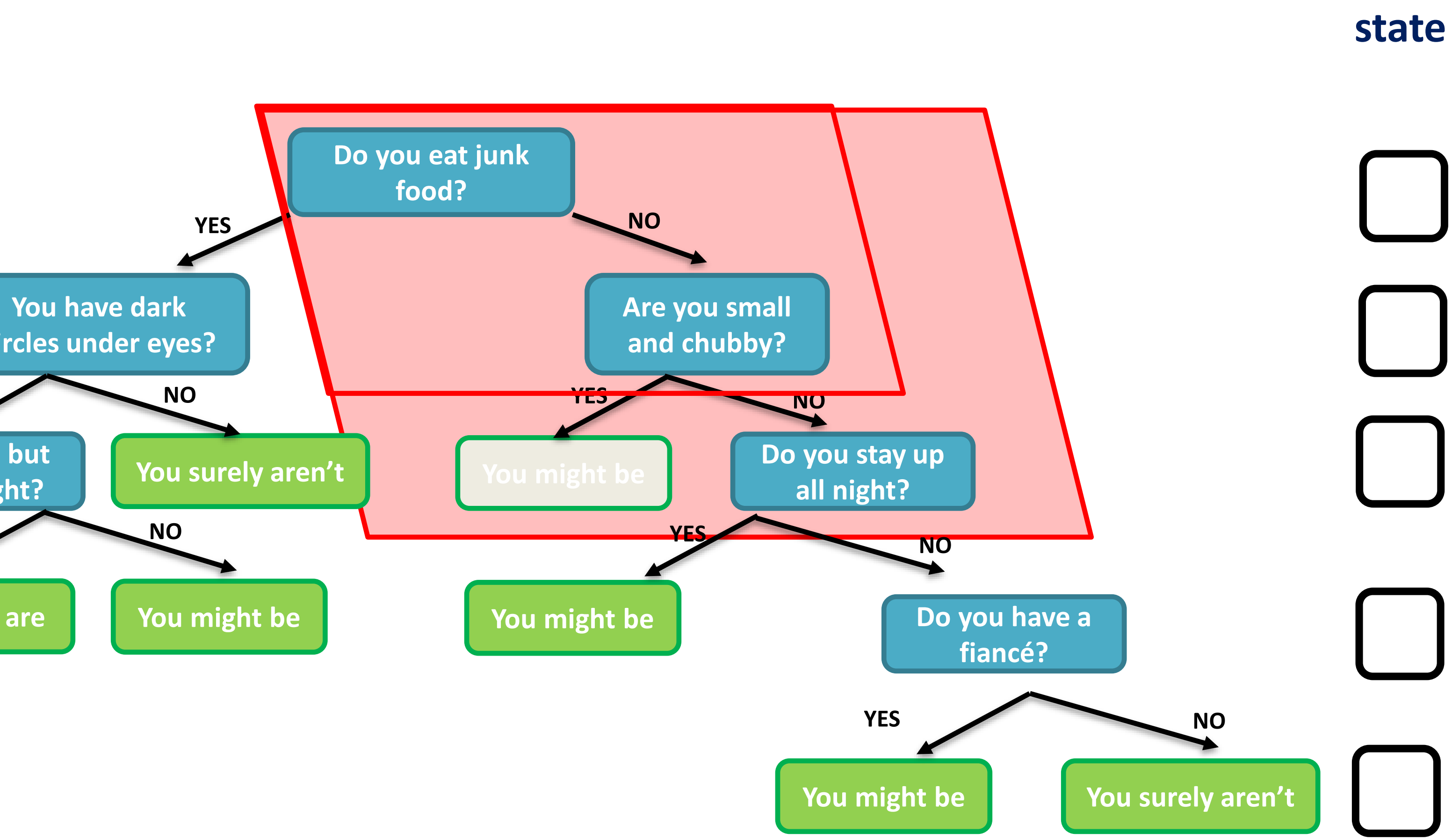

**It starts from the root and goes down to the leaves extending the size of the subproblems. With each move it updates some state of up to $D$ values. The update is related to the coefficient $\frac{|C|!(|A|-|C|-1)!}{|A|!}$ in the Shapley value and to sizes of coalitions, in particular.**

# Intuition behind TreeSHAP



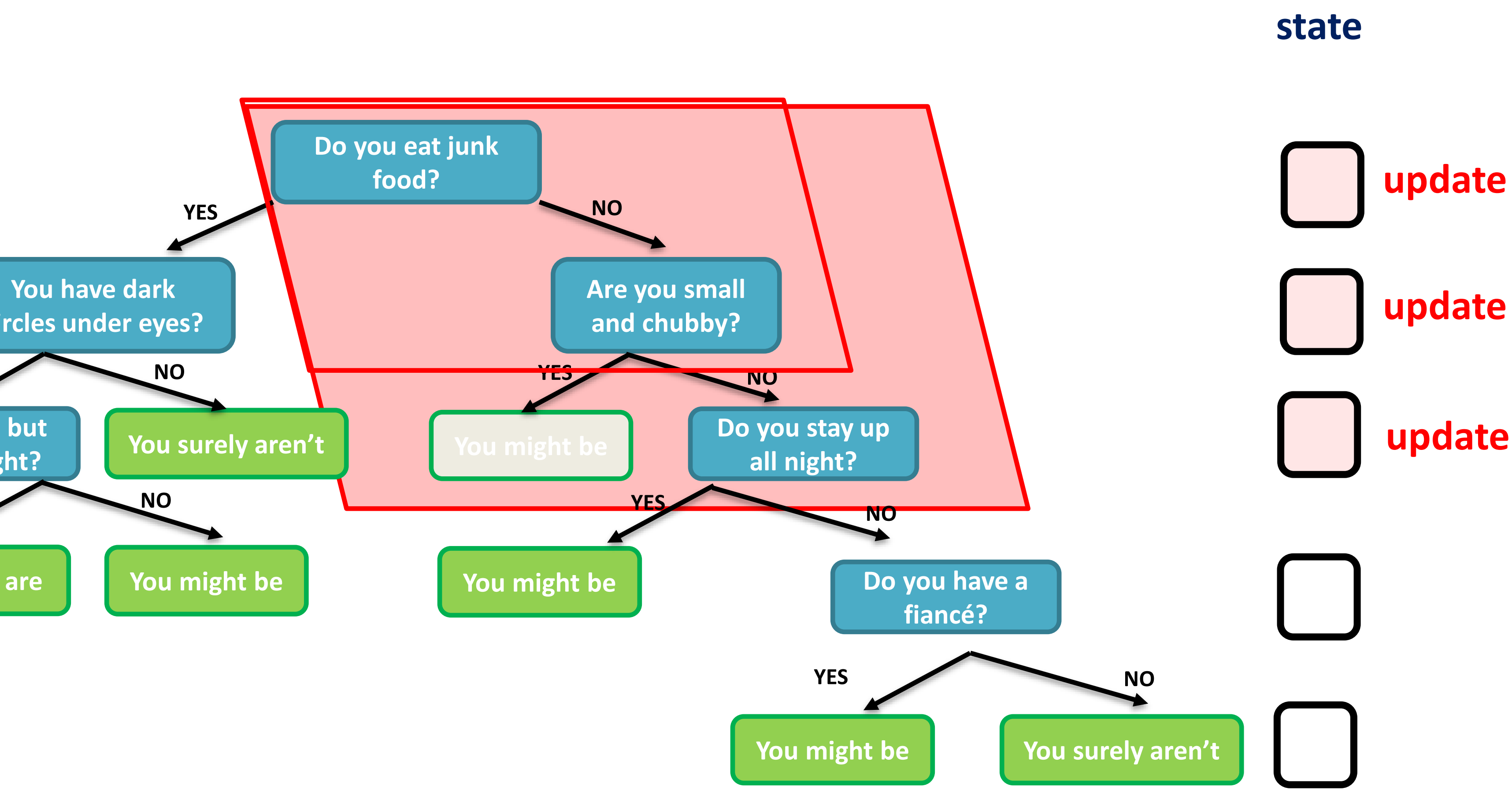

**It starts from the root and goes down to the leaves extending the size of the subproblems. With each move it updates some state of up to $D$ values. The update is related to the coefficient $\frac{|C|!(|A|-|C|-1)!}{|A|!}$ in the Shapley value and to sizes of coalitions, in particular.**

# Intuition behind TreeSHAP



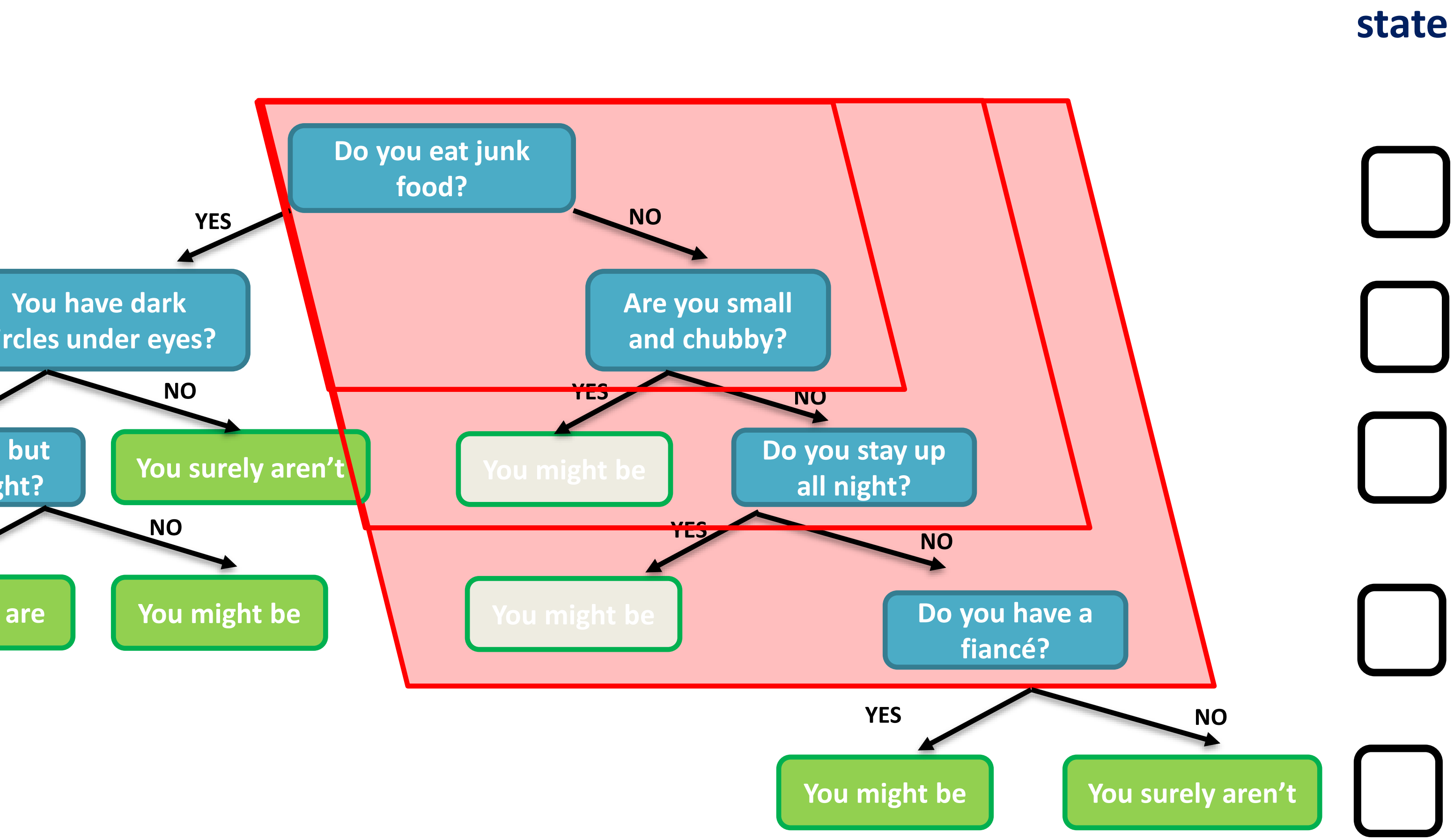

It starts from the root and goes down to the leaves extending the size of the subproblems. With each move it updates some state of up to $D$ values. The update is related to the coefficient $\frac{|C|!(|A|-|C|-1)!}{|A|!}$ in the Shapley value and to sizes of coalitions, in particular.

# Intuition behind TreeSHAP



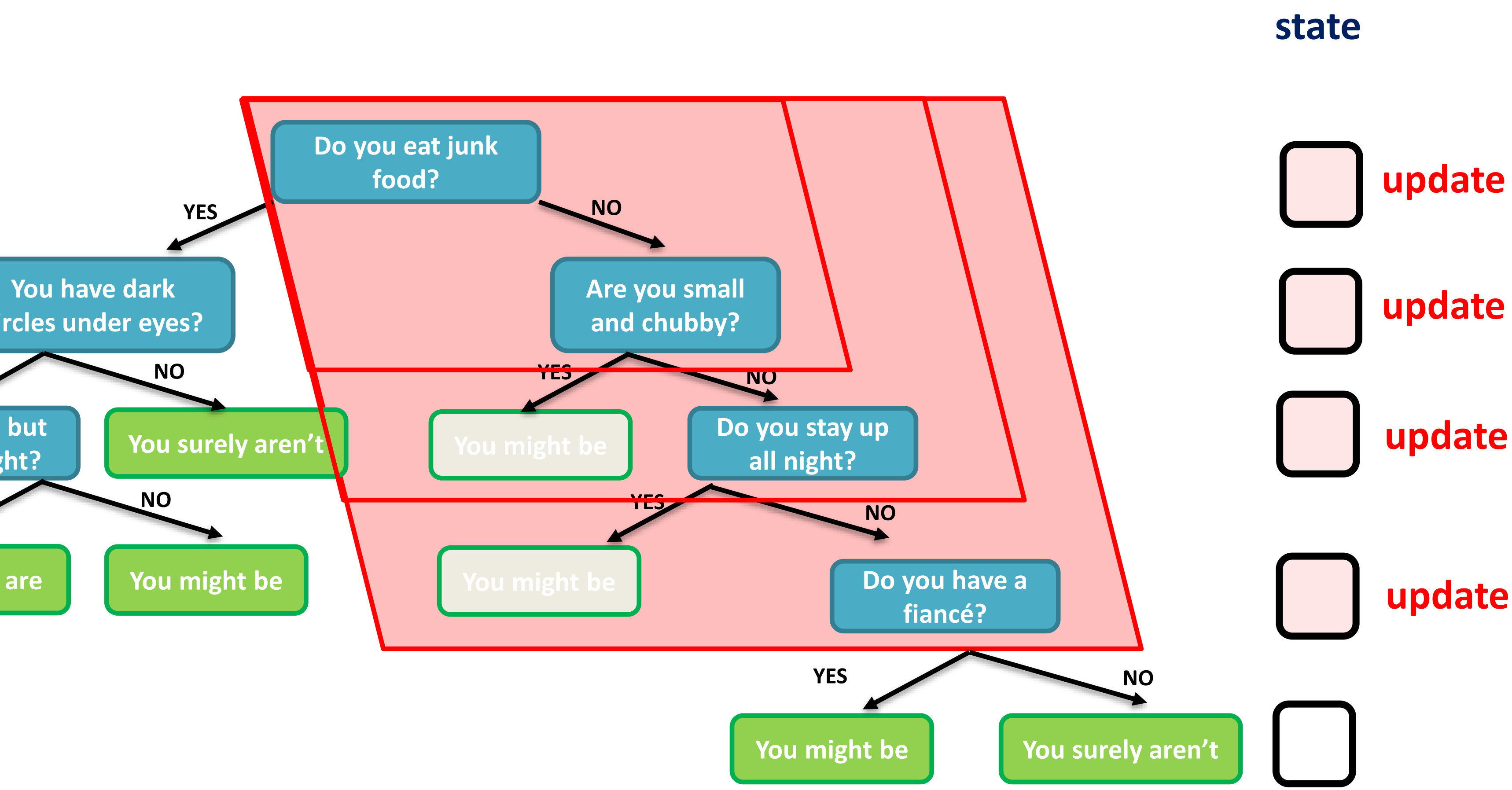

**It starts from the root and goes down to the leaves extending the size of the subproblems. With each move it updates some state of up to $D$ values. The update is related to the coefficient $\frac{|C|!(|A|-|C|-1)!}{|A|!}$ in the Shapley value and to sizes of coalitions, in particular.**

# Intuition behind TreeSHAP



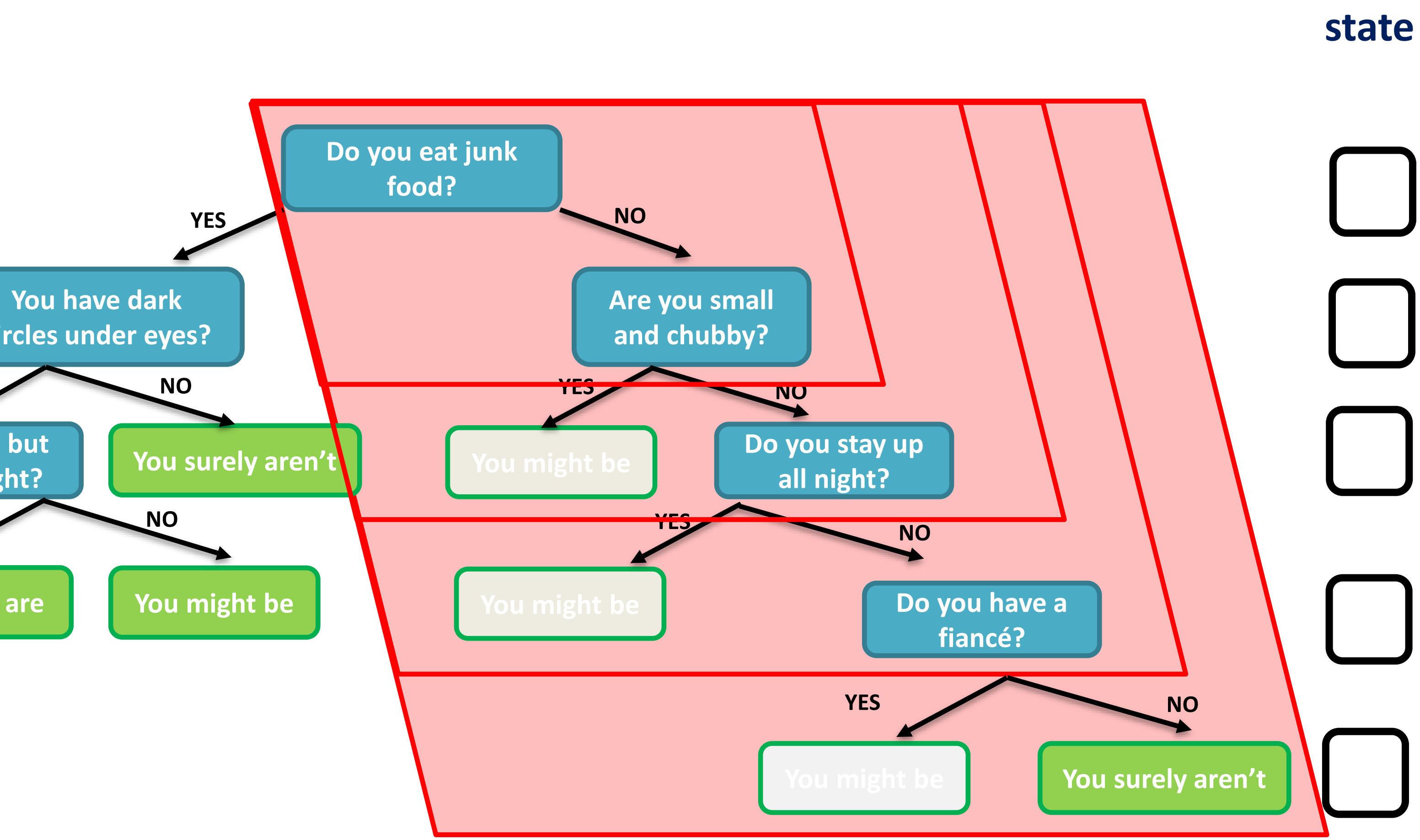

It starts from the root and goes down to the leaves extending the size of the subproblems. With each move it updates some state of up to $D$ values. The update is related to the coefficient $\frac{|C|!(|A|-|C|-1)!}{|A|!}$ in the Shapley value and to sizes of coalitions, in particular.

# Intuition behind TreeSHAP



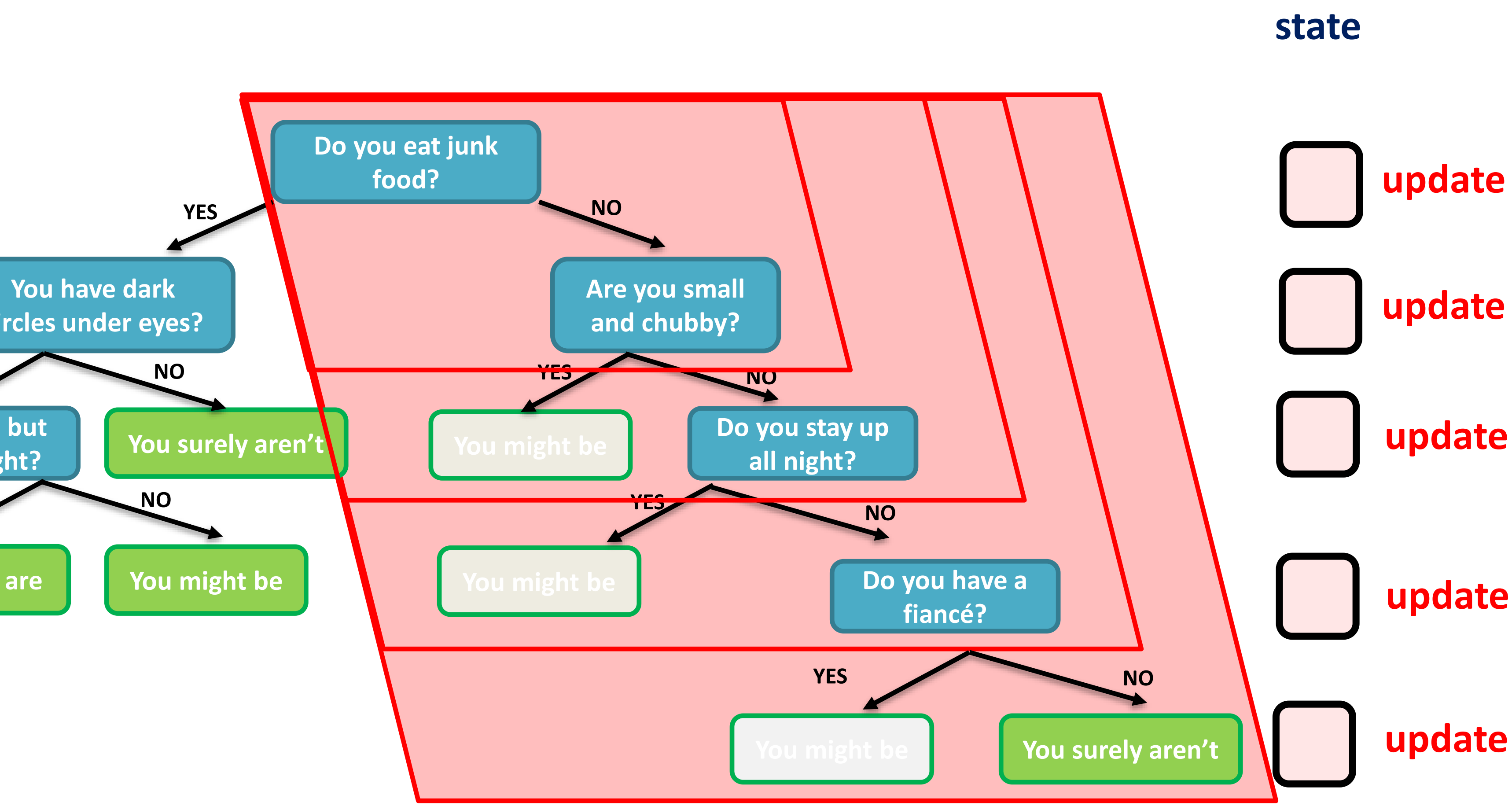

It starts from the root and goes down to the leaves extending the size of the subproblems. With each move it updates some state of up to $D$ values. The update is related to the coefficient $\frac{|C|!(|A|-|C|-1)!}{|A|!}$ in the Shapley value and to sizes of coalitions, in particular.

# Intuition behind TreeSHAP



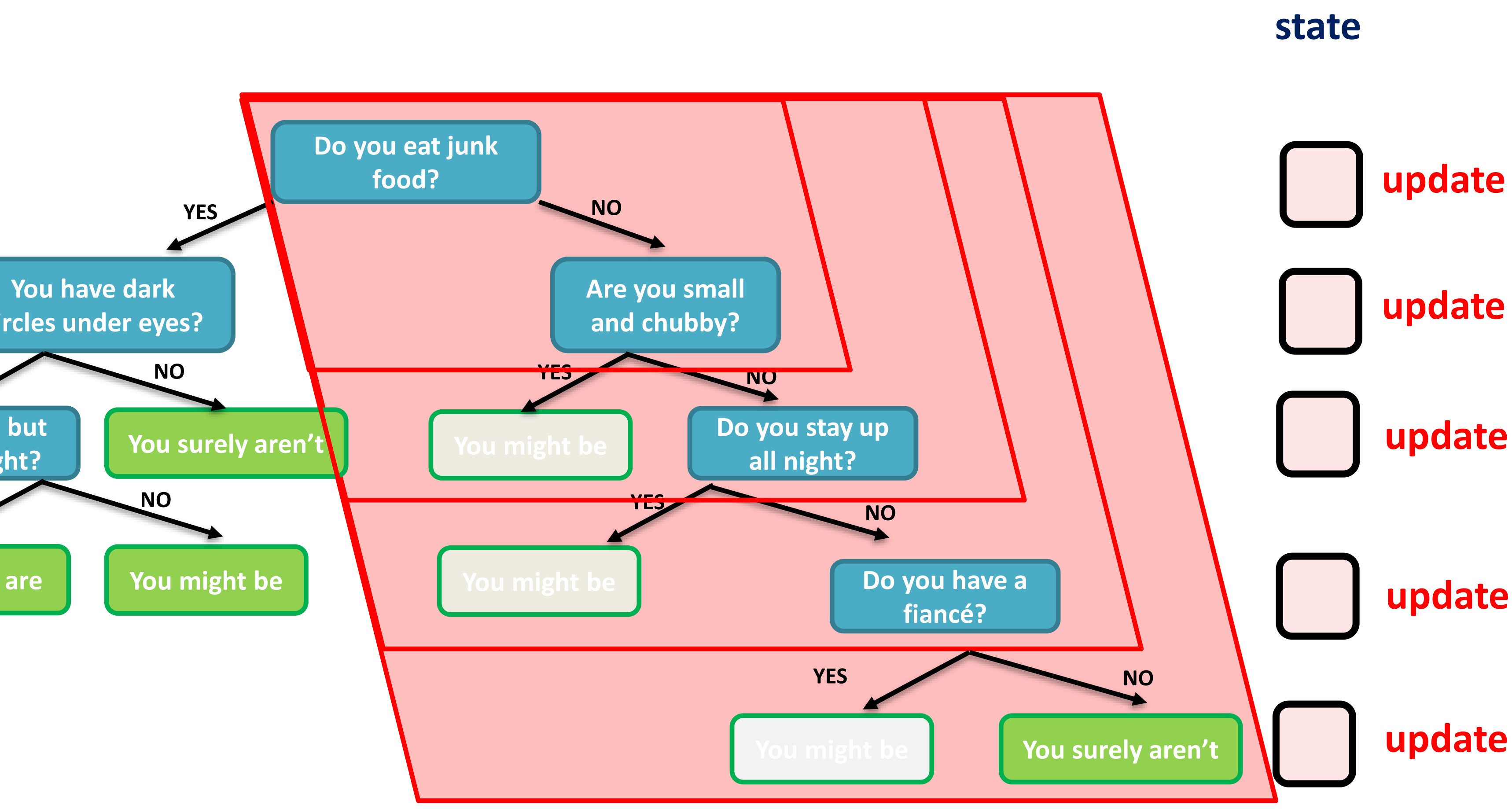

It starts from the root and goes down to the leaves extending the size of the subproblems. With each move it updates some state of up to $D$ values. The update is related to the coefficient $\frac{|C|!(|A|-|C|-1)!}{|A|!}$ in the Shapley value and to sizes of coalitions, in particular.

# Intuition behind TreeSHAP




It starts from the root and goes down to the leaves extending the size of the subproblems. With each move it updates some state of up to $D$ values. The update is related to the coefficient $\frac{|C|!(|A|-|C|-1)!}{|A|!}$ in the Shapley value and to sizes of coalitions, in particular.

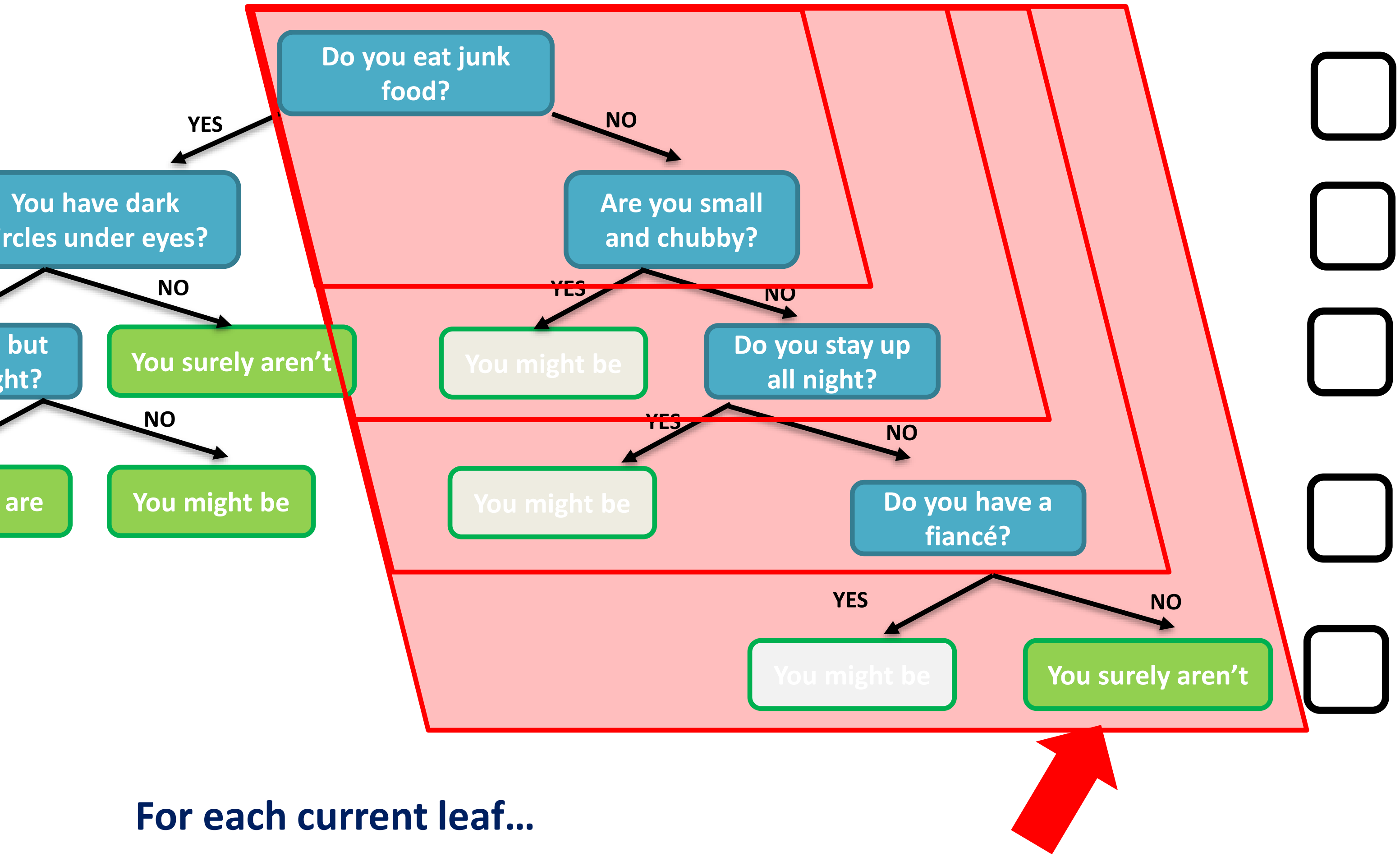
# Intuition behind TreeSHAP




The complexity so far is $O(TLD)$.

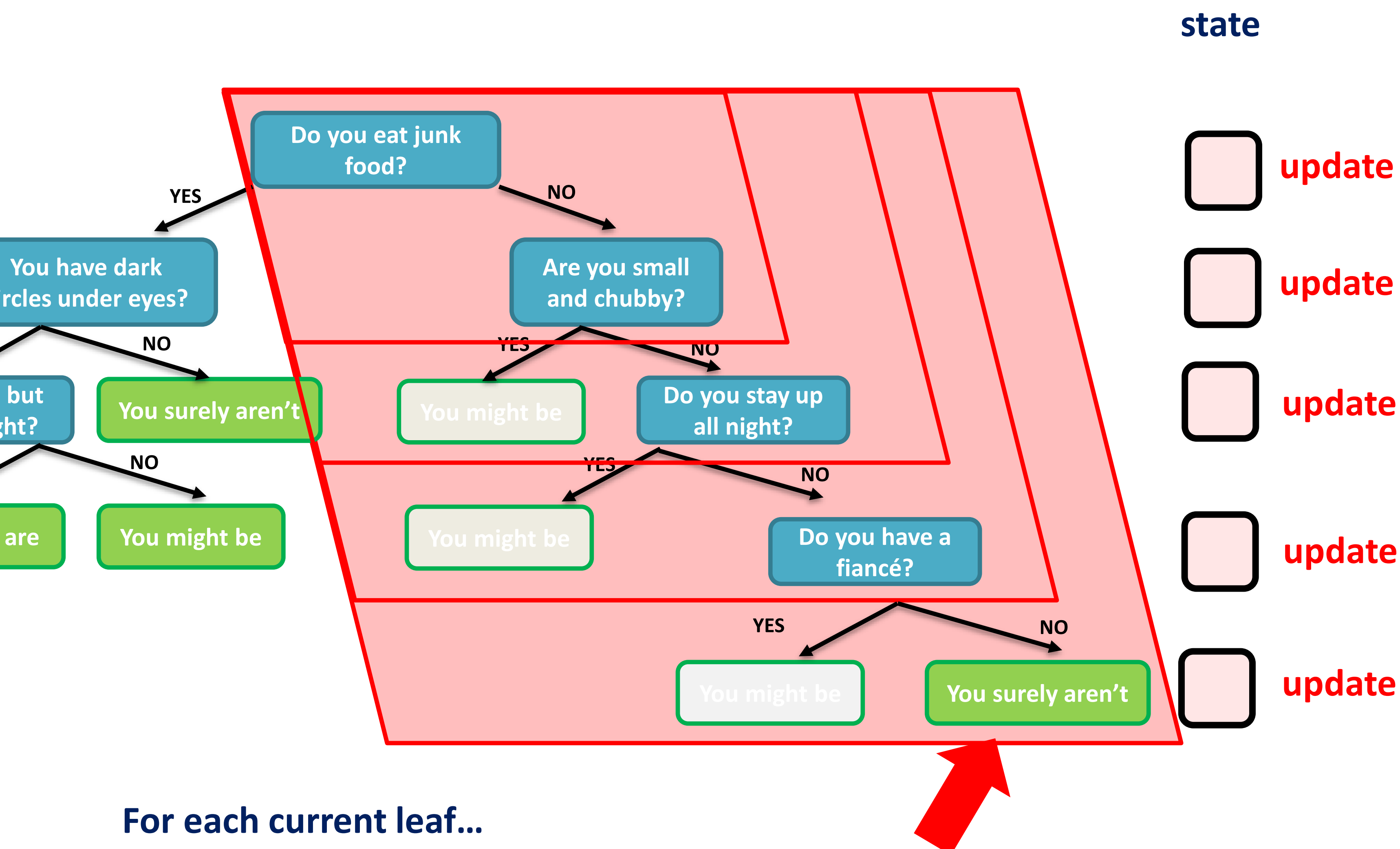
# Intuition behind TreeSHAP

state



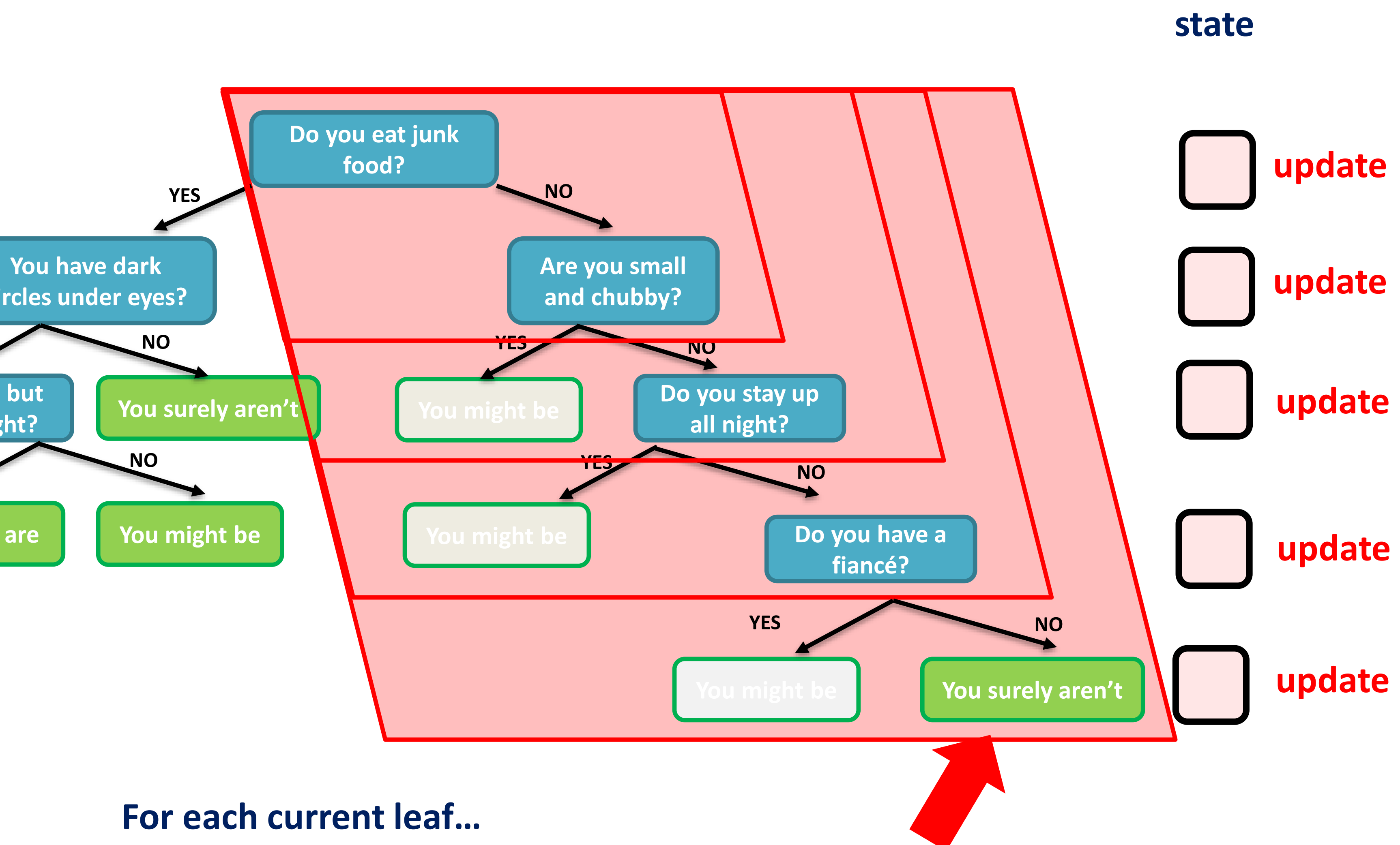

For each current leaf...

# Intuition behind TreeSHAP




For each current leaf…
The algorithm does another $D$ updates that are related to the subtractions in the marginal contributions.

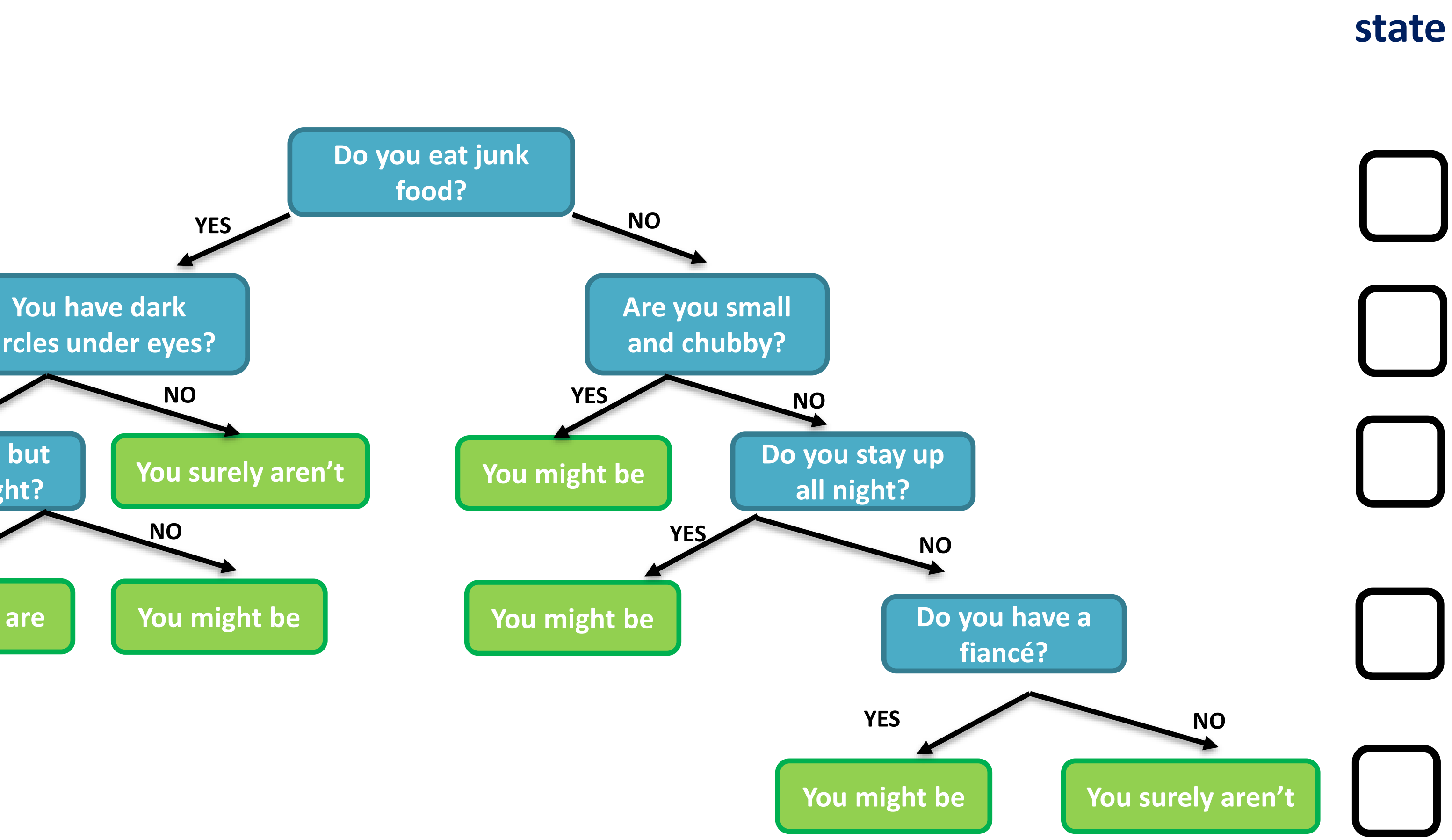
# Intuition behind TreeSHAP



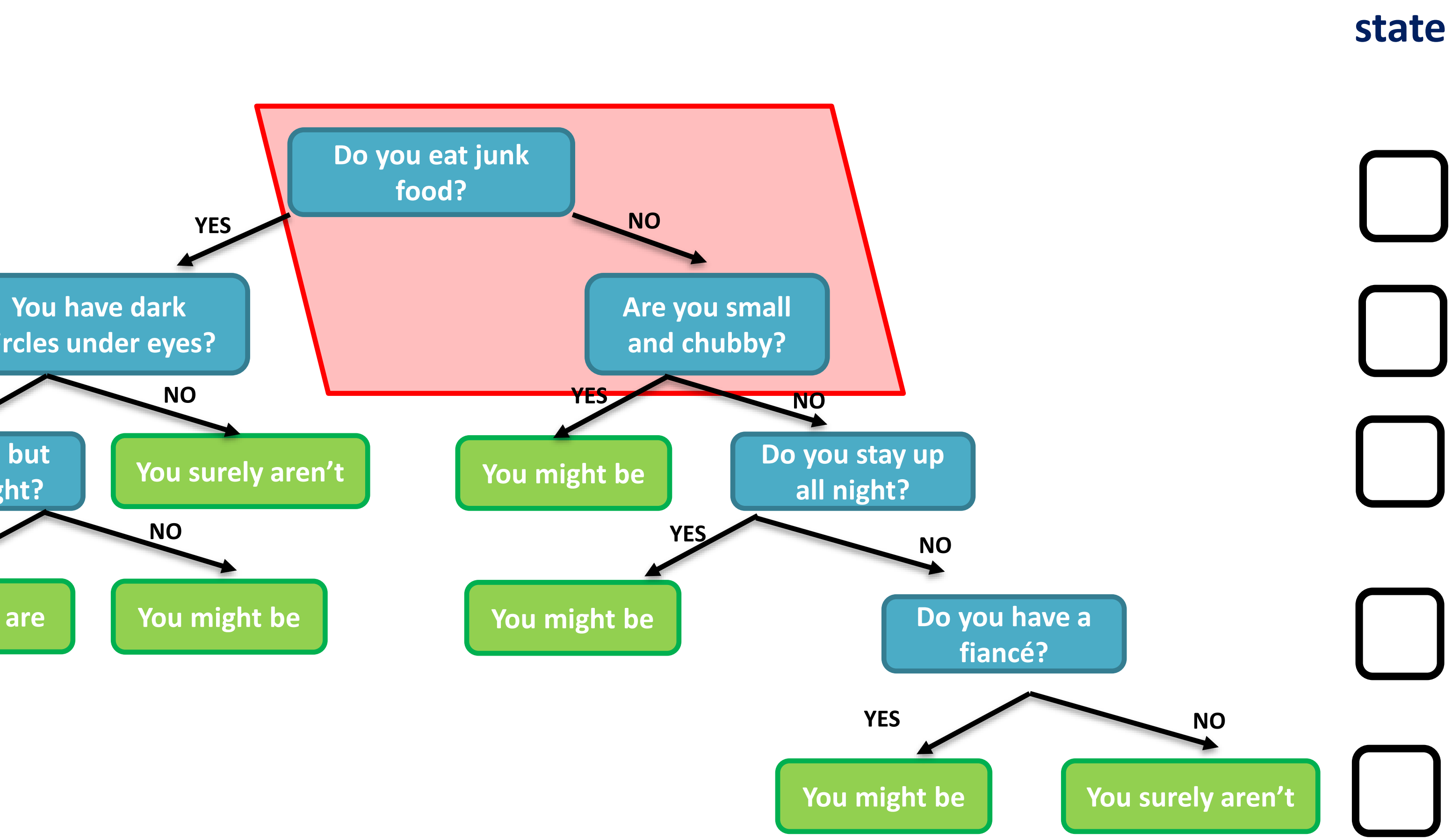

For each current leaf…

The algorithm does another $D$ updates that are related to the subtractions in the marginal contributions. The complexity is $O(TLD^2)$.

# Intuition behind our Algorithm

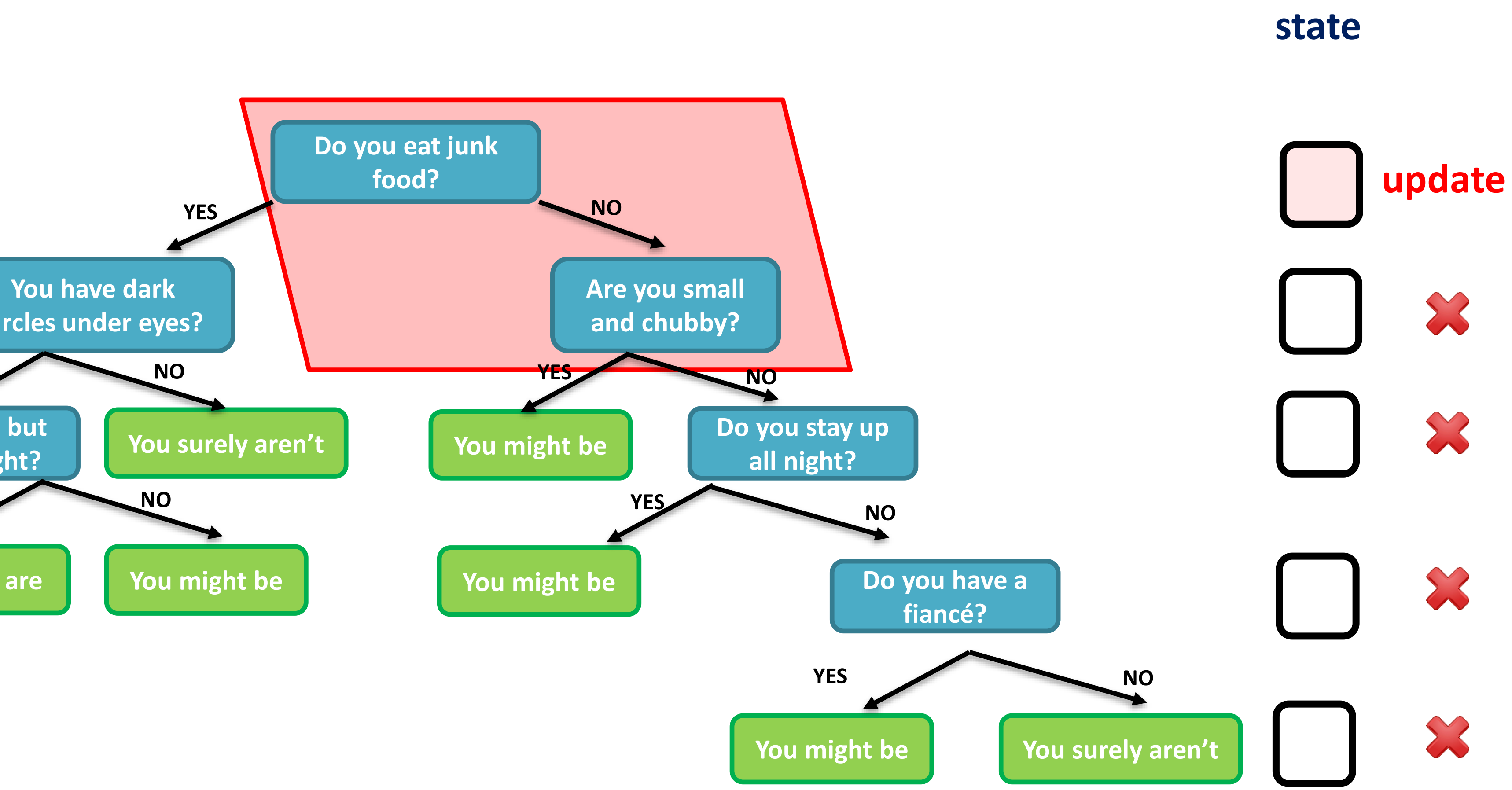


# Intuition behind our Algorithm




Since we don't have the weight coefficient $\frac{|C|!(|A|-|C|-1)!}{|A|!}$ in the Banzhaf value, we only update a single value for each node.

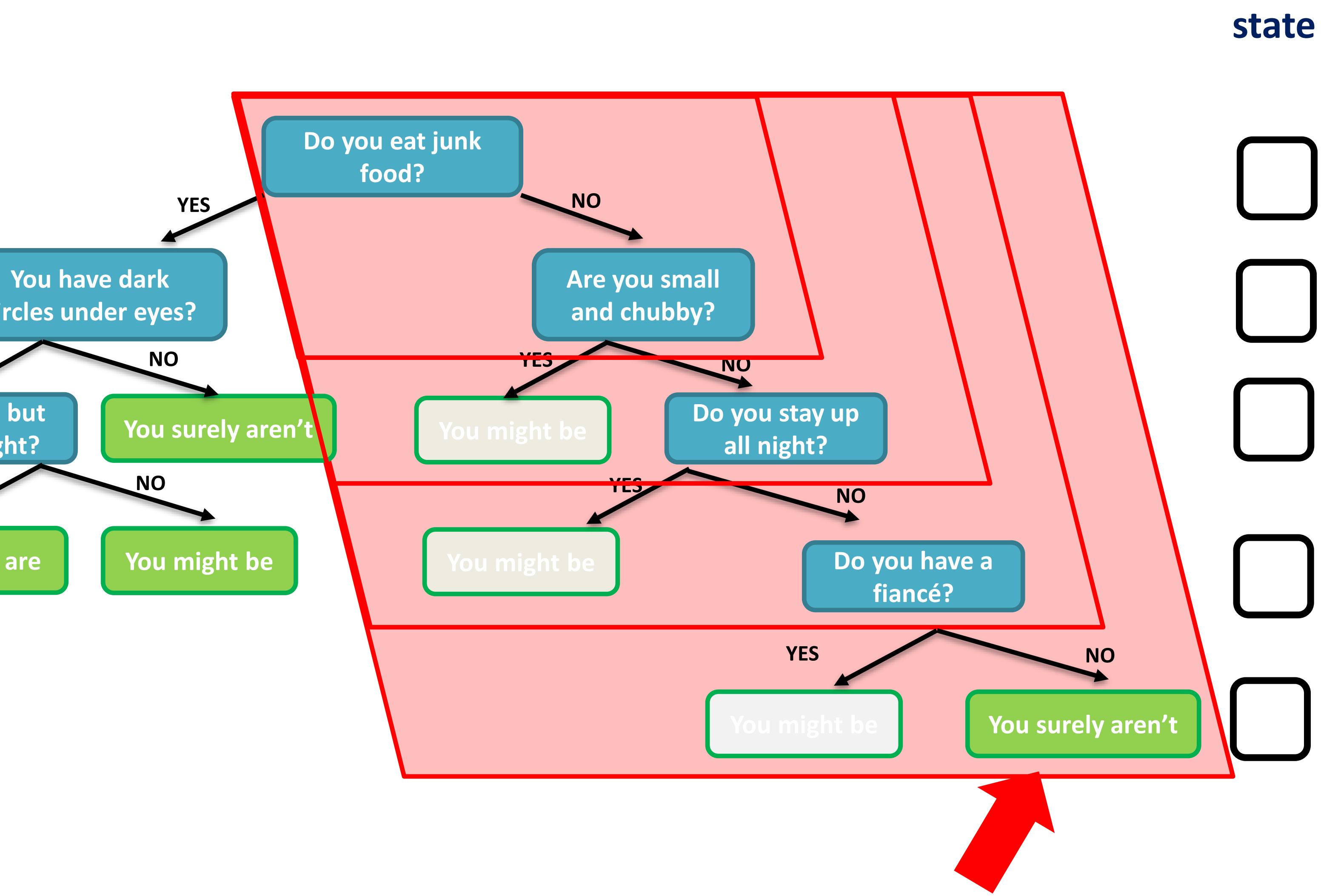
# Intuition behind our Algorithm




Since we don't have the weight coefficient $\frac{|C|!(|A|-|C|-1)!}{|A|!}$ in the Banzhaf value, we only update a single value for each node.

Here the improvement comes from the definition of the Banzhaf value.

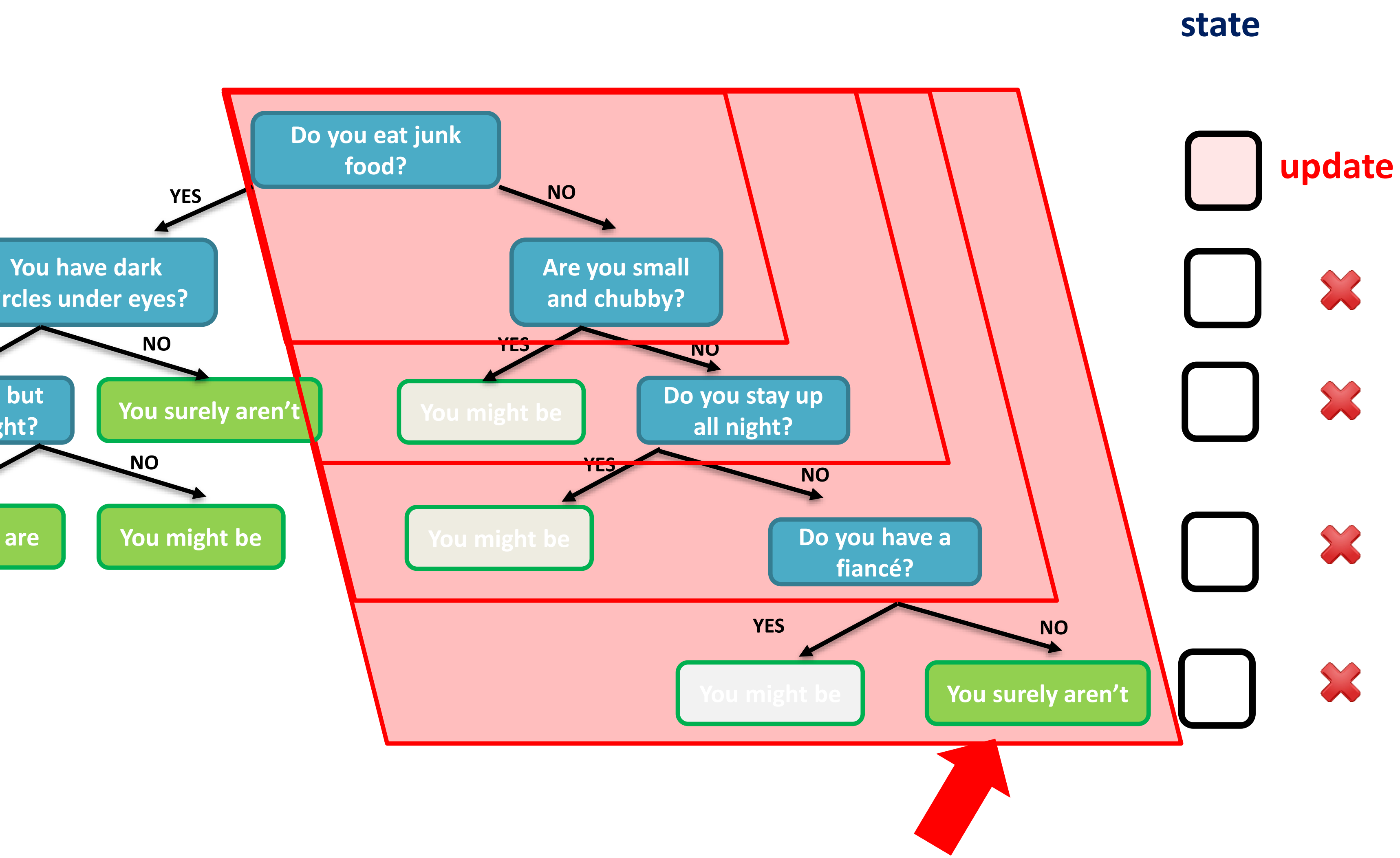
# Intuition behind TreeSHAP




As for updating the partial values when we reach each leaf.

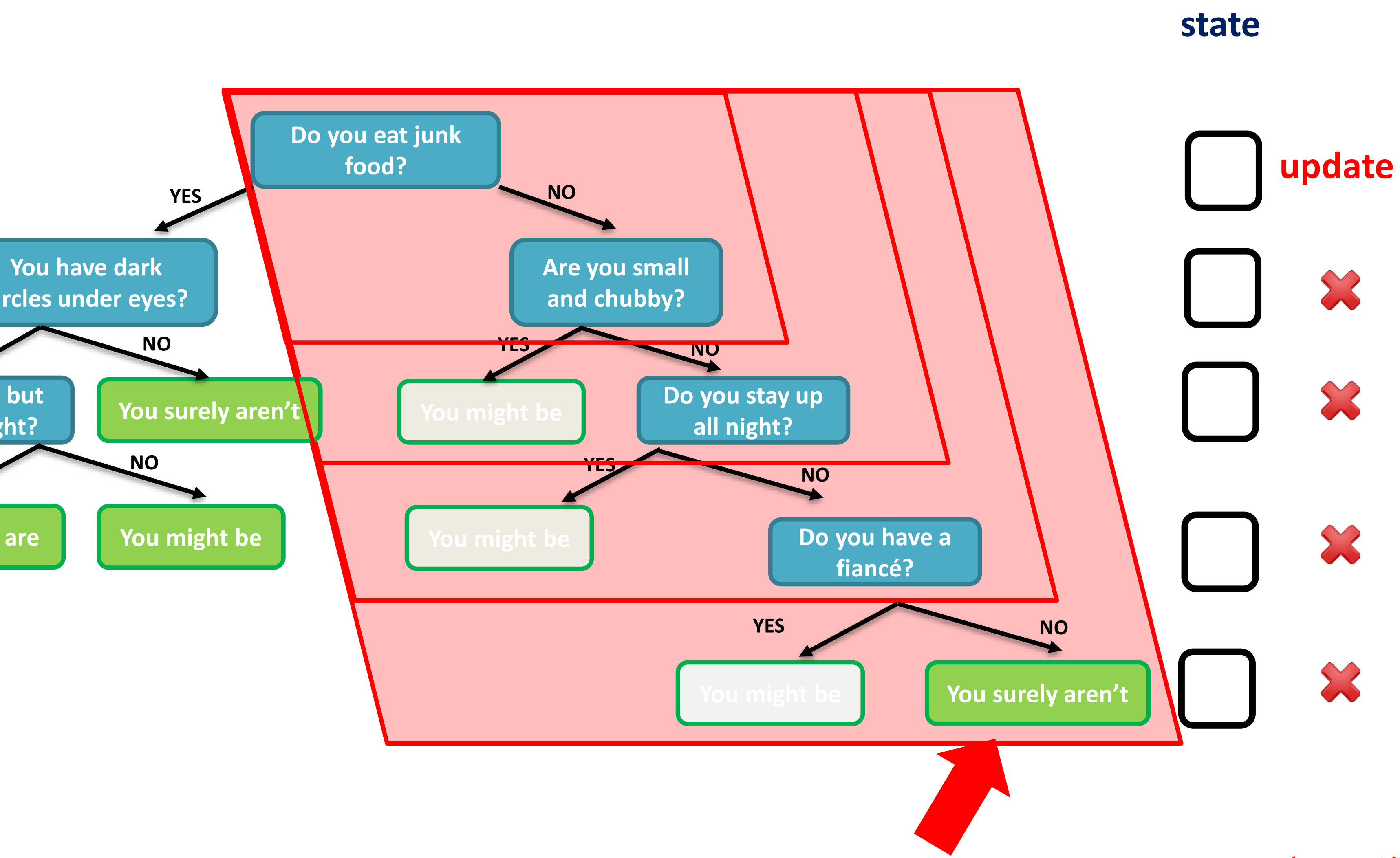
Here, we figured out a method to do only a single update per leaf on average when we backtrack.

# Intuition behind TreeSHAP




As for updating the partial values when we reach each leaf.
Here, we figured out a method to do only a single update per leaf on average when we backtrack. Our complexity is $O(TL)$.

# Intuition behind TreeSHAP




This last trick can be applied to improve the TreeSHAP algorithm from $O(TLD^2)$ to $O(TLD)$.

# Plan of the Talk

1. Values in Cooperative Game Theory
2. Our algorithm for the Banzhaf value vs. TreeSHAP
3. Advantages of the Banzhaf value for tree models - experimental analysis

# Algorithms for DT & Datasets

**Two arguably most popular algorithms for generating decision trees:**

- **sklearn implementation of Decision Trees (DT)**
- **xgboost implementation of Gradient Boosting Decision Trees (GB)**

| Name | Description | Approx. size |
| --- | --- | --- |
| BOSTON *(BS)* | This small prediction dataset contains information concerning housing in thearea of Boston Massachusetts. The task is to predict the price of the house. | 506 rows, 13 features |
| NHANES *(NH)* | One of the most widely-used datasets describing the health and socioeconomic status of people residing in the US. | 8023 rows, 79 features |
| HEALTH_INSURANCE *(HI)* | A medium size dataset for predicting who might be interested in health insurance purchase. | 304887 rows, 14 features |
| FLIGHTS *(FL)* | A large dataset for predicting the flights' delays. | 1543718 rows, 647 features |

| Dataset | xgboost | | | Decision tree |
| --- | --- | --- | --- | --- |
| | Iterations | Max depth | Learning rate | tree_depth |
| BOSTON *(BS)* | 100 | 6 | 0.01 | 10 |
| NHANES *(NH)* | 250 | 4 | 0.2 | 40 |
| HEALTH_INS. *(HI)* | 250 | 4 | 0.2 | 60 |
| FLIGHTS *(FL)* | 250 | 10 | 0.2 | 100 |

# Experimental Results: Running Times

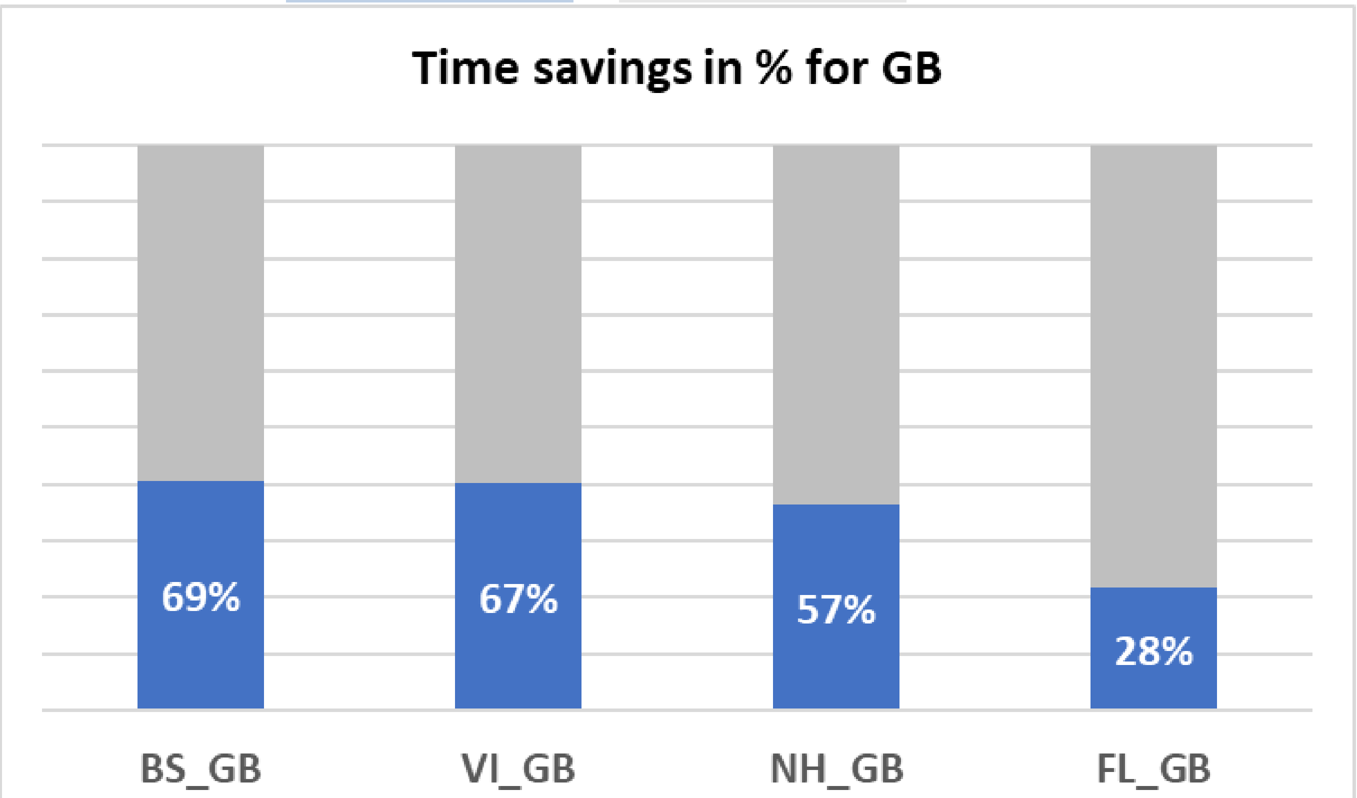
| | BANZHAF | TREESHAP |
|---|---|---|
| BS_GB | 0.48 s | 0.70 s |
| VI_GB | 23.63 s | 35.32 s |
| NH_GB | 50.20 s | 1 m 28 s |
| FL_GB | 13 m 18 s | 48 m 8 s |

| | BANZHAF | TREESHAP |
|---|---|---|
| BS_DT | 0.41 s | 0.41 s |
| NH_DT | 3.57 s | 42.87 s |
| VI_DT | 4 m 55 s | 30 m 55 s |
| FL_DT | 14 m 28 s | 5 h 9 m |

# Experimental Results: Running Times

| | BANZHAF | TREESHAP |
|---|---|---|
| BS_GB | 0.48 s | 0.70 s |
| VI_GB | 23.63 s | 35.32 s |
| NH_GB | 50.20 s | 1 m 28 s |
| FL_GB | 13 m 18 s | 48 m 8 s |

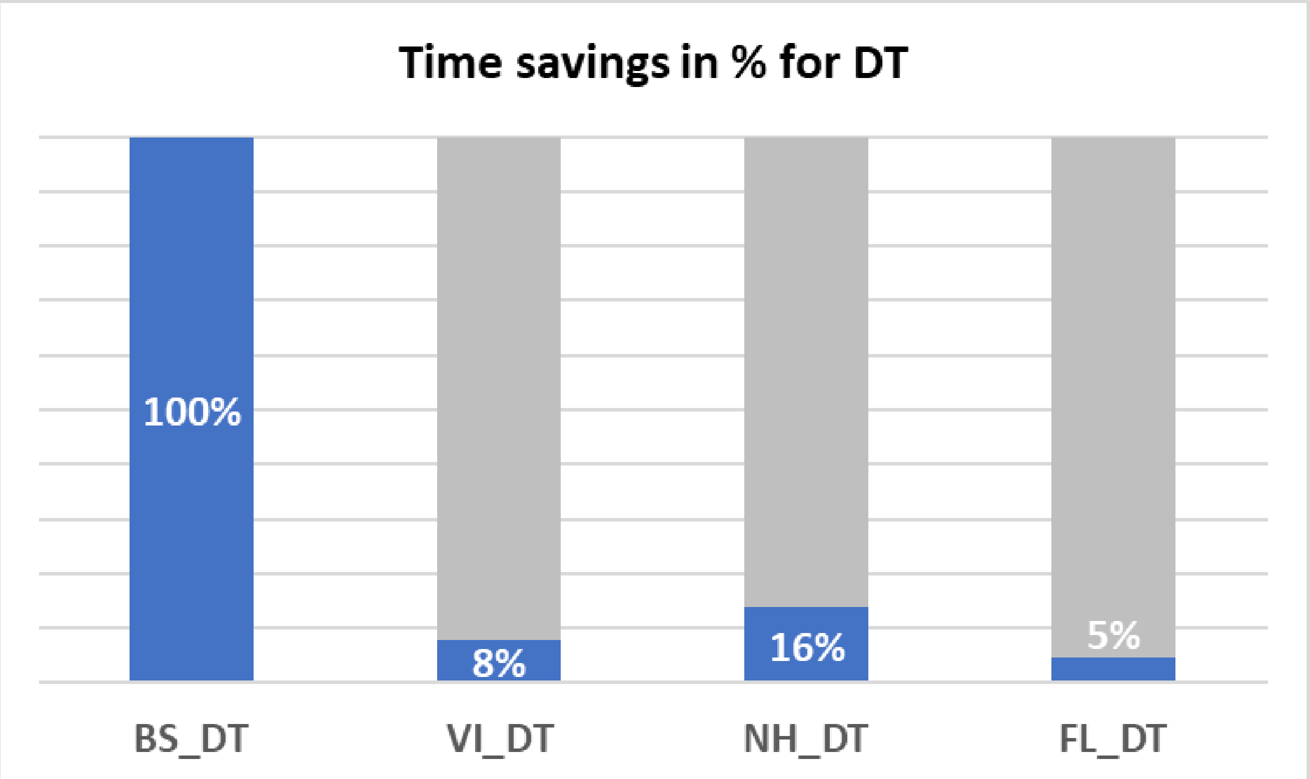
| | BANZHAF | TREESHAP |
|---|---|---|
| BS_DT | 0.41 s | 0.41 s |
| NH_DT | 3.57 s | 42.87 s |
| VI_DT | 4 m 55 s | 30 m 55 s |
| FL_DT | 14 m 28 s | 5 h 9 m |



# Experimental Results: Running Times

| | BANZHAF | TREESHAP |
|---|---|---|
| BS_GB | 0.48 s | 0.70 s |
| VI_GB | 23.63 s | 35.32 s |
| NH_GB | 50.20 s | 1 m 28 s |
| FL_GB | 13 m 18 s | 48 m 8 s |

| | BANZHAF | TREESHAP |
|---|---|---|
| BS_DT | 0.41 s | 0.41 s |
| NH_DT | 3.57 s | 42.87 s |
| VI_DT | 4 m 55 s | 30 m 55 s |
| FL_DT | 14 m 28 s | 5 h 9 m |



# Global impact: Qualitative Difference?

**Global impact – we use the same measure of global impact as Lundberg et al. (2020).**

$\mathcal{D}$ **- a dataset.**
$i \in A$ **- feature**

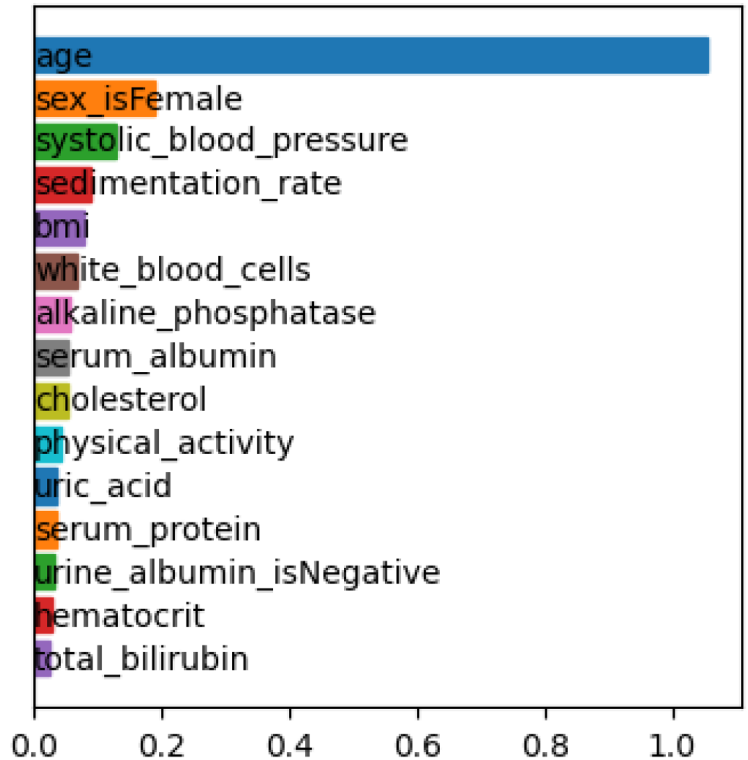
**Shapley global impact:**

$$\Gamma_i^{\mathrm{Sh}} = \sum_{x \in \mathcal{D}} |Sh_i(x)|$$

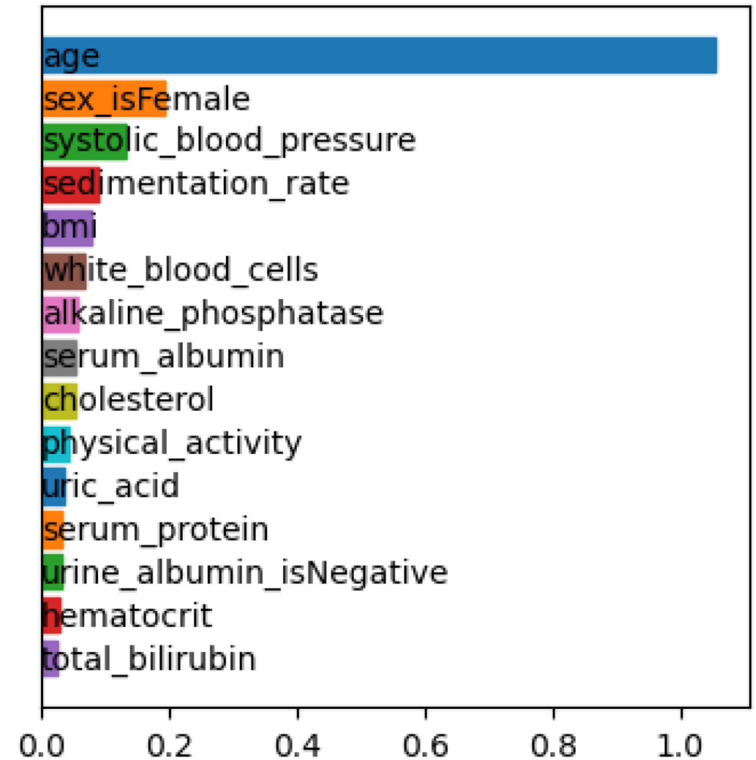
**Banzhaf global impact:**

$$\Gamma_i^{\mathrm{Sh}} = \sum_{x \in \mathcal{D}} |Bh_i(x)|$$

# Global Impacts: NH_GB



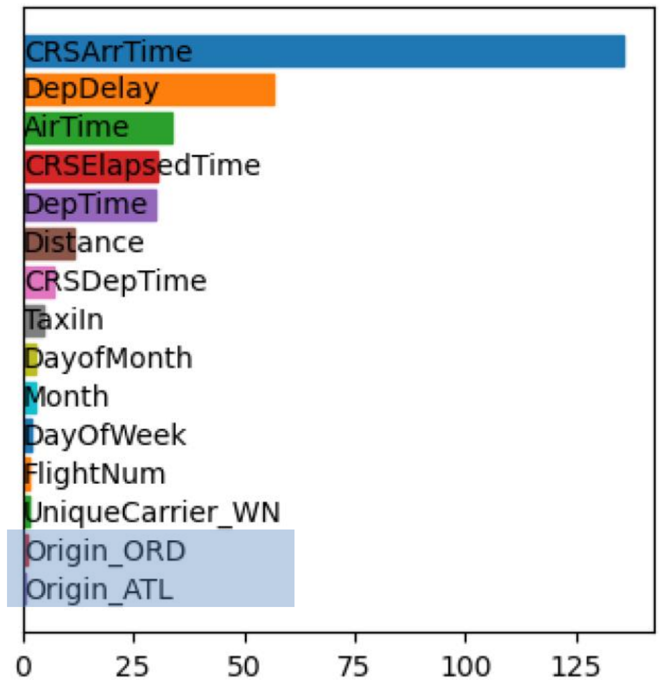

(a) Global Shapley impact obtained with TREESHAP_PATH.



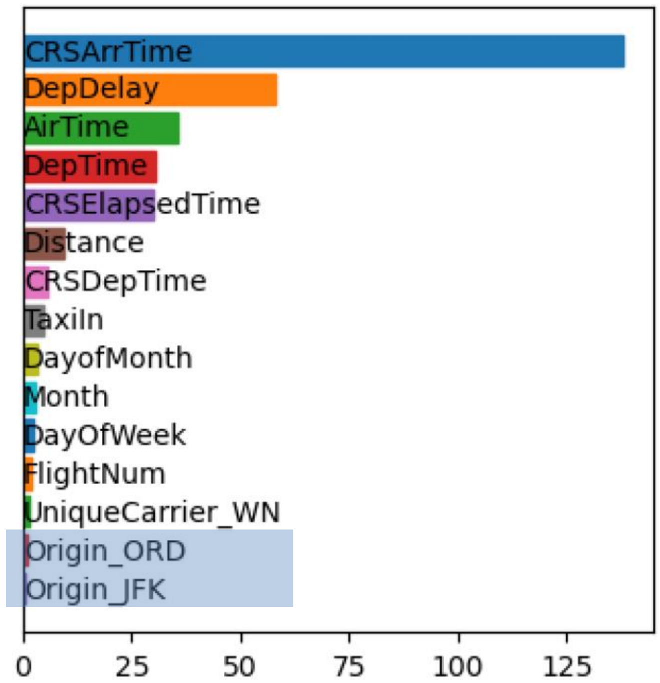

(b) Global Banzhaf impact obtained with BANZHAF.

**The same ordering.**
**Banzhaf results are virtually indistinguishable from the Shapley results.**
**The same holds for BS and VI datasets, both GB and DT.**

# Global Impacts: FL_GB




(a) The original Shapley value.



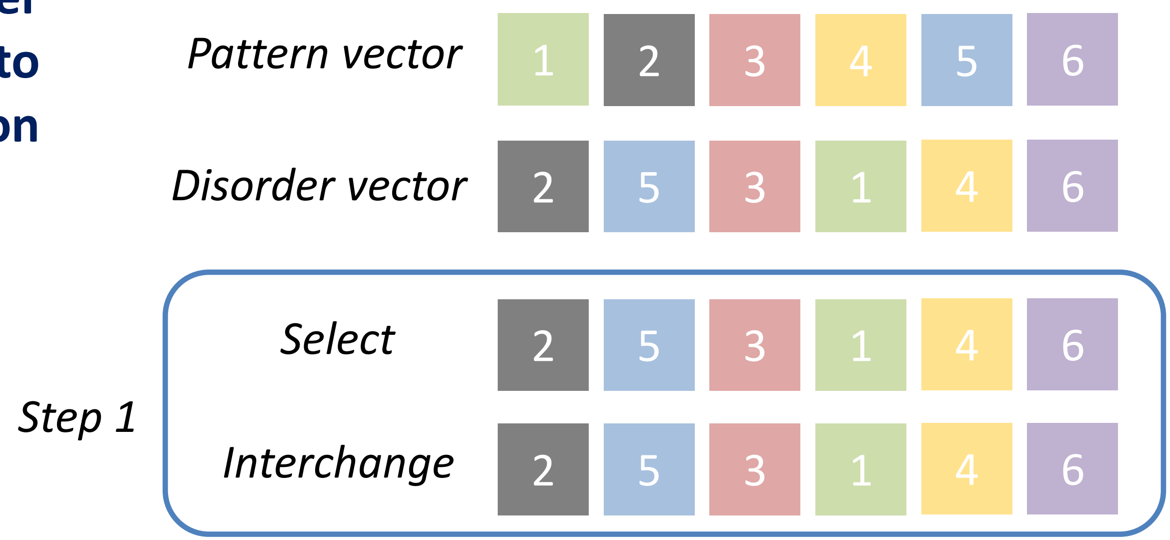

(b) The Banzhaf value.

Figure 6: The global impacts of the individual features for the `FLIGHTS_GB` dataset. We observe small differences in the ordering.

**For the largest dataset with the deepest tree, only very small differences in the ordering of features by importance can be observed for both, both GB and DT.**

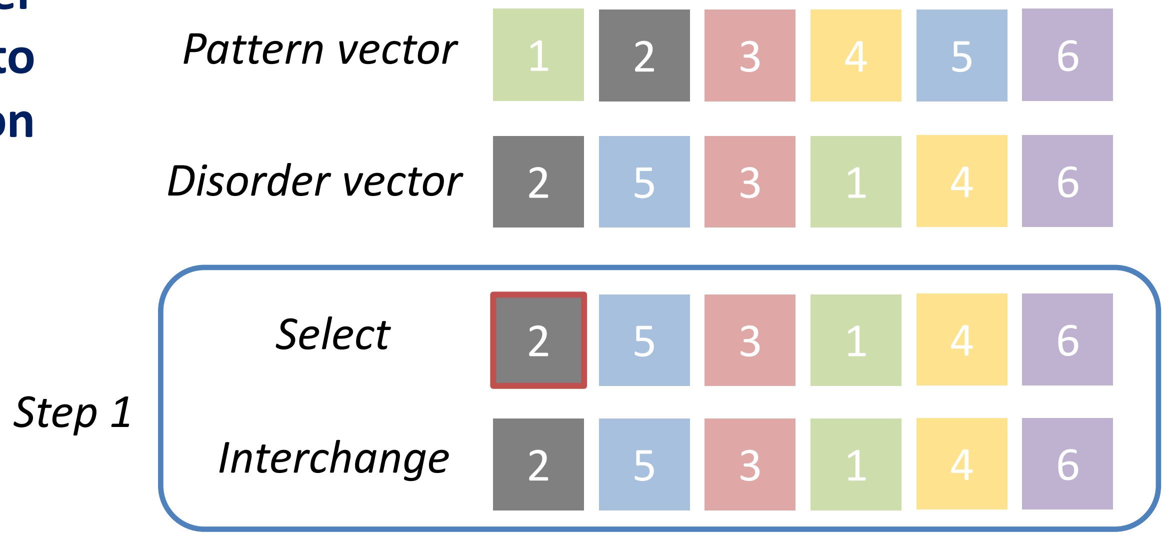
# Individual Data Points: Qualitative Difference?

**Cayley distance – the numer of swaps that are needed to generate one permutation from another**



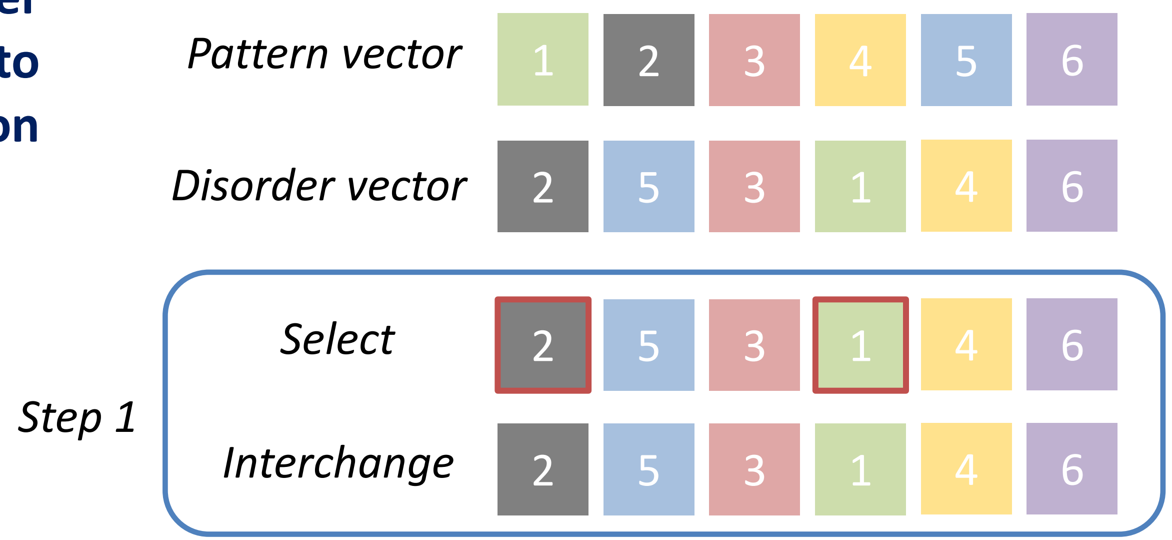
# Are the Banzhaf results qualitatively different?

**Cayley distance – the numer of swaps that are needed to generate one permutation from another**



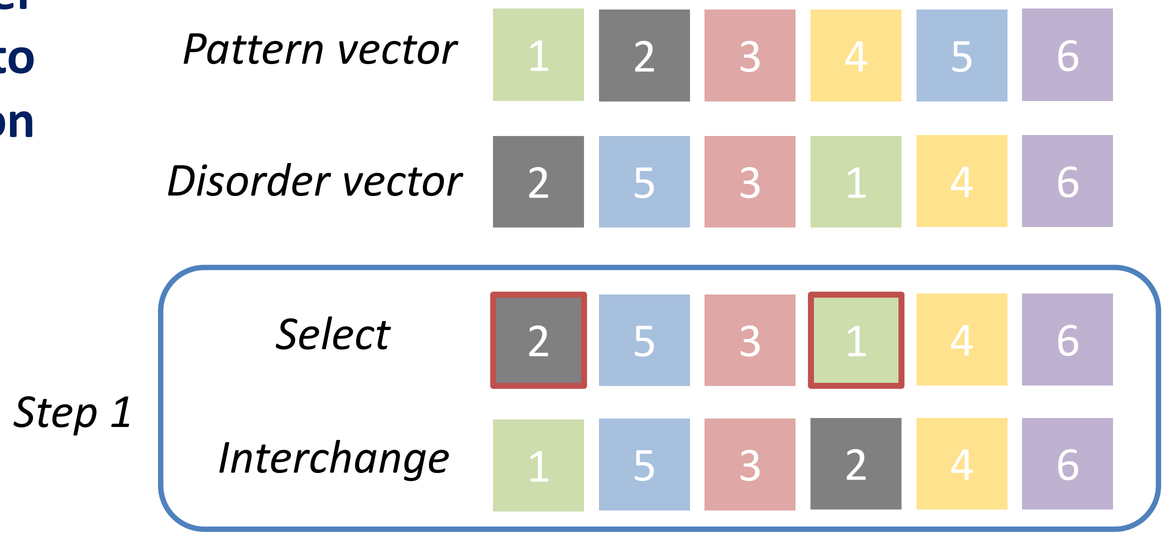
# Are the Banzhaf results qualitatively different?

**Cayley distance – the numer of swaps that are needed to generate one permutation from another**



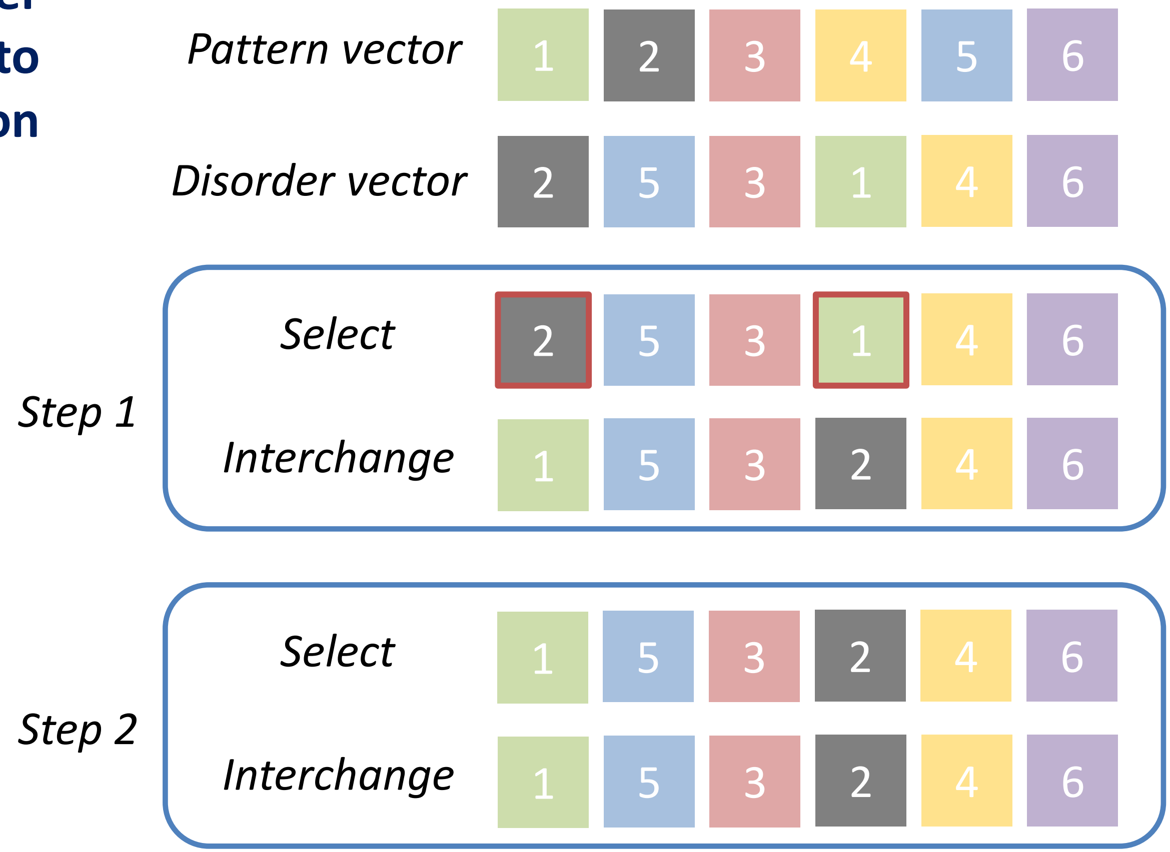
# Are the Banzhaf results qualitatively different?

**Cayley distance – the numer of swaps that are needed to generate one permutation from another**



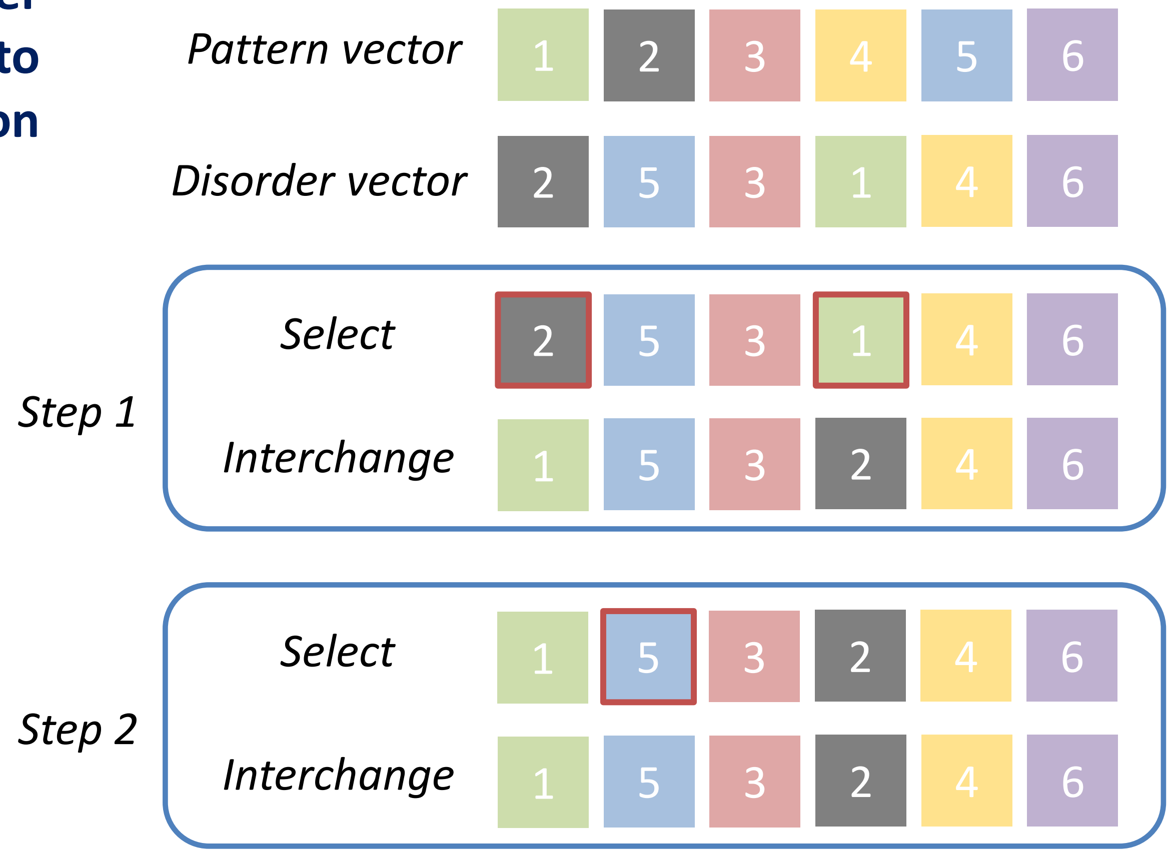
# Are the Banzhaf results qualitatively different?

**Cayley distance – the numer of swaps that are needed to generate one permutation from another**



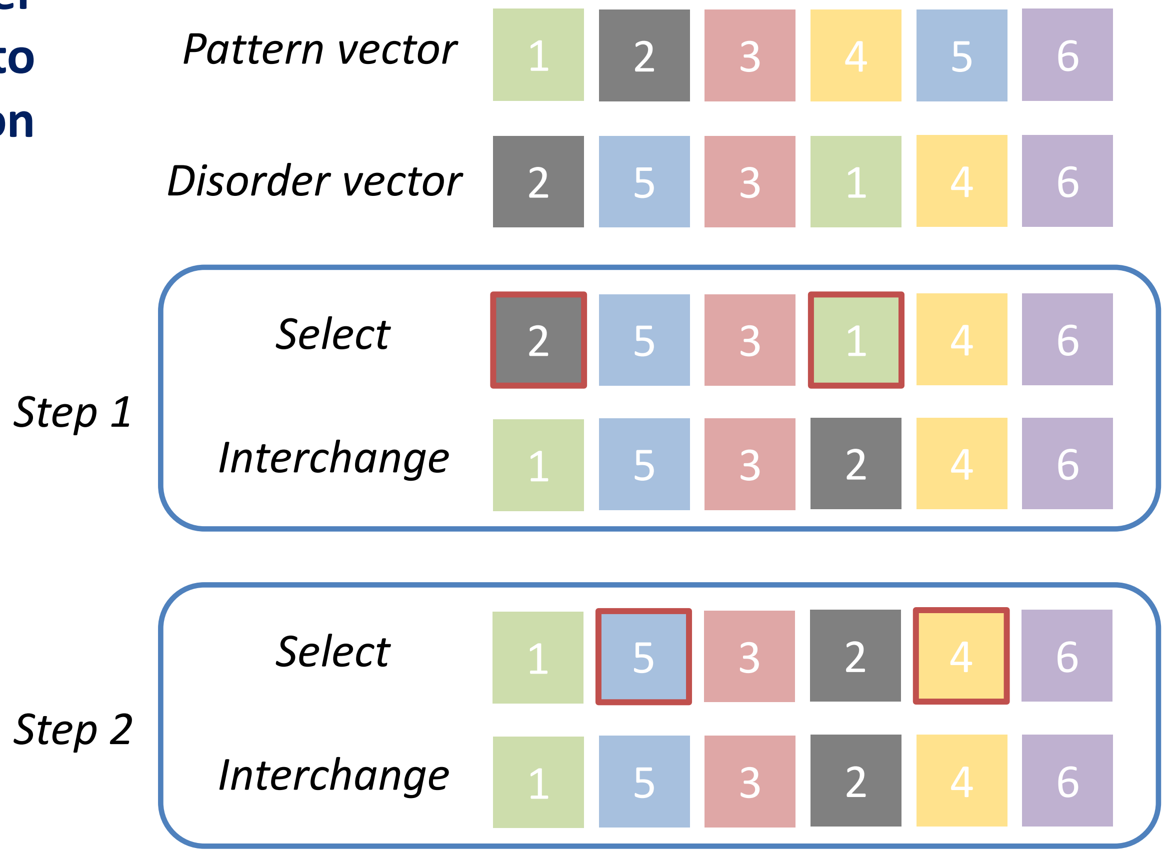
# Are the Banzhaf results qualitatively different?

**Cayley distance – the numer of swaps that are needed to generate one permutation from another**



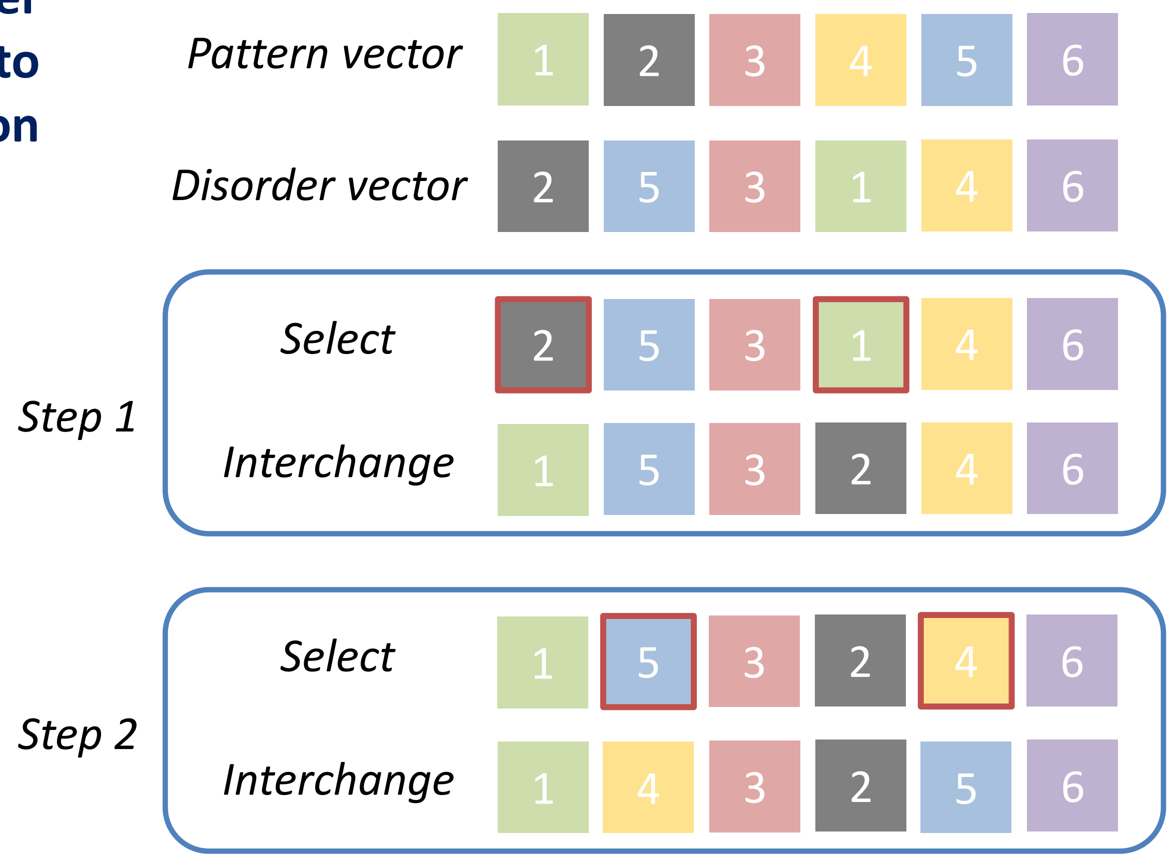
# Are the Banzhaf results qualitatively different?

**Cayley distance – the numer of swaps that are needed to generate one permutation from another**



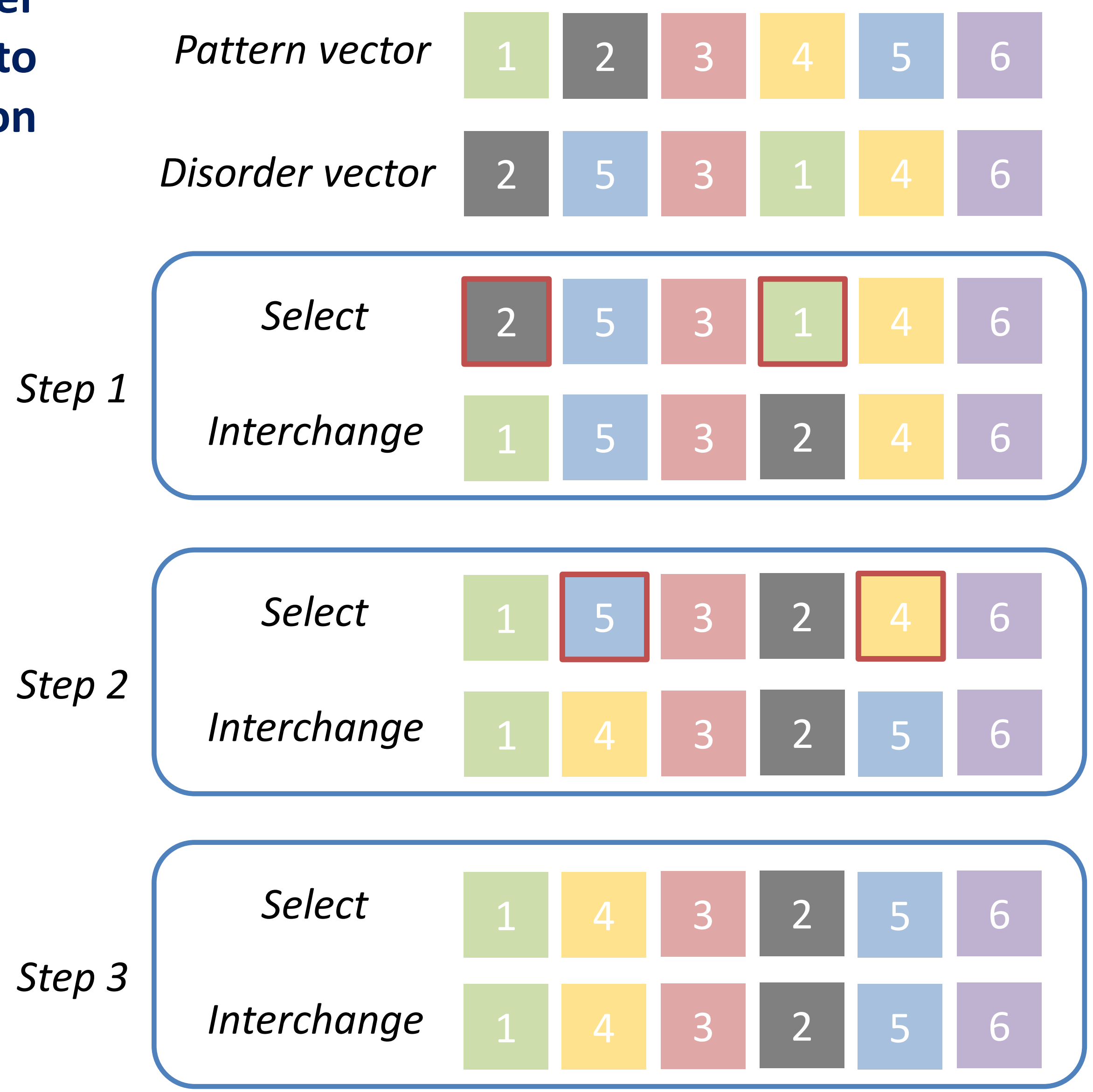
# Are the Banzhaf results qualitatively different?

**Cayley distance – the numer of swaps that are needed to generate one permutation from another**



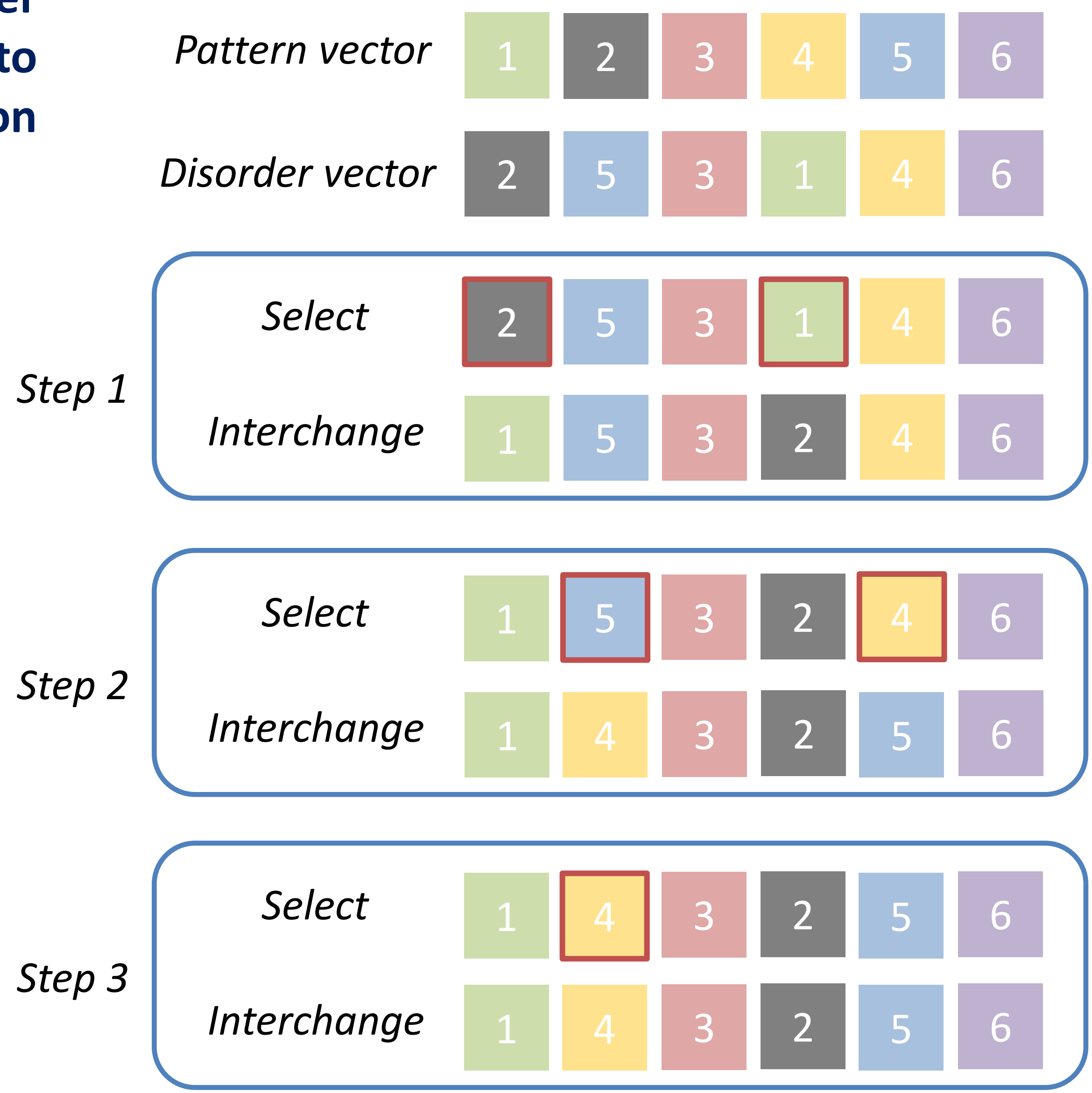
# Are the Banzhaf results qualitatively different?

**Cayley distance – the numer of swaps that are needed to generate one permutation from another**



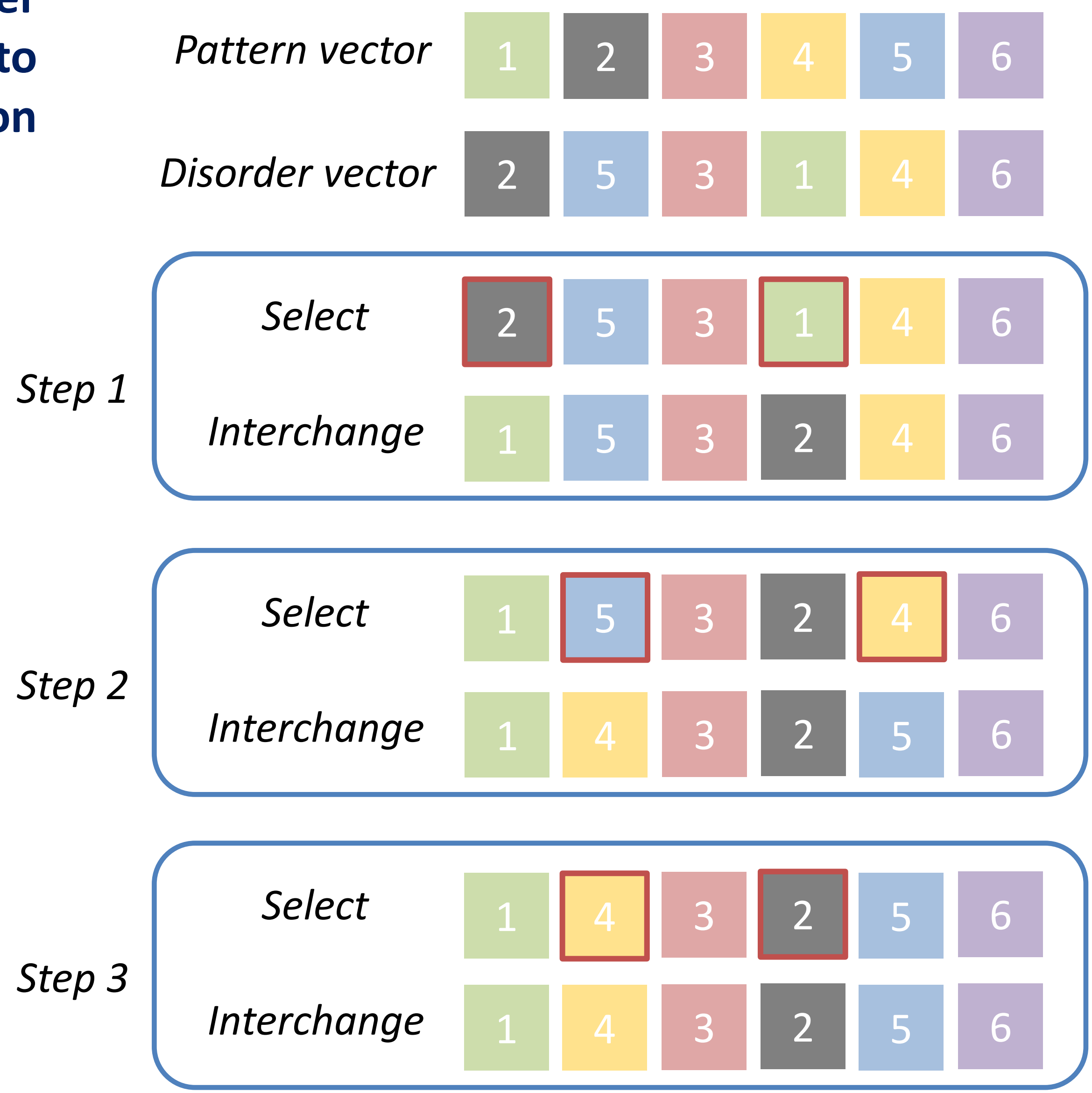
# Are the Banzhaf results qualitatively different?

**Cayley distance – the numer of swaps that are needed to generate one permutation from another**



# Are the Banzhaf results qualitatively different?

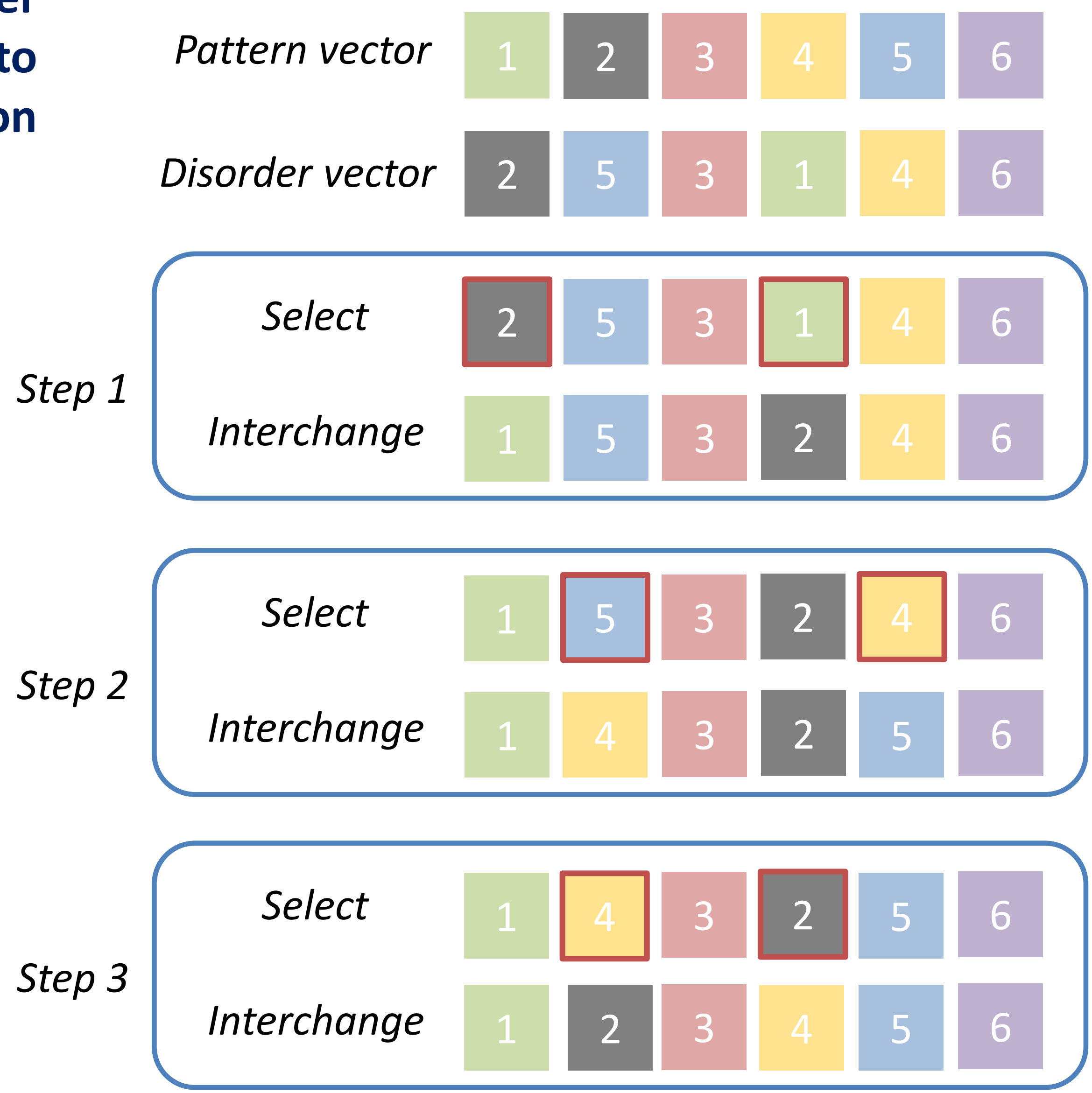
**Cayley distance – the numer of swaps that are needed to generate one permutation from another**



# Are the Banzhaf results qualitatively different?

**Cayley distance – the numer of swaps that are needed to generate one permutation from another**

**Cayley distance = 3**

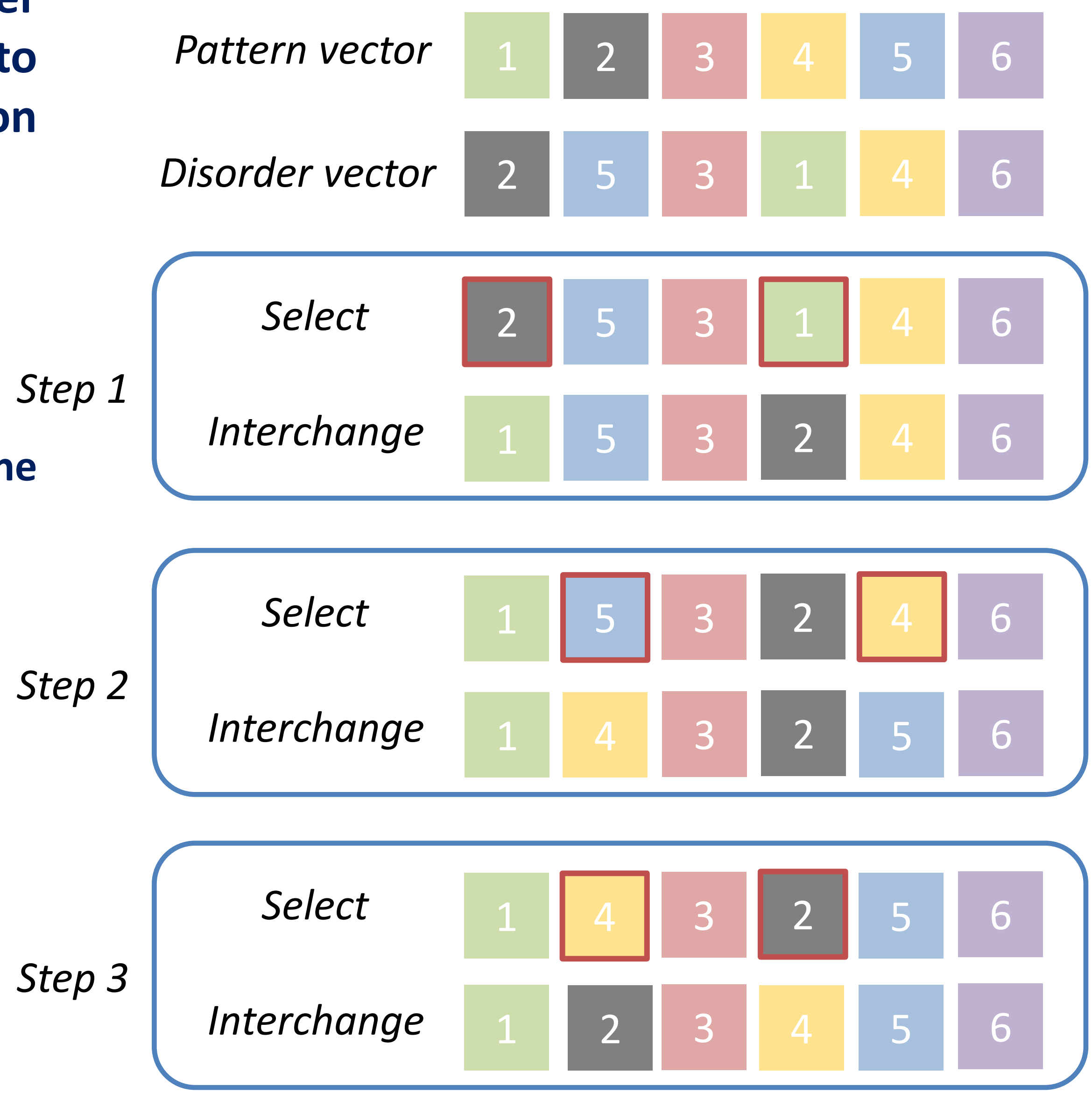


# Are the Banzhaf results qualitatively different?

**Cayley distance – the numer of swaps that are needed to generate one permutation from another**

**Cayley distance = 3**

**Missing features ➔**
**➔ added at the end of the permutation.**



# Average Cayley Distance over all datapoints

| Ins/n | 3 | 10 | 20 |
|---|---|---|---|
| BOS_GB | 0.02 | 1.05 | |
| NH_GB | 0.01 | 0.34 | 1.53 |
| VI_GB | 0.02 | 0.73 | |
| FL_GB | 0.4 | 3.08 | 8.63 |

| Ins/n | 3 | 10 | 20 |
|---|---|---|---|
| BOS_DT | 0.08 | 1.7 | |
| NH_DT | 0.29 | 3.69 | 10.79 |
| VI_DT | 0.13 | 2.60 | |
| FL_DT | 0.18 | 3.38 | 10.59 |

# Average Cayley Distance over all datapoints

| Ins/n | 3 | 10 | 20 |
|---|---|---|---|
| BOS_GB | 0.02 | [illegible] | |
| NH_GB | 0.01 | [illegible] | [illegible] |
| VI_GB | 0.02 | [illegible] | |
| FL_GB | 0.4 | 3.08 | 8.63 |

| Ins/n | 3 | 10 | 20 |
|---|---|---|---|
| BOS_DT | 0.08 | 1.7 | |
| NH_DT | 0.29 | 3.69 | 10.79 |
| VI_DT | 0.13 | 2.60 | |
| FL_DT | 0.18 | 3.38 | 10.59 |

Cayley Distance for 3 most important features

# Average Cayley Distance over all datapoints

| Ins/n | 3 | 10 | 20 |
|---|---|---|---|
| BOS_GB | 0.02 | 1.05 | |
| NH_GB | 0.01 | 0.34 | 1.53 |
| VI_GB | 0.02 | 0.73 | |
| FL_GB | 0.4 | 3.08 | 8.63 |

| Ins/n | 3 | 10 | 20 |
|---|---|---|---|
| BOS_DT | 0.08 | 1.7 | |
| NH_DT | 0.29 | 3.69 | 10.79 |
| VI_DT | 0.13 | 2.60 | |
| FL_DT | 0.18 | 3.38 | 10.59 |

# Average Cayley Distance over all datapoints

| Ins/n | 3 | 10 | 20 |
|---|---|---|---|
| BOS_GB | 0.02 | 1.05 | |
| NH_GB | 0.01 | 0.34 | 1.53 |
| VI_GB | 0.02 | 0.73 | |
| FL_GB | 0.4 | 3.08 | 8.63 |

| Ins/n | 3 | 10 | 20 |
|---|---|---|---|
| BOS_DT | 0.08 | 1.7 | |
| NH_DT | 0.29 | 3.69 | 10.79 |
| VI_DT | 0.13 | 2.60 | |
| FL_DT | 0.18 | 3.38 | 10.59 |

# Average Cayley Distance over all datapoints

| Ins/n | 3 | 10 | 20 |
|---|---|---|---|
| BOS_GB | 0.02 | 1.05 | |
| NH_GB | 0.01 | 0.34 | 1.53 |
| VI_GB | 0.02 | 0.73 | |
| FL_GB | 0.4 | 3.08 | 8.63 |

| Ins/n | 3 | 10 | 20 |
|---|---|---|---|
| BOS_DT | 0.08 | 1.7 | |
| NH_DT | 0.29 | 3.69 | 10.79 |
| VI_DT | 0.13 | 2.60 | |
| FL_DT | 0.18 | 3.38 | 10.59 |

For 98% of the data points, the respective 3 top features and their order matched.

# Average Cayley Distance over all datapoints

| Ins/n | 3 | 10 | 20 |
|---|---|---|---|
| BOS_GB | 0.02 | 1.05 | |
| NH_GB | 0.01 | 0.34 | 1.53 |
| VI_GB | 0.02 | 0.73 | |
| FL_GB | 0.4 | 3.08 | 8.63 |

| Ins/n | 3 | 10 | 20 |
|---|---|---|---|
| BOS_DT | 0.08 | 1.7 | |
| NH_DT | 0.29 | 3.69 | 10.79 |
| VI_DT | 0.13 | 2.60 | |
| FL_DT | 0.18 | 3.38 | 10.59 |

**The orderings deviation was generally larger for DT instances, where larger tree depths were allowed.**

**We also studied per-feature average differences. We consider both:**

- **MAE (Mean Average Error) less than 5% for smaller and 20% for larger models (for top features)**
- **RMSE (Root Mean Square Error) – for the large model the difference reached 50% even for top features**

# Numerical Accuracy

**We can prove the following very pessimistic statement about Banzhaf values.**

**Lemma:** *The Banzhaf values can be computed with relative error at most* $(1+\varepsilon)^{O(D)}-1$*, where* $\varepsilon$ *is machine epsilon and D is the tree depth.*

**This bound is quite pessimistic and at the same time not very large if double precision is used and the tree depth D is small enough.**

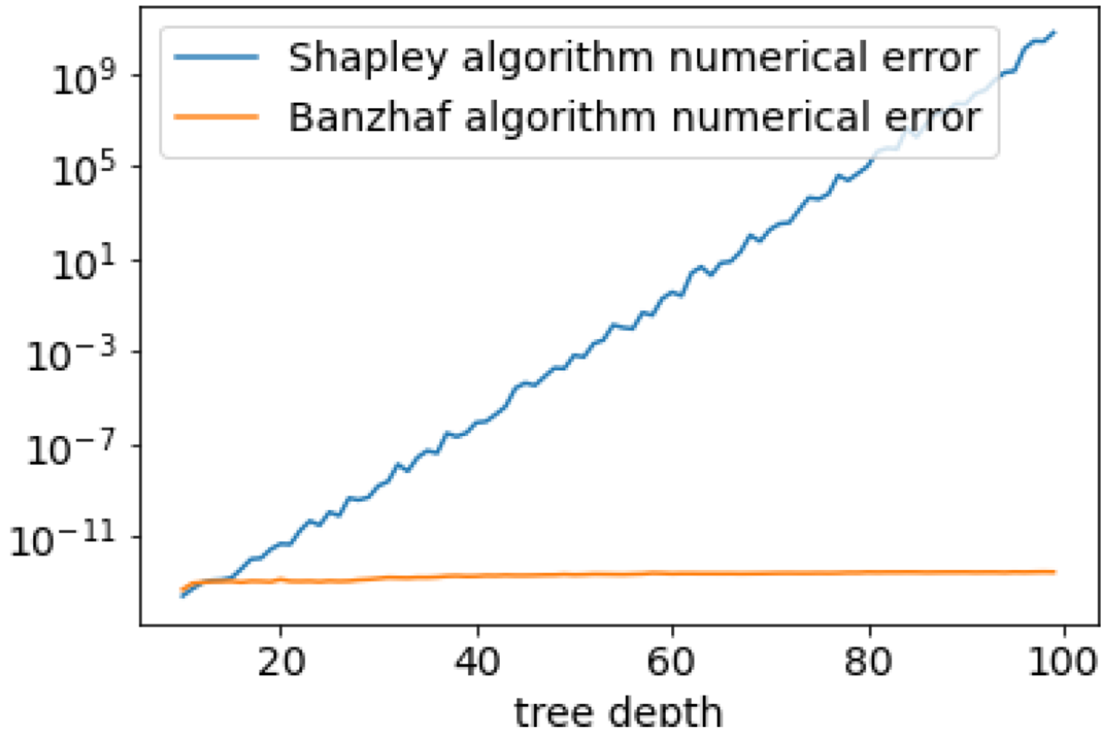
**Similar bound is impossible for TreeShap as it requires subtraction of intermediate values, which can lead to so-called catastrophic cancellations.**

**A much slower impractical algorithm for TreeShap that avoids subtractions behaves similarly to Banzhaf in experiments.**

# Numerical Accuracy

As more significant differences arose for large models ➔ the algorithms might suffer **numerical problems**

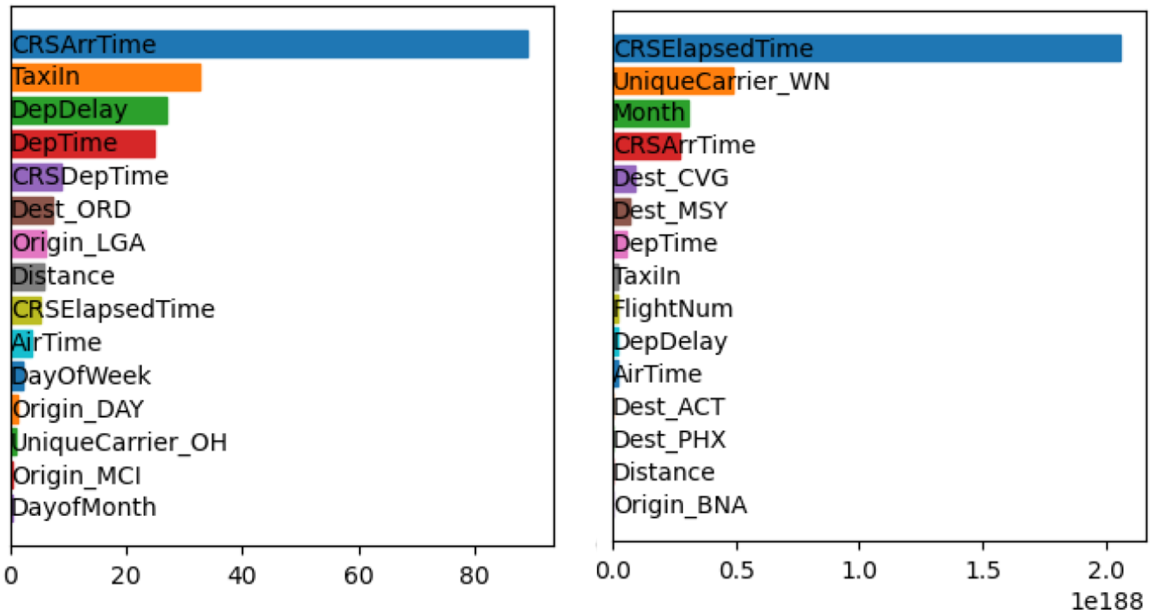
We compare **numerical stability** on a simple artificially prepared instance SYNTHETIC_SPARSE for which we know the answer for both the Shapley value and the Banzhaf value.



# Numerical Accuracy

**As more significant differences arose for large models ➔ the algorithms might suffer numerical problems**

**When comparing different implementations of SHAP with respect to point explanations we can spot significant differences.**




(a) shap_orig (b) shap_fast

# Take away message

**Technical contribution:**

We advocate the Banzhaf value for tree models:

1. It can be computed noticeably faster than the Shapley value
2. Probably more numerically stable
3. Both methods deliver:
   - essentially the same global impacts
   - close explanations of individual predictions

**Meta level:**

- Game theory and algorithmic view
- Interplay of the above areas with AI is growing
- Many more interesting problems to come

# Thank you


## Questions?