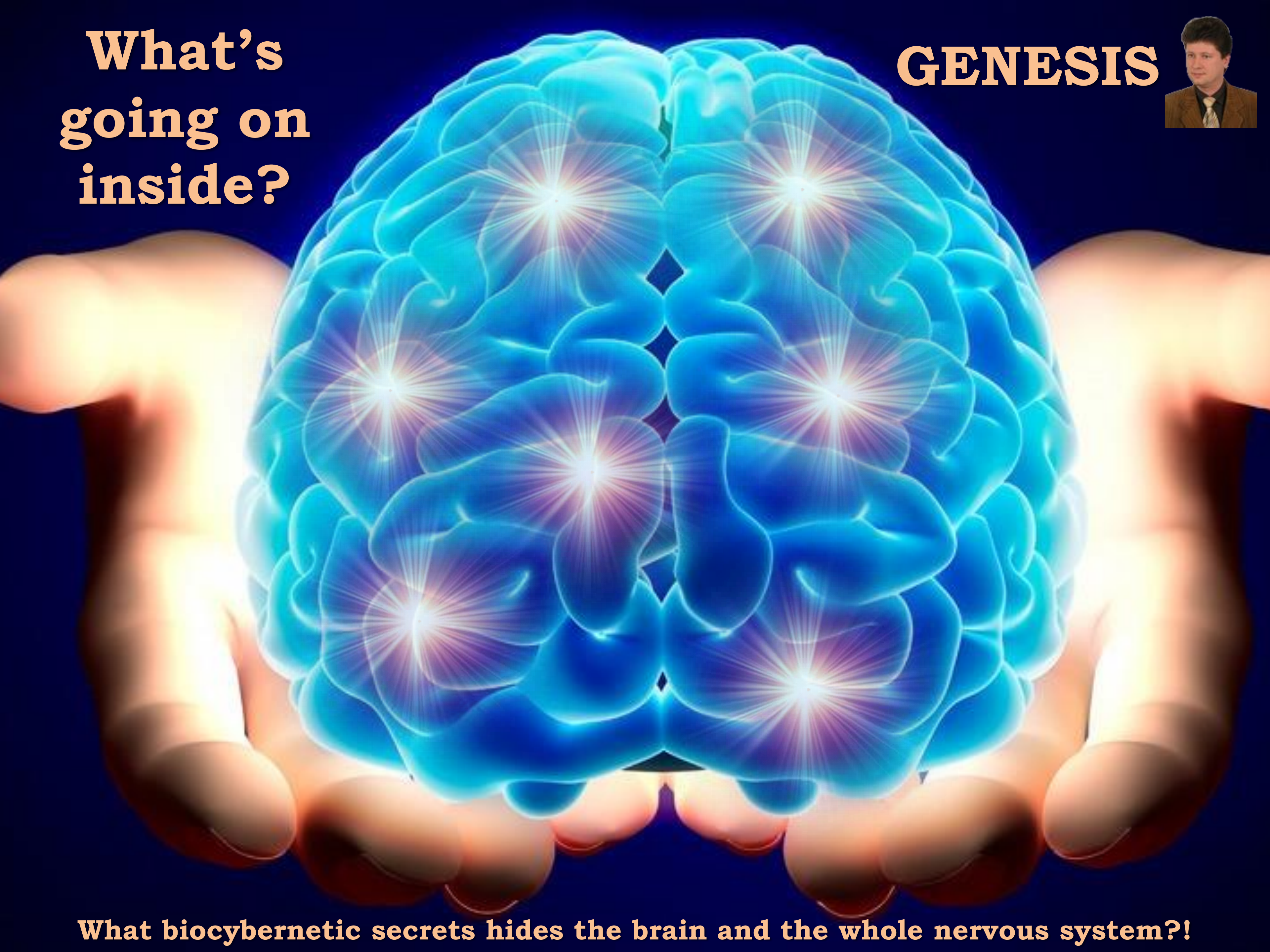
COMPUTATIONAL INTELLIGENCE

Associative Data Structures and Associative Neural Graphs

AGH

Adrian Horzyk


What's going on inside?
GENESIS
What biocybernetic secrets hides the brain and the whole nervous system?!

BRAIN

Internal associative representation of the data and fast associative processes allows the brain to quickly conclude and anticipate!

Brain – a biocybernetic structure forming knowledge and intelligence

Each brain can automatically and in the best-known way:

- memorize relations between data and form associations representing them,
- automatically form and broaden knowledge on the basis of the incoming data,
- remember various patterns and generalize about them,
- store important relations between data,
- work and recall facts in an associative way,
- easily use related data and information,
- quickly and context-sensitive recall adequate pieces of information,
- automatically recognize similarities and use them in thinking processes,
- transfer properties and behavior among similar objects,
- create new rules, methods, and algorithms based on the remembered ones.

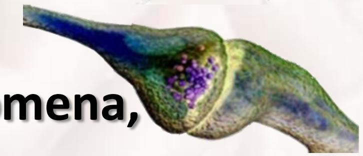
Every event and experience of our lives is changing our brains to a certain extent, its way of working, and influencing future associations and actions!

The dynamics and biocybernetic capabilities of our brains do not currently have a decent cybernetic equivalent or model in computational intelligence!

Brain – a dynamically changing biocybernetic structure

The brain is an unusual „computing” machine because it changes both its hardware and software as a result of the interaction with the data coming to it in the form of different stimuli. These changes concern:

- In the way of its further operation,
- In the process of further data processing,
- In its structure and properties of connections,
- In parameters of construction and functioning of neurons,
- In the previously memorized facts, rules, and objects,
- In the representation of various objects, actions, and phenomena,
- In the way of associating and remembering facts, rules, and routines.



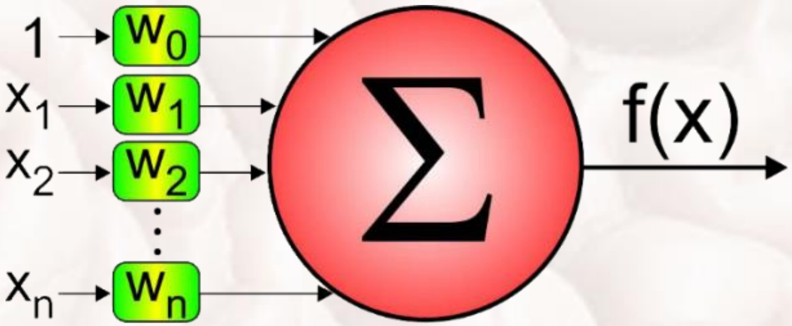
The brain allows us to **memorize**, but not everything and not permanently.

Definitions and ways of understanding different objects **can grow, narrow, update, and even totally change** throughout our lives.

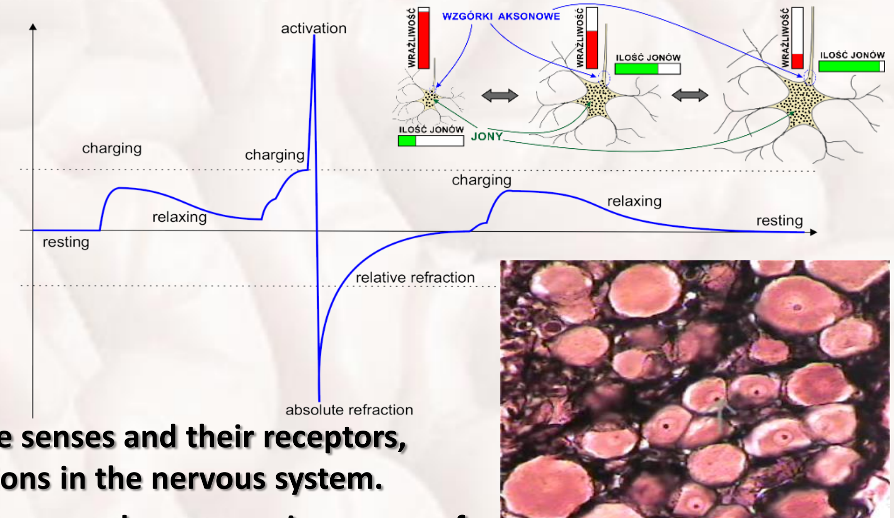
The way it works is related to a nerve structure that allows it **to act in an associative way** and to **selectively represent** relations between data, objects, their groups, sequences, and classes.

BIOLOGICAL AND ARTIFICIAL NEURON

Artificial neurons used today in computational intelligence are <u>very poor models of</u> biological neurons, distorting in their way of acting, plastic changes inside them, and the way of adaptation to incoming data:



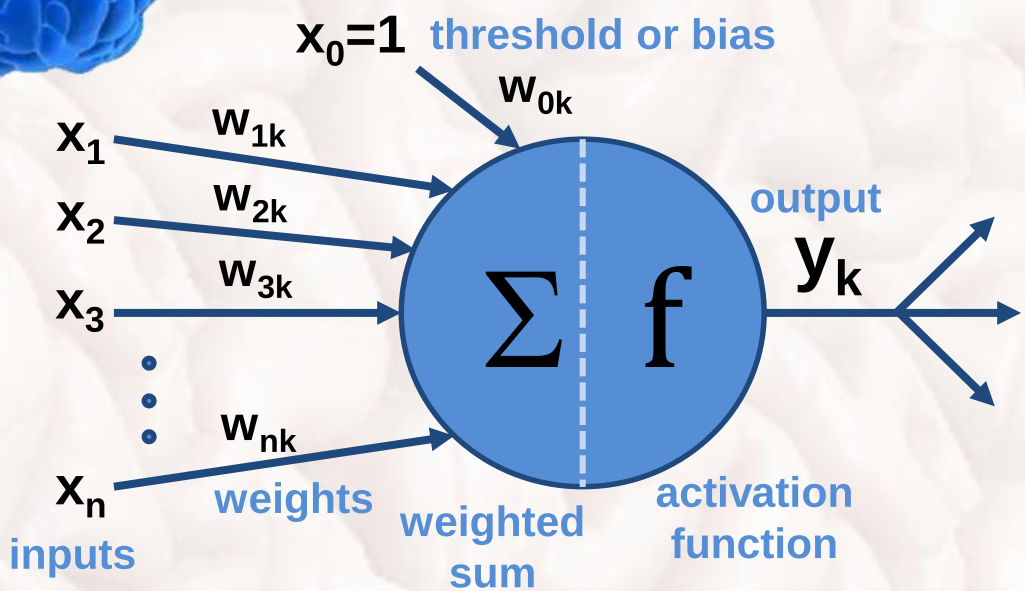
X



Artificial neurons usually:

1. Are detached from the rest nervous system components as the senses and their receptors, cerebrospinal fluid, glial cells and their non-neglectable functions in the nervous system.
2. Compute sums of weighted input signals without taking into account the automatic process of restoring insufficiently stimulated neurons and refracted neurons to the resting state after some time.
3. Neglect and do not define or use their position in the network structure (except a few types of networks, e.g. SOM).
4. Diminish the significance of an activation threshold by bringing it to another weight with constant stimulation (bias), except spiking neurons.
5. Change the natural ability of neurons to be activated into continuous and differential activation functions.
6. Do not take into account the different and variable size of neurons that affect its sensitivity and specialization.
7. Bring down synapses to an adaptive balance that can amplify input signals many times (biologically not plausible).
8. Do not take into account various periods of various internal neuronal processes taking place in biological neurons.
9. Are mostly connected on the each-to-each basis between layers, which usually prevents them from specializing in the selected input groups.
10. Do not make any automatic connection or functional changes during adaptation (learning) process, bringing them to nonlinear functions that can be combined in the artificial neural network creating complex approximators.

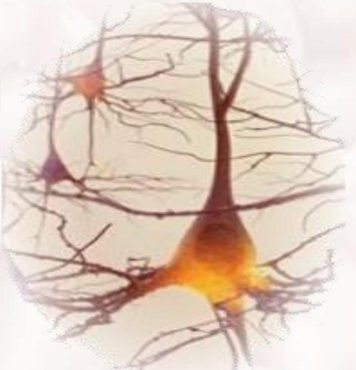
ARTIFICIAL NEURON



$$y_k = f\left(\sum_{i=0}^{n} w_{ik} x_i\right)$$

- Input data x_1 … x_n typically simultaneously affect an artificial neuron.
- Previous states of artificial neurons have no influence on their current state, only current stimulation and weights w_{0k}, w_{1k},…, w_{nk} are taken into account.
- No temporal relationships between the states are considered.
- The response of an artificial neuron is immediate and calculated after an activation function which value depends on the weighted sum of current inputs x_{0k}, x_{1k},…, x_{nk} and weights w_{0k}, w_{1k},…, w_{nk}.

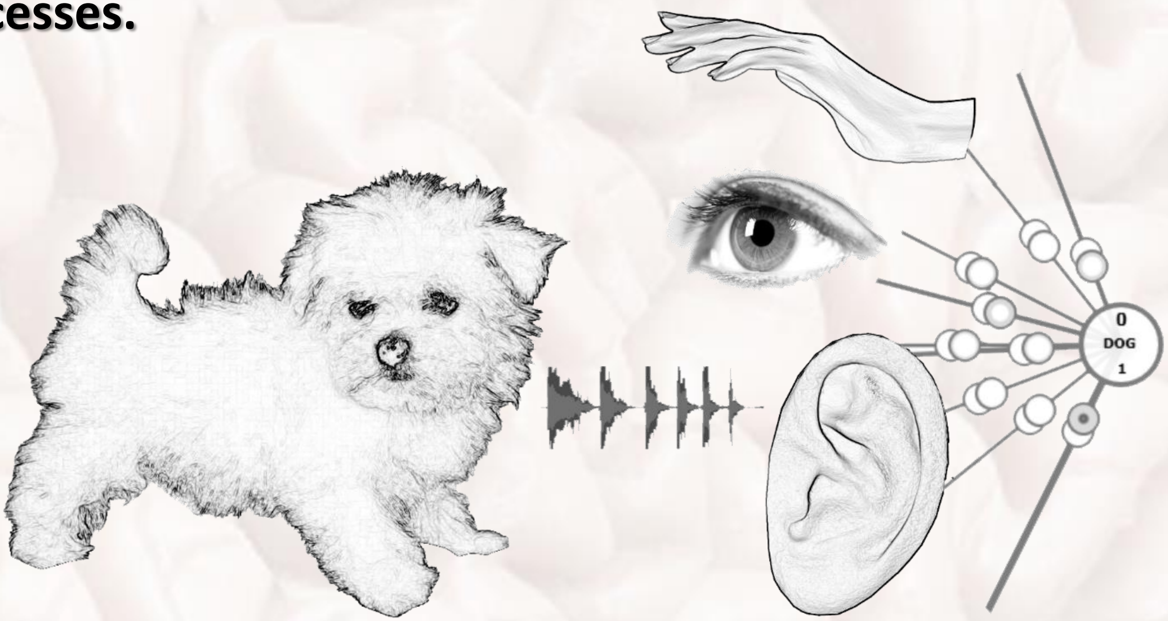
?



THE SENSES AND RECEPTORS

- They provide the nervous system with the necessary stimuli for its functioning, development, and adaptation.
- They stimulate the neurons with certain combinations of input stimuli, which we call training patterns.

- The brain would not be able to develop without the senses and their receptors.
- The stimuli coming from the receptors form some stimulus combinations.
- Such combinations can be further associated and memorized.
- The created associations are used as a context for future associations and mental processes.

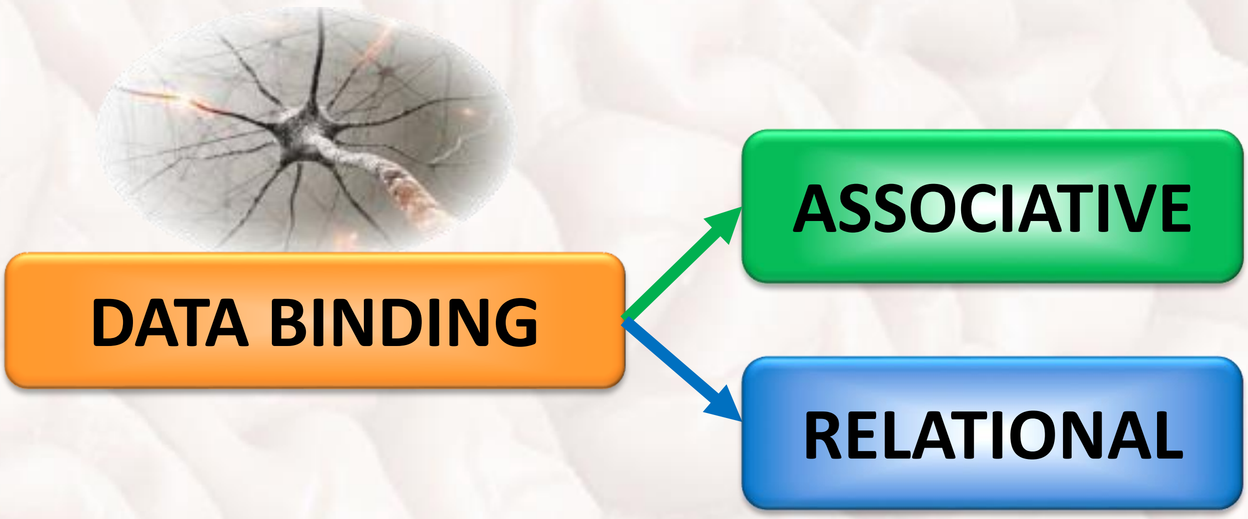


REASONS FOR THE ASSOCIATIVE REPRESENTATION OF DATA, OBJECTS, ACTIONS, AND FEATURES

Knowledge and intelligence allow us to quickly draw conclusions and make wise decisions thanks to the associations created and remembered in our minds.

The associative data representation is much richer and gives us far more possibilities than the most commonly used relational representation used in relational databases:

- **Relational databases – allow us only for horizontal data binding thanks to the primary and foreign keys representing relations between objects.**
- **Associative systems – allow us for both horizontal and vertical data binding combined with the aggregated representation of duplicates, which results in significant memory and computational time savings! Graph neural structures the with automatic vertical representation of data relationships <u>replace a lot of time-consuming operations</u>, which we have to perform on a relational database!**



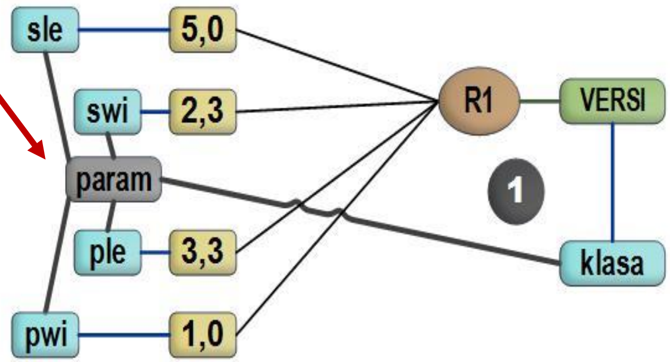
IRIS PATTERNS

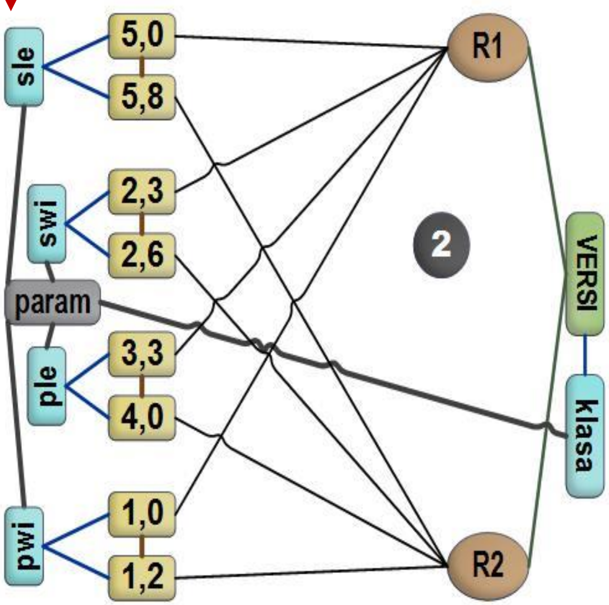
param	sle	swi	ple	pwi	klasa
R1	5,0	2,3	3,3	1,0	VERSI
R2	5,8	2,6	4,0	1,2	VERSI
R3	5,4	3,0	4,5	1,5	VERSI
R4	6,3	3,3	4,7	1,6	VERSI
R5	6,0	2,7	5,1	1,6	VERSI
R6	6,7	3,0	5,0	1,7	VERSI
R7	5,9	3,2	4,8	1,8	VERSI
R8	6,0	2,2	5,0	1,5	VIRGIN
R9	4,9	2,5	4,5	1,7	VIRGIN
R10	6,0	3,0	4,8	1,8	VIRGIN
R11	5,8	2,7	5,1	1,9	VIRGIN
R12	5,7	2,5	5,0	2,0	VIRGIN
R13	6,5	3,2	5,1	2,0	VIRGIN

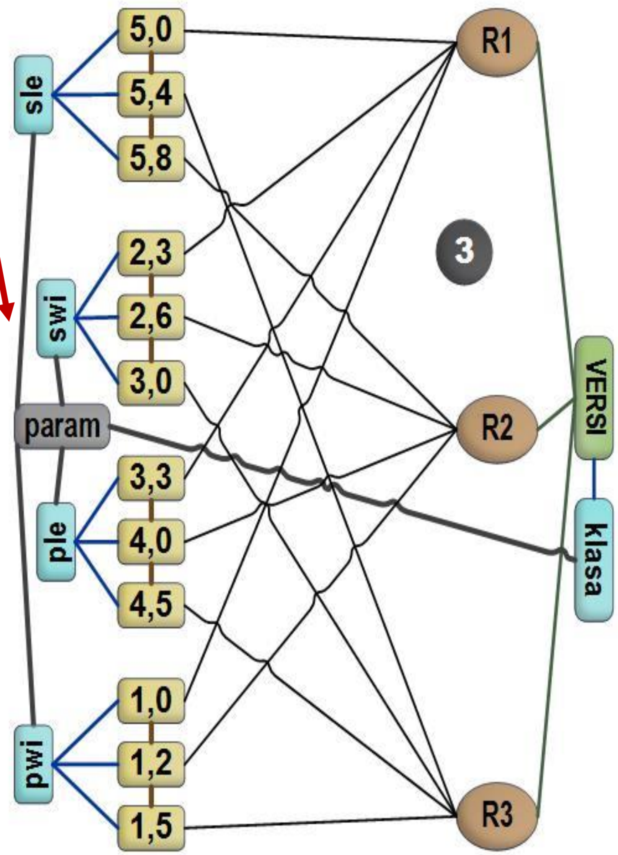
PASSIVE ASSOCIATIVE GRAPH DATA STRUCTURE - AGDS

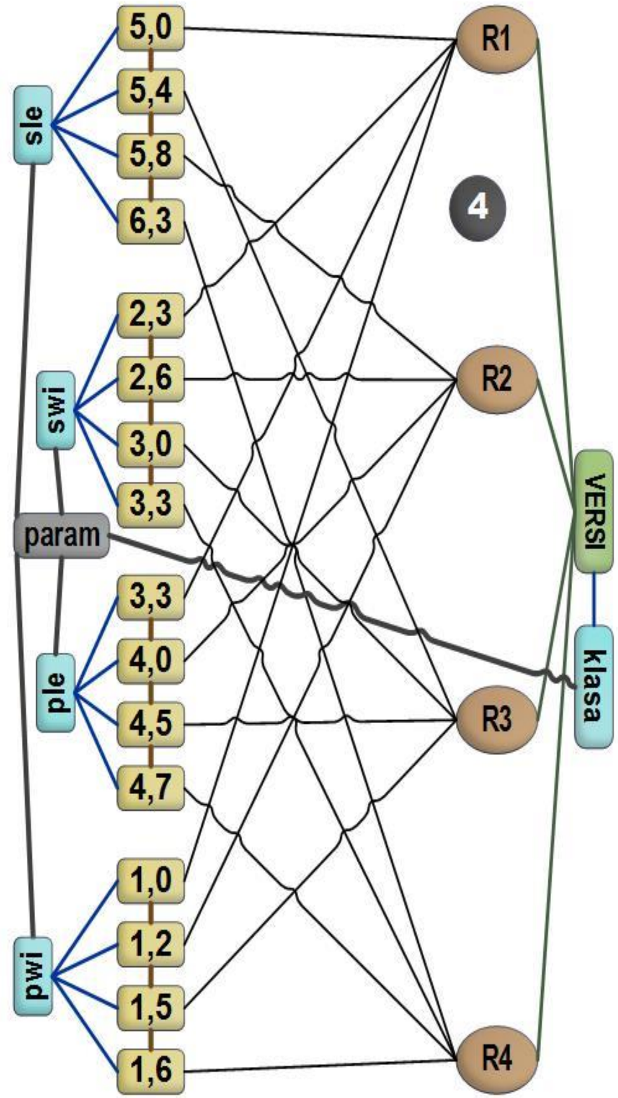
IRIS PATTERNS

param	sle	swi	ple	pwi	klasa
R1	5,0	2,3	3,3	1,0	VERSI
R2	5,8	2,6	4,0	1,2	VERSI
R3	5,4	3,0	4,5	1,5	VERSI
R4	6,3	3,3	4,7	1,6	VERSI
R5	6,0	2,7	5,1	1,6	VERSI
R6	6,7	3,0	5,0	1,7	VERSI
R7	5,9	3,2	4,8	1,8	VERSI
R8	6,0	2,2	5,0	1,5	VIRGIN
R9	4,9	2,5	4,5	1,7	VIRGIN
R10	6,0	3,0	4,8	1,8	VIRGIN
R11	5,8	2,7	5,1	1,9	VIRGIN
R12	5,7	2,5	5,0	2,0	VIRGIN
R13	6,5	3,2	5,1	2,0	VIRGIN

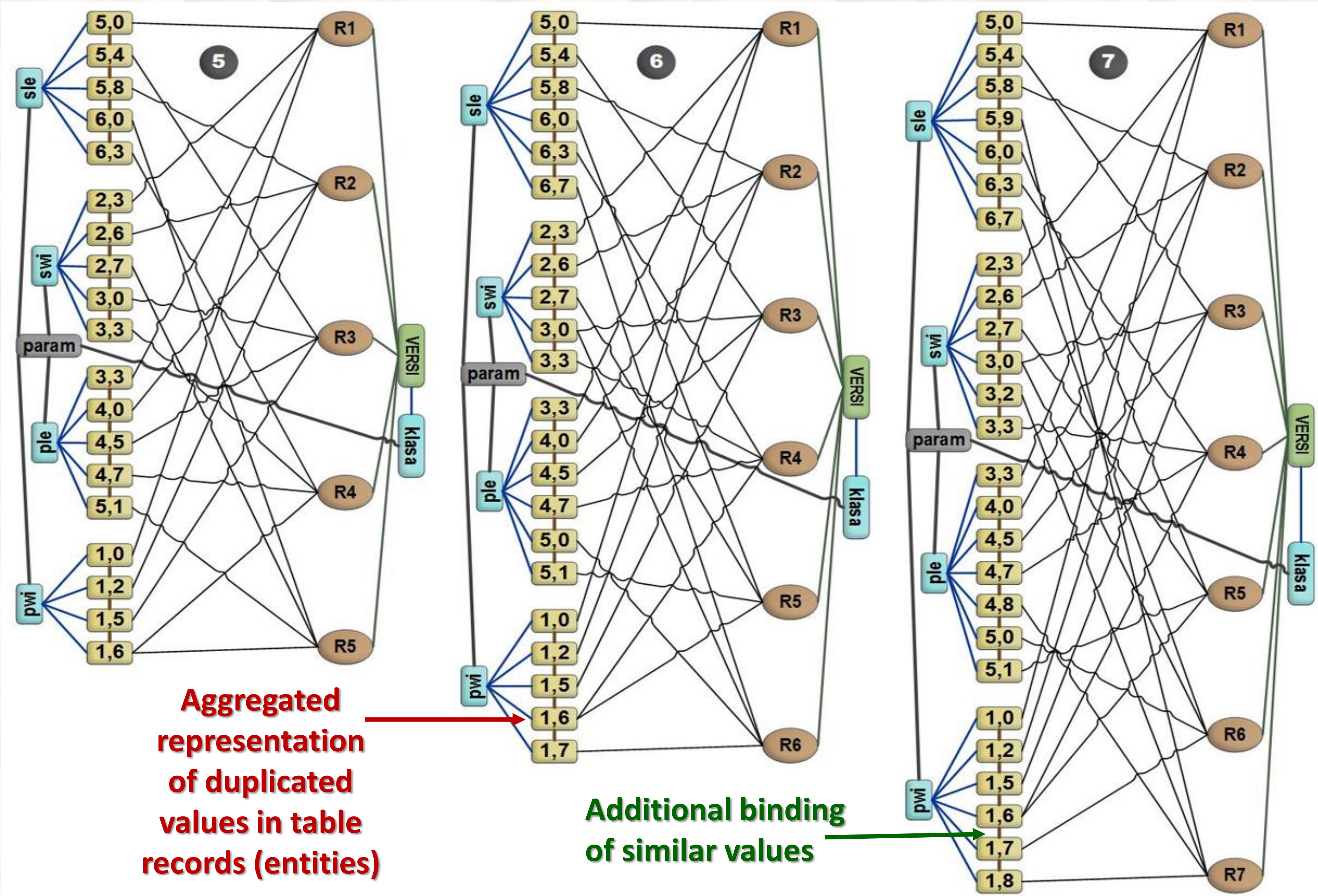








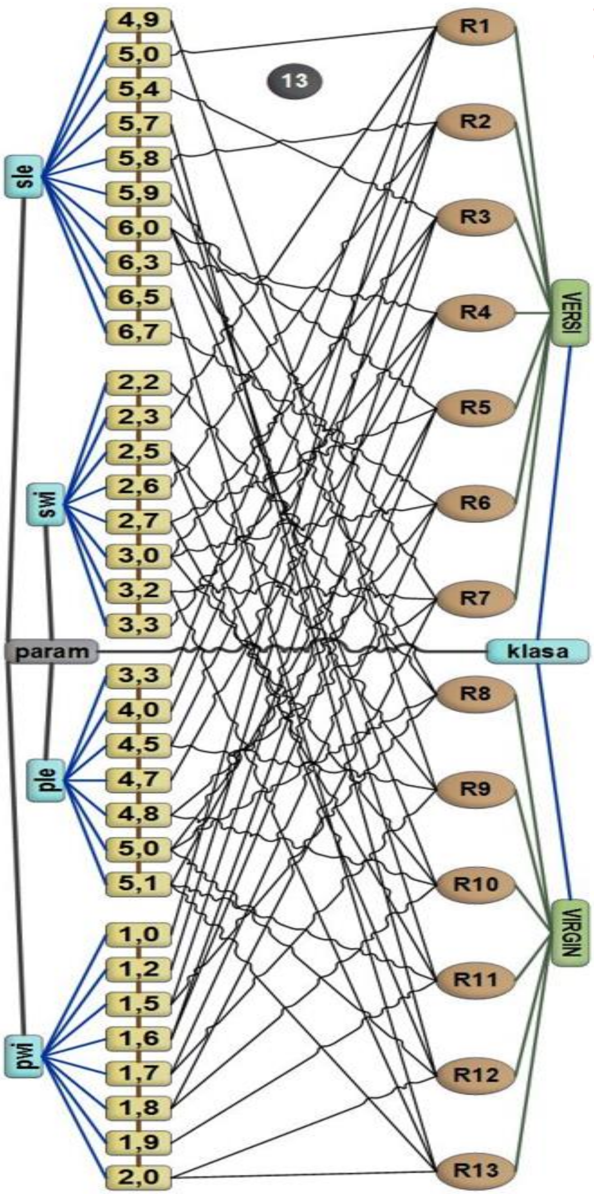
PASSIVE ASSOCIATIVE GRAPH DATA STRUCTURE - AGDS

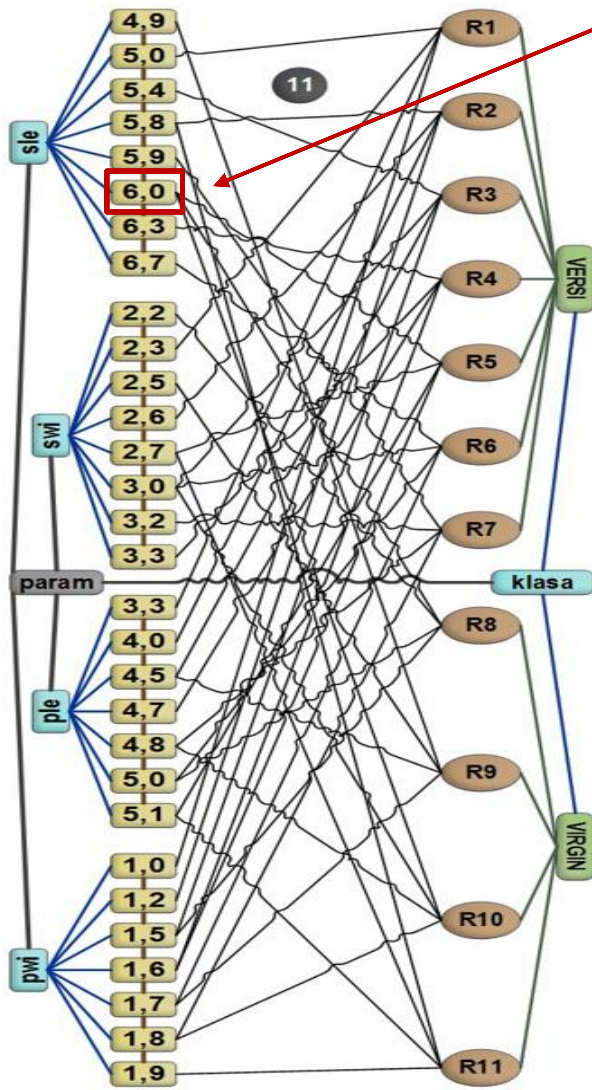


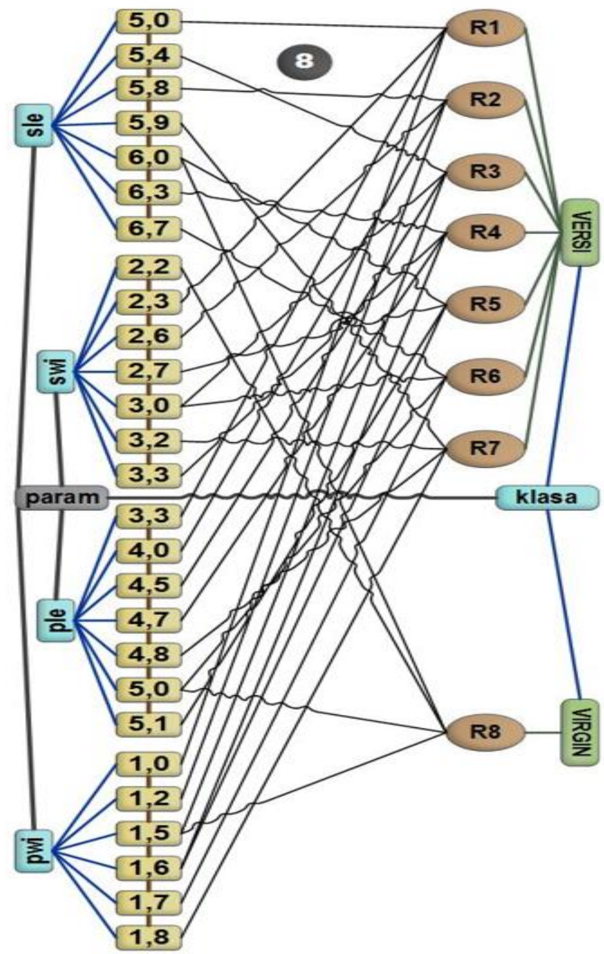
PASSIVE ASSOCIATIVE GRAPH DATA STRUCTURE - AGDS

The more data, the greater efficiency of these structures, cost-effectiveness and lossless compression of data representation.

R1	5,0	2,3	3,3	1,0	VERSI
R2	5,8	2,6	4,0	1,2	VERSI
R3	5,4	3,0	4,5	1,5	VERSI
R4	6,3	3,3	4,7	1,6	VERSI
R5	6,0	2,7	5,1	1,6	VERSI
R6	6,7	3,0	5,0	1,7	VERSI
R7	5,9	3,2	4,8	1,8	VERSI
R8	6,0	2,2	5,0	1,5	VIRGIN
R9	4,9	2,5	4,5	1,7	VIRGIN
R10	6,0	3,0	4,8	1,8	VIRGIN
R11	5,8	2,7	5,1	1,9	VIRGIN
R12	5,7	2,5	5,0	2,0	VIRGIN
R13	6,5	3,2	5,1	2,0	VIRGIN







VERTICAL SIMILARITIES BETWEEN OBJECTS

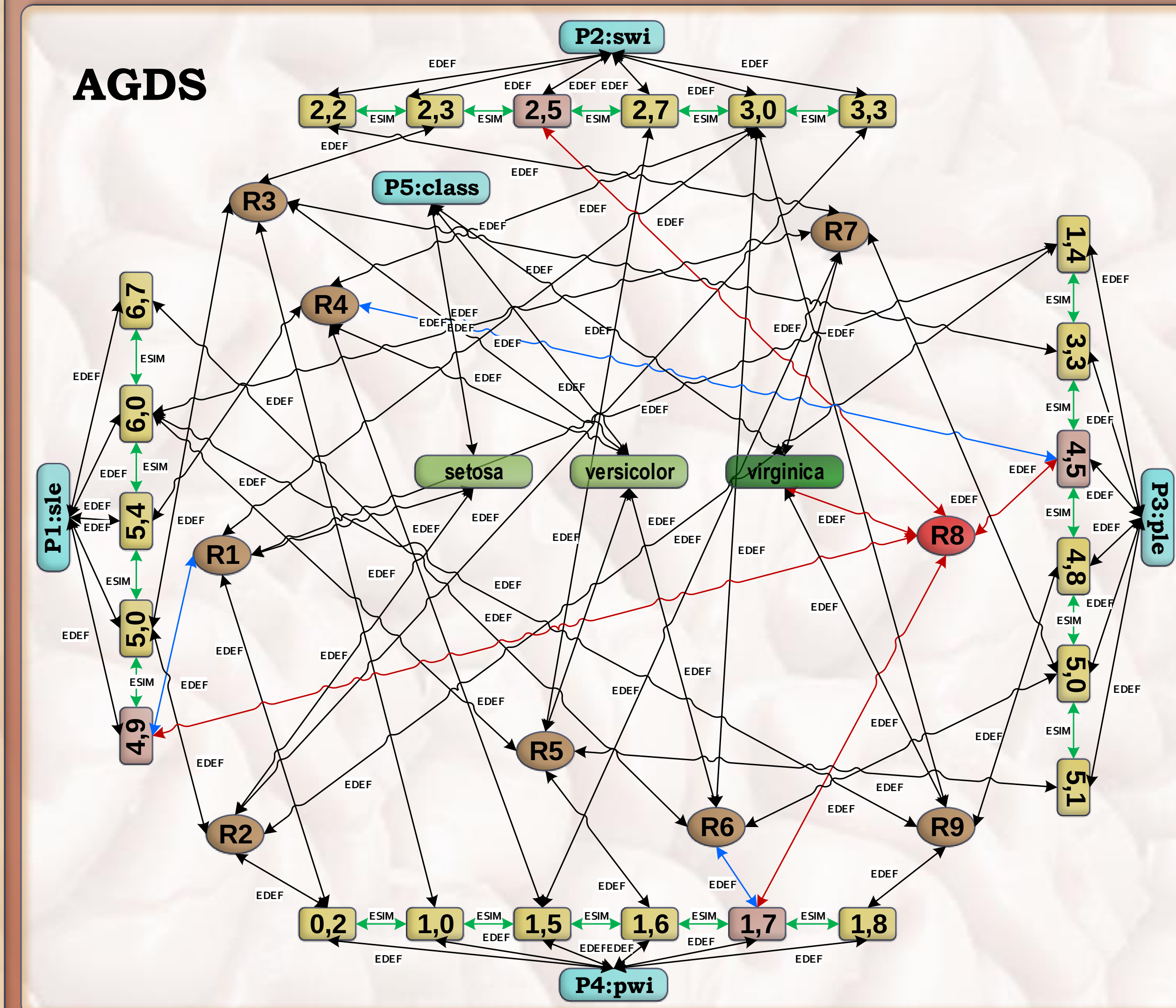
The connections point out related objects and similar data:

REPRESENTATION OF 9 IRIS SAMPLES IN TWO DATA STRUCTURES

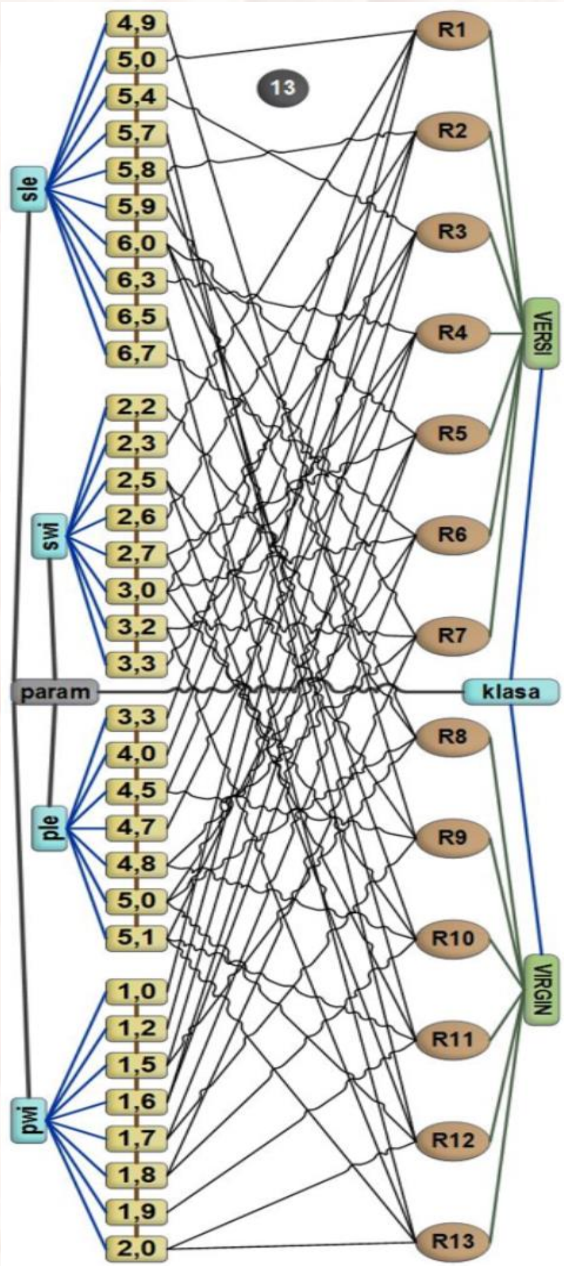
TABLE OF SAMPLES

	sle	swi	ple	pwi	class
R1	4,9	3,0	1,4	0,2	setosa
R2	5,0	3,3	1,4	0,2	setosa
R3	5,0	2,3	3,3	1,0	versicolor
R4	5,4	3,0	4,5	1,5	versicolor
R5	6,0	2,7	5,1	1,6	versicolor
R6	6,7	3,0	5,0	1,7	versicolor
R7	6,0	2,2	5,0	1,5	virginica
R8	4,9	2,5	4,5	1,7	virginica
R9	6,0	3,0	4,8	1,8	virginica

All similarities to other samples are immediately identified in the AGDS structure!



COMPARE STRUCTURES AND DRAW CONCLUSIONS



- What **data relations** can be **simply read** from these data structures and which **must be found**?
- What are **the pros and cons** of these structures?
- How do these structures affect the **computational efficiency** of operations on the stored data?
- Which structure is more suitable for efficient **knowledge exploration** and **data mining**?

IRIS PATTERNS

param	sle	swi	ple	pwi	klasa
R1	5,0	2,3	3,3	1,0	VERSI
R2	5,8	2,6	4,0	1,2	VERSI
R3	5,4	3,0	4,5	1,5	VERSI
R4	6,3	3,3	4,7	1,6	VERSI
R5	6,0	2,7	5,1	1,6	VERSI
R6	6,7	3,0	5,0	1,7	VERSI
R7	5,9	3,2	4,8	1,8	VERSI
R8	6,0	2,2	5,0	1,5	VIRGIN
R9	4,9	2,5	4,5	1,7	VIRGIN
R10	6,0	3,0	4,8	1,8	VIRGIN
R11	5,8	2,7	5,1	1,9	VIRGIN
R12	5,7	2,5	5,0	2,0	VIRGIN
R13	6,5	3,2	5,1	2,0	VIRGIN

CONNECTION WEIGHTS IN THE AGDS STRUCTURES

The AGDS nodes representing neighboring (subsequent) values of each attribute a_k are connected and the weight of this connection (edge) is computed by the following formula:

$$w_{v_i^{a_k}, v_j^{a_k}} = 1 - \frac{\left|v_i^{a_k} - v_j^{a_k}\right|}{r^{a_k}}$$

where

$v_i^{a_k}, v_j^{a_k}$ - are values represented by the neighboring attribute nodes, which are connected by an edge in the AGDS graph,

$r^{a_k} = v_{max}^{a_k} - v_{min}^{a_k}$- is the current range of values of the attribute a_k.

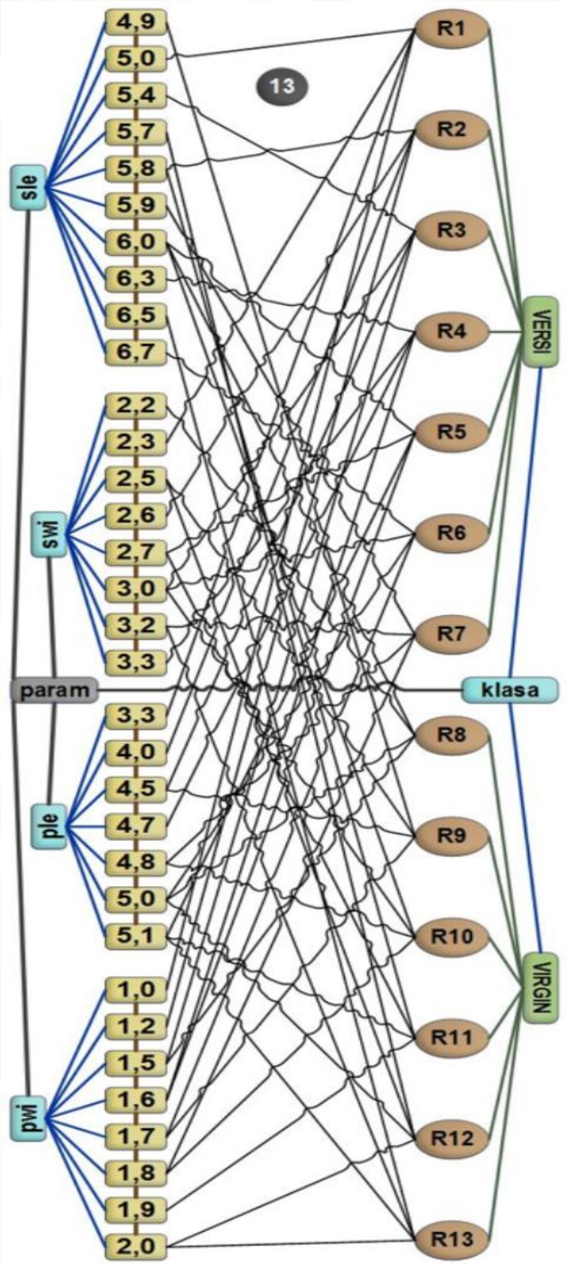
The weight of the connection from the value node $v_i^{a_k}$ of the attribute a_k to the object node R_m is determined after the number of occurrences $N_i^{a_k}$ of this value ($v_i^{a_k}$) in all objects:

$$w_{v_i^{a_k}, R_m} = \frac{1}{N_i^{a_k}} = \frac{1}{\left\|v_i^{a_k}\right\|}$$

These numbers ($N_i^{a_k} = \left\|v_i^{a_k}\right\|$) are stored in the individual value nodes of each attribute. This number is equal to the number or all connections of this value node to all object nodes if there are no duplicated objects in the table used to create the AGDS structure.

In the opposite direction, the weights of connections from the object nodes to the value nodes are always equal to one:

$$w_{R_m, v_i^{a_k}} = 1$$

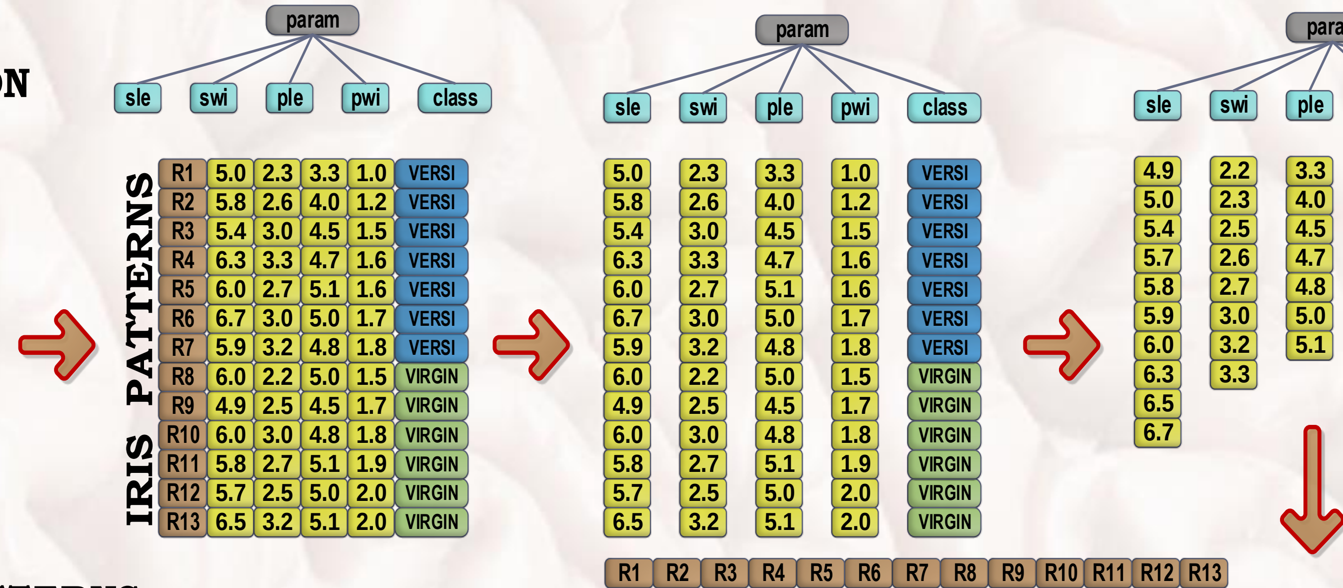


CREATION OF AGDS FOR A SINGLE DATABASE TABLE

ASSOCIATIVE TRANSFORMATION

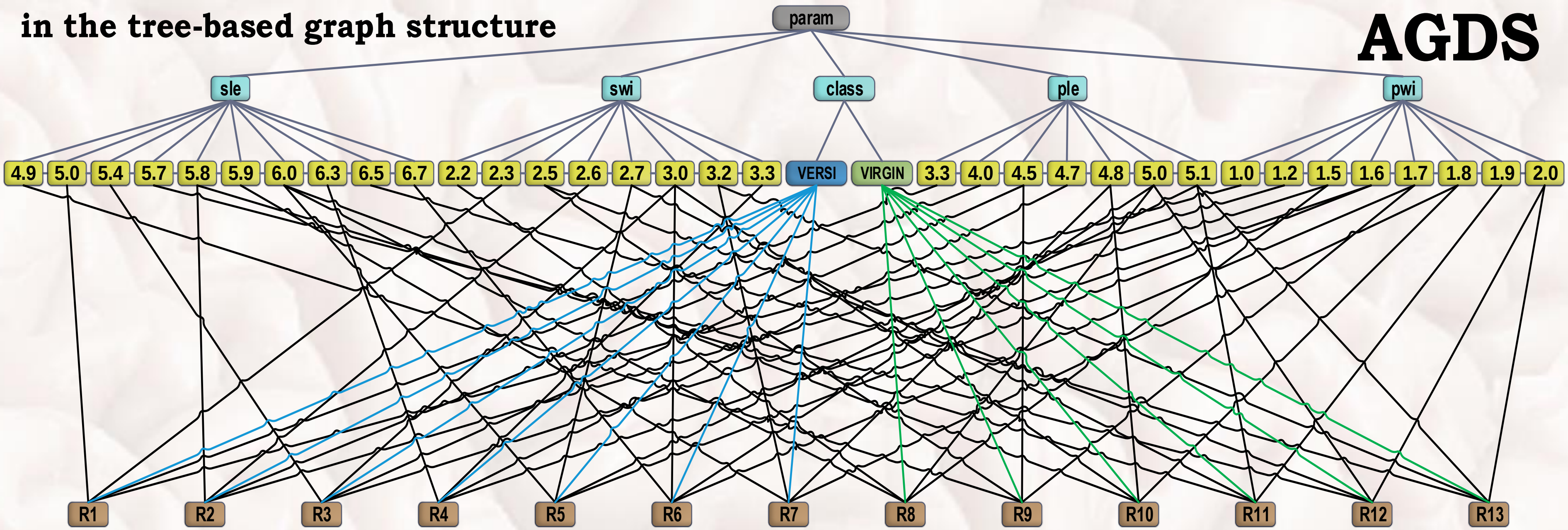
IRIS PATTERNS

param	sle	swi	ple	pwi	class
R1	5.0	2.3	3.3	1.0	VERSI
R2	5.8	2.6	4.0	1.2	VERSI
R3	5.4	3.0	4.5	1.5	VERSI
R4	6.3	3.3	4.7	1.6	VERSI
R5	6.0	2.7	5.1	1.6	VERSI
R6	6.7	3.0	5.0	1.7	VERSI
R7	5.9	3.2	4.8	1.8	VERSI
R8	6.0	2.2	5.0	1.5	VIRGIN
R9	4.9	2.5	4.5	1.7	VIRGIN
R10	6.0	3.0	4.8	1.8	VIRGIN
R11	5.8	2.7	5.1	1.9	VIRGIN
R12	5.7	2.5	5.0	2.0	VIRGIN
R13	6.5	3.2	5.1	2.0	VIRGIN



IRIS PATTERNS in the tree-based graph structure

AGDS

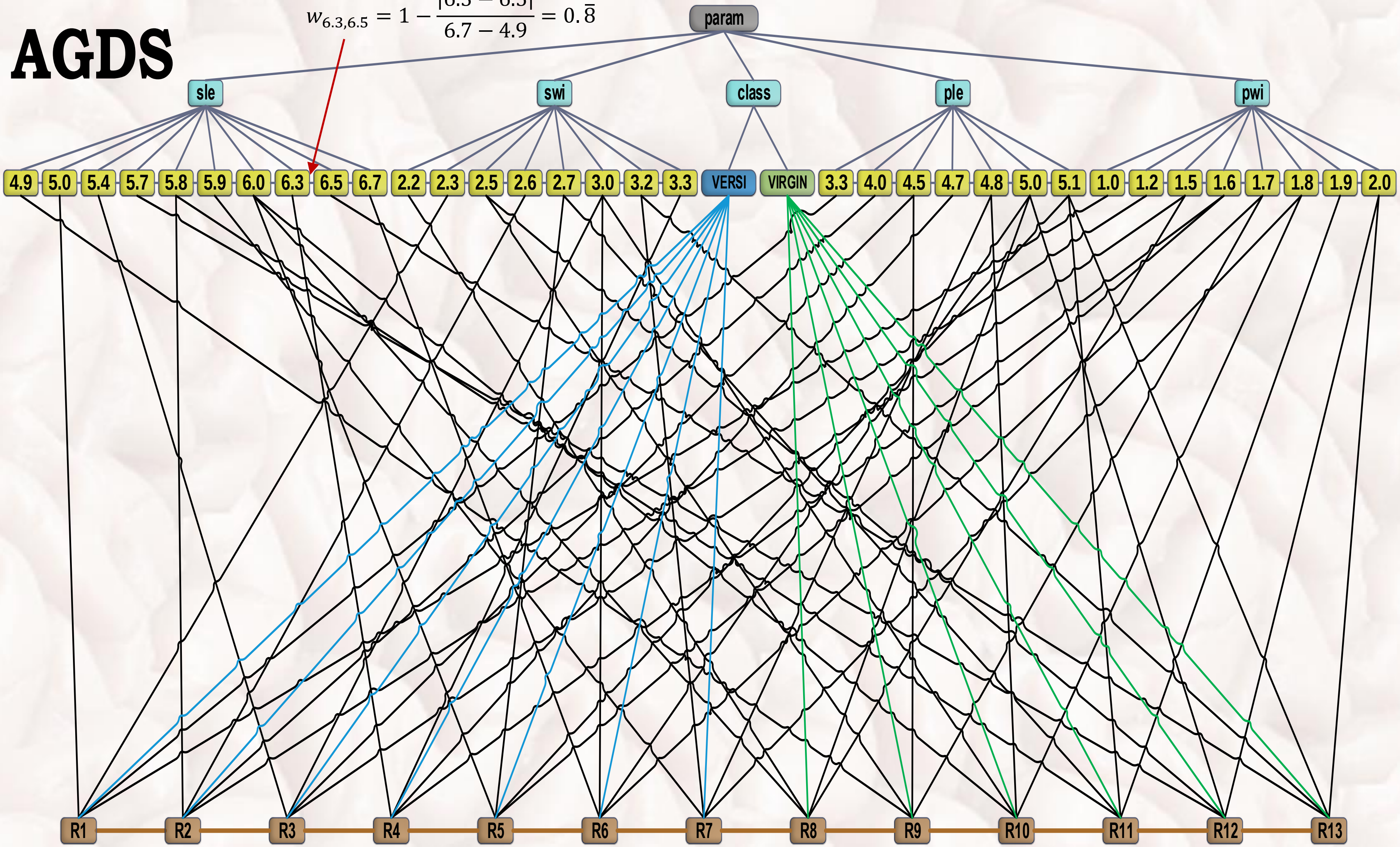


All elements can be quickly accessed through the param root node that has connections to all parameters etc.

THE TREE STRUCTURE USED IN PARALLEL COMPUTING

AGDS

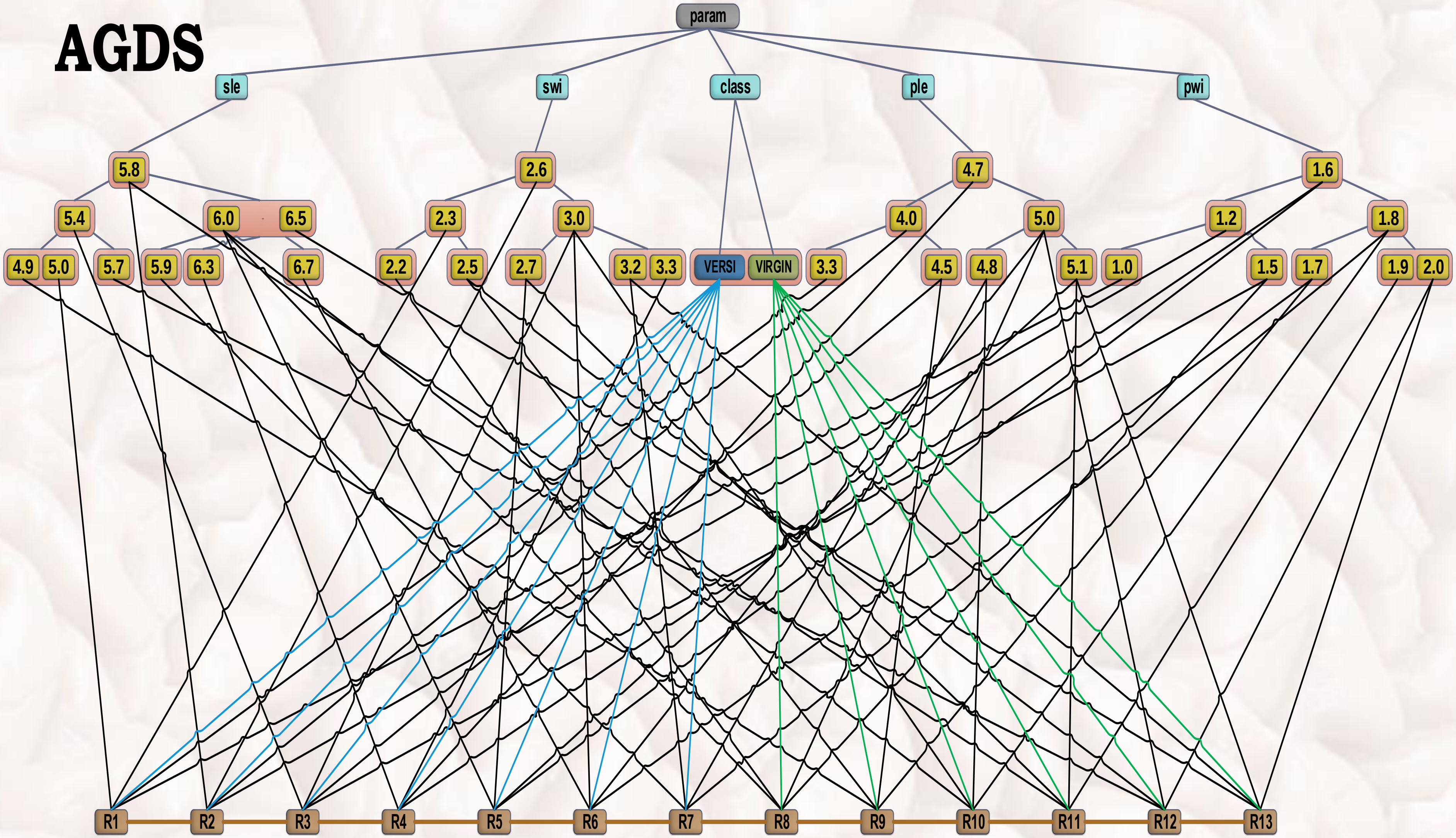
$$w_{6.3,6.5} = 1 - \frac{|6.3 - 6.5|}{6.7 - 4.9} = 0.\bar{8}$$



This tree-based graph gives you a very fast access to any data or relationships between these related and linked data. You can also draw various conclusions very fast.

ATTRIBUTE VALUE STRUCTURE IS BASED ON AVB-TREES

AGDS



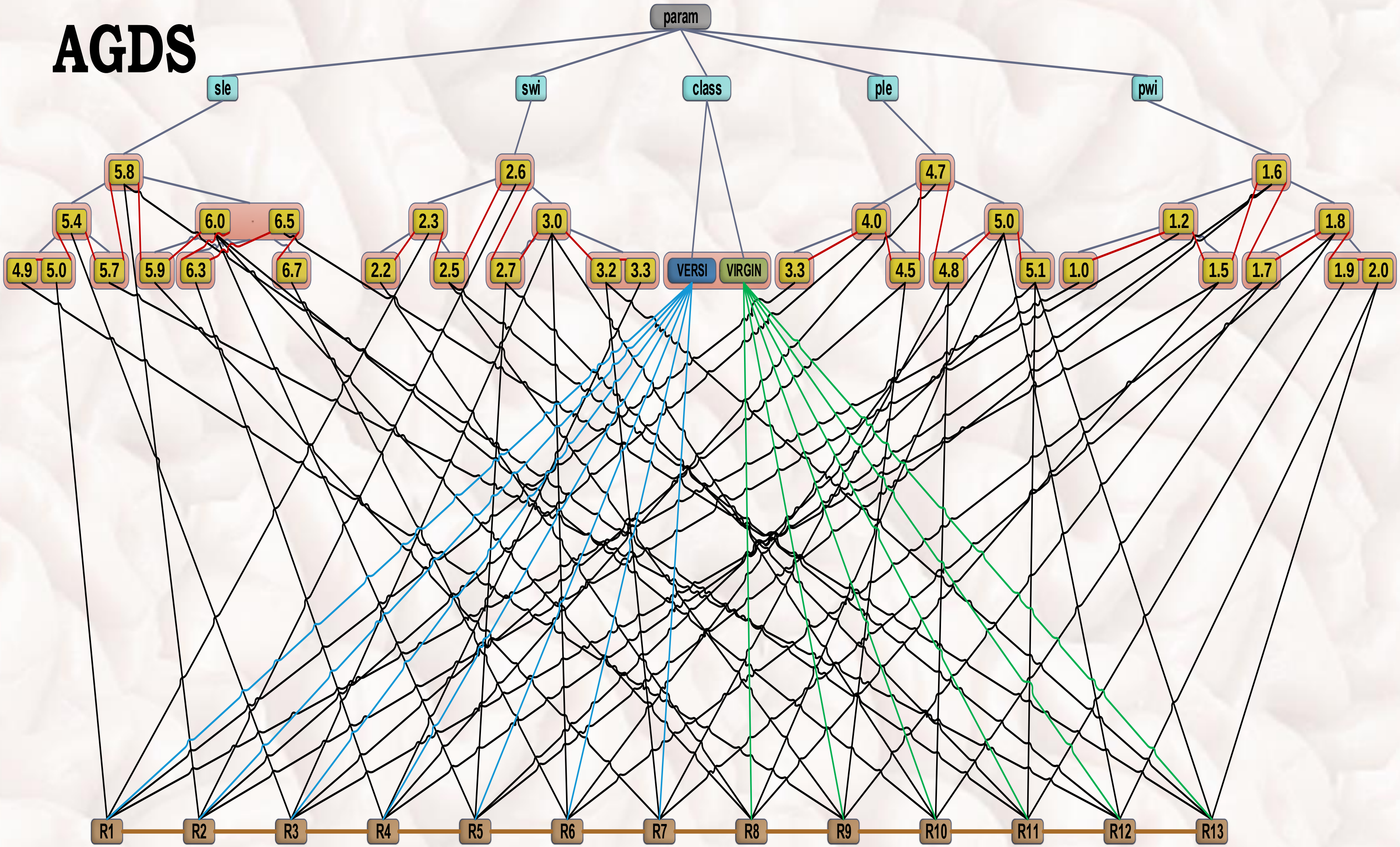
In the case of sequential (non-parallel) implementation of the AGDS structure, AVB-trees are used.

The AVB-trees are the simple modification of B-trees, which aggregate representation of duplicates.

The AVB-trees contain only unique attribute values for efficient access to them; duplicates are reduced.

ATTRIBUTE VALUE STRUCTURE IS BASED ON AVB+TREES

AGDS

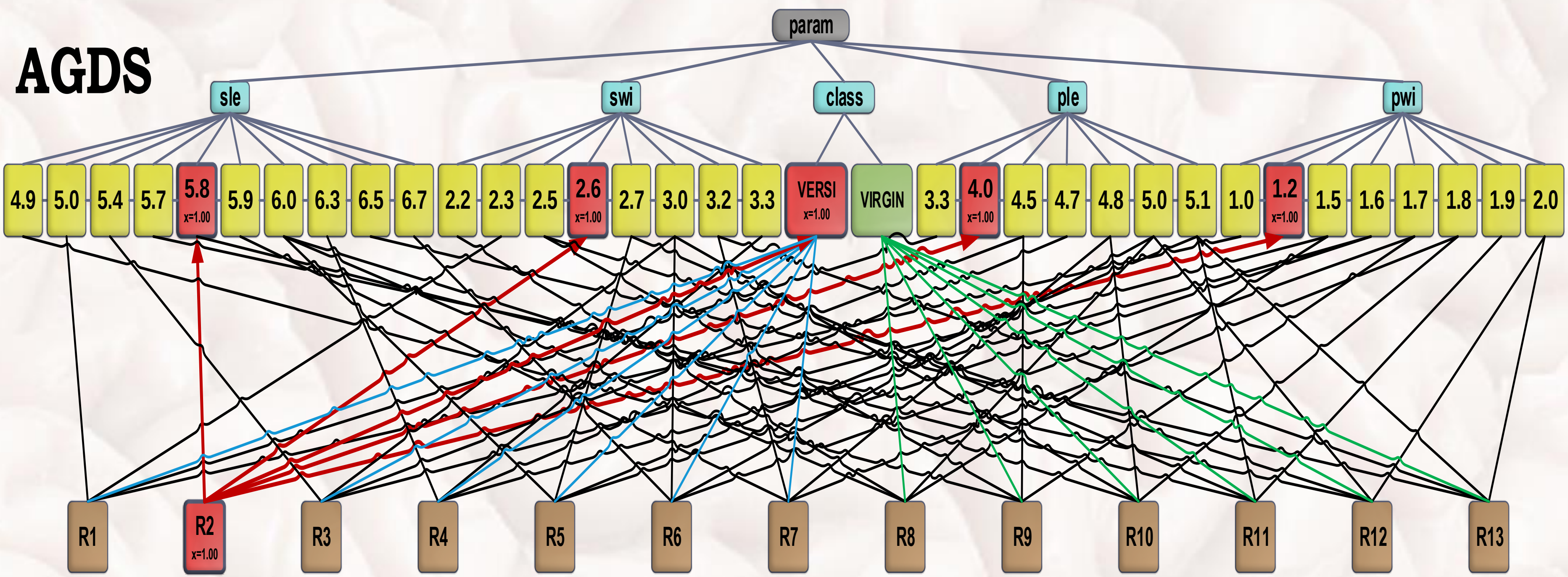


The subsequent values (keys of AVB+trees) can be additionally connected to reproduce proximity between represented unique keys, however, we can also use the AVB-tree structure to quickly find them.

ASSOCIATIVE INFERENCE USING AGDS STRUCTURES

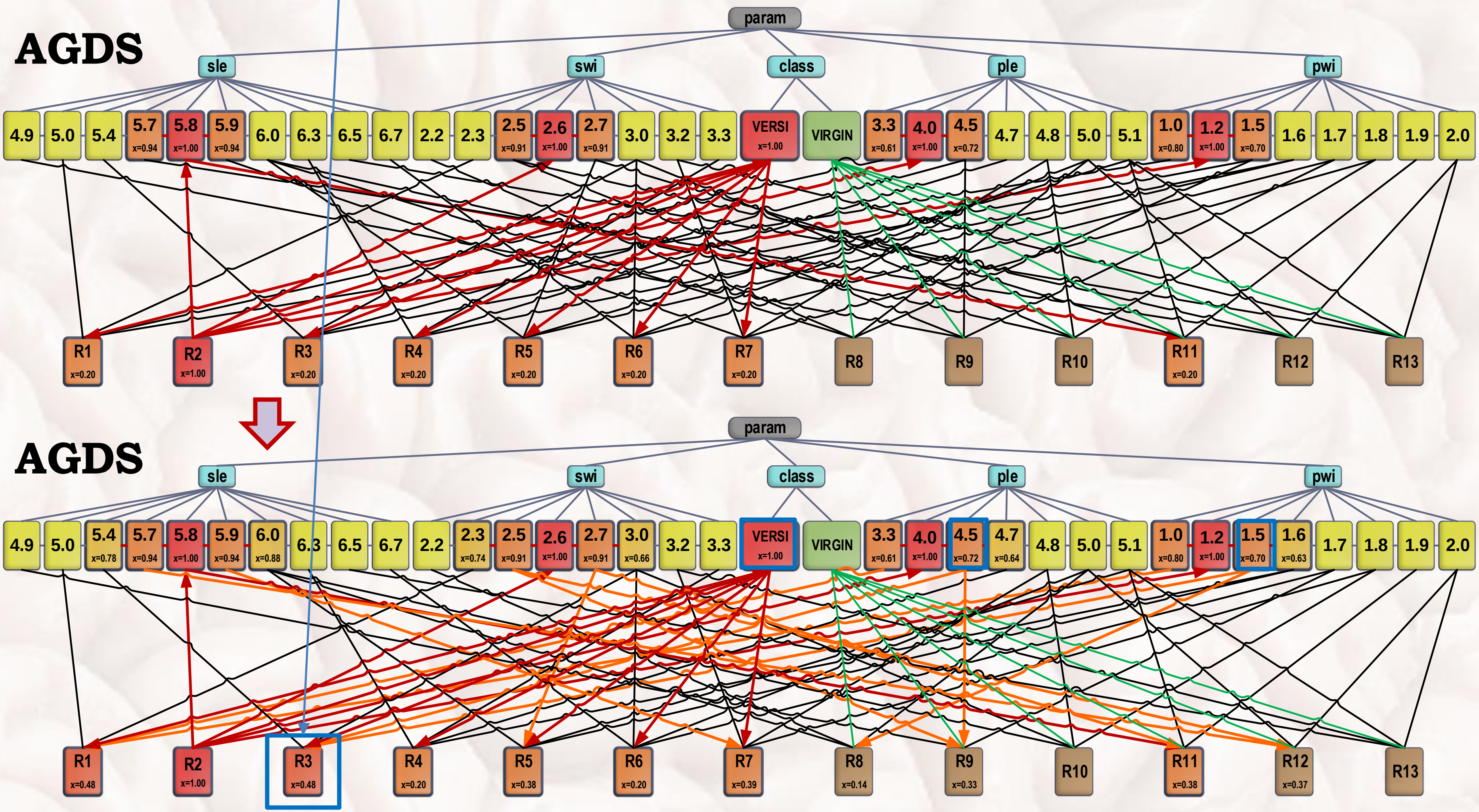
Associative data structures AGDS can be now used for associative inference, which is based on moving along the connections to the connected nodes and computing some values in these nodes on the basis of the send values multiplied by weights of these connections. In such a way we get the information about, e.g. similarity of objects represented by other nodes of the same kind or about the objects that satisfy some given conditions defined by the represented attribute values. Let's use our AGDS graph created for 13 Irises for such inference looking for objects (Irises) Rx which are most similar to R2.

1. **We start in the node R2 which assumes the similarity value x=1.0 because this node is 100% similar to itself.**
2. **Next, we assign values x of the connected nodes representing the following values: 5.8, 2.6, VERSI, 4.0, and 1.2 by multiplying the value coming from the node R2 with the connection weights, which are equal 1.0. So, as a result, we achieve x=1.0 for all these connected nodes.**



ASSOCIATIVE INFERENCE USING AGDS STRUCTURES

3. Subsequently, the values computed for these nodes are multiplied by next connection weights and send to the neighbor connected value nodes, for which we also compute their similarity values x.
4. Similarly, we compute the similarity values x for connected object nodes with regards to the necessity to add the passed weighted values to the sums already stored in these nodes, e.g. for the node R3 we compute $x = 1.0 * 0.2 + 0.72 * 0.2 + 0.7 * 0.2 = 0.48$

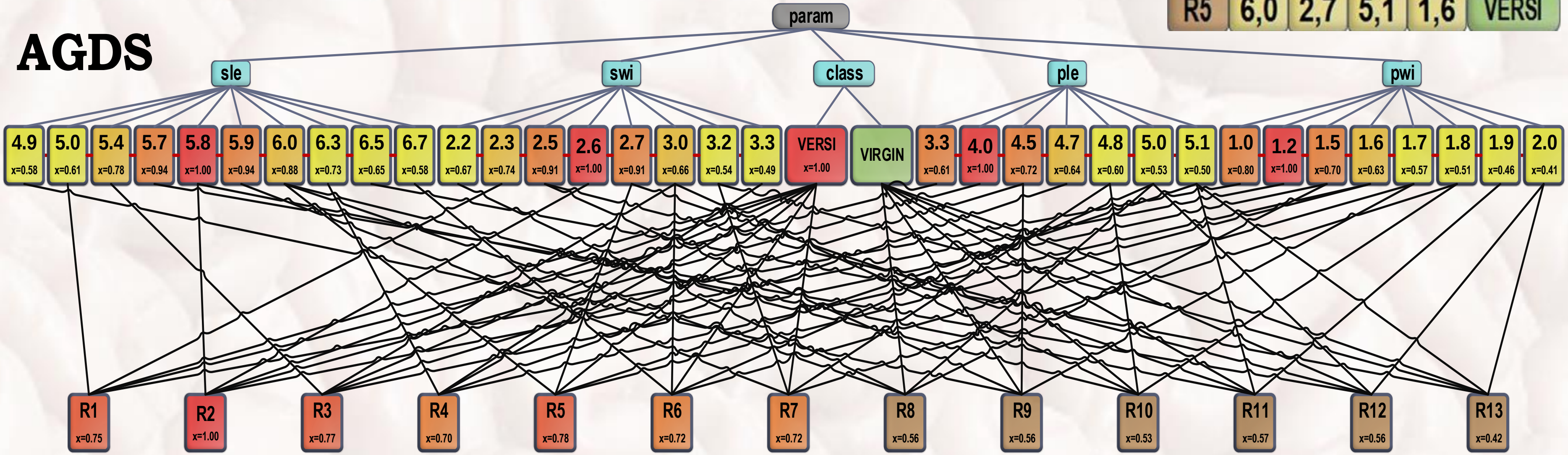


ASSOCIATIVE INFERENCE USING AGDS STRUCTURES

5. Finally, when we go through all the connected (associated) values nodes computing theirs values of similarities by multiplying the sender similarity values by connection weights. We also computed weighted sums for all object nodes, where these weights are the same w = 1/5 = 0.2. The computed similarity values for the nodes Rx can be used to compare and designate the most similar objects to the object R2: R5 (78%), R3 (77%), R1 (75%), …

R1	5,0	2,3	3,3	1,0	VERSI
R2	5,8	2,6	4,0	1,2	VERSI
R3	5,4	3,0	4,5	1,5	VERSI
R4	6,3	3,3	4,7	1,6	VERSI
R5	6,0	2,7	5,1	1,6	VERSI

AGDS



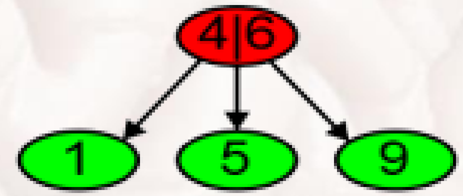
It is also worth noting that AGDS graphs are not neural structures, so we are not obligated to multiply the nodes similarity values by connection weights, but we can also use other formulas, e.g. we can subtract the complement of the connection weight value from the similarity value represented by the sender: x' = x - (1 – w).

Consequently, we get another measure of similarity represented by the value nodes and object nodes.

We can also use DASNG graph formulas to calculate weights between value nodes and object nodes to emphasize the rarity of the value using the frequency of connections coming out from value nodes: w = 1 / the number of outgoing connections.

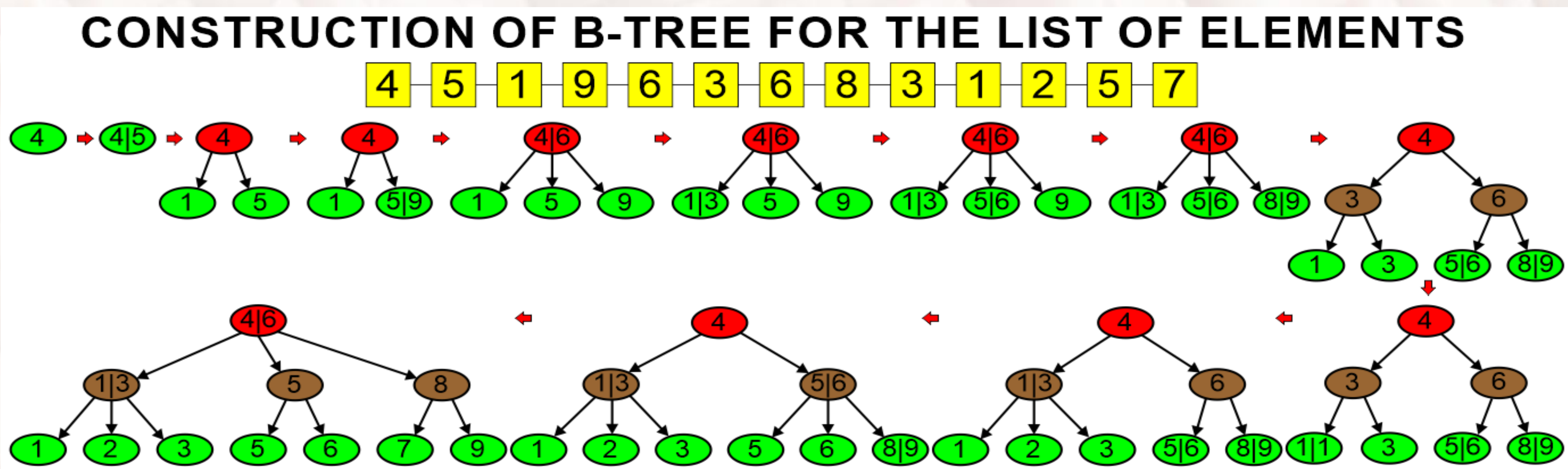
CONSTRUCTION OF B-TREES

B-trees are often used to created indices for attributes in relational databases. The construction of B-trees is a complex process that requires performing specific operations to restore assumptions and conditions: https://www.cs.usfca.edu/~galles/visualization/BTree.html



The addition of a new element to the B-tree consists of several steps:

1. Go from the root of the tree to one of its leaves after the following rules:
   - Go to the left if the key is less or equal to the left key value of the parent node,
   - Go to the right if the key is bigger than the right key value of the parent node,
   - Go in the middle if the key is bigger than the left key value of the parent node and less or equal to the right key value of the parent node.
2. When you get to the leaf, add the new key to it in order if it does not yet store two keys.
3. If it already contains two keys, divide this node into two nodes, leaving the smallest key in its left node, the biggest key in its right node, and pass the middle key to its parent node. If the parent node does not yet exist, create it. The parent node will be connected to these two child nodes.
4. If the parent node contains already two keys, the passed key is added in order and the parent node is also divided in the same way, creating two nodes and passing its middle key to its parent or creating it.

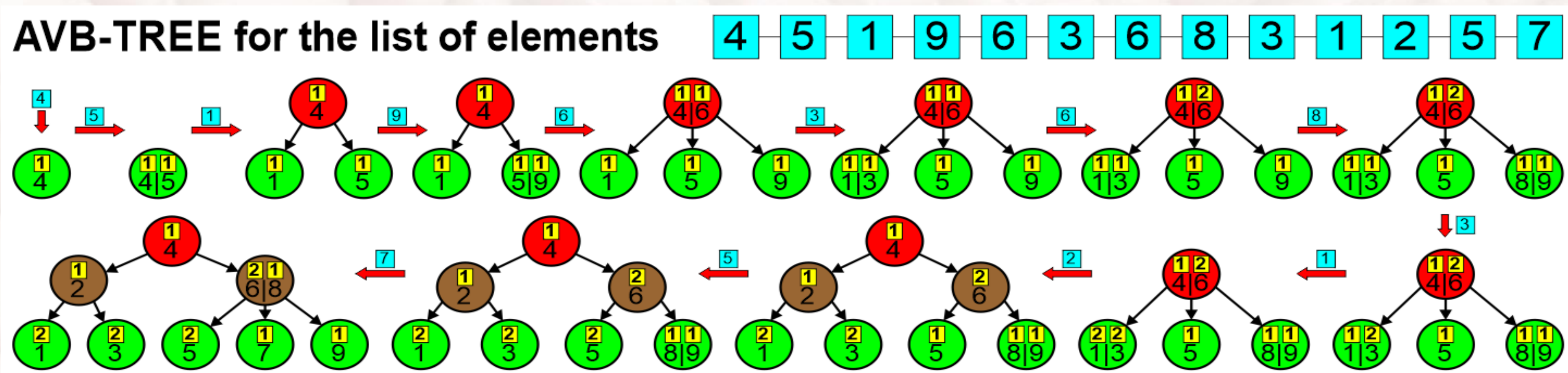


The operation of removal the keys from the B-TREE structure cannot violate the B-tree properties.

CONSTRUCTION OF AVB-TREES

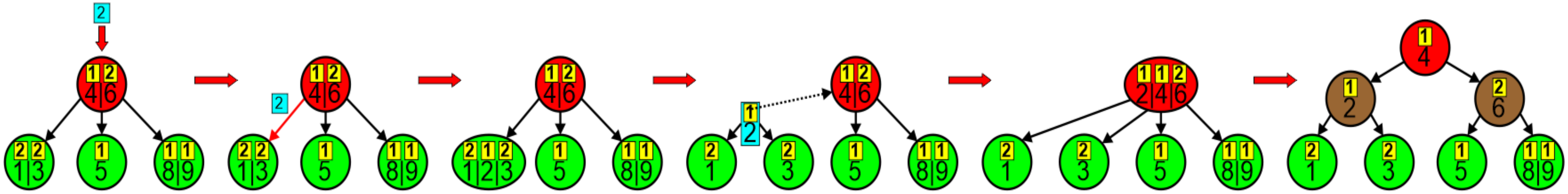
AVB-trees are a simple modification of B-trees. AVB-trees aggregate the same (duplicated) values, represent them in a single node and count them up in order to know how many values have been aggregated to be able to remove the key representing several aggregated values correctly. Addition of a new key to the AVB-tree:

1. Go from the root of the tree to one of its leaves after the following rules:
   - Go to the left if the key is less to the left key value of the parent node,
   - Go to the right if the key is bigger than the right key value of the parent node,
   - Go in the middle if the key is bigger than the left key value and less than the right key value of the parent node.
   - Increment the counter of the left or right key of parent node if the added element is equal to one of them, and stop the descent process to the leaves.
2. When you get to the leaf, and the element is not equal to any key in it, add the new key to it in order if it does not yet store two keys.
3. If it already contains two keys, divide this node into two nodes, leaving the smallest key in its left node, the biggest key in its right node, and pass the middle key to its parent node. If the parent node does not yet exist, create it. The parent node will be connected to these two child nodes.
4. If the parent node contains already two keys, the passed key is added in order and the parent node is also divided in the same way, creating two nodes and passing its middle key to its parent or creating it.
5. If the leaf contains a key that is equal to the added element, increment its counter.



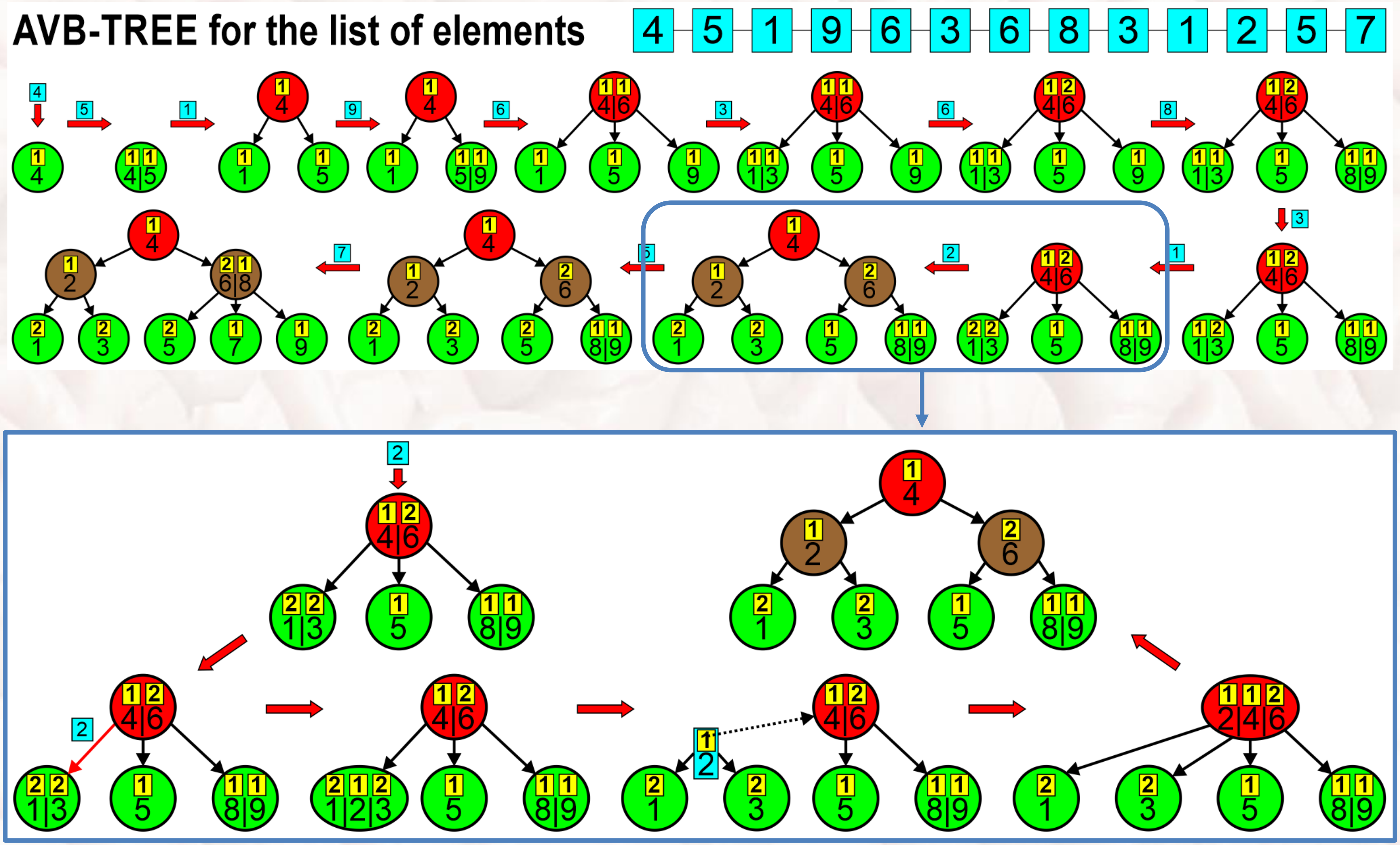
INSERTION OF THE KEY TO THE AVB-TREE

1. Start from the root and go recursively down along the edges to the descendants until the leaf is not achieved after the following rules:
   - a) Go to the left if the inserted key is less than the most left key in the node.
   - b) Go to the right if the inserted key is greater than the most right key in the node.
   - c) Go in the middle if the node contains two keys an the inserted key is greater than the left key and less then the right key.
   - d) Increment the counter of the key in the node that equals to the inserted keys.
2. When the leaf is achieved:
   - a) and if the inserted key is equal one of the keys in this leaf, increment the counter of this key.
   - b) else insert the inserted key to the keys stored in this leaf in the increasing order, initialize its counter to one, and go to the step 3.
3. If the number of all keys stored in this leaf is greater then two, divide this node into two nodes: let the new left leaf represent the left (least) key together with its counter, the new right leaf represent the right (greatest) key together with its counter, and the middle key together with its counter and the pointers the new leaves pass to the parent node if it exists, and go to the step 4; if the parent node does not exist, create it (a new root of the AVB tree) and let it represent this middle key together with its counter and create new edges for the passed pointers to the new leaves.
4. Insert the passed key together with its counter to the key(s) represented in this node in the increasing order: if the key comes from the left branch, insert it on the left side of the key(s); if the key comes from the right branch, insert it on the right side of the key(s); if the key comes from the middle branch, insert it between the keys.
5. If the number of all keys stored in this node is equal to two, create two new edges for the passed pointers to the two divided nodes, where the edges are appropriately connected before and after the passed key in order instead of the edge that passed the key.
6. If the number of all keys stored in this node is greater then two, divide this node into two nodes: let the new left node represent the left (least) key together with its counter and connect the two leftmost edges to it; the new right node represent the right (greatest) key together with its counter and connect the two rightmost edges to it; and the middle key together with its counter and the pointers to the divided nodes pass to the parent node if it exists, and go to the step 4; if the parent node does not exist, create it (a new root of the AVB tree) and let it represent this middle key together with its counter, create new edges, and connect them to the divided nodes.



INTERMEDIATE STEPS OF PASSING THE MIDDLE KEYS

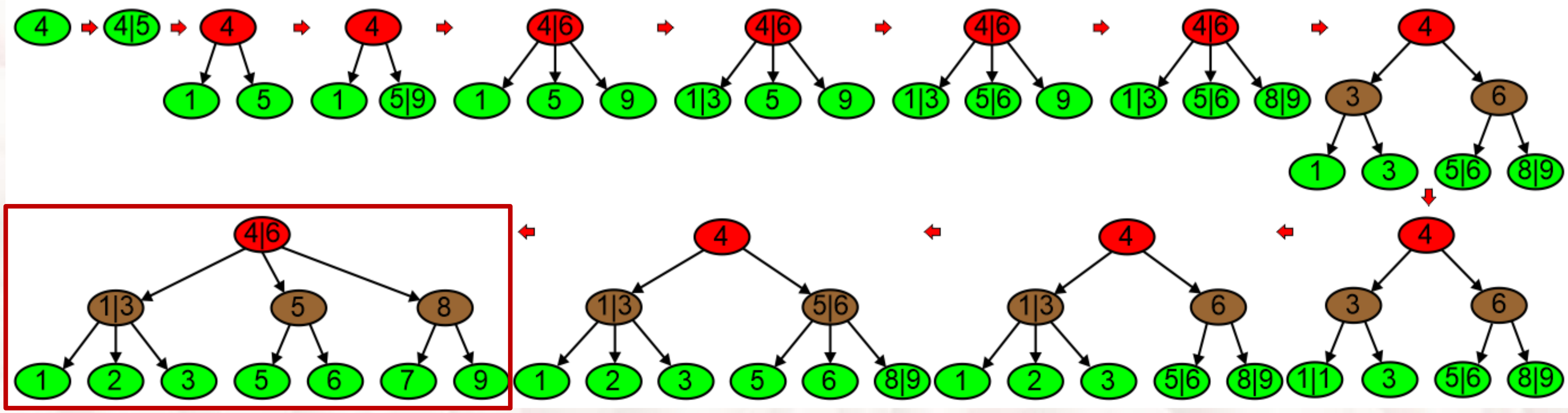
The intermediate steps of passing the middle key together with its counter and pointers to the new edges of the divided leaves/nodes to the parent node after the division of a leaf or a node or the creation of anew root.



COMPARISON OF THE B-TREES AND AVB-TREES

CONSTRUCTION OF B-TREE FOR THE LIST OF ELEMENTS

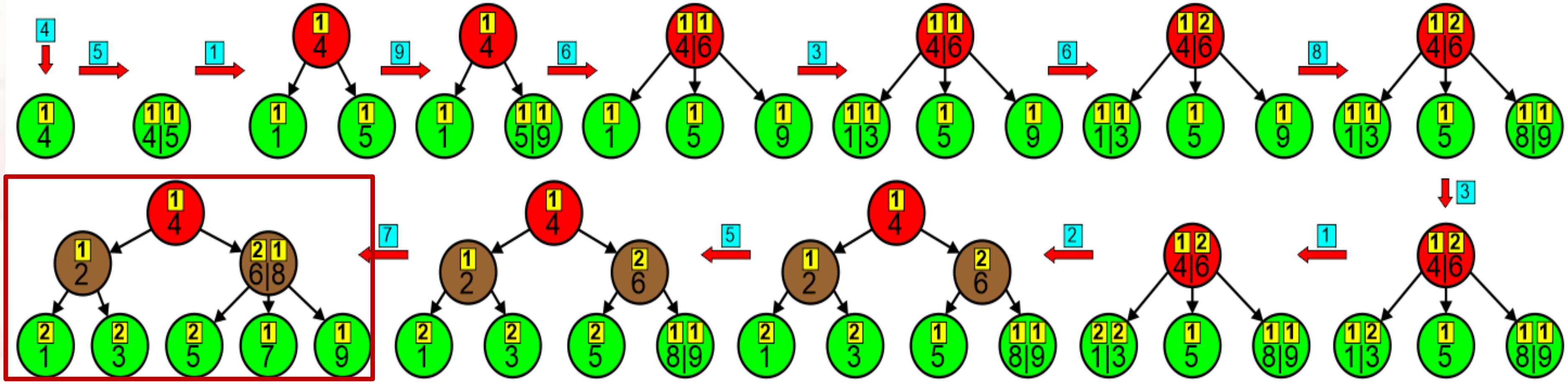
4 5 1 9 6 3 6 8 3 1 2 5 7



AVB-trees are smaller and more cost-effective.

AVB-TREE for the list of elements

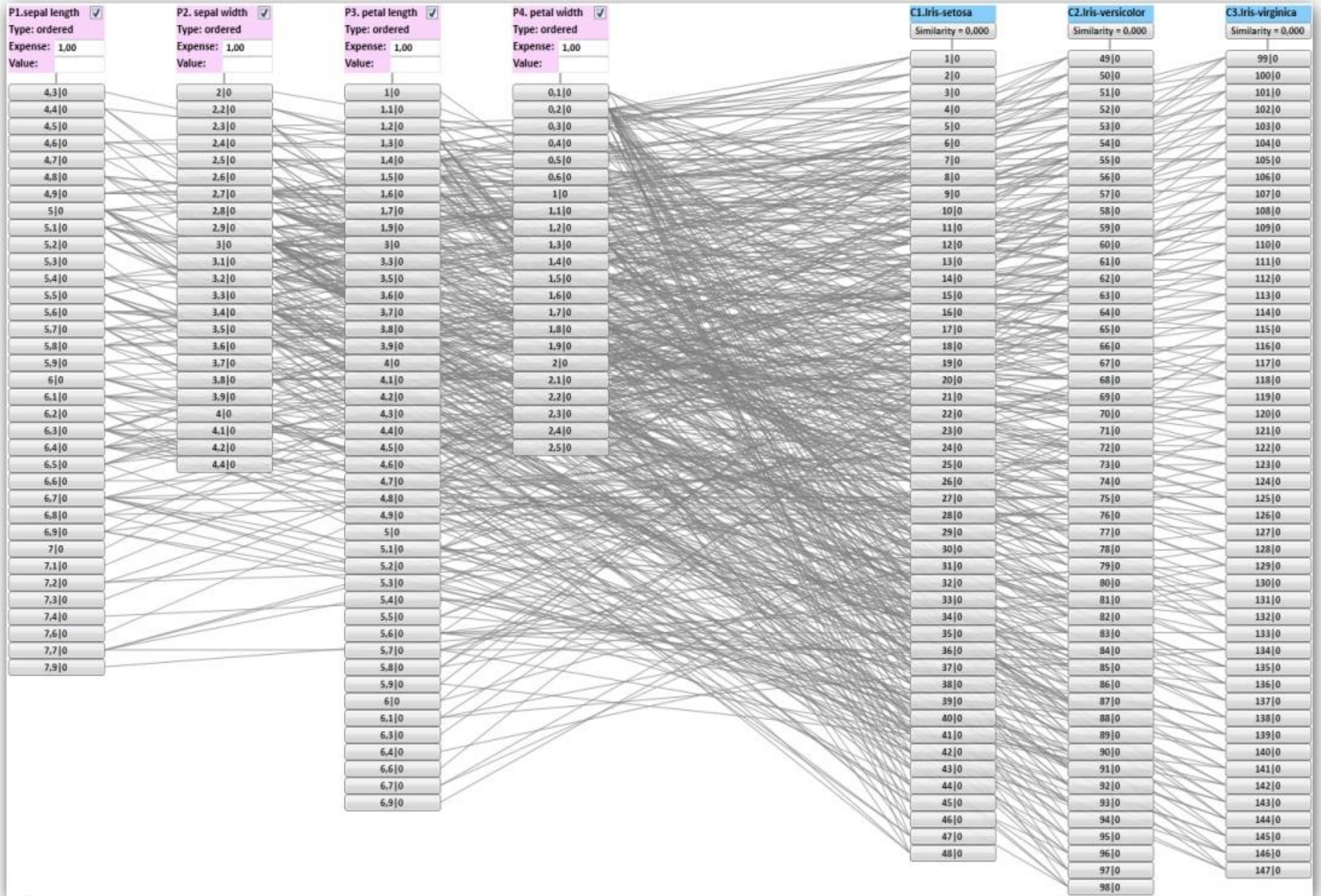
4 5 1 9 6 3 6 8 3 1 2 5 7



The AGDS for the IRIS data from ML Repository

Lp.	klasa	długość liścia	szerokość liścia	długość płatka	szerokość płatka	Lp.	klasa	długość liścia	szerokość liścia	długość płatka	szerokość płatka	Lp.	klasa	długość liścia	szerokość liścia	długość płatka	szerokość płatka
1	Iris Setosa	5,1	3,5	1,4	0,2	51	Iris Versicolor	7,0	3,2	4,7	1,4	101	Iris Virginica	6,3	3,3	6,0	2,5
2	Iris Setosa	4,9	3,0	1,4	0,2	52	Iris Versicolor	6,4	3,2	4,5	1,5	102	Iris Virginica	5,8	2,7	5,1	1,9
3	Iris Setosa	4,7	3,2	1,3	0,2	53	Iris Versicolor	6,9	3,1	4,9	1,5	103	Iris Virginica	7,1	3,0	5,9	2,1
4	Iris Setosa	4,6	3,1	1,5	0,2	54	Iris Versicolor	5,5	2,3	4,0	1,3	104	Iris Virginica	6,3	2,9	5,6	1,8
5	Iris Setosa	5,0	3,6	1,4	0,2	55	Iris Versicolor	6,5	2,8	4,6	1,5	105	Iris Virginica	6,5	3,0	5,8	2,2
6	Iris Setosa	5,4	3,9	1,7	0,4	56	Iris Versicolor	5,7	2,8	4,5	1,3	106	Iris Virginica	7,6	3,0	6,6	2,1
7	Iris Setosa	4,6	3,4	1,4	0,3	57	Iris Versicolor	6,3	3,3	4,7	1,6	107	Iris Virginica	4,9	2,5	4,5	1,7
8	Iris Setosa	5,0	3,4	1,5	0,2	58	Iris Versicolor	4,9	2,4	3,3	1,0	108	Iris Virginica	7,3	2,9	6,3	1,8
9	Iris Setosa	4,4	2,9	1,4	0,2	59	Iris Versicolor	6,6	2,9	4,6	1,3	109	Iris Virginica	6,7	2,5	5,8	1,8
10	Iris Setosa	4,9	3,1	1,5	0,1	60	Iris Versicolor	5,2	2,7	3,9	1,4	110	Iris Virginica	7,2	3,6	6,1	2,5
11	Iris Setosa	5,4	3,7	1,5	0,2	61	Iris Versicolor	5,0	2,0	3,5	1,0	111	Iris Virginica	6,5	3,2	5,1	2,0
12	Iris Setosa	4,8	3,4	1,6	0,2	62	Iris Versicolor	5,9	3,0	4,2	1,5	112	Iris Virginica	6,4	2,7	5,3	1,9
13	Iris Setosa	4,8	3,0	1,4	0,1	63	Iris Versicolor	6,0	2,2	4,0	1,0	113	Iris Virginica	6,8	3,0	5,5	2,1
14	Iris Setosa	4,3	3,0	1,1	0,1	64	Iris Versicolor	6,1	2,9	4,7	1,4	114	Iris Virginica	5,7	2,5	5,0	2,0
15	Iris Setosa	5,8	4,0	1,2	0,2	65	Iris Versicolor	5,6	2,9	3,6	1,3	115	Iris Virginica	5,8	2,8	5,1	2,4
16	Iris Setosa	5,7	4,4	1,5	0,4	66	Iris Versicolor	6,7	3,1	4,4	1,4	116	Iris Virginica	6,4	3,2	5,3	2,3
17	Iris Setosa	5,4	3,9	1,3	0,4	67	Iris Versicolor	5,6	3,0	4,5	1,5	117	Iris Virginica	6,5	3,0	5,5	1,8
18	Iris Setosa	5,1	3,5	1,4	0,3	68	Iris Versicolor	5,8	2,7	4,1	1,0	118	Iris Virginica	7,7	3,8	6,7	2,2
19	Iris Setosa	5,7	3,8	1,7	0,3	69	Iris Versicolor	6,2	2,2	4,5	1,5	119	Iris Virginica	7,7	2,6	6,9	2,3
20	Iris Setosa	5,1	3,8	1,5	0,3	70	Iris Versicolor	5,6	2,5	3,9	1,1	120	Iris Virginica	6,0	2,2	5,0	1,5
21	Iris Setosa	5,4	3,4	1,7	0,2	71	Iris Versicolor	5,9	3,2	4,8	1,8	121	Iris Virginica	6,9	3,2	5,7	2,3
22	Iris Setosa	5,1	3,7	1,5	0,4	72	Iris Versicolor	6,1	2,8	4,0	1,3	122	Iris Virginica	5,6	2,8	4,9	2,0
23	Iris Setosa	4,6	3,6	1,0	0,2	73	Iris Versicolor	6,3	2,5	4,9	1,5	123	Iris Virginica	7,7	2,8	6,7	2,0
24	Iris Setosa	5,1	3,3	1,7	0,5	74	Iris Versicolor	6,1	2,8	4,7	1,2	124	Iris Virginica	6,3	2,7	4,9	1,8
25	Iris Setosa	4,8	3,4	1,9	0,2	75	Iris Versicolor	6,4	2,9	4,3	1,3	125	Iris Virginica	6,7	3,3	5,7	2,1
26	Iris Setosa	5,0	3,0	1,6	0,2	76	Iris Versicolor	6,6	3,0	4,4	1,4	126	Iris Virginica	7,2	3,2	6,0	1,8
27	Iris Setosa	5,0	3,4	1,6	0,4	77	Iris Versicolor	6,8	2,8	4,8	1,4	127	Iris Virginica	6,2	2,8	4,8	1,8
28	Iris Setosa	5,2	3,5	1,5	0,2	78	Iris Versicolor	6,7	3,0	5,0	1,7	128	Iris Virginica	6,1	3,0	4,9	1,8
29	Iris Setosa	5,2	3,4	1,4	0,2	79	Iris Versicolor	6,0	2,9	4,5	1,5	129	Iris Virginica	6,4	2,8	5,6	2,1
30	Iris Setosa	4,7	3,2	1,6	0,2	80	Iris Versicolor	5,7	2,6	3,5	1,0	130	Iris Virginica	7,2	3,0	5,8	1,6
31	Iris Setosa	4,8	3,1	1,6	0,2	81	Iris Versicolor	5,5	2,4	3,8	1,1	131	Iris Virginica	7,4	2,8	6,1	1,9
32	Iris Setosa	5,4	3,4	1,5	0,4	82	Iris Versicolor	5,5	2,4	3,7	1,0	132	Iris Virginica	7,9	3,8	6,4	2,0
33	Iris Setosa	5,2	4,1	1,5	0,1	83	Iris Versicolor	5,8	2,7	3,9	1,2	133	Iris Virginica	6,4	2,8	5,6	2,2
34	Iris Setosa	5,5	4,2	1,4	0,2	84	Iris Versicolor	6,0	2,7	5,1	1,6	134	Iris Virginica	6,3	2,8	5,1	1,5
35	Iris Setosa	4,9	3,1	1,5	0,1	85	Iris Versicolor	5,4	3,0	4,5	1,5	135	Iris Virginica	6,1	2,6	5,6	1,4
36	Iris Setosa	5,0	3,2	1,2	0,2	86	Iris Versicolor	6,0	3,4	4,5	1,6	136	Iris Virginica	7,7	3,0	6,1	2,3
37	Iris Setosa	5,5	3,5	1,3	0,2	87	Iris Versicolor	6,7	3,1	4,7	1,5	137	Iris Virginica	6,3	3,4	5,6	2,4
38	Iris Setosa	4,9	3,1	1,5	0,1	88	Iris Versicolor	6,3	2,3	4,4	1,3	138	Iris Virginica	6,4	3,1	5,5	1,8
39	Iris Setosa	4,4	3,0	1,3	0,2	89	Iris Versicolor	5,6	3,0	4,1	1,3	139	Iris Virginica	6,0	3,0	4,8	1,8
40	Iris Setosa	5,1	3,4	1,5	0,2	90	Iris Versicolor	5,5	2,5	4,0	1,3	140	Iris Virginica	6,9	3,1	5,4	2,1
41	Iris Setosa	5,0	3,5	1,3	0,3	91	Iris Versicolor	5,5	2,6	4,4	1,2	141	Iris Virginica	6,7	3,1	5,6	2,4
42	Iris Setosa	4,5	2,3	1,3	0,3	92	Iris Versicolor	6,1	3,0	4,6	1,4	142	Iris Virginica	6,9	3,1	5,1	2,3
43	Iris Setosa	4,4	3,2	1,3	0,2	93	Iris Versicolor	5,8	2,6	4,0	1,2	143	Iris Virginica	5,8	2,7	5,1	1,9
44	Iris Setosa	5,0	3,5	1,6	0,6	94	Iris Versicolor	5,0	2,3	3,3	1,0	144	Iris Virginica	6,8	3,2	5,9	2,3
45	Iris Setosa	5,1	3,8	1,9	0,4	95	Iris Versicolor	5,6	2,7	4,2	1,3	145	Iris Virginica	6,7	3,3	5,7	2,5
46	Iris Setosa	4,8	3,0	1,4	0,3	96	Iris Versicolor	5,7	3,0	4,2	1,2	146	Iris Virginica	6,7	3,0	5,2	2,3
47	Iris Setosa	5,1	3,8	1,6	0,2	97	Iris Versicolor	5,7	2,9	4,2	1,3	147	Iris Virginica	6,3	2,5	5,0	1,9
48	Iris Setosa	4,6	3,2	1,4	0,2	98	Iris Versicolor	6,2	2,9	4,3	1,3	148	Iris Virginica	6,5	3,0	5,2	2,0
49	Iris Setosa	5,3	3,7	1,5	0,2	99	Iris Versicolor	5,1	2,5	3,0	1,1	149	Iris Virginica	6,2	3,4	5,4	2,3
50	Iris Setosa	5,0	3,3	1,4	0,2	100	Iris Versicolor	5,7	2,8	4,1	1,3	150	Iris Virginica	5,9	3,0	5,1	1,8

ATTRIBUTE VALUES ON THE LEFT and OBJECTS ON THE RIGHT

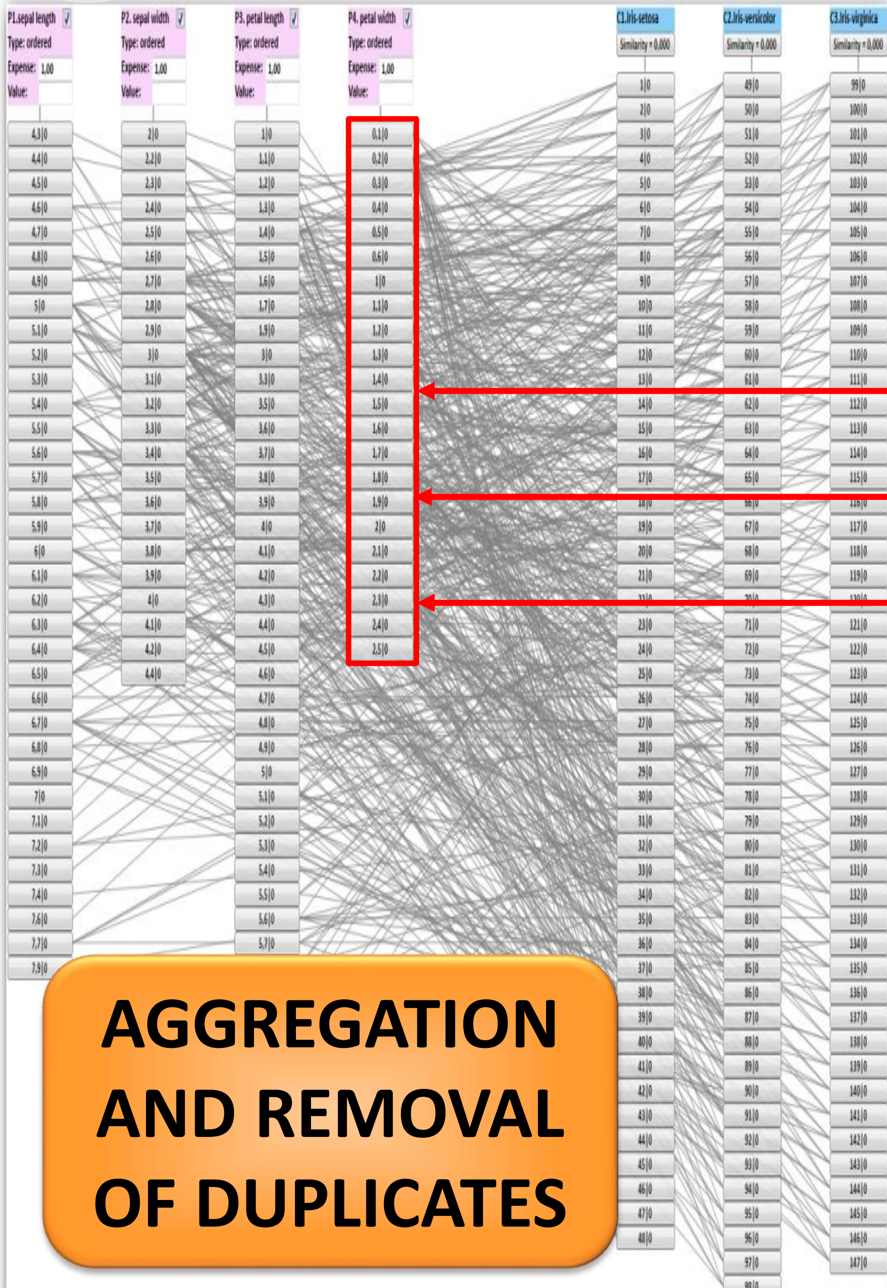


REMOVING OF REDUNDANCY (REDUCTION OF DUPLICATES)

REMOVAL OF REDUNDANCY

Lp.	klasa	długość liścia	szerokość liścia	długość płatka	szerokość płatka	Lp.	klasa	długość liścia	szerokość liścia	długość płatka	szerokość płatka	Lp.	klasa	długość liścia	szerokość liścia	długość płatka	szerokość płatka
1	Iris Setosa	5,1	3,5	1,4	0,2	51	Iris Versicolor	7,0	3,2	4,7	1,4	101	Iris Virginica	6,3	3,3	6,0	2,5
2	Iris Setosa	4,9	3,0	1,4	0,2	52	Iris Versicolor	6,4	3,2	4,5	1,5	102	Iris Virginica	5,8	2,7	5,1	1,9
3	Iris Setosa	4,7	3,2	1,3	0,2	53	Iris Versicolor	6,9	3,1	4,9	1,5	103	Iris Virginica	7,1	3,0	5,9	2,1
4	Iris Setosa	4,6	3,1	1,5	0,2	54	Iris Versicolor	5,5	2,3	4,0	1,3	104	Iris Virginica	6,3	2,9	5,6	1,8
5	Iris Setosa	5,0	3,6	1,4	0,2	55	Iris Versicolor	6,5	2,8	4,6	1,5	105	Iris Virginica	6,5	3,0	5,8	2,2
6	Iris Setosa	5,4	3,9	1,7	0,4	56	Iris Versicolor	5,7	2,8	4,5	1,3	106	Iris Virginica	7,6	3,0	6,6	2,1
7	Iris Setosa	4,6	3,4	1,4	0,3	57	Iris Versicolor	6,3	3,3	4,7	1,6	107	Iris Virginica	4,9	2,5	4,5	1,7
8	Iris Setosa	5,0	3,4	1,5	0,2	58	Iris Versicolor	4,9	2,4	3,3	1,0	108	Iris Virginica	7,3	2,9	6,3	1,8
9	Iris Setosa	4,4	2,9	1,4	0,2	59	Iris Versicolor	6,6	2,9	4,6	1,3	109	Iris Virginica	6,7	2,5	5,8	1,8
10	Iris Setosa	4,9	3,1	1,5	0,1	60	Iris Versicolor	5,2	2,7	3,9	1,4	110	Iris Virginica	7,2	3,6	6,1	2,5
11	Iris Setosa	5,4	3,7	1,5	0,2	61	Iris Versicolor	5,0	2,0	3,5	1,0	111	Iris Virginica	6,5	3,2	5,1	2,0
12	Iris Setosa	4,8	3,4	1,6	0,2	62	Iris Versicolor	5,9	3,0	4,2	1,5	112	Iris Virginica	6,4	2,7	5,3	1,9
13	Iris Setosa	4,8	3,0	1,4	0,1	63	Iris Versicolor	6,0	2,2	4,0	1,0	113	Iris Virginica	6,8	3,0	5,5	2,1
14	Iris Setosa	4,3	3,0	1,1	0,1	64	Iris Versicolor	6,1	2,9	4,7	1,4	114	Iris Virginica	5,7	2,5	5,0	2,0
15	Iris Setosa	5,8	4,0	1,2	0,2	65	Iris Versicolor	5,6	2,9	3,6	1,3	115	Iris Virginica	5,8	2,8	5,1	2,4
16	Iris Setosa	5,7	4,4	1,5	0,4	66	Iris Versicolor	6,7	3,1	4,4	1,4	116	Iris Virginica	6,4	3,2	5,3	2,3
17	Iris Setosa	5,4	3,9	1,3	0,4	67	Iris Versicolor	5,6	3,0	4,5	1,5	117	Iris Virginica	6,5	3,0	5,5	1,8
18	Iris Setosa	5,1	3,5	1,4	0,3	68	Iris Versicolor	5,8	2,7	4,1	1,0	118	Iris Virginica	7,7	3,8	6,7	2,2
19	Iris Setosa	5,7	3,8	1,7	0,3	69	Iris Versicolor	6,2	2,2	4,5	1,5	119	Iris Virginica	7,7	2,6	6,9	2,3
20	Iris Setosa	5,1	3,8	1,5	0,3	70	Iris Versicolor	5,6	2,5	3,9	1,1	120	Iris Virginica	6,0	2,2	5,0	1,5
21	Iris Setosa	5,4	3,4	1,7	0,2	71	Iris Versicolor	5,9	3,2	4,8	1,8	121	Iris Virginica	6,9	3,2	5,7	2,3
22	Iris Setosa	5,1	3,7	1,5	0,4	72	Iris Versicolor	6,1	2,8	4,0	1,3	122	Iris Virginica	5,6	2,8	4,9	2,0
23	Iris Setosa	4,6	3,6	1,0	0,2	73	Iris Versicolor	6,3	2,5	4,9	1,5	123	Iris Virginica	7,7	2,8	6,7	2,0
24	Iris Setosa	5,1	3,3	1,7	0,5	74	Iris Versicolor	6,1	2,8	4,7	1,2	124	Iris Virginica	6,3	2,7	4,9	1,8
25	Iris Setosa	4,8	3,4	1,9	0,2	75	Iris Versicolor	6,4	2,9	4,3	1,3	125	Iris Virginica	6,7	3,3	5,7	2,1
26	Iris Setosa	5,0	3,0	1,6	0,2	76	Iris Versicolor	6,6	3,0	4,4	1,4	126	Iris Virginica	7,2	3,2	6,0	1,8
27	Iris Setosa	5,0	3,4	1,6	0,4	77	Iris Versicolor	6,8	2,8	4,8	1,4	127	Iris Virginica	6,2	2,8	4,8	1,8
28	Iris Setosa	5,2	3,5	1,5	0,2	78	Iris Versicolor	6,7	3,0	5,0	1,7	128	Iris Virginica	6,1	3,0	4,9	1,8
29	Iris Setosa	5,2	3,4	1,4	0,2	79	Iris Versicolor	6,0	2,9	4,5	1,5	129	Iris Virginica	6,4	2,8	5,6	2,1
30	Iris Setosa	4,7	3,2	1,6	0,2	80	Iris Versicolor	5,7	2,6	3,5	1,0	130	Iris Virginica	7,2	3,0	5,8	1,6
31	Iris Setosa	4,8	3,1	1,6	0,2	81	Iris Versicolor	5,5	2,4	3,8	1,1	131	Iris Virginica	7,4	2,8	6,1	1,9
32	Iris Setosa	5,4	3,4	1,5	0,4	82	Iris Versicolor	5,5	2,4	3,7	1,0	132	Iris Virginica	7,9	3,8	6,4	2,0
33	Iris Setosa	5,2	4,1	1,5	0,1	83	Iris Versicolor	5,8	2,7	3,9	1,2	133	Iris Virginica	6,4	2,8	5,6	2,2
34	Iris Setosa	5,5	4,2	1,4	0,2	84	Iris Versicolor	6,0	2,7	5,1	1,6	134	Iris Virginica	6,3	2,8	5,1	1,5
35	Iris Setosa	4,9	3,1	1,5	0,1	85	Iris Versicolor	5,4	3,0	4,5	1,5	135	Iris Virginica	6,1	2,6	5,6	1,4
36	Iris Setosa	5,0	3,2	1,2	0,2	86	Iris Versicolor	6,0	3,4	4,5	1,6	136	Iris Virginica	7,7	3,0	6,1	2,3
37	Iris Setosa	5,5	3,5	1,3	0,2	87	Iris Versicolor	6,7	3,1	4,7	1,5	137	Iris Virginica	6,3	3,4	5,6	2,4
38	Iris Setosa	4,9	3,1	1,5	0,1	88	Iris Versicolor	6,3	2,3	4,4	1,3	138	Iris Virginica	6,4	3,1	5,5	1,8
39	Iris Setosa	4,4	3,0	1,3	0,2	89	Iris Versicolor	5,6	3,0	4,1	1,3	139	Iris Virginica	6,0	3,0	4,8	1,8
40	Iris Setosa	5,1	3,4	1,5	0,2	90	Iris Versicolor	5,5	2,5	4,0	1,3	140	Iris Virginica	6,9	3,1	5,4	2,1
41	Iris Setosa	5,0	3,5	1,3	0,3	91	Iris Versicolor	5,5	2,6	4,4	1,2	141	Iris Virginica	6,7	3,1	5,6	2,4
42	Iris Setosa	4,5	2,3	1,3	0,3	92	Iris Versicolor	6,1	3,0	4,6	1,4	142	Iris Virginica	6,9	3,1	5,1	2,3
43	Iris Setosa	4,4	3,2	1,3	0,2	93	Iris Versicolor	5,8	2,6	4,0	1,2	143	Iris Virginica	5,8	2,7	5,1	1,9
44	Iris Setosa	5,0	3,5	1,6	0,6	94	Iris Versicolor	5,0	2,3	3,3	1,0	144	Iris Virginica	6,8	3,2	5,9	2,3
45	Iris Setosa	5,1	3,8	1,9	0,4	95	Iris Versicolor	5,6	2,7	4,2	1,3	145	Iris Virginica	6,7	3,3	5,7	2,5
46	Iris Setosa	4,8	3,0	1,4	0,3	96	Iris Versicolor	5,7	3,0	4,2	1,2	146	Iris Virginica	6,7	3,0	5,2	2,3
47	Iris Setosa	5,1	3,8	1,6	0,2	97	Iris Versicolor	5,7	2,9	4,2	1,3	147	Iris Virginica	6,3	2,5	5,0	1,9
48	Iris Setosa	4,6	3,2	1,4	0,2	98	Iris Versicolor	6,2	2,9	4,3	1,3	148	Iris Virginica	6,5	3,0	5,2	2,0
49	Iris Setosa	5,3	3,7	1,5	0,2	99	Iris Versicolor	5,1	2,5	3,0	1,1	149	Iris Virginica	6,2	3,4	5,4	2,3
50	Iris Setosa	5,0	3,3	1,4	0,2	100	Iris Versicolor	5,7	2,8	4,1	1,3	150	Iris Virginica	5,9	3,0	5,1	1,8

AGGREGATION AND REMOVAL OF REDUNDANCY IN AGDS



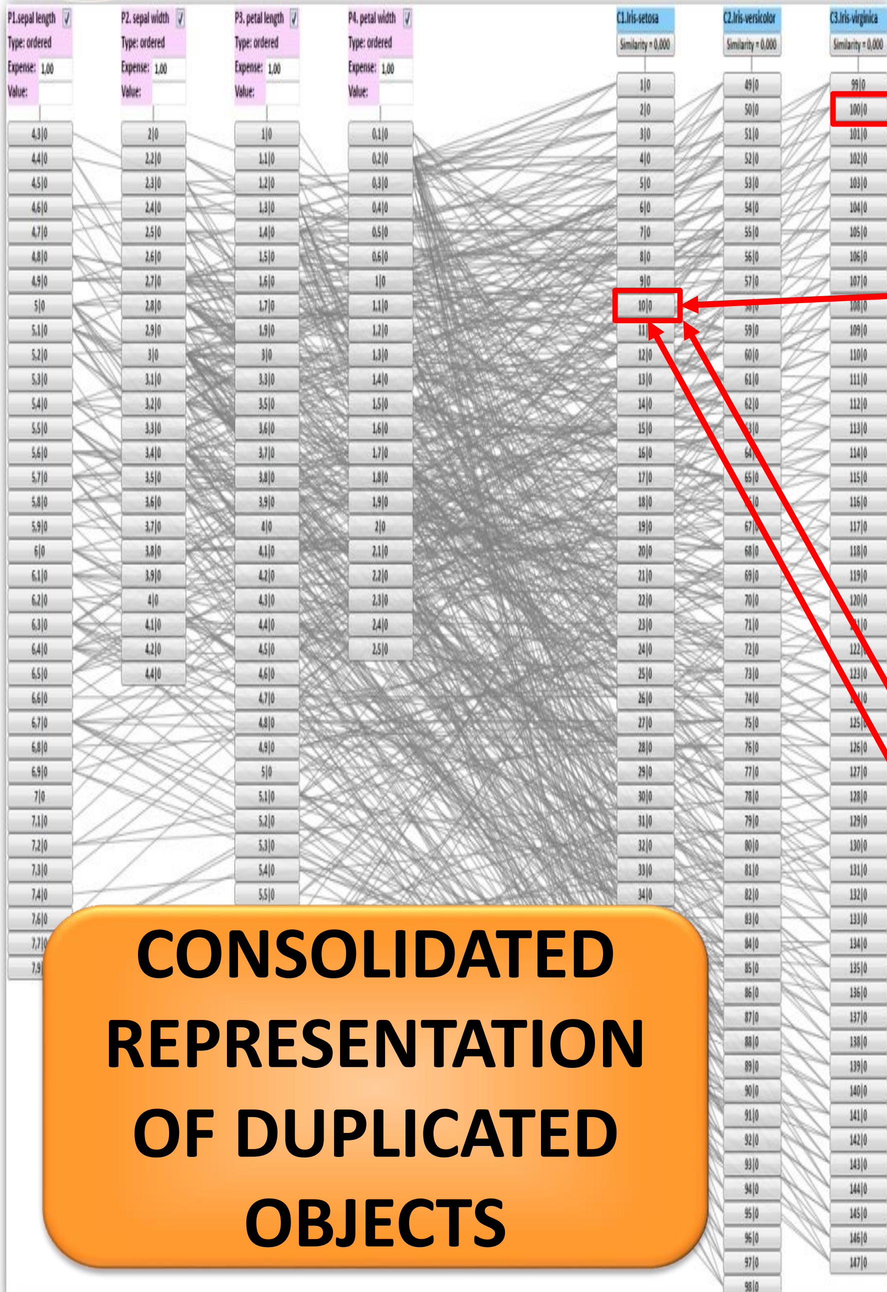
AGGREGATION AND REMOVAL OF DUPLICATES

Lp.	klasa	długość liścia	szerokość liścia	długość płatka	szerokość płatka
1	Iris Setosa	5,1	3,5	1,4	0,2
2	Iris Setosa	4,9	3,0	1,4	0,2
3	Iris Setosa	4,7	3,2	1,3	0,2
4	Iris Setosa	4,6	3,1	1,5	0,2
5	Iris Setosa	5,0	3,6	1,4	0,2
6	Iris Setosa	5,4	3,9	1,7	0,4
7	Iris Setosa	4,6	3,4	1,4	0,3
8	Iris Setosa	5,0	3,4	1,5	0,2
9	Iris Setosa	4,4	2,9	1,4	0,2
10	Iris Setosa	4,9	3,1	1,5	0,1
11	Iris Setosa	5,4	3,7	1,5	0,2
12	Iris Setosa	4,8	3,4	1,6	0,2
13	Iris Setosa	4,8	3,0	1,4	0,1
14	Iris Setosa	4,3	3,0	1,1	0,1
15	Iris Setosa	5,8	4,0	1,2	0,2
16	Iris Setosa	5,7	4,4	1,5	0,4
17	Iris Setosa	5,4	3,9	1,3	0,4
18	Iris Setosa	5,1	3,5	1,4	0,3
19	Iris Setosa	5,7	3,8	1,7	0,3
20	Iris Setosa	5,1	3,8	1,5	0,3
21	Iris Setosa	5,4	3,4	1,7	0,2
22	Iris Setosa	5,1	3,7	1,5	0,4
23	Iris Setosa	4,6	3,6	1,0	0,2
24	Iris Setosa	5,1	3,3	1,7	0,5
25	Iris Setosa	4,8	3,4	1,9	0,2
26	Iris Setosa	5,0	3,0	1,6	0,2
27	Iris Setosa	5,0	3,4	1,6	0,4
28	Iris Setosa	5,2	3,5	1,5	0,2
29	Iris Setosa	5,2	3,4	1,4	0,2
30	Iris Setosa	4,7	3,2	1,6	0,2
31	Iris Setosa	4,8	3,1	1,6	0,2
32	Iris Setosa	5,4	3,4	1,5	0,4
33	Iris Setosa	5,2	4,1	1,5	0,1
34	Iris Setosa	5,5	4,2	1,4	0,2
35	Iris Setosa	4,9	3,1	1,5	0,1
36	Iris Setosa	5,0	3,2	1,2	0,2
37	Iris Setosa	5,5	3,5	1,3	0,2
38	Iris Setosa	4,9	3,1	1,5	0,1
39	Iris Setosa	4,4	3,0	1,3	0,2
40	Iris Setosa	5,1	3,4	1,5	0,2
41	Iris Setosa	5,0	3,5	1,3	0,3
42	Iris Setosa	4,5	2,3	1,3	0,3
43	Iris Setosa	4,4	3,2	1,3	0,2
44	Iris Setosa	5,0	3,5	1,6	0,6
45	Iris Setosa	5,1	3,8	1,9	0,4
46	Iris Setosa	4,8	3,0	1,4	0,3
47	Iris Setosa	5,1	3,8	1,6	0,2
48	Iris Setosa	4,6	3,2	1,4	0,2
49	Iris Setosa	5,3	3,7	1,5	0,2
50	Iris Setosa	5,0	3,3	1,4	0,2

Lp.	klasa	długość liścia	szerokość liścia	długość płatka	szerokość płatka
51	Iris Versicolor	7,0	3,2	4,7	1,4
52	Iris Versicolor	6,4	3,2	4,5	1,5
53	Iris Versicolor	6,9	3,1	4,9	1,5
54	Iris Versicolor	5,5	2,3	4,0	1,3
55	Iris Versicolor	6,5	2,8	4,6	1,5
56	Iris Versicolor	5,7	2,8	4,5	1,3
57	Iris Versicolor	6,3	3,3	4,7	1,6
58	Iris Versicolor	4,9	2,4	3,3	1,0
59	Iris Versicolor	6,6	2,9	4,6	1,3
60	Iris Versicolor	5,2	2,7	3,9	1,4
61	Iris Versicolor	5,0	2,0	3,5	1,0
62	Iris Versicolor	5,9	3,0	4,2	1,5
63	Iris Versicolor	6,0	2,2	4,0	1,0
64	Iris Versicolor	6,1	2,9	4,7	1,4
65	Iris Versicolor	5,6	2,9	3,6	1,3
66	Iris Versicolor	6,7	3,1	4,4	1,4
67	Iris Versicolor	5,6	3,0	4,5	1,5
68	Iris Versicolor	5,8	2,7	4,1	1,0
69	Iris Versicolor	6,2	2,2	4,5	1,5
70	Iris Versicolor	5,6	2,5	3,9	1,1
71	Iris Versicolor	5,9	3,2	4,8	1,8
72	Iris Versicolor	6,1	2,8	4,0	1,3
73	Iris Versicolor	6,3	2,5	4,9	1,5
74	Iris Versicolor	6,1	2,8	4,7	1,2
75	Iris Versicolor	6,4	2,9	4,3	1,3
76	Iris Versicolor	6,6	3,0	4,4	1,4
77	Iris Versicolor	6,8	2,8	4,8	1,4
78	Iris Versicolor	6,7	3,0	5,0	1,7
79	Iris Versicolor	6,0	2,9	4,5	1,5
80	Iris Versicolor	5,7	2,6	3,5	1,0
81	Iris Versicolor	5,5	2,4	3,8	1,1
82	Iris Versicolor	5,5	2,4	3,7	1,0
83	Iris Versicolor	5,8	2,7	3,9	1,2
84	Iris Versicolor	6,0	2,7	5,1	1,6
85	Iris Versicolor	5,4	3,0	4,5	1,5
86	Iris Versicolor	6,0	3,4	4,5	1,6
87	Iris Versicolor	6,7	3,1	4,7	1,5
88	Iris Versicolor	6,3	2,3	4,4	1,3
89	Iris Versicolor	5,6	3,0	4,1	1,3
90	Iris Versicolor	5,5	2,5	4,0	1,3
91	Iris Versicolor	5,5	2,6	4,4	1,2
92	Iris Versicolor	6,1	3,0	4,6	1,4
93	Iris Versicolor	5,8	2,6	4,0	1,2
94	Iris Versicolor	5,0	2,3	3,3	1,0
95	Iris Versicolor	5,6	2,7	4,2	1,3
96	Iris Versicolor	5,7	3,0	4,2	1,2
97	Iris Versicolor	5,7	2,9	4,2	1,3
98	Iris Versicolor	6,2	2,9	4,3	1,3
99	Iris Versicolor	5,1	2,5	3,0	1,1
100	Iris Versicolor	5,7	2,8	4,1	1,3

Lp.	klasa	długość liścia	szerokość liścia	długość płatka	szerokość płatka
101	Iris Virginica	6,3	3,3	6,0	2,5
102	Iris Virginica	5,8	2,7	5,1	1,9
103	Iris Virginica	7,1	3,0	5,9	2,1
104	Iris Virginica	6,3	2,9	5,6	1,8
105	Iris Virginica	6,5	3,0	5,8	2,2
106	Iris Virginica	7,6	3,0	6,6	2,1
107	Iris Virginica	4,9	2,5	4,5	1,7
108	Iris Virginica	7,3	2,9	6,3	1,8
109	Iris Virginica	6,7	2,5	5,8	1,8
110	Iris Virginica	7,2	3,6	6,1	2,5
111	Iris Virginica	6,5	3,2	5,1	2,0
112	Iris Virginica	6,4	2,7	5,3	1,9
113	Iris Virginica	6,8	3,0	5,5	2,1
114	Iris Virginica	5,7	2,5	5,0	2,0
115	Iris Virginica	5,8	2,8	5,1	2,4
116	Iris Virginica	6,4	3,2	5,3	2,3
117	Iris Virginica	6,5	3,0	5,5	1,8
118	Iris Virginica	7,7	3,8	6,7	2,2
119	Iris Virginica	7,7	2,6	6,9	2,3
120	Iris Virginica	6,0	2,2	5,0	1,5
121	Iris Virginica	6,9	3,2	5,7	2,3
122	Iris Virginica	5,6	2,8	4,9	2,0
123	Iris Virginica	7,7	2,8	6,7	2,0
124	Iris Virginica	6,3	2,7	4,9	1,8
125	Iris Virginica	6,7	3,3	5,7	2,1
126	Iris Virginica	7,2	3,2	6,0	1,8
127	Iris Virginica	6,2	2,8	4,8	1,8
128	Iris Virginica	6,1	3,0	4,9	1,8
129	Iris Virginica	6,4	2,8	5,6	2,1
130	Iris Virginica	7,2	3,0	5,8	1,6
131	Iris Virginica	7,4	2,8	6,1	1,9
132	Iris Virginica	7,9	3,8	6,4	2,0
133	Iris Virginica	6,4	2,8	5,6	2,2
134	Iris Virginica	6,3	2,8	5,1	1,5
135	Iris Virginica	6,1	2,6	5,6	1,4
136	Iris Virginica	7,7	3,0	6,1	2,3
137	Iris Virginica	6,3	3,4	5,6	2,4
138	Iris Virginica	6,4	3,1	5,5	1,8
139	Iris Virginica	6,0	3,0	4,8	1,8
140	Iris Virginica	6,9	3,1	5,4	2,1
141	Iris Virginica	6,7	3,1	5,6	2,4
142	Iris Virginica	6,9	3,1	5,1	2,3
143	Iris Virginica	5,8	2,7	5,1	1,9
144	Iris Virginica	6,8	3,2	5,9	2,3
145	Iris Virginica	6,7	3,3	5,7	2,5
146	Iris Virginica	6,7	3,0	5,2	2,3
147	Iris Virginica	6,3	2,5	5,0	1,9
148	Iris Virginica	6,5	3,0	5,2	2,0
149	Iris Virginica	6,2	3,4	5,4	2,3
150	Iris Virginica	5,9	3,0	5,1	1,8

AGGREGATION OF DUPLICATED OBJECTS IN AGDS



CONSOLIDATED REPRESENTATION OF DUPLICATED OBJECTS

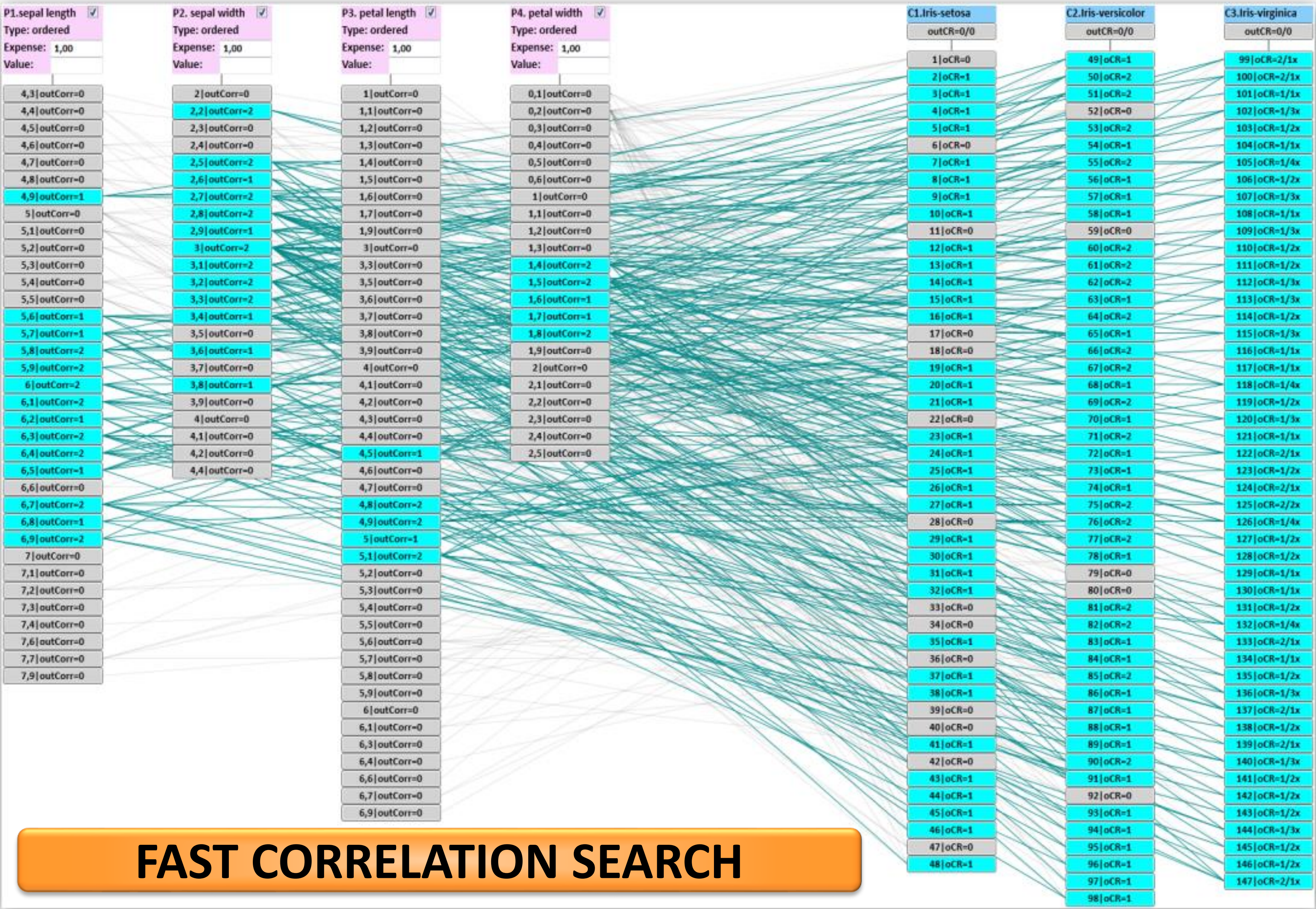
Lp.	klasa	długość liścia	szerokość liścia	długość płatka	szerokość płatka	Lp.	klasa	długość liścia	szerokość liścia	długość płatka	szerokość płatka	Lp.	klasa	długość liścia	szerokość liścia	długość płatka	szerokość płatka
1	Iris Setosa	5,1	3,5	1,4	0,2	51	Iris Versicolor	7,0	3,2	4,7	1,4	101	Iris Virginica	6,3	3,3	6,0	2,5
2	Iris Setosa	4,9	3,0	1,4	0,2	52	Iris Versicolor	6,4	3,2	4,5	1,5	102	Iris Virginica	5,8	2,7	5,1	1,9
3	Iris Setosa	4,7	3,2	1,3	0,2	53	Iris Versicolor	6,9	3,1	4,9	1,5	103	Iris Virginica	7,1	3,0	5,9	2,1
4	Iris Setosa	4,6	3,1	1,5	0,2	54	Iris Versicolor	5,5	2,3	4,0	1,3	104	Iris Virginica	6,3	2,9	5,6	1,8
5	Iris Setosa	5,0	3,6	1,4	0,2	55	Iris Versicolor	6,5	2,8	4,6	1,5	105	Iris Virginica	6,5	3,0	5,8	2,2
6	Iris Setosa	5,4	3,9	1,7	0,4	56	Iris Versicolor	5,7	2,8	4,5	1,3	106	Iris Virginica	7,6	3,0	6,6	2,1
7	Iris Setosa	4,6	3,4	1,4	0,3	57	Iris Versicolor	6,3	3,3	4,7	1,6	107	Iris Virginica	4,9	2,5	4,5	1,7
8	Iris Setosa	5,0	3,4	1,5	0,2	58	Iris Versicolor	4,9	2,4	3,3	1,0	108	Iris Virginica	7,3	2,9	6,3	1,8
9	Iris Setosa	4,4	2,9	1,4	0,2	59	Iris Versicolor	6,6	2,9	4,6	1,3	109	Iris Virginica	6,7	2,5	5,8	1,8
10	Iris Setosa	4,9	3,1	1,5	0,1	60	Iris Versicolor	5,2	2,7	3,9	1,4	110	Iris Virginica	7,2	3,6	6,1	2,5
11	Iris Setosa	5,4	3,7	1,5	0,2	61	Iris Versicolor	5,0	2,0	3,5	1,0	111	Iris Virginica	6,5	3,2	5,1	2,0
12	Iris Setosa	4,8	3,4	1,6	0,2	62	Iris Versicolor	5,9	3,0	4,2	1,5	112	Iris Virginica	6,4	2,7	5,3	1,9
13	Iris Setosa	4,8	3,0	1,4	0,1	63	Iris Versicolor	6,0	2,2	4,0	1,0	113	Iris Virginica	6,8	3,0	5,5	2,1
14	Iris Setosa	4,3	3,0	1,1	0,1	64	Iris Versicolor	6,1	2,9	4,7	1,4	114	Iris Virginica	5,7	2,5	5,0	2,0
15	Iris Setosa	5,8	4,0	1,2	0,2	65	Iris Versicolor	5,6	2,9	3,6	1,3	115	Iris Virginica	5,8	2,8	5,1	2,4
16	Iris Setosa	5,7	4,4	1,5	0,4	66	Iris Versicolor	6,7	3,1	4,4	1,4	116	Iris Virginica	6,4	3,2	5,3	2,3
17	Iris Setosa	5,4	3,9	1,3	0,4	67	Iris Versicolor	5,6	3,0	4,5	1,5	117	Iris Virginica	6,5	3,0	5,5	1,8
18	Iris Setosa	5,1	3,5	1,4	0,3	68	Iris Versicolor	5,8	2,7	4,1	1,0	118	Iris Virginica	7,7	3,8	6,7	2,2
19	Iris Setosa	5,7	3,8	1,7	0,3	69	Iris Versicolor	6,2	2,2	4,5	1,5	119	Iris Virginica	7,7	2,6	6,9	2,3
20	Iris Setosa	5,1	3,8	1,5	0,3	70	Iris Versicolor	5,6	2,5	3,9	1,1	120	Iris Virginica	6,0	2,2	5,0	1,5
21	Iris Setosa	5,4	3,4	1,7	0,2	71	Iris Versicolor	5,9	3,2	4,8	1,8	121	Iris Virginica	6,9	3,2	5,7	2,3
22	Iris Setosa	5,1	3,7	1,5	0,4	72	Iris Versicolor	6,1	2,8	4,0	1,3	122	Iris Virginica	5,6	2,8	4,9	2,0
23	Iris Setosa	4,6	3,6	1,0	0,2	73	Iris Versicolor	6,3	2,5	4,9	1,5	123	Iris Virginica	7,7	2,8	6,7	2,0
24	Iris Setosa	5,1	3,3	1,7	0,5	74	Iris Versicolor	6,1	2,8	4,7	1,2	124	Iris Virginica	6,3	2,7	4,9	1,8
25	Iris Setosa	4,8	3,4	1,9	0,2	75	Iris Versicolor	6,4	2,9	4,3	1,3	125	Iris Virginica	6,7	3,3	5,7	2,1
26	Iris Setosa	5,0	3,0	1,6	0,2	76	Iris Versicolor	6,6	3,0	4,4	1,4	126	Iris Virginica	7,2	3,2	6,0	1,8
27	Iris Setosa	5,0	3,4	1,6	0,4	77	Iris Versicolor	6,8	2,8	4,8	1,4	127	Iris Virginica	6,2	2,8	4,8	1,8
28	Iris Setosa	5,2	3,5	1,5	0,2	78	Iris Versicolor	6,7	3,0	5,0	1,7	128	Iris Virginica	6,1	3,0	4,9	1,8
29	Iris Setosa	5,2	3,4	1,4	0,2	79	Iris Versicolor	6,0	2,9	4,5	1,5	129	Iris Virginica	6,4	2,8	5,6	2,1
30	Iris Setosa	4,7	3,2	1,6	0,2	80	Iris Versicolor	5,7	2,6	3,5	1,0	130	Iris Virginica	7,2	3,0	5,8	1,6
31	Iris Setosa	4,8	3,1	1,6	0,2	81	Iris Versicolor	5,5	2,4	3,8	1,1	131	Iris Virginica	7,4	2,8	6,1	1,9
32	Iris Setosa	5,4	3,4	1,5	0,4	82	Iris Versicolor	5,5	2,4	3,7	1,0	132	Iris Virginica	7,9	3,8	6,4	2,0
33	Iris Setosa	5,2	4,1	1,5	0,1	83	Iris Versicolor	5,8	2,7	3,9	1,2	133	Iris Virginica	6,4	2,8	5,6	2,2
34	Iris Setosa	5,5	4,2	1,4	0,2	84	Iris Versicolor	6,0	2,7	5,1	1,6	134	Iris Virginica	6,3	2,8	5,1	1,5
35	Iris Setosa	4,9	3,1	1,5	0,1	85	Iris Versicolor	5,4	3,0	4,5	1,5	135	Iris Virginica	6,1	2,6	5,6	1,4
36	Iris Setosa	5,0	3,2	1,2	0,2	86	Iris Versicolor	6,0	3,4	4,5	1,6	136	Iris Virginica	7,7	3,0	6,1	2,3
37	Iris Setosa	5,5	3,5	1,3	0,2	87	Iris Versicolor	6,7	3,1	4,7	1,5	137	Iris Virginica	6,3	3,4	5,6	2,4
38	Iris Setosa	4,9	3,1	1,5	0,1	88	Iris Versicolor	6,3	2,3	4,4	1,3	138	Iris Virginica	6,4	3,1	5,5	1,8
39	Iris Setosa	4,4	3,0	1,3	0,2	89	Iris Versicolor	5,6	3,0	4,1	1,3	139	Iris Virginica	6,0	3,0	4,8	1,8
40	Iris Setosa	5,1	3,4	1,5	0,2	90	Iris Versicolor	5,5	2,5	4,0	1,3	140	Iris Virginica	6,9	3,1	5,4	2,1
41	Iris Setosa	5,0	3,5	1,3	0,3	91	Iris Versicolor	5,5	2,6	4,4	1,2	141	Iris Virginica	6,7	3,1	5,6	2,4
42	Iris Setosa	4,5	2,3	1,3	0,3	92	Iris Versicolor	6,1	3,0	4,6	1,4	142	Iris Virginica	6,9	3,1	5,1	2,3
43	Iris Setosa	4,4	3,2	1,3	0,2	93	Iris Versicolor	5,8	2,6	4,0	1,2	143	Iris Virginica	5,8	2,7	5,1	1,9
44	Iris Setosa	5,0	3,5	1,6	0,6	94	Iris Versicolor	5,0	2,3	3,3	1,0	144	Iris Virginica	6,8	3,2	5,9	2,3
45	Iris Setosa	5,1	3,8	1,9	0,4	95	Iris Versicolor	5,6	2,7	4,2	1,3	145	Iris Virginica	6,7	3,3	5,7	2,5
46	Iris Setosa	4,8	3,0	1,4	0,3	96	Iris Versicolor	5,7	3,0	4,2	1,2	146	Iris Virginica	6,7	3,0	5,2	2,3
47	Iris Setosa	5,1	3,8	1,6	0,2	97	Iris Versicolor	5,7	2,9	4,2	1,3	147	Iris Virginica	6,3	2,5	5,0	1,9
48	Iris Setosa	4,6	3,2	1,4	0,2	98	Iris Versicolor	6,2	2,9	4,3	1,3	148	Iris Virginica	6,5	3,0	5,2	2,0
49	Iris Setosa	5,3	3,7	1,5	0,2	99	Iris Versicolor	5,1	2,5	3,0	1,1	149	Iris Virginica	6,2	3,4	5,4	2,3
50	Iris Setosa	5,0	3,3	1,4	0,2	100	Iris Versicolor	5,7	2,8	4,1	1,3	150	Iris Virginica	5,9	3,0	5,1	1,8

ELIMINATION OF REDUNDANCY IN AGDS REPRESENTATION

NO REDUNDANCY AND DUPLICATES

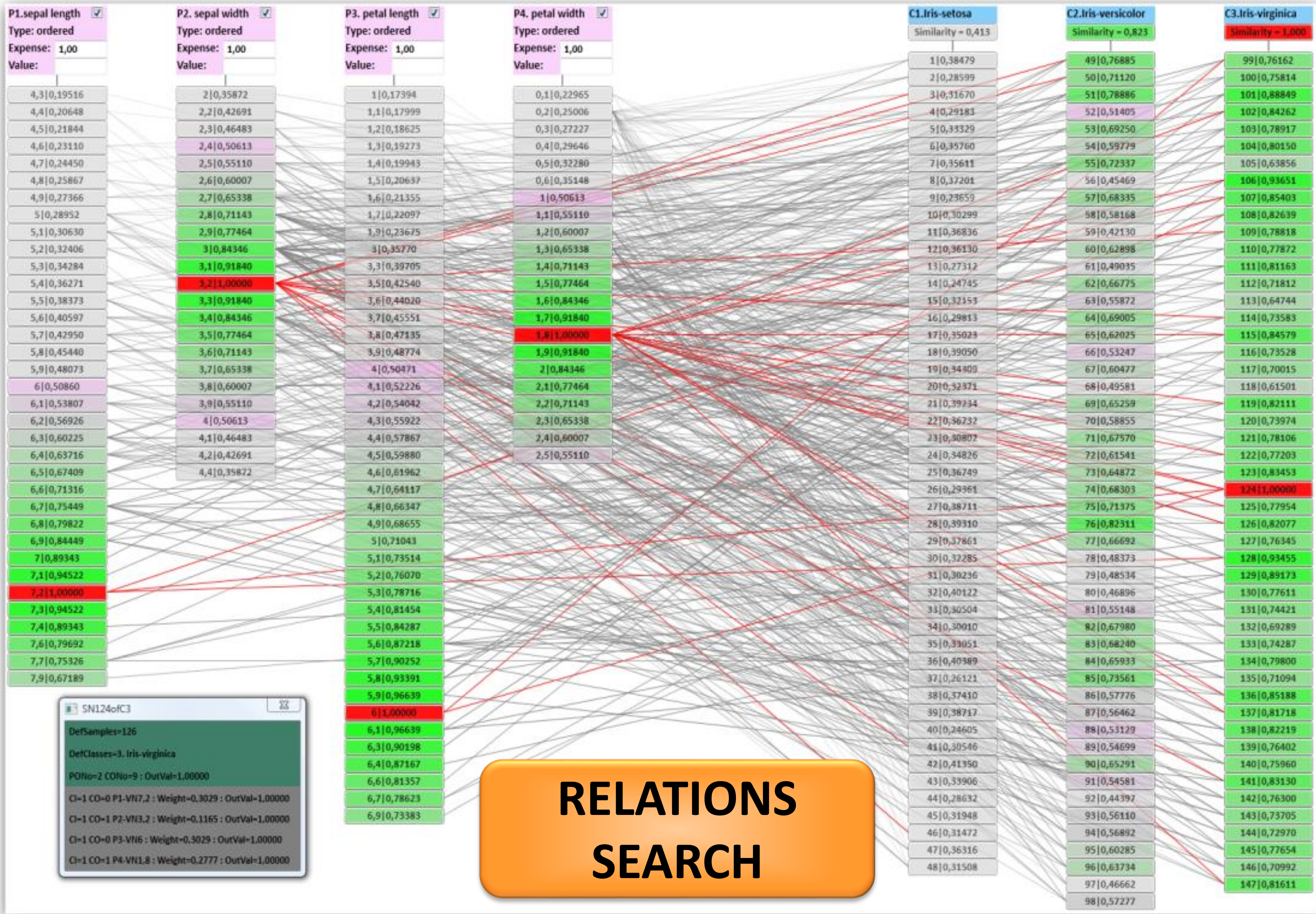
Lp.	klasa	długość liścia	szerokość liścia	długość płatka	szerokość płatka	Lp.	klasa	długość liścia	szerokość liścia	długość płatka	szerokość płatka	Lp.	klasa	długość liścia	szerokość liścia	długość płatka	szerokość płatka
1	Iris Setosa	5,1	3,5	1,4	0,2	51	Iris Versicolor	7,0	3,2	4,7	1,4	101	Iris Virginica	6,3	3,3	6,0	2,5
2	Iris Setosa	4,9	3,0	1,4	0,2	52	Iris Versicolor	6,4	3,2	4,5	1,5	102	Iris Virginica	5,8	2,7	5,1	1,9
3	Iris Setosa	4,7	3,2	1,3	0,2	53	Iris Versicolor	6,9	3,1	4,9	1,5	103	Iris Virginica	7,1	3,0	5,9	2,1
4	Iris Setosa	4,6	3,1	1,5	0,2	54	Iris Versicolor	5,5	2,3	4,0	1,3	104	Iris Virginica	6,3	2,9	5,6	1,8
5	Iris Setosa	5,0	3,6	1,4	0,2	55	Iris Versicolor	6,5	2,8	4,6	1,5	105	Iris Virginica	6,5	3,0	5,8	2,2
6	Iris Setosa	5,4	3,9	1,7	0,4	56	Iris Versicolor	5,7	2,8	4,5	1,3	106	Iris Virginica	7,6	3,0	6,6	2,1
7	Iris Setosa	4,6	3,4	1,4	0,3	57	Iris Versicolor	6,3	3,3	4,7	1,6	107	Iris Virginica	4,9	2,5	4,5	1,7
8	Iris Setosa	5,0	3,4	1,5	0,2	58	Iris Versicolor	4,9	2,4	3,3	1,0	108	Iris Virginica	7,3	2,9	6,3	1,8
9	Iris Setosa	4,4	2,9	1,4	0,2	59	Iris Versicolor	6,6	2,9	4,6	1,3	109	Iris Virginica	6,7	2,5	5,8	1,8
10	Iris Setosa	4,9	3,1	1,5	0,1	60	Iris Versicolor	5,2	2,7	3,9	1,4	110	Iris Virginica	7,2	3,6	6,1	2,5
11	Iris Setosa	5,4	3,7	1,5	0,2	61	Iris Versicolor	5,0	2,0	3,5	1,0	111	Iris Virginica	6,5	3,2	5,1	2,0
12	Iris Setosa	4,8	3,4	1,6	0,2	62	Iris Versicolor	5,9	3,0	4,2	1,5	112	Iris Virginica	6,4	2,7	5,3	1,9
13	Iris Setosa	4,8	3,0	1,4	0,1	63	Iris Versicolor	6,0	2,2	4,0	1,0	113	Iris Virginica	6,8	3,0	5,5	2,1
14	Iris Setosa	4,3	3,0	1,1	0,1	64	Iris Versicolor	6,1	2,9	4,7	1,4	114	Iris Virginica	5,7	2,5	5,0	2,0
15	Iris Setosa	5,8	4,0	1,2	0,2	65	Iris Versicolor	5,6	2,9	3,6	1,3	115	Iris Virginica	5,8	2,8	5,1	2,4
16	Iris Setosa	5,7	4,4	1,5	0,4	66	Iris Versicolor	6,7	3,1	4,4	1,4	116	Iris Virginica	6,4	3,2	5,3	2,3
17	Iris Setosa	5,4	3,9	1,3	0,4	67	Iris Versicolor	5,6	3,0	4,5	1,5	117	Iris Virginica	6,5	3,0	5,5	1,8
18	Iris Setosa	5,1	3,5	1,4	0,3	68	Iris Versicolor	5,8	2,7	4,1	1,0	118	Iris Virginica	7,7	3,8	6,7	2,2
19	Iris Setosa	5,7	3,8	1,7	0,3	69	Iris Versicolor	6,2	2,2	4,5	1,5	119	Iris Virginica	7,7	2,6	6,9	2,3
20	Iris Setosa	5,1	3,8	1,5	0,3	70	Iris Versicolor	5,6	2,5	3,9	1,1	120	Iris Virginica	6,0	2,2	5,0	1,5
21	Iris Setosa	5,4	3,4	1,7	0,2	71	Iris Versicolor	5,9	3,2	4,8	1,8	121	Iris Virginica	6,9	3,2	5,7	2,3
22	Iris Setosa	5,1	3,7	1,5	0,4	72	Iris Versicolor	6,1	2,8	4,0	1,3	122	Iris Virginica	5,6	2,8	4,9	2,0
23	Iris Setosa	4,6	3,6	1,0	0,2	73	Iris Versicolor	6,3	2,5	4,9	1,5	123	Iris Virginica	7,7	2,8	6,7	2,0
24	Iris Setosa	5,1	3,3	1,7	0,5	74	Iris Versicolor	6,1	2,8	4,7	1,2	124	Iris Virginica	6,3	2,7	4,9	1,8
25	Iris Setosa	4,8	3,4	1,9	0,2	75	Iris Versicolor	6,4	2,9	4,3	1,3	125	Iris Virginica	6,7	3,3	5,7	2,1
26	Iris Setosa	5,0	3,0	1,6	0,2	76	Iris Versicolor	6,6	3,0	4,4	1,4	126	Iris Virginica	7,2	3,2	6,0	1,8
27	Iris Setosa	5,0	3,4	1,6	0,4	77	Iris Versicolor	6,8	2,8	4,8	1,4	127	Iris Virginica	6,2	2,8	4,8	1,8
28	Iris Setosa	5,2	3,5	1,5	0,2	78	Iris Versicolor	6,7	3,0	5,0	1,7	128	Iris Virginica	6,1	3,0	4,9	1,8
29	Iris Setosa	5,2	3,4	1,4	0,2	79	Iris Versicolor	6,0	2,9	4,5	1,5	129	Iris Virginica	6,4	2,8	5,6	2,1
30	Iris Setosa	4,7	3,2	1,6	0,2	80	Iris Versicolor	5,7	2,6	3,5	1,0	130	Iris Virginica	7,2	3,0	5,8	1,6
31	Iris Setosa	4,8	3,1	1,6	0,2	81	Iris Versicolor	5,5	2,4	3,8	1,1	131	Iris Virginica	7,4	2,8	6,1	1,9
32	Iris Setosa	5,4	3,4	1,5	0,4	82	Iris Versicolor	5,5	2,4	3,7	1,0	132	Iris Virginica	7,9	3,8	6,4	2,0
33	Iris Setosa	5,2	4,1	1,5	0,1	83	Iris Versicolor	5,8	2,7	3,9	1,2	133	Iris Virginica	6,4	2,8	5,6	2,2
34	Iris Setosa	5,5	4,2	1,4	0,2	84	Iris Versicolor	6,0	2,7	5,1	1,6	134	Iris Virginica	6,3	2,8	5,1	1,5
35	Iris Setosa	4,9	3,1	1,5	0,1	85	Iris Versicolor	5,4	3,0	4,5	1,5	135	Iris Virginica	6,1	2,6	5,6	1,4
36	Iris Setosa	5,0	3,2	1,2	0,2	86	Iris Versicolor	6,0	3,4	4,5	1,6	136	Iris Virginica	7,7	3,0	6,1	2,3
37	Iris Setosa	5,5	3,5	1,3	0,2	87	Iris Versicolor	6,7	3,1	4,7	1,5	137	Iris Virginica	6,3	3,4	5,6	2,4
38	Iris Setosa	4,9	3,1	1,5	0,1	88	Iris Versicolor	6,3	2,3	4,4	1,3	138	Iris Virginica	6,4	3,1	5,5	1,8
39	Iris Setosa	4,4	3,0	1,3	0,2	89	Iris Versicolor	5,6	3,0	4,1	1,3	139	Iris Virginica	6,0	3,0	4,8	1,8
40	Iris Setosa	5,1	3,4	1,5	0,2	90	Iris Versicolor	5,5	2,5	4,0	1,3	140	Iris Virginica	6,9	3,1	5,4	2,1
41	Iris Setosa	5,0	3,5	1,3	0,3	91	Iris Versicolor	5,5	2,6	4,4	1,2	141	Iris Virginica	6,7	3,1	5,6	2,4
42	Iris Setosa	4,5	2,3	1,3	0,3	92	Iris Versicolor	6,1	3,0	4,6	1,4	142	Iris Virginica	6,9	3,1	5,1	2,3
43	Iris Setosa	4,4	3,2	1,3	0,2	93	Iris Versicolor	5,8	2,6	4,0	1,2	143	Iris Virginica	5,8	2,7	5,1	1,9
44	Iris Setosa	5,0	3,5	1,6	0,6	94	Iris Versicolor	5,0	2,3	3,3	1,0	144	Iris Virginica	6,8	3,2	5,9	2,3
45	Iris Setosa	5,1	3,8	1,9	0,4	95	Iris Versicolor	5,6	2,7	4,2	1,3	145	Iris Virginica	6,7	3,3	5,7	2,5
46	Iris Setosa	4,8	3,0	1,4	0,3	96	Iris Versicolor	5,7	3,0	4,2	1,2	146	Iris Virginica	6,7	3,0	5,2	2,3
47	Iris Setosa	5,1	3,8	1,6	0,2	97	Iris Versicolor	5,7	2,9	4,2	1,3	147	Iris Virginica	6,3	2,5	5,0	1,9
48	Iris Setosa	4,6	3,2	1,4	0,2	98	Iris Versicolor	6,2	2,9	4,3	1,3	148	Iris Virginica	6,5	3,0	5,2	2,0
49	Iris Setosa	5,3	3,7	1,5	0,2	99	Iris Versicolor	5,1	2,5	3,0	1,1	149	Iris Virginica	6,2	3,4	5,4	2,3
50	Iris Setosa	5,0	3,3	1,4	0,2	100	Iris Versicolor	5,7	2,8	4,1	1,3	150	Iris Virginica	5,9	3,0	5,1	1,8

FAST SEARCH FOR RELATIONS AND CORRELATIONS

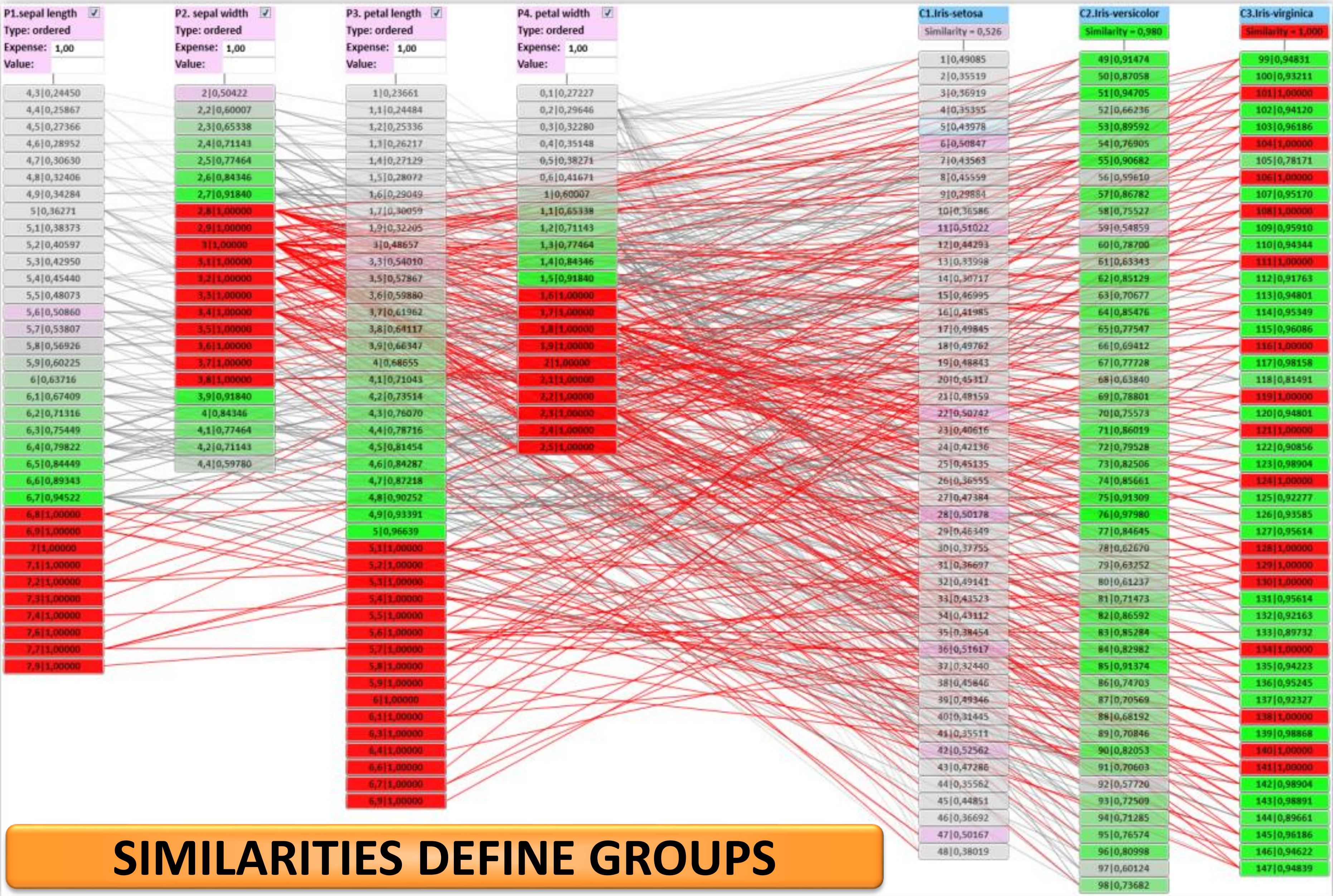


FAST CORRELATION SEARCH

FAST SEARCH FOR RELATIONS BETWEEN OBJECTS



FAST FINDING OF VARIOUS GROUPS AND CLASSIFICATION



SIMILARITIES DEFINE GROUPS

DATA STRUCTURE AND EFFICIENCY OF DATA PROCESSING

The introduced associative graph data structures (AGDS) essentially reduce the speed of data access and eliminates loops that have to be used on data organized in tables. Thus, the applied data structures have fundamental importance in data mining and its efficacy. Appropriately organized data can also facilitate various cognitive processes as well as intelligent inference.

In the AGDS structures, there is possible to:

- Storing always sorted data for all attributes at the same time,
- Lossless compression of data by removing any redundancy by eliminating all duplicates of attribute values and objects,
- Linking attribute data through additional relationships not presented in tabular structures, mapping different vertical relationships, e.g. similarity, differences, order, minima, maxima, and thus also additional relationships between objects,
- Instantaneous data access (in constant time),
- Automatic grouping of similar data and objects is built-in this structure and accessible in constant time.

REPLACING OPERATIONS BY THE ASSOCIATIVE STRUCTURE

The AGDS is not only another way of storing data in the graph structure, but it also replaces many operations and methods that have to be executed on tabular structures, looking for vertical relationships, e.g.:

- ✓ search for similar, different, correlated, inverse, neighbor, or duplicate objects,
- ✓ filter and search for various groups (e.g. clusters, classes) against given restrictions or constraints, selected attribute values, or their ranges,
- ✓ organize objects by all attributes simultaneously.

DEFINITION: We say that the structure replaces operations performer on another data structure when the computational complexity of the resulting data decreases to constant computational complexity O(1).

Generally speaking, if you have reached the computational goal in constant time (as in AGDS structures without loops) then your structure replaces more time-consuming operations that must be processed on another structure.

Due to the fact that in computer science we lose most of the time for data search operations, the AGDS structure can accelerate many operations and applications several dozens, hundred, or even thousand Times depending on the size of the browsed data! Intelligence demands such an efficiency!

ACTIVE ASSOCIATIVE NEURONAL GRAPHS – AANG

- In the human brain, we find reactive neurons and active neuronal structures that not only quickly and effectively associate data but also are able to actively respond or react to incoming data from senses, i.e. receptors.
- Despite the slow-acting neurons in relations to the clocking speed of modern processors, mental processes are rapidly overwhelmed by the constant computational complexity of neuronal associative and recall processes.
- Such structures in the human brain do not have to go through the processes of crawling, searching, comparing, and exploring data in many nested loops, nor passive tables are used for storing data as in relational databases.
- Biological processes of knowledge formation, data storage, information, and reasoning are based on plastic associative processes that reach for relevant data if they are fixed in them through learning, experience, introspection, inference, or other cognitive processes in our minds.
- In addition, the human mind has the ability to compile various triggers from the memory of events regardless of the actual place and time of their occurrence. This ability allows you to create new cognitive contexts for next thought processes as well as provide creativity and generosity at the high mental, logical, emotional, and abstract levels.

Modern computer science is very expensive in finding and exploring large amounts of data. That is why we talk about BIG DATA PROBLEMS!

ACTIVE ASSOCIATIVE NEURONAL GRAPHS – AANG

AGDS
passive

→

AANG
reactive

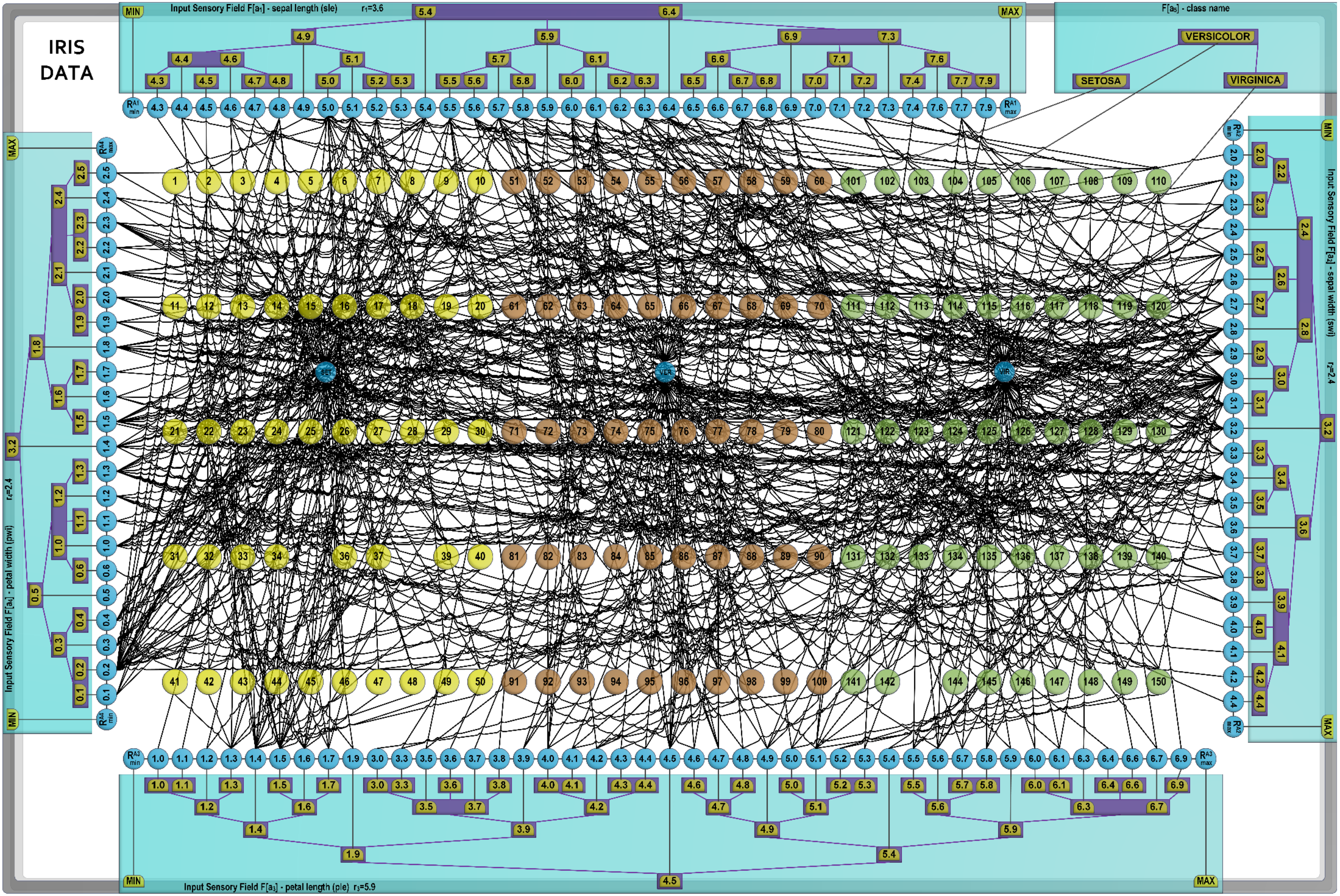
PASSIVE DATA STRUCTURES

are designed to store data
in their intact form

REACTIVE DATA STRUCTURES

react to new data and allow data
to interact with each other automatically

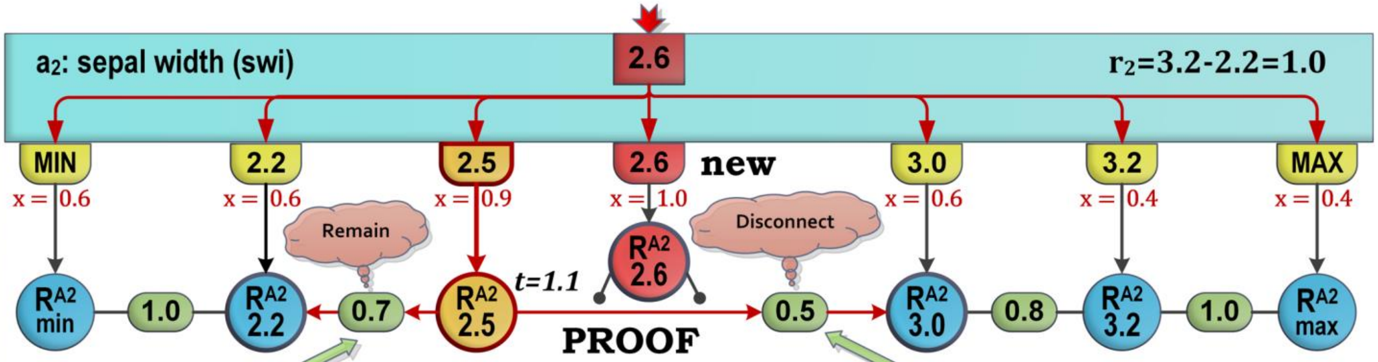
The AANG constructed for all Iris data from ML Repository using AVB-trees for representation of all attribute values



CREATION OF REACTIVE ASSOCIATIVE NEURONAL STRUCTURES

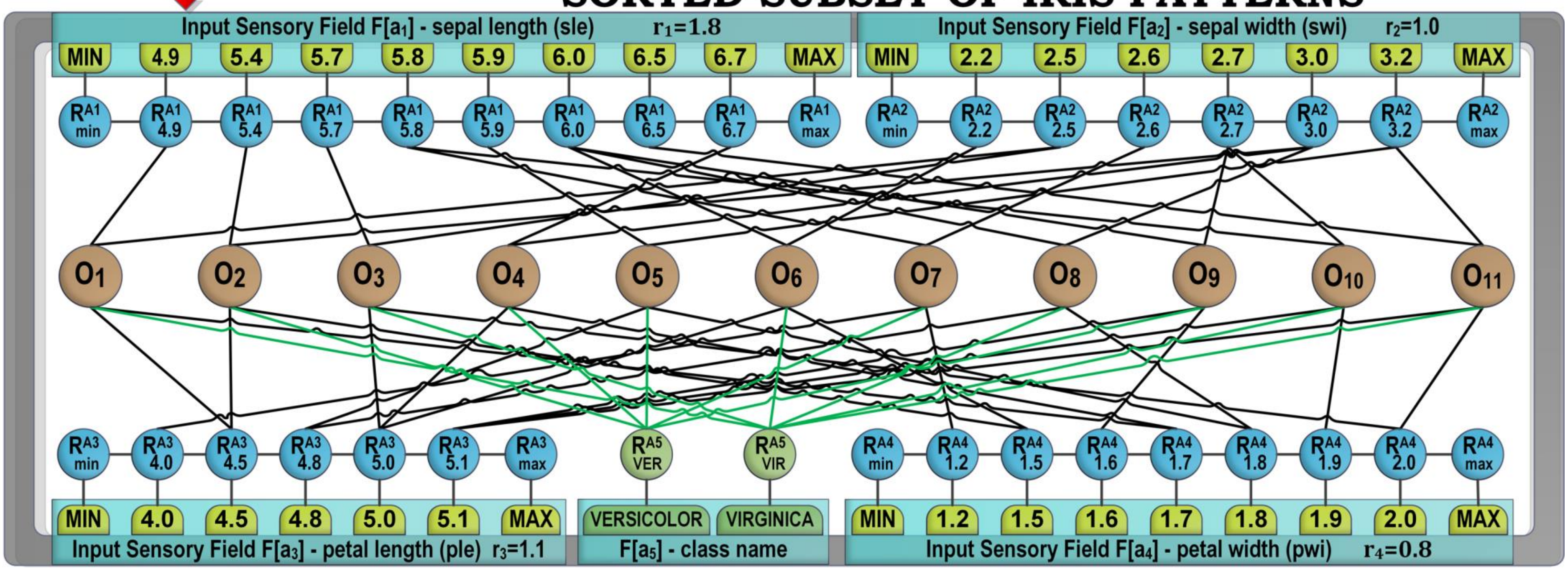
The ASSORT-2 algorithm automatically creates the basic associative neuronal structure for any table.

	Attributes $a_1,\ldots,a_5$				
	sle	swi	ple	pwi	class name
O_1	4.9	2.5	4.5	1.7	VIRGINICA
O_2	5.4	3.0	4.5	1.5	VERSICOLOR
O_3	5.7	2.5	5.0	2.0	VIRGINICA
O_4	6.7	3.0	5.0	1.7	VERSICOLOR
O_5	5.9	3.2	4.8	1.8	VERSICOLOR
O_6	6.0	2.2	5.0	1.5	VIRGINICA
O_7	5.8	2.6	4.0	1.2	VERSICOLOR
O_8	6.0	3.0	4.8	1.8	VIRGINICA
O_9	5.8	2.7	5.1	1.9	VIRGINICA
O_{10}	6.0	2.7	5.1	1.6	VERSICOLOR
O_{11}	6.5	3.2	5.1	2.0	VIRGINICA



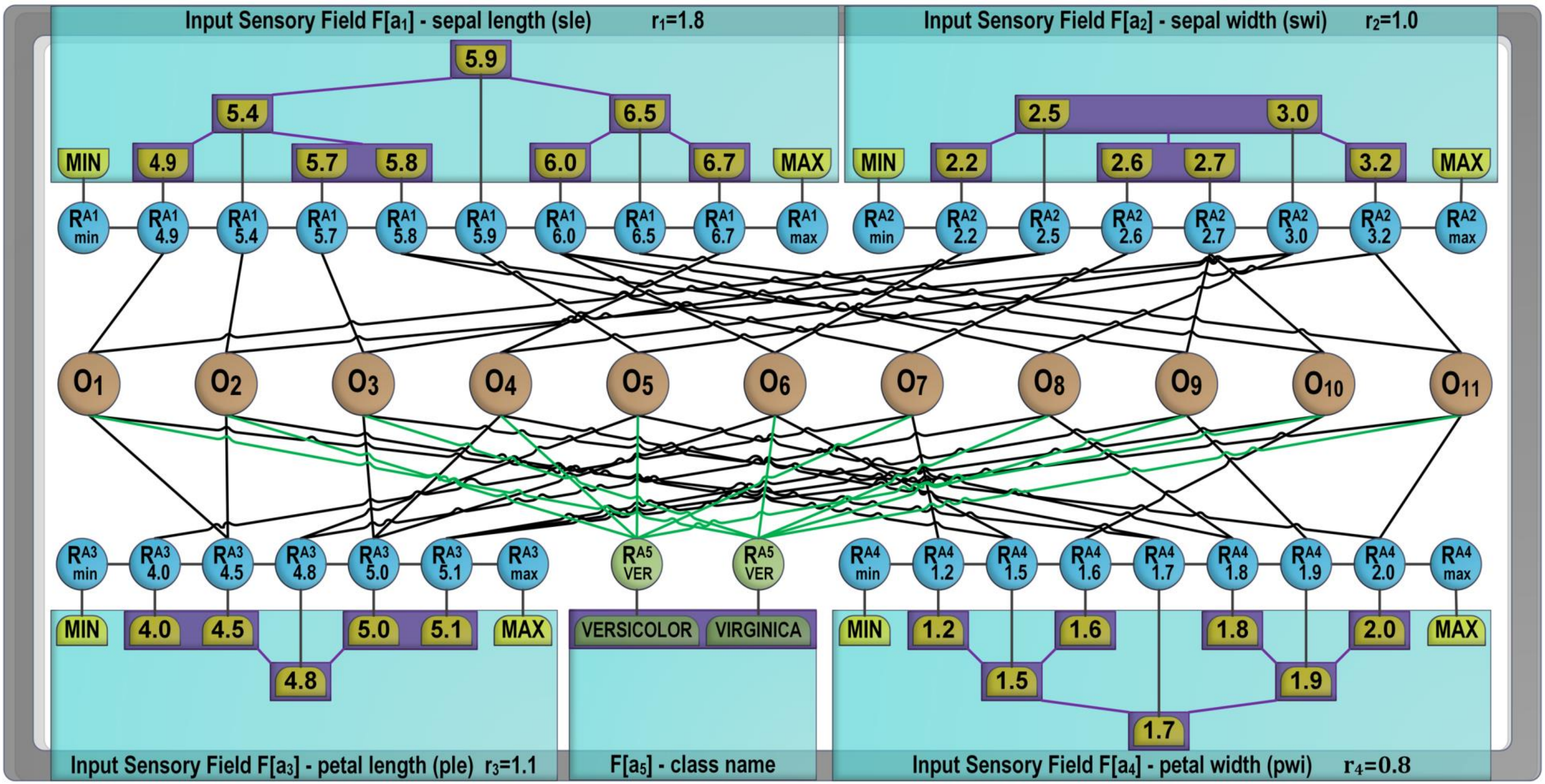
$$1-\frac{|2.2-2.6|}{1.0} = 0.6 \not> 0.7 = 1\cdot\left(1-\frac{|2.5-2.2|}{1.0}\right) > 0 \quad 0 < 1\cdot\left(1-\frac{|2.5-3.0|}{1.0}\right) = 0.5 < 0.6 = 1-\frac{|3.0-2.6|}{1.0} \qquad (10)$$

SORTED SUBSET OF IRIS PATTERNS



REACTIVE ASSOCIATIVE NEURAL STRUCTURES ON SEQUENTIAL MACHINES

Buy in large, contemporary computers work sequentially, have sequential cores in processors, sequential memories and sequential ways of executing operations and programs. Brains are parallel, and all internal processes run simultaneously. When implementing reactive neuronal structures on contemporary computers, we have to keep in mind these limitations and use AVB-trees to efficiently organize and access attribute data represented by sensors in sensory



CREATION OF ACTIVE ASSOCIATIVE NEURAL GRAPHS

Suppose we have objects $o_1, \ldots, o_N$ defined by the attributes $a_1, \ldots, a_K$ in such a way that each object is defined by a set of values of these attributes (K is a number of all attributes):

$$o_n = \left(v_{n_1}^{a_1}, \ldots, v_{n_K}^{a_K}\right)$$

Let these values react to certain sensory fields, modeling the senses, having sensors, modeling the receptors, enabling them to react to these values at a certain intensity.

Determination of ranges of represented values by the input sensory fields is computed after:

$$r^{a_k} = v_{max}^{a_k} - v_{min}^{a_k} \qquad where\ v_{max}^{a_k} = max\{v_i^{a_k}\},\ v_{min}^{a_k} = min\{v_i^{a_k}\}$$

Sensors in the sensory fields are created after the presentation of the stimulus that is not yet represented by any of the existing sensors, i.e. none of the existing sensors does not react enough, i.e. the distance between the presented and represented the value by this sensor is bigger than a defined certain minimum sensitivity:

$$d(v^{a_k}, v_i^{a_k}) = \left|v^{a_k} - v_i^{a_k}\right| \qquad \forall_i\, d(v^{a_k}, v_i^{a_k}) > \varepsilon^{a_k}$$

In case, when one of the sensors recognizes a certain value of the stimulus with a certain force, then the new sensor is not created:

$$d(v^{a_k}, v_i^{a_k}) \le \varepsilon^{a_k}$$

To the extreme (minimum and maximum) values of external stimuli react extreme sensors:

$$S_{min}^{a_k} \text{ and } S_{max}^{a_k}$$

CREATION OF ACTIVE ASSOCIATIVE NEURAL GRAPHS

Sensors react (respond) to external stimuli with a specific force depending on the proximity of the stimulus value to the value represented by that sensor, for which it is the most sensitive. Extreme sensors $S^{a_k}_{min}$ and $S^{a_k}_{max}$ use the following formulas to compute their responses:

$$x^{a_k}_{min} = \begin{cases} \frac{v^{a_k}_{max} - v^{a_k}}{r^{a_k}} & if\ r^{a_k} > 0 \\ v^{a_k}_{min} - v^{a_k} + 1 & if\ r^{a_k} = 0 \end{cases}$$

$$x^{a_k}_{max} = \begin{cases} \frac{v^{a_k} - v^{a_k}_{min}}{r^{a_k}} & if\ r^{a_k} > 0 \\ v^{a_k} - v^{a_k}_{max} + 1 & if\ r^{a_k} = 0 \end{cases}$$

Value sensors $S^{a_k}_{v_i}$ represent attribute values and calculate their responses on the sensory input stimulations v^{a_k} on the basis of the following formula:

$$x^{a_k}_{v_i} = \begin{cases} 1 - \frac{|v^{a_k}_i - v^{a_k}|}{r^{a_k}} & if\ r^{a_k} > 0 \\ \frac{|v^{a_k}_i|}{|v^{a_k}_i| + |v^{a_k}_i - v^{a_k}|} & if\ r^{a_k} = 0 \end{cases}$$

The stimulated sensors $S^{a_k}_{v_i}$ start to stimulate connected sensory neurons $R^{a_k}_{v_i}$ with the computed strength $x^{a_k}_{v_i}$ as long as the value v^{a_k} is presented at the input sensory field. This can lead to activation of the connected neurons after a certain period of time which can be computed after:

$$t_{v^{a_k}_i} = \begin{cases} \frac{r^{a_k}}{\theta_{R^{a_k}_{v_i}} \left(r^{a_k} - |v^{a_k}_i - v^{a_k}| \right)} & if\ |v^{a_k}_i - v^{a_k}| < r^{a_k} \\ \infty & if\ |v^{a_k}_i - v^{a_k}| = r^{a_k} \\ 1 + \frac{|v^{a_k}_i - v^{a_k}|}{|v^{a_k}_i|} & if\ r^{a_k} = 0 \end{cases}$$

CREATION OF
ACTIVE ASSOCIATIVE NEURAL GRAPHS

Next, extreme neurons $R_{min}^{a_k}$ and $R_{max}^{a_k}$ can react to extreme values according to their stimulation by extreme sensors. Their reactions can be divided into three categories (ranges):

$x_{min}^{a_k}$, $x_{max}^{a_k}$:

- < 1 for non-extreme values
- = 1 to the values equal to the current extremum (minimum or maximum)
- > 1 to the values that are new extremum to the current one

$$y_{min}^{a_k} = f(x_{min}^{a_k}) = \begin{cases} 1 \; if \; x_{min}^{a_k} \geq \theta_{min}^{a_k} \\ 0 \; if \; x_{min}^{a_k} < \theta_{min}^{a_k} \end{cases}$$

$$y_{max}^{a_k} = f(x_{max}^{a_k}) = \begin{cases} 1 \; if \; x_{max}^{a_k} \geq \theta_{max}^{a_k} \\ 0 \; if \; x_{max}^{a_k} < \theta_{max}^{a_k} \end{cases}$$

Extreme neurons are connected to the value neurons representing extreme values. The connection weights are always equal the activation thresholds of the connected neurons:

$$w_{R_{min}^{a_k}, R_{v_{min}}^{a_k}} = \theta_{R_{v_{min}}^{a_k}} \qquad w_{R_{max}^{a_k}, R_{v_{max}}^{a_k}} = \theta_{R_{v_{max}}^{a_k}}$$

$$w_{R_{v_{min}}^{a_k}, R_{min}^{a_k}} = \theta_{R_{min}^{a_k}} \qquad w_{R_{v_{max}}^{a_k}, R_{max}^{a_k}} = \theta_{R_{max}^{a_k}}$$

The activation thresholds in this model are always equal one (θ = 1).

CREATION OF ACTIVE ASSOCIATIVE NEURAL GRAPHS

Sensory neurons react to stimulations coming from sensors $S_{v_i}^{a_k}$, neighbor sensory neurons $R_{v_i}^{a_k}$, and object neurons O_n according to the following formula:

$$X_{R_{v_i}^{a_k}} = t_{v_i^{a_k}} \cdot x_{v_i}^{a_k} + \sum_{j}^{R_{v_j}^{a_k} \rightsquigarrow R_{v_i}^{a_k}} y_{R_{v_j}^{a_k}} \cdot w_{R_{v_j}^{a_k}, R_{v_i}^{a_k}} + \sum_{n}^{O_n \rightsquigarrow R_{v_i}^{a_k}} y_{O_n} \cdot w_{O_n, R_{v_i}^{a_k}}$$

And calculate their output value depending on the achievement of their activation thresholds:

$$y_{R_{v_j}^{a_k}} = \begin{cases} 1 & if\ X_{R_{v_j}^{a_k}} \geq \theta_{R_{v_j}^{a_k}} \\ 0 & if\ X_{R_{v_j}^{a_k}} < \theta_{R_{v_j}^{a_k}} \end{cases}$$

While sensors can stimulate them for some time, charging them until they reach their activation thresholds which is determined by the following formula:

$$t_{v_i^{a_k}} = \begin{cases} \dfrac{r^{a_k}}{\theta_{R_{v_i}^{a_k}} \left(r^{a_k} - \left| v_i^{a_k} - v^{a_k} \right| \right)} & if\ \left| v_i^{a_k} - v^{a_k} \right| < r^{a_k} \\ \infty & if\ \left| v_i^{a_k} - v^{a_k} \right| = r^{a_k} \\ 1 + \dfrac{\left| v_i^{a_k} - v^{a_k} \right|}{\left| v_i^{a_k} \right|} & if\ r^{a_k} = 0 \end{cases}$$

CREATION OF ACTIVE ASSOCIATIVE NEURAL GRAPHS

Sensory neurons are connected by synapses which weights are determined by:

$$w_{R_{v_i}^{a_k}, R_{v_j}^{a_k}} = 1 - \frac{\left| v_i^{a_k} - v_j^{a_k} \right|}{r^{a_k}}$$

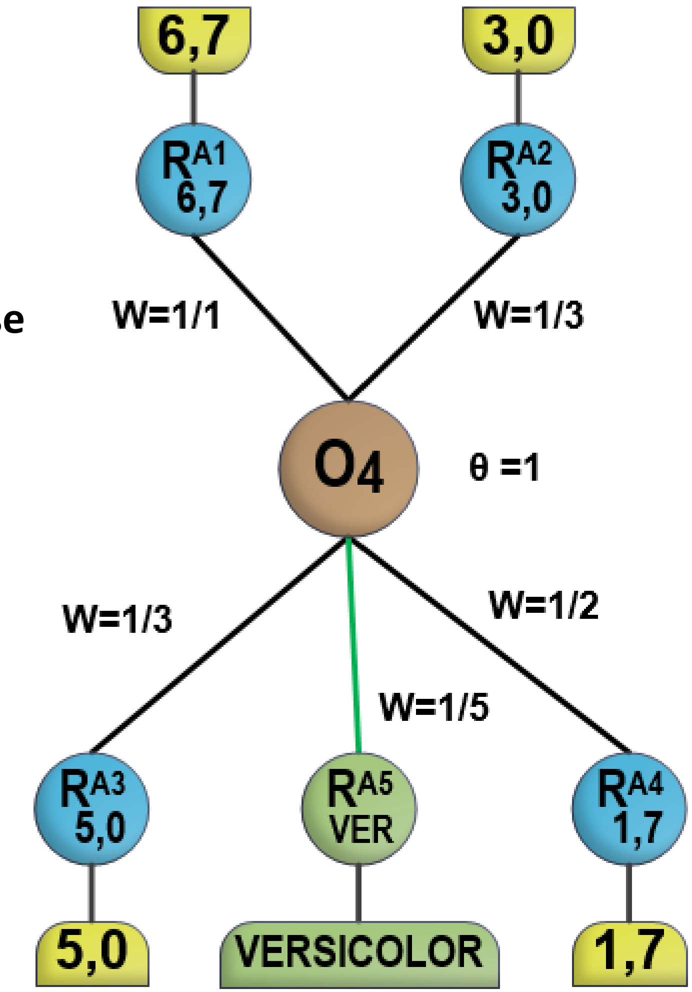
Sensory neurons are connected to object neurons representing objects defined by values represented by these sensory neurons.

The weights of synaptic connections leading from sensory neurons to object neurons are determined by:

$$w_{R_{v_i}^{a_k}, O_n} = \frac{1}{\left\| v_i^{a_k} \right\|}$$

The weights of synaptic connections leading from object neurons to sensory neurons are equal their activation thresholds:

$$w_{O_n, R_{v_i}^{a_k}} = \theta_{R_{v_i}^{a_k}} = 1$$



CREATION OF ACTIVE ASSOCIATIVE NEURAL GRAPHS

The stimulation of object neurons is determined by:

$$X_{O_n} = \sum_{k}^{R_{v_{n_k}}^{a_k} \rightsquigarrow O_n} y_{R_{v_{n_k}}^{a_k}} \cdot w_{R_{v_{n_k}}^{a_k}, O_n}$$

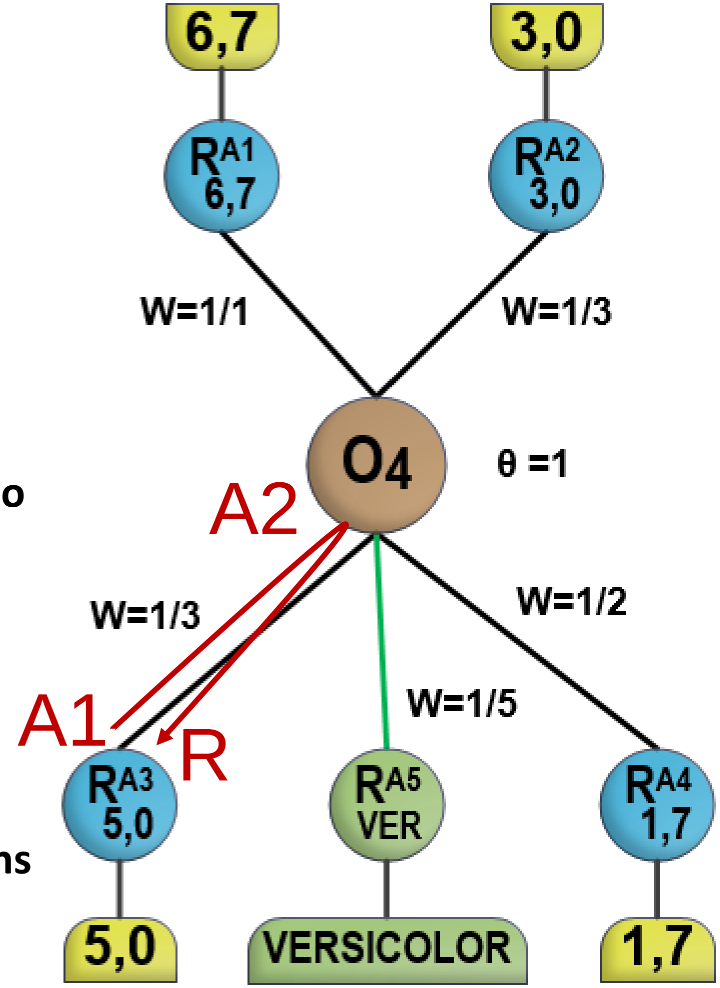
And their output value is computed as follows:

$$y_{O_n} = \begin{cases} 1 & if\ X_{O_n} \geq \theta_{O_n} \\ 0 & if\ X_{O_n} < \theta_{O_n} \end{cases}$$

Where the neuron activation thresholds are initially equal to one:

$$\theta_{O_n} = 1$$

Thanks this, if there is presented an input combination defining a known object on the sensory input fields, the neuron representing this combination will activate at first. The other neurons representing similar combinations will activate later if the input combination is further presented on the input sensory fields of the AANG.



Neurons, which were activated (e.g. A1) are for some time in the refractory states (R), so they are not reactive to any stimulations, e.g. the one coming back from the neuron A2.

CREATION OF ACTIVE ASSOCIATIVE NEURAL GRAPHS

Sensory neurons should not only react to sensory stimuli of a specific intensity but also stimulate other connected sensory neurons with the most similar values.

Hence, there is necessary the self-organizing capability of the AANG network.

The sensory connective plasticity rule determines in which cases the plasticity results in the creation or reconfiguration of existing connections between sensory neurons.

The sensory connective plasticity rule between sensory neurons says that the sensory neuron $R_{v_j}^{a_k}$ will disconnect with the neuron $R_{v_i}^{a_k}$ which stimulates it weaker than the connected sensor $S_{v_j}^{a_k}$:

$$0 < y_{R_{v_i}^{a_k}} \cdot w_{R_{v_i}^{a_k}, R_{v_j}^{a_k}} < x_{v_j^{a_k}} - \varepsilon^{a_k}$$

Sensory neurons are thus programmed to require precisely two connections with the remaining sensory neurons or extreme neurons.

Disconnection thus triggers the neuronal process of connective plasticity, which will look for other neurons that wish to connect at a given moment.

Therefore, if a new sensor and its new sensory neuron for a not yet represented new value in a given sensory field is created then this new sensory neuron will try to connect to these two disconnected neurons.

In result, the new sensory neuron representing the value v^{a_k} will join the others in an orderly way:

$$v_j^{a_k} < v^{a_k} < v_i^{a_k} \; or \; v_j^{a_k} > v^{a_k} > v_i^{a_k}$$

CREATION OF ACTIVE ASSOCIATIVE NEURAL GRAPHS

However, this plasticity is only possible in the sensory neuron stimulated simultaneously by the sensor and another previously activated sensory neuron.

Therefore, it is important to take into consideration and computation the time and the order of activations of the individual sensory neurons in time to make such plasticity.

The activation time of sensory neurons as a result of their stimulation by the connected sensors will vary depending on the similarity of represented values by sensors to the presented value on their input sensory fields:

$$t_{v_i^{a_k}} = \begin{cases} \dfrac{r^{a_k}}{\theta_{R_{v_i}^{a_k}}\left(r^{a_k} - \left|v_i^{a_k} - v^{a_k}\right|\right)} & if\ \left|v_i^{a_k} - v^{a_k}\right| < r^{a_k} \\ \infty & if\ \left|v_i^{a_k} - v^{a_k}\right| = r^{a_k} \\ 1 + \dfrac{\left|v_i^{a_k} - v^{a_k}\right|}{\left|v_i^{a_k}\right|} & if\ r^{a_k} = 0 \end{cases}$$

The neuron, which activates first as a result of such sensory stimulation, sends a weighted signal to the two connected sensory (or extreme) neurons. Always only one of it will satisfy the connective plasticity condition:

$$0 < y_{R_{v_i}^{a_k}} \cdot w_{R_{v_i}^{a_k}, R_{v_j}^{a_k}} = 1 \cdot \left(1 - \frac{\left|v_i^{a_k} - v_j^{a_k}\right|}{r^{a_k}}\right) < 1 - \frac{\left|v_j^{a_k} - v^{a_k}\right|}{r^{a_k}} = x_{v_j^{a_k}}$$

and breaks its connection to this neuron because:

$$\left|v_i^{a_k} - v_j^{a_k}\right| > \left|v_j^{a_k} - v^{a_k}\right| if\ and\ only\ if\ v_j^{a_k} < v^{a_k} < v_i^{a_k}\ or\ v_j^{a_k} > v^{a_k} > v_i^{a_k}$$

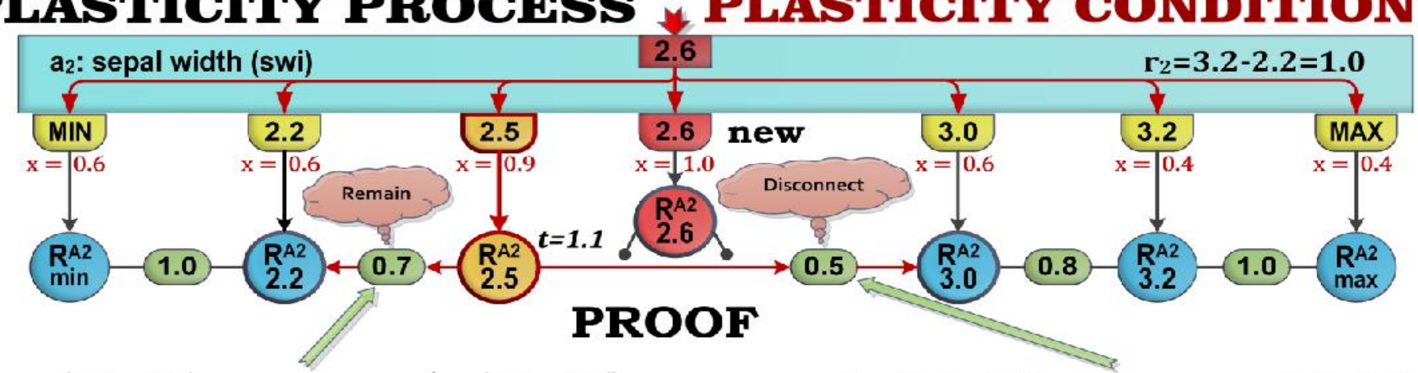
The presented algorithm is called the ASSORT-2 associative sort algorithm and is used for the automatic and incremental construction of the AANG neural network for any set of patterns.

CREATION OF ACTIVE ASSOCIATIVE NEURAL GRAPHS

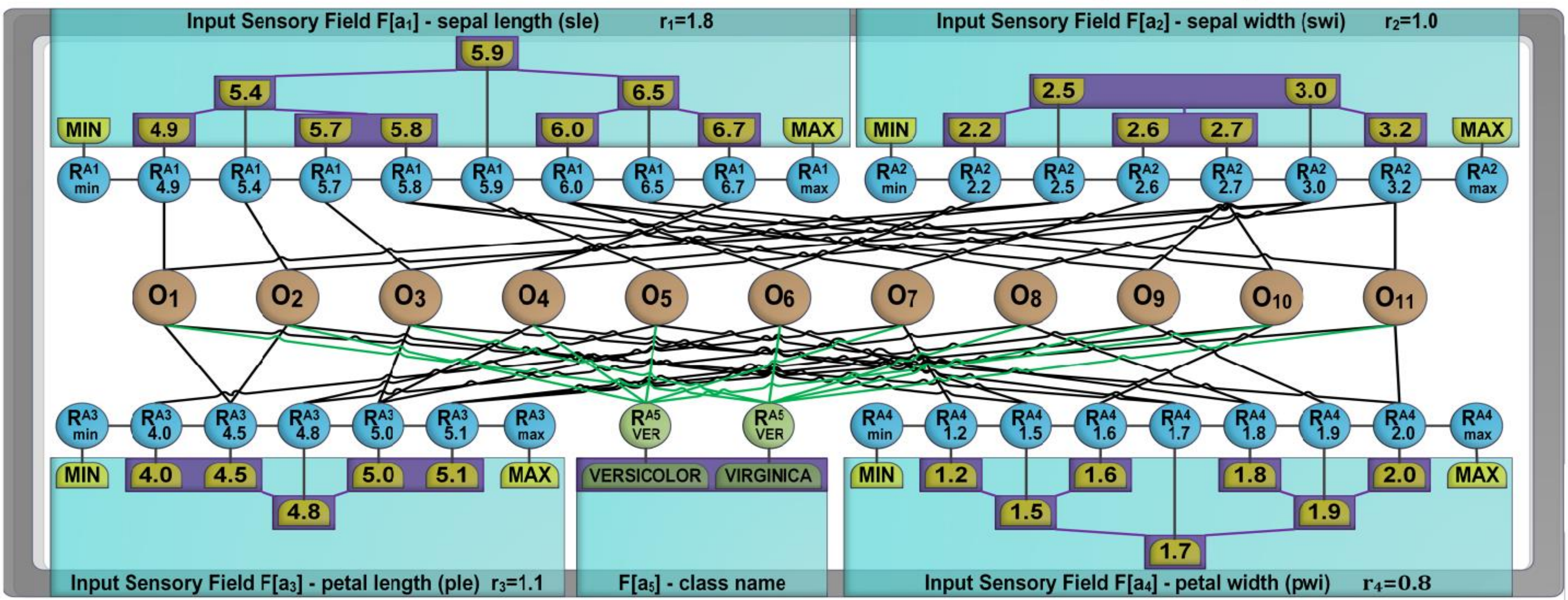
We can get the following graph structure built by the ASSORT-2 algorithm:

CONDITIONAL PLASTICITY PROCESS — PLASTICITY CONDITION

	Attributes $a_1,\dots,a_5$				
	sle	swi	ple	pwi	class name
O_1	4.9	2.5	4.5	1.7	VIRGINICA
O_2	5.4	3.0	4.5	1.5	VERSICOLOR
O_3	5.7	2.5	5.0	2.0	VIRGINICA
O_4	6.7	3.0	5.0	1.7	VERSICOLOR
O_5	5.9	3.2	4.8	1.8	VERSICOLOR
O_6	6.0	2.2	5.0	1.5	VIRGINICA
O_7	5.8	2.6	4.0	1.2	VERSICOLOR
O_8	6.0	3.0	4.8	1.8	VIRGINICA
O_9	5.8	2.7	5.1	1.9	VIRGINICA
O_{10}	6.0	2.7	5.1	1.6	VERSICOLOR
O_{11}	6.5	3.2	5.1	2.0	VIRGINICA



$$1-\frac{|2.2-2.6|}{1.0} = 0.6 \ngtr 0.7 = 1\cdot\left(1-\frac{|2.5-2.2|}{1.0}\right) > 0 \qquad 0 < 1\cdot\left(1-\frac{|2.5-3.0|}{1.0}\right) = 0.5 < 0.6 = 1-\frac{|3.0-2.6|}{1.0}$$



ASSORT FOR A SELECTED SUBSET OF IRIS DATA

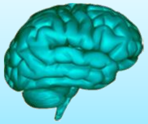
SORTED SUBSET OF IRIS PATTERNS

	Attributes				
	sle	swi	ple	pwi	class name
R1	5.8	2.6	4.0	1.2	VERSICOLOR
R2	5.4	3.0	4.5	1.5	VERSICOLOR
R3	6.0	2.7	5.1	1.6	VERSICOLOR
R4	6.7	3.0	5.0	1.7	VERSICOLOR
R5	5.9	3.2	4.8	1.8	VERSICOLOR
R6	6.0	2.2	5.0	1.5	VIRGINICA
R7	4.9	2.5	4.5	1.7	VIRGINICA
R8	6.0	3.0	4.8	1.8	VIRGINICA
R9	5.8	2.7	5.1	1.9	VIRGINICA
R10	5.7	2.5	5.0	2.0	VIRGINICA
R11	6.5	3.2	5.1	2.0	VIRGINICA

STEP 1

STEP 1. CREATION OF A NEW GRAPH

Create a new AANG graph for a set of objects stored in a tabular structure (table).

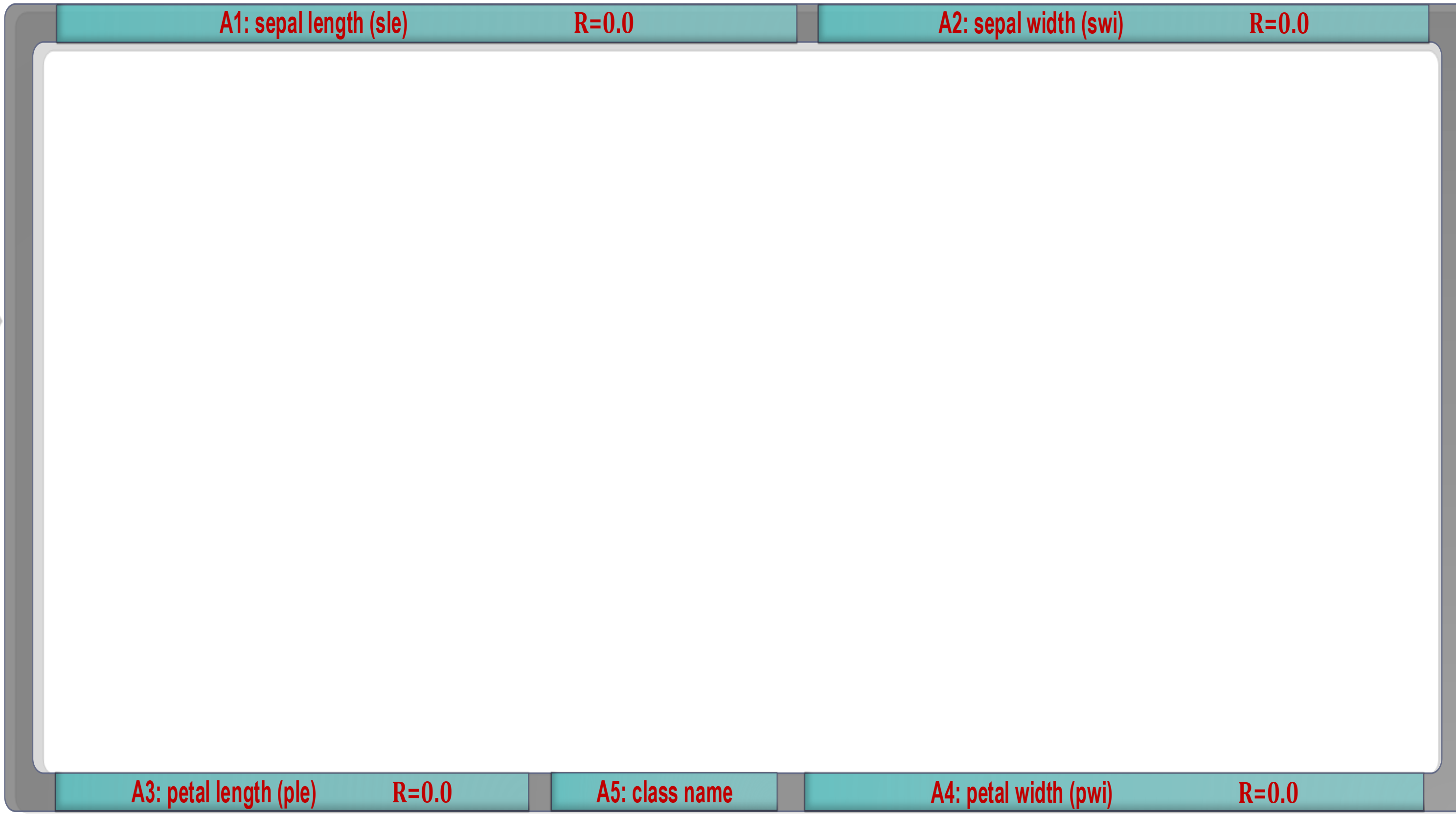


ASSORT FOR A SELECTED SUBSET OF IRIS DATA

SORTED SUBSET OF IRIS PATTERNS

	Attributes				
	sle	swi	ple	pwi	class name
R1	5.8	2.6	4.0	1.2	VERSICOLOR
R2	5.4	3.0	4.5	1.5	VERSICOLOR
R3	6.0	2.7	5.1	1.6	VERSICOLOR
R4	6.7	3.0	5.0	1.7	VERSICOLOR
R5	5.9	3.2	4.8	1.8	VERSICOLOR
R6	6.0	2.2	5.0	1.5	VIRGINICA
R7	4.9	2.5	4.5	1.7	VIRGINICA
R8	6.0	3.0	4.8	1.8	VIRGINICA
R9	5.8	2.7	5.1	1.9	VIRGINICA
R10	5.7	2.5	5.0	2.0	VIRGINICA
R11	6.5	3.2	5.1	2.0	VIRGINICA

STEP 2



STEP 2. CREATION OF SENSORY FIELDS IN THE GRAPH (INPUT INTERFACES FOR THE AANG)

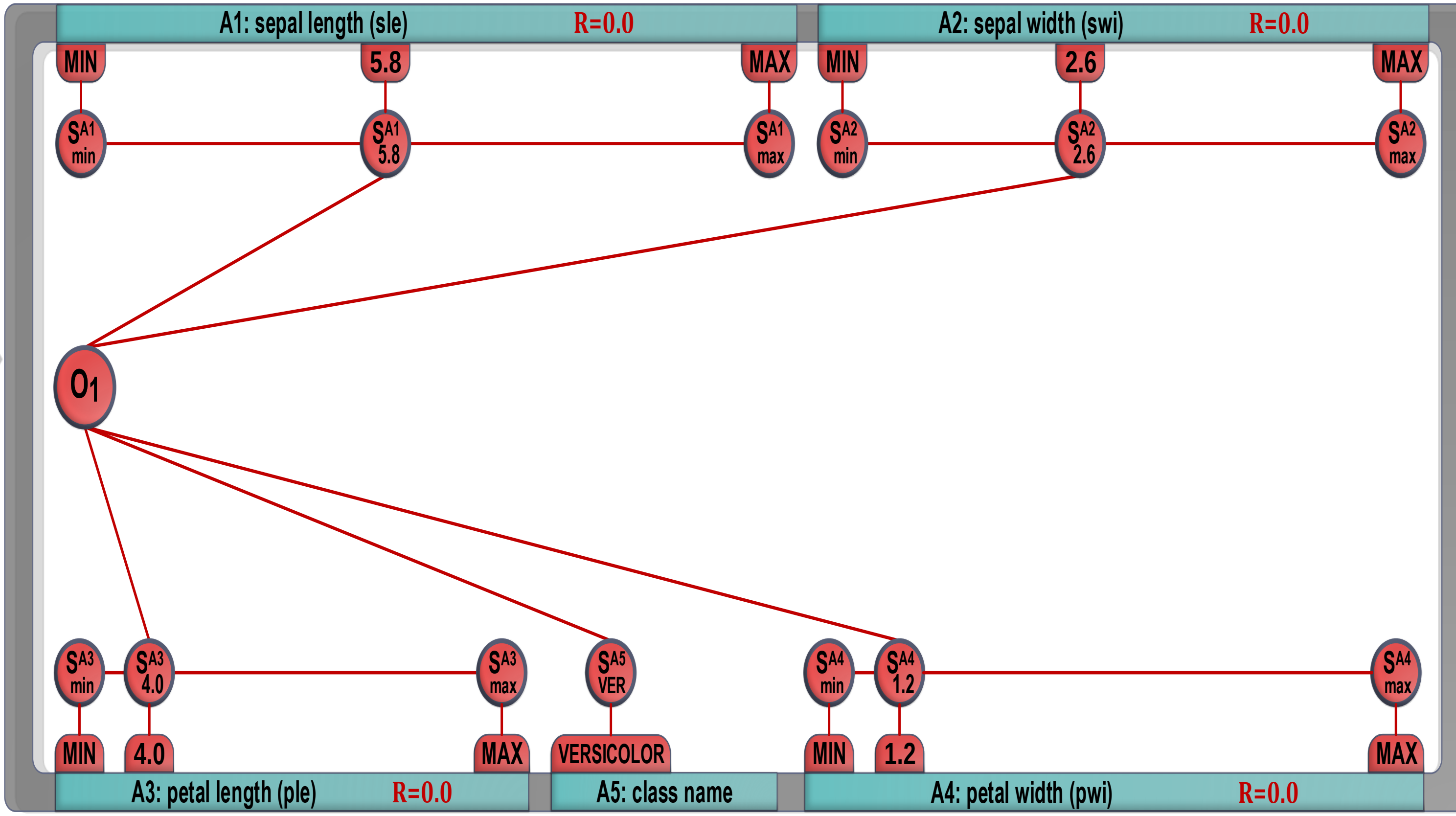
Create new sensory fields for all known attributes defined in the transformed table.

ASSORT FOR A SELECTED SUBSET OF IRIS DATA

SORTED SUBSET OF IRIS PATTERNS

	Attributes				
	sle	swi	ple	pwi	class name
R1	5.8	2.6	4.0	1.2	VERSICOLOR
R2	5.4	3.0	4.5	1.5	VERSICOLOR
R3	6.0	2.7	5.1	1.6	VERSICOLOR
R4	6.7	3.0	5.0	1.7	VERSICOLOR
R5	5.9	3.2	4.8	1.8	VERSICOLOR
R6	6.0	2.2	5.0	1.5	VIRGINICA
R7	4.9	2.5	4.5	1.7	VIRGINICA
R8	6.0	3.0	4.8	1.8	VIRGINICA
R9	5.8	2.7	5.1	1.9	VIRGINICA
R10	5.7	2.5	5.0	2.0	VIRGINICA
R11	6.5	3.2	5.1	2.0	VIRGINICA

STEP 3



STEP 3. CREATION OF ASSOCIATIVE NEURONAL REPRESENTATION OF THE OBJECT (RECORD) R1

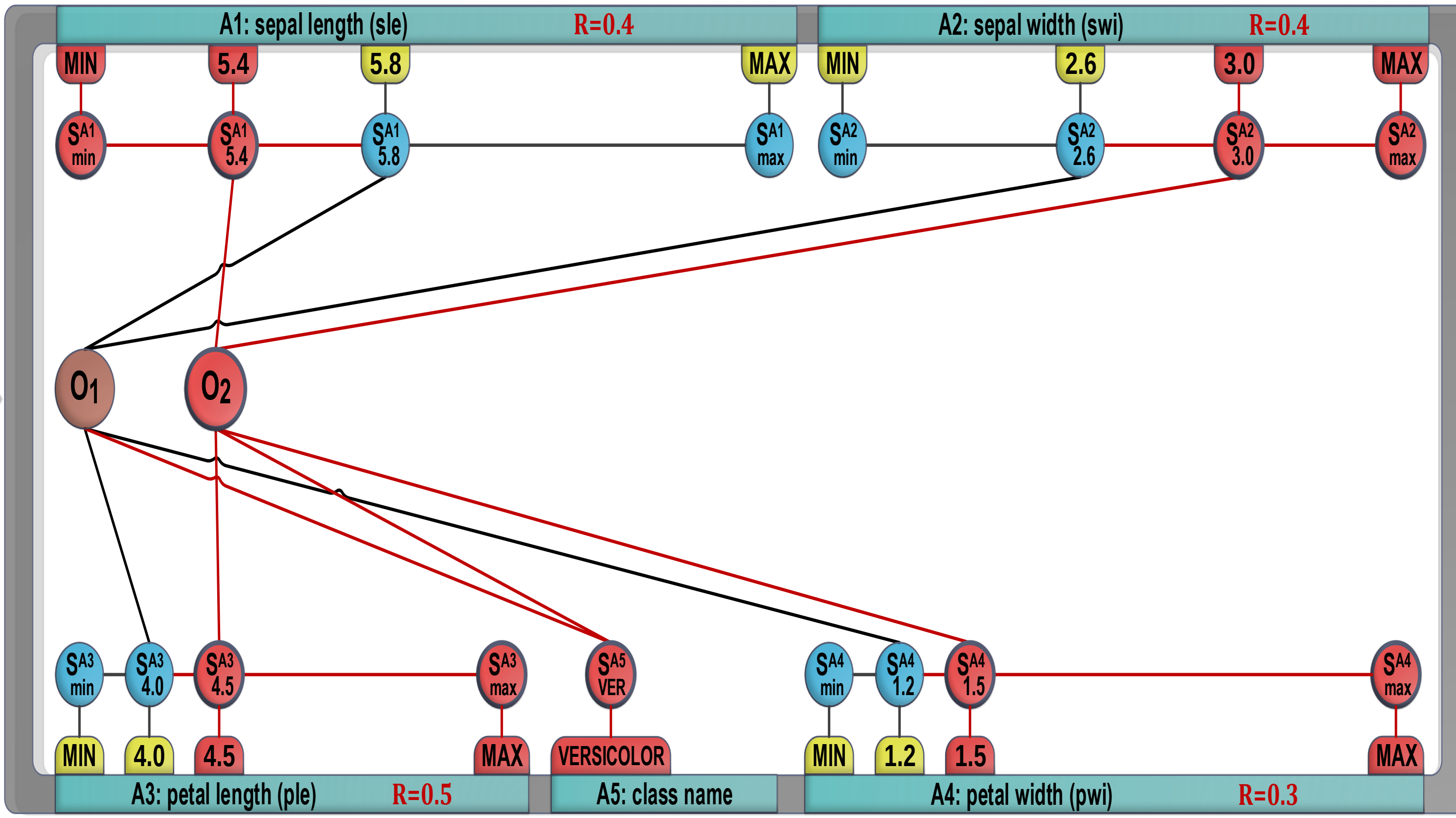
Create an object representation related to new sensory neurons and sensors using ASSORT **the sensory connective plasticity rule**, which connects new sensory neurons in order.

ASSORT FOR A SELECTED SUBSET OF IRIS DATA

SORTED SUBSET OF IRIS PATTERNS

	Attributes				
	sle	swi	ple	pwi	class name
R1	5.8	2.6	4.0	1.2	VERSICOLOR
R2	5.4	3.0	4.5	1.5	VERSICOLOR
R3	6.0	2.7	5.1	1.6	VERSICOLOR
R4	6.7	3.0	5.0	1.7	VERSICOLOR
R5	5.9	3.2	4.8	1.8	VERSICOLOR
R6	6.0	2.2	5.0	1.5	VIRGINICA
R7	4.9	2.5	4.5	1.7	VIRGINICA
R8	6.0	3.0	4.8	1.8	VIRGINICA
R9	5.8	2.7	5.1	1.9	VIRGINICA
R10	5.7	2.5	5.0	2.0	VIRGINICA
R11	6.5	3.2	5.1	2.0	VIRGINICA

STEP 4



STEP 4. CREATION OF ASSOCIATIVE NEURONAL REPRESENTATION OF THE OBJECT (RECORD) R2

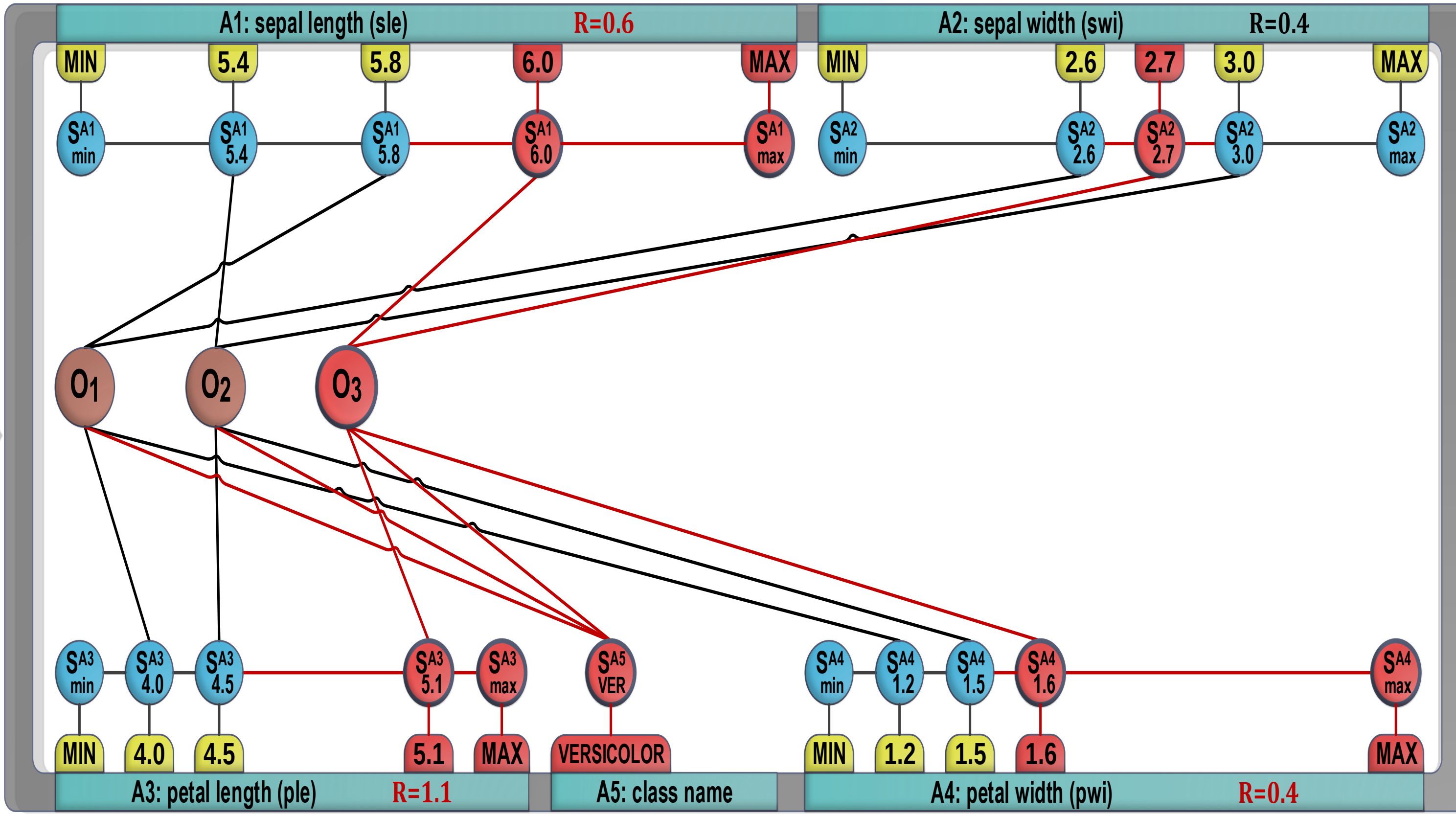
Create a representation of another object (R2) in the AANG structure using already created sensors and sensory neurons, aggregating, not duplicating, representation of the same values (VERSICOLOR).

ASSORT FOR A SELECTED SUBSET OF IRIS DATA

SORTED SUBSET OF IRIS PATTERNS

	Attributes				
	sle	swi	ple	pwi	class name
R1	5.8	2.6	4.0	1.2	VERSICOLOR
R2	5.4	3.0	4.5	1.5	VERSICOLOR
R3	6.0	2.7	5.1	1.6	VERSICOLOR
R4	6.7	3.0	5.0	1.7	VERSICOLOR
R5	5.9	3.2	4.8	1.8	VERSICOLOR
R6	6.0	2.2	5.0	1.5	VIRGINICA
R7	4.9	2.5	4.5	1.7	VIRGINICA
R8	6.0	3.0	4.8	1.8	VIRGINICA
R9	5.8	2.7	5.1	1.9	VIRGINICA
R10	5.7	2.5	5.0	2.0	VIRGINICA
R11	6.5	3.2	5.1	2.0	VIRGINICA

STEP 5



STEP 5. CREATION OF ASSOCIATIVE NEURONAL REPRESENTATION OF THE OBJECT (RECORD) R3

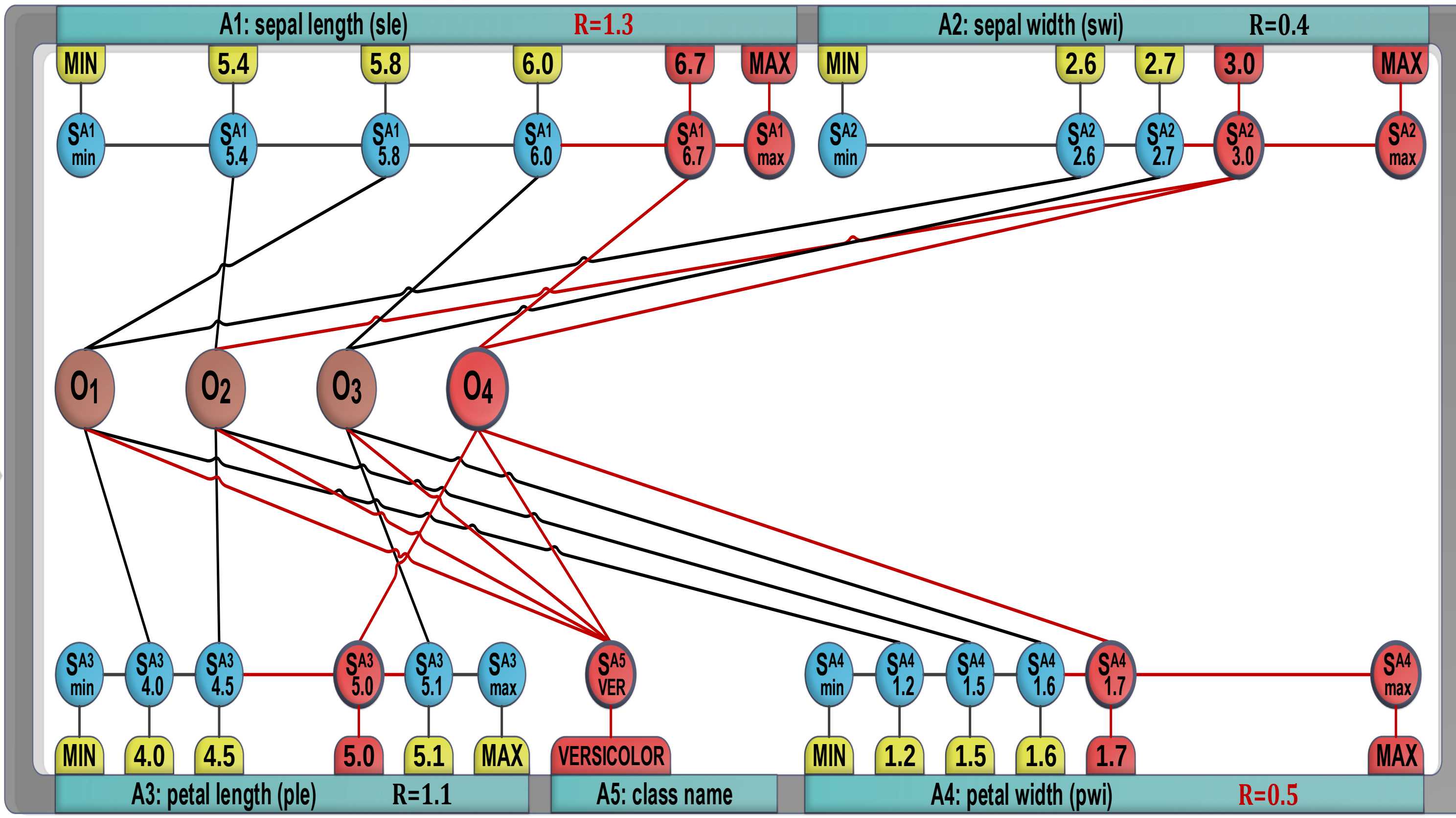
Presentation of further objects causes stimulation and activation of sensory neurons if they represent presented values or addition of new ones if the presented values are not yet represented.

ASSORT FOR A SELECTED SUBSET OF IRIS DATA

SORTED SUBSET OF IRIS PATTERNS

	Attributes				
	sle	swi	ple	pwi	class name
R1	5.8	2.6	4.0	1.2	VERSICOLOR
R2	5.4	3.0	4.5	1.5	VERSICOLOR
R3	6.0	2.7	5.1	1.6	VERSICOLOR
R4	6.7	3.0	5.0	1.7	VERSICOLOR
R5	5.9	3.2	4.8	1.8	VERSICOLOR
R6	6.0	2.2	5.0	1.5	VIRGINICA
R7	4.9	2.5	4.5	1.7	VIRGINICA
R8	6.0	3.0	4.8	1.8	VIRGINICA
R9	5.8	2.7	5.1	1.9	VIRGINICA
R10	5.7	2.5	5.0	2.0	VIRGINICA
R11	6.5	3.2	5.1	2.0	VIRGINICA

STEP 6



STEP 6. CREATION OF ASSOCIATIVE NEURONAL REPRESENTATION OF THE OBJECT (RECORD) R4

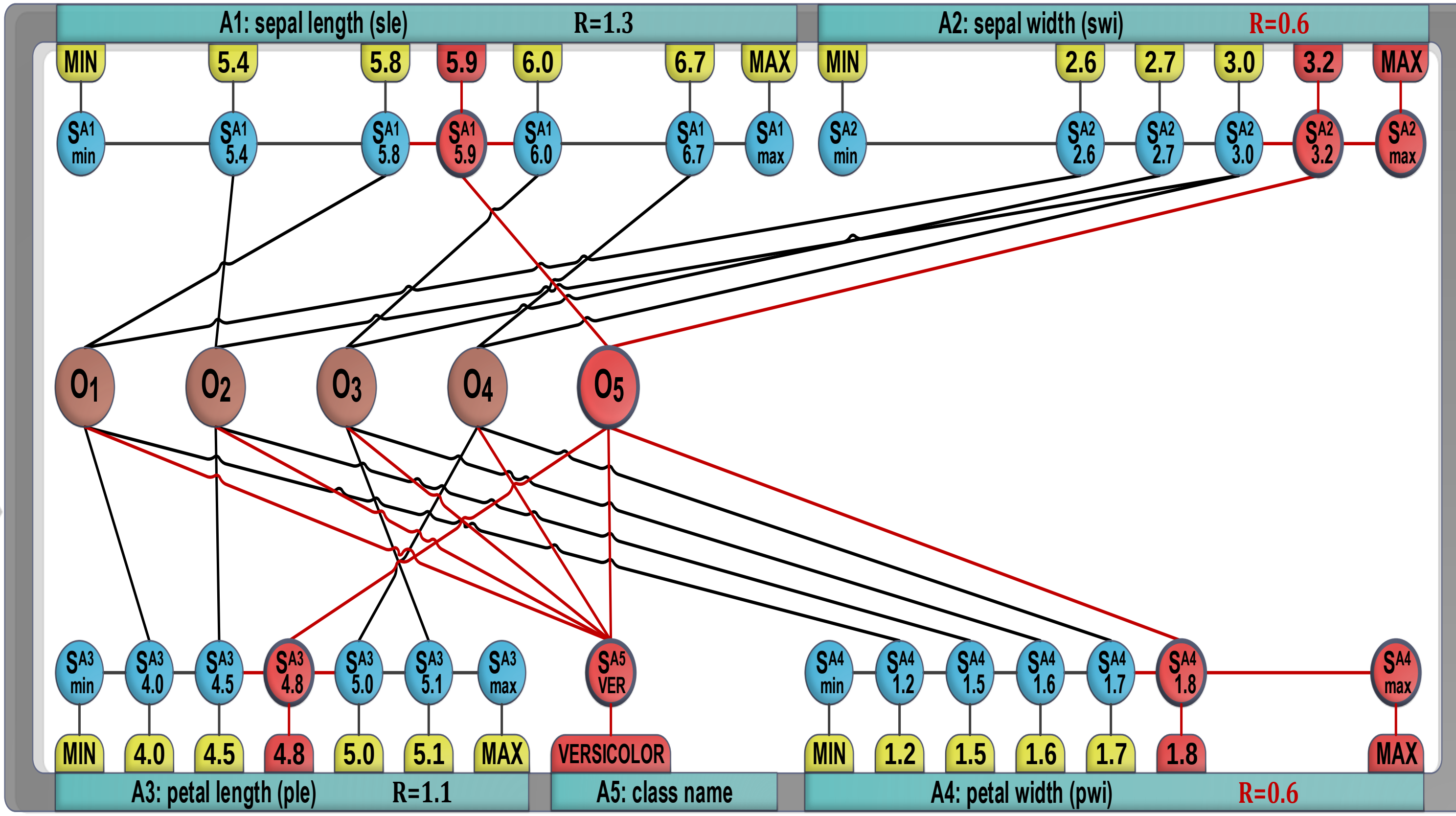
There is visible the aggregated representation of the same attribute values by the same sensors and sensory neurons, e.g. 3.0 for the attribute A2: sepal width.

ASSORT FOR A SELECTED SUBSET OF IRIS DATA

SORTED SUBSET OF IRIS PATTERNS

	Attributes				
	sle	swi	ple	pwi	class name
R1	5.8	2.6	4.0	1.2	VERSICOLOR
R2	5.4	3.0	4.5	1.5	VERSICOLOR
R3	6.0	2.7	5.1	1.6	VERSICOLOR
R4	6.7	3.0	5.0	1.7	VERSICOLOR
R5	5.9	3.2	4.8	1.8	VERSICOLOR
R6	6.0	2.2	5.0	1.5	VIRGINICA
R7	4.9	2.5	4.5	1.7	VIRGINICA
R8	6.0	3.0	4.8	1.8	VIRGINICA
R9	5.8	2.7	5.1	1.9	VIRGINICA
R10	5.7	2.5	5.0	2.0	VIRGINICA
R11	6.5	3.2	5.1	2.0	VIRGINICA

STEP 7



STEP 7. CREATION OF ASSOCIATIVE NEURONAL REPRESENTATION OF THE OBJECT (RECORD) R5

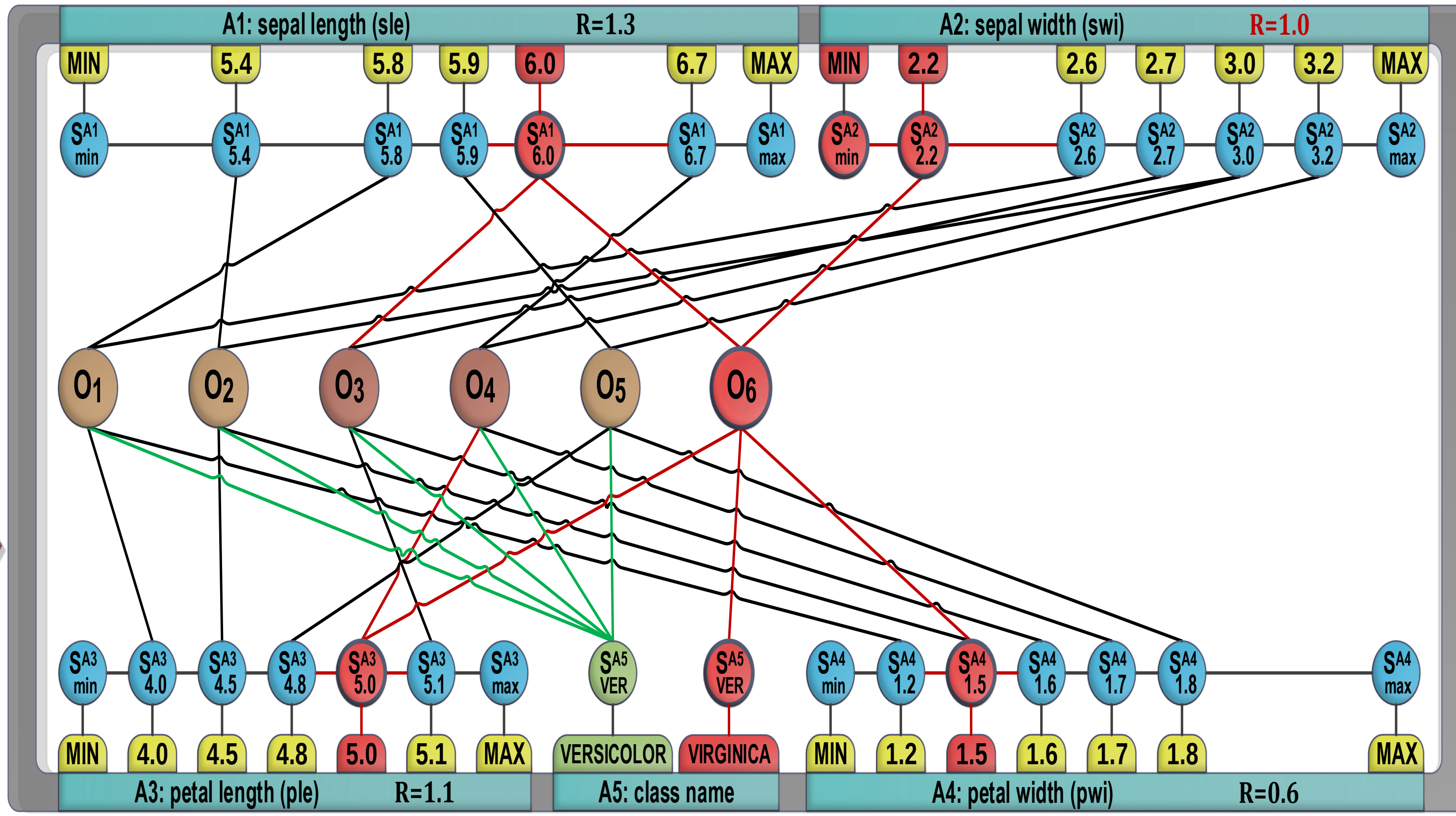
Sometimes MIN or MAX sensors are also activated when the values presented on the sensory input fields are minimum or maximum in the range of given attributes.

ASSORT FOR A SELECTED SUBSET OF IRIS DATA

SORTED SUBSET OF IRIS PATTERNS

	Attributes				
	sle	swi	ple	pwi	class name
R1	5.8	2.6	4.0	1.2	VERSICOLOR
R2	5.4	3.0	4.5	1.5	VERSICOLOR
R3	6.0	2.7	5.1	1.6	VERSICOLOR
R4	6.7	3.0	5.0	1.7	VERSICOLOR
R5	5.9	3.2	4.8	1.8	VERSICOLOR
R6	6.0	2.2	5.0	1.5	VIRGINICA
R7	4.9	2.5	4.5	1.7	VIRGINICA
R8	6.0	3.0	4.8	1.8	VIRGINICA
R9	5.8	2.7	5.1	1.9	VIRGINICA
R10	5.7	2.5	5.0	2.0	VIRGINICA
R11	6.5	3.2	5.1	2.0	VIRGINICA

STEP 8



STEP 8. CREATION OF ASSOCIATIVE NEURONAL REPRESENTATION OF THE OBJECT (RECORD) R6

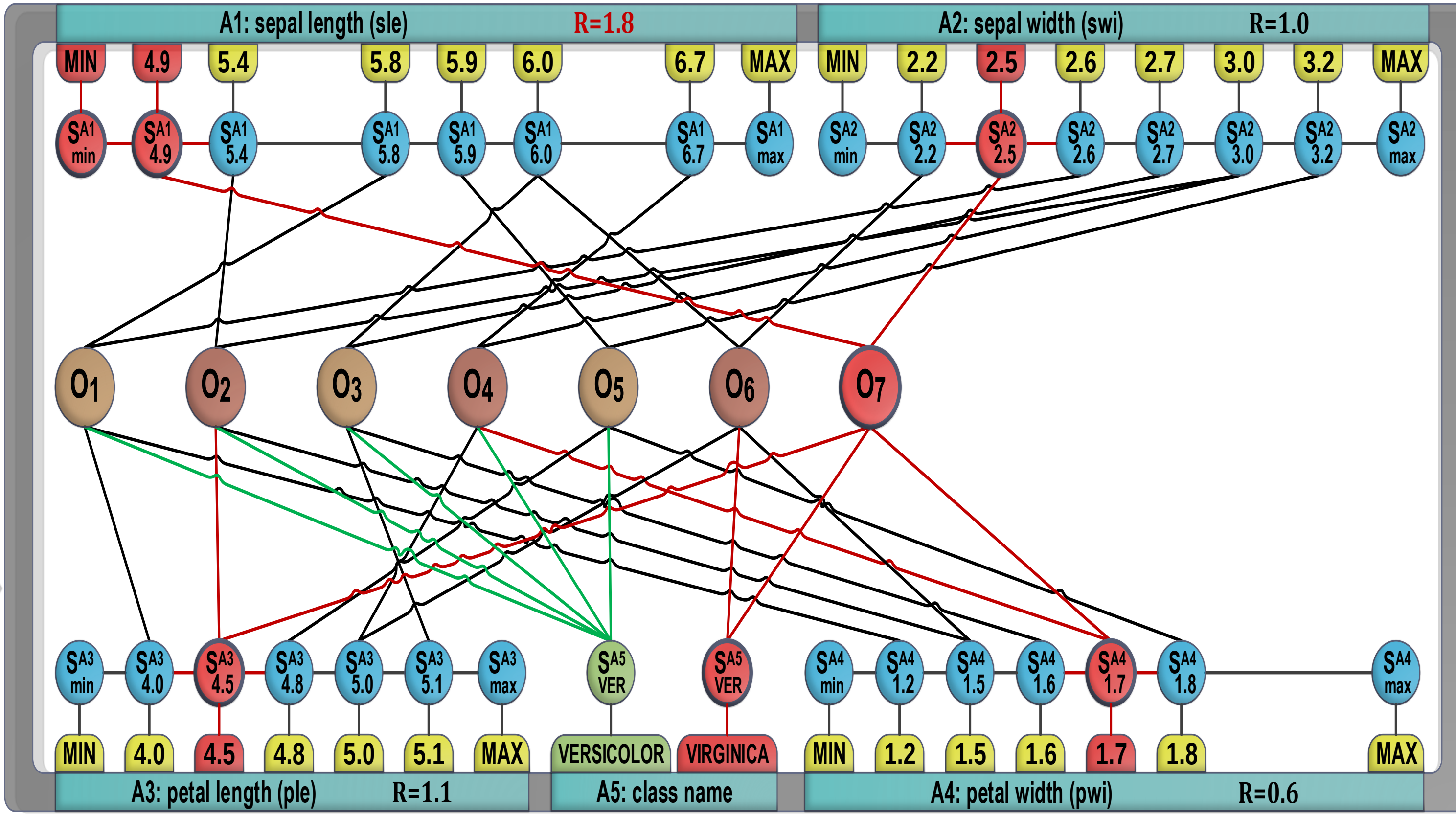
The number and level of aggregations will grow together with the number of represented objects, e.g. 6.0 for the attribute A1 and 5.0 for the attribute A3), which reduces the cost of representation.

ASSORT FOR A SELECTED SUBSET OF IRIS DATA

SORTED SUBSET OF IRIS PATTERNS

	Attributes				
	sle	swi	ple	pwi	class name
R1	5.8	2.6	4.0	1.2	VERSICOLOR
R2	5.4	3.0	4.5	1.5	VERSICOLOR
R3	6.0	2.7	5.1	1.6	VERSICOLOR
R4	6.7	3.0	5.0	1.7	VERSICOLOR
R5	5.9	3.2	4.8	1.8	VERSICOLOR
R6	6.0	2.2	5.0	1.5	VIRGINICA
R7	4.9	2.5	4.5	1.7	VIRGINICA
R8	6.0	3.0	4.8	1.8	VIRGINICA
R9	5.8	2.7	5.1	1.9	VIRGINICA
R10	5.7	2.5	5.0	2.0	VIRGINICA
R11	6.5	3.2	5.1	2.0	VIRGINICA

STEP 9



STEP 9. CREATION OF ASSOCIATIVE NEURONAL REPRESENTATION OF THE OBJECT (RECORD) R7

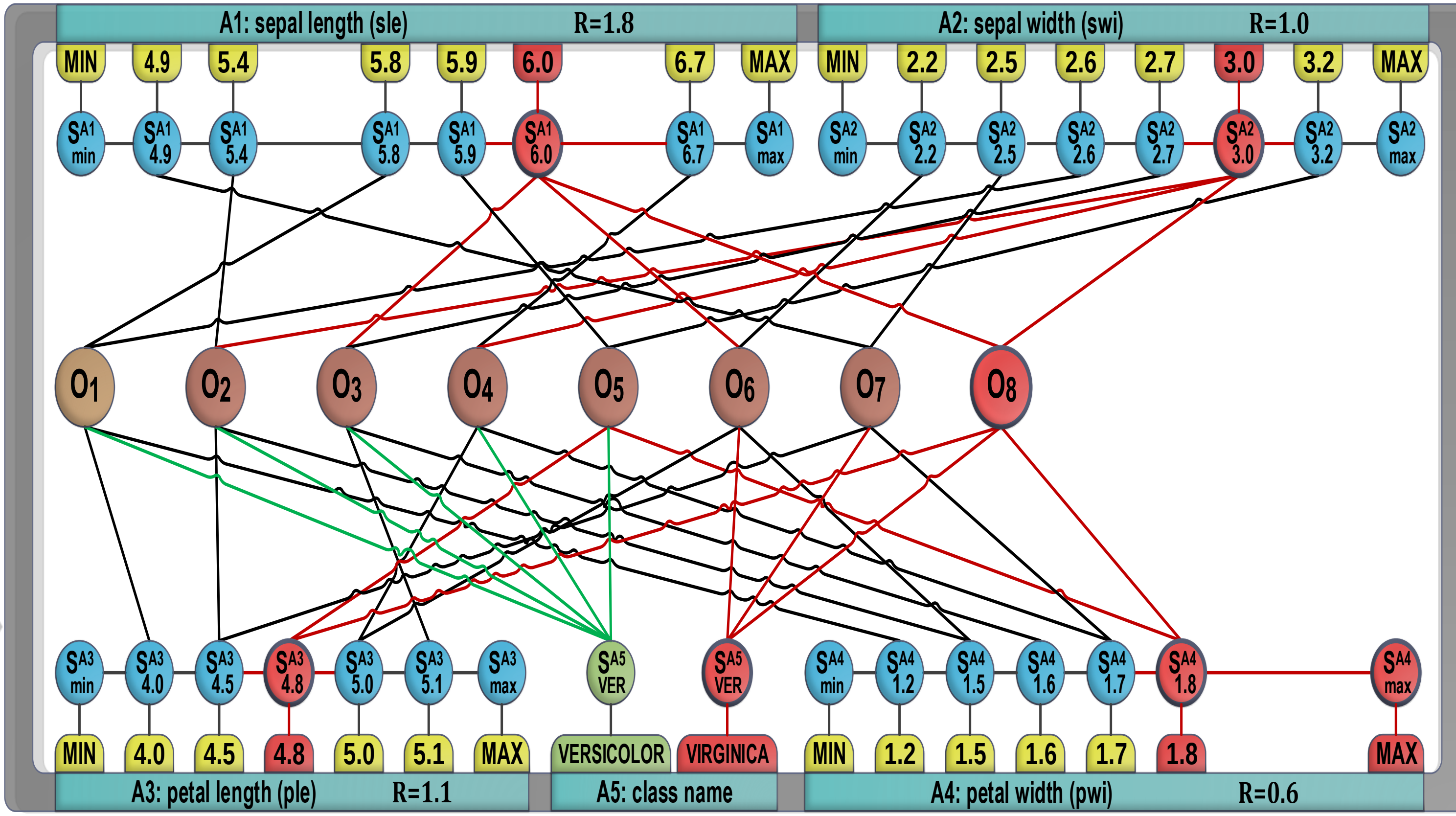
Subsequent object aggregations enable automatic associations between objects, e.g. the sensory neuron 4.5 for the attribute A3 links together objects R2 and R7, and 1.7 for the attribute A4 links together objects R3 and R7.

ASSORT FOR A SELECTED SUBSET OF IRIS DATA

SORTED SUBSET OF IRIS PATTERNS

	Attributes				
	sle	swi	ple	pwi	class name
R1	5.8	2.6	4.0	1.2	VERSICOLOR
R2	5.4	3.0	4.5	1.5	VERSICOLOR
R3	6.0	2.7	5.1	1.6	VERSICOLOR
R4	6.7	3.0	5.0	1.7	VERSICOLOR
R5	5.9	3.2	4.8	1.8	VERSICOLOR
R6	6.0	2.2	5.0	1.5	VIRGINICA
R7	4.9	2.5	4.5	1.7	VIRGINICA
R8	6.0	3.0	4.8	1.8	VIRGINICA
R9	5.8	2.7	5.1	1.9	VIRGINICA
R10	5.7	2.5	5.0	2.0	VIRGINICA
R11	6.5	3.2	5.1	2.0	VIRGINICA

STEP 10



STEP 10. CREATION OF ASSOCIATIVE NEURONAL REPRESENTATION OF THE OBJECT (RECORD) R8

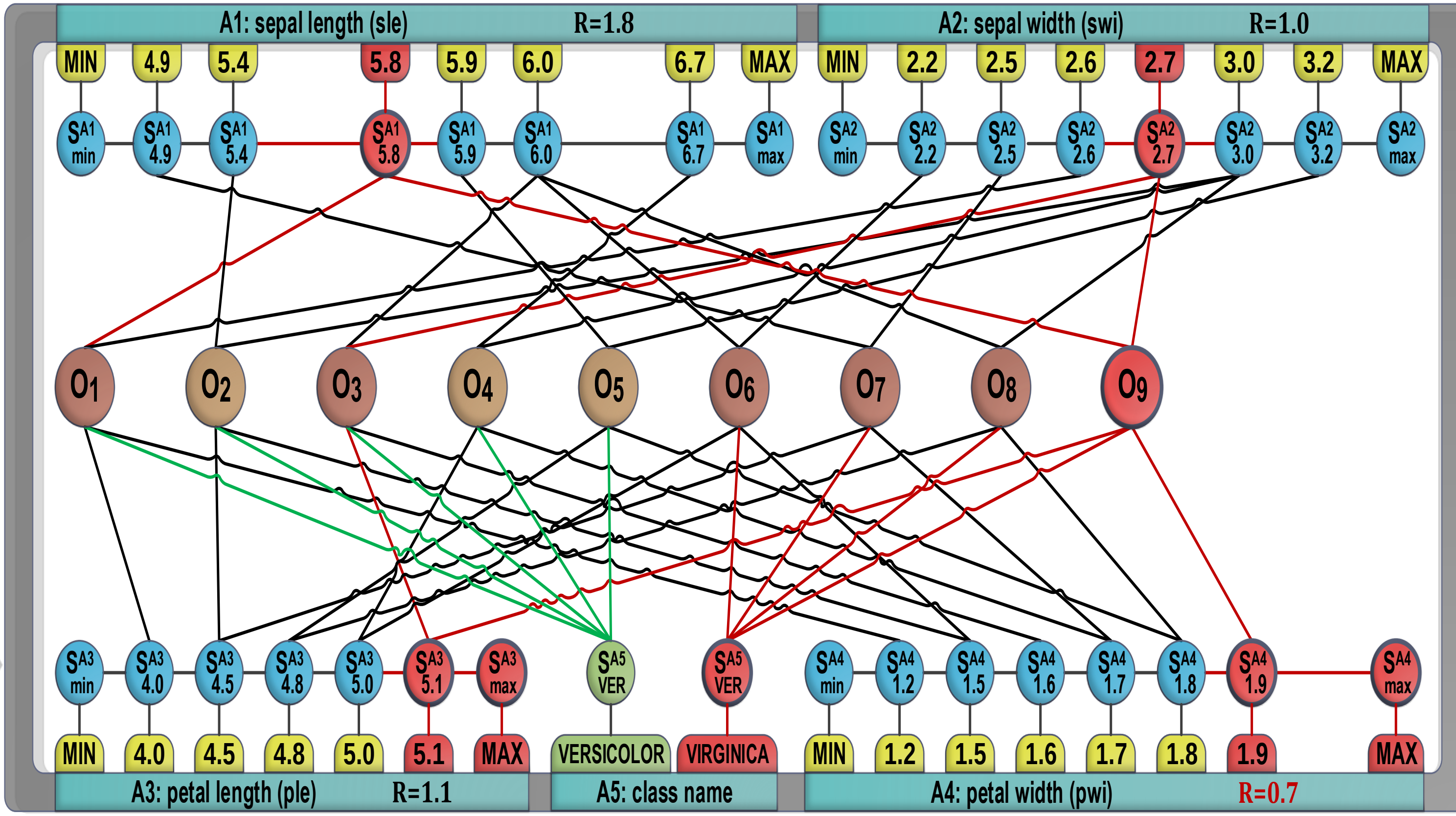
Neuronal aggregates also allow for automatic grouping (clustering and classification) of objects against any combination of input values as well as similar values that are linked together.

ASSORT FOR A SELECTED SUBSET OF IRIS DATA

SORTED SUBSET OF IRIS PATTERNS

	Attributes				
	sle	swi	ple	pwi	class name
R1	5.8	2.6	4.0	1.2	VERSICOLOR
R2	5.4	3.0	4.5	1.5	VERSICOLOR
R3	6.0	2.7	5.1	1.6	VERSICOLOR
R4	6.7	3.0	5.0	1.7	VERSICOLOR
R5	5.9	3.2	4.8	1.8	VERSICOLOR
R6	6.0	2.2	5.0	1.5	VIRGINICA
R7	4.9	2.5	4.5	1.7	VIRGINICA
R8	6.0	3.0	4.8	1.8	VIRGINICA
R9	5.8	2.7	5.1	1.9	VIRGINICA
R10	5.7	2.5	5.0	2.0	VIRGINICA
R11	6.5	3.2	5.1	2.0	VIRGINICA

STEP 11



STEP 11. CREATION OF ASSOCIATIVE NEURONAL REPRESENTATION OF THE OBJECT (RECORD) R9

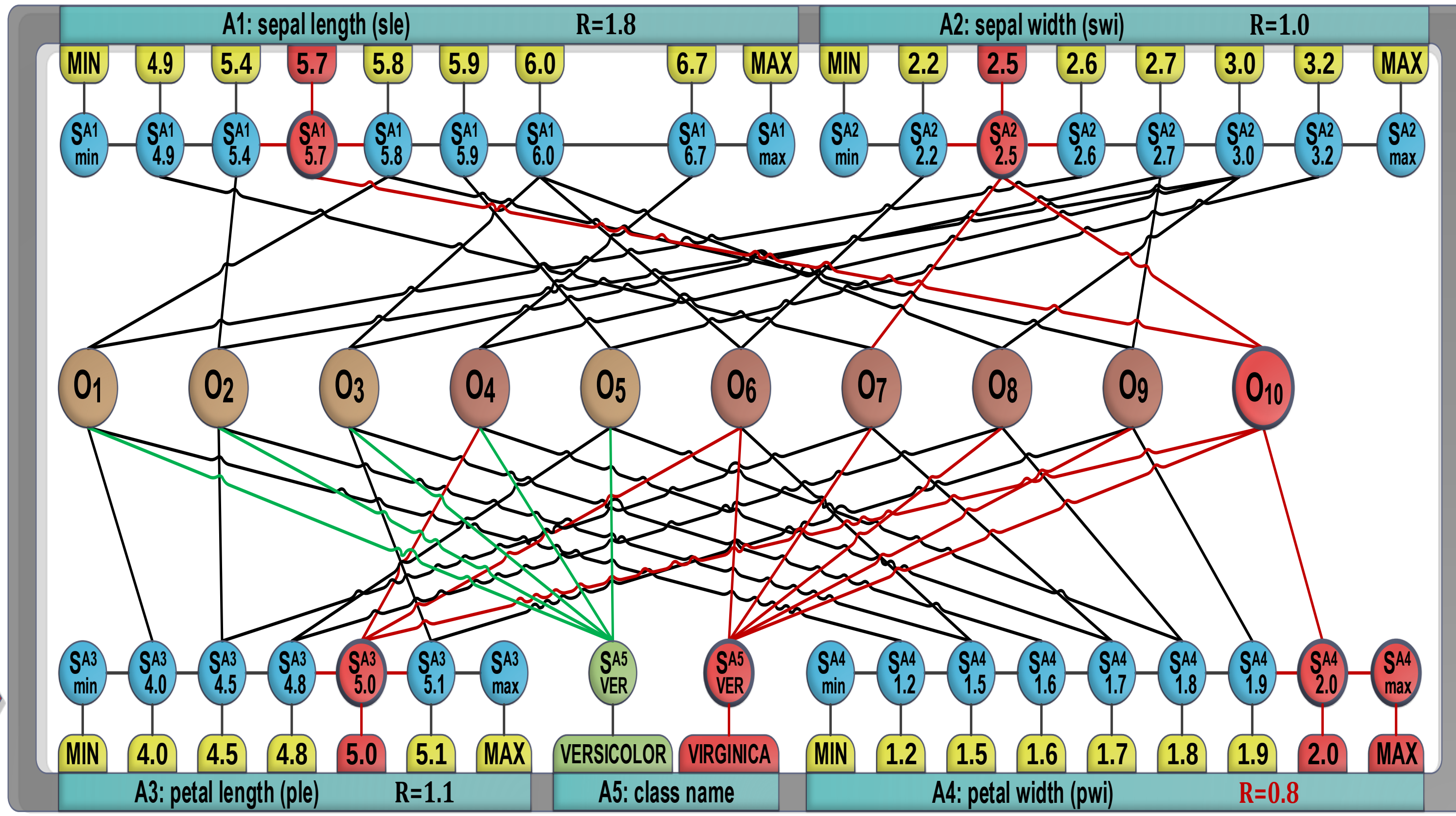
Most aggregations occur where there are natural classes (e.g. VERSICOLOR or VIRGINICA), but in this model, the class can be defined by any aggregation of the same values or any range of them. The AANG graph allows to quickly process any grouping or filtration.

ASSORT FOR A SELECTED SUBSET OF IRIS DATA

SORTED SUBSET OF IRIS PATTERNS

	Attributes				
	sle	swi	ple	pwi	class name
R1	5.8	2.6	4.0	1.2	VERSICOLOR
R2	5.4	3.0	4.5	1.5	VERSICOLOR
R3	6.0	2.7	5.1	1.6	VERSICOLOR
R4	6.7	3.0	5.0	1.7	VERSICOLOR
R5	5.9	3.2	4.8	1.8	VERSICOLOR
R6	6.0	2.2	5.0	1.5	VIRGINICA
R7	4.9	2.5	4.5	1.7	VIRGINICA
R8	6.0	3.0	4.8	1.8	VIRGINICA
R9	5.8	2.7	5.1	1.9	VIRGINICA
R10	5.7	2.5	5.0	2.0	VIRGINICA
R11	6.5	3.2	5.1	2.0	VIRGINICA

STEP 12



STEP 12. CREATION OF ASSOCIATIVE NEURONAL REPRESENTATION OF THE OBJECT (RECORD) R10

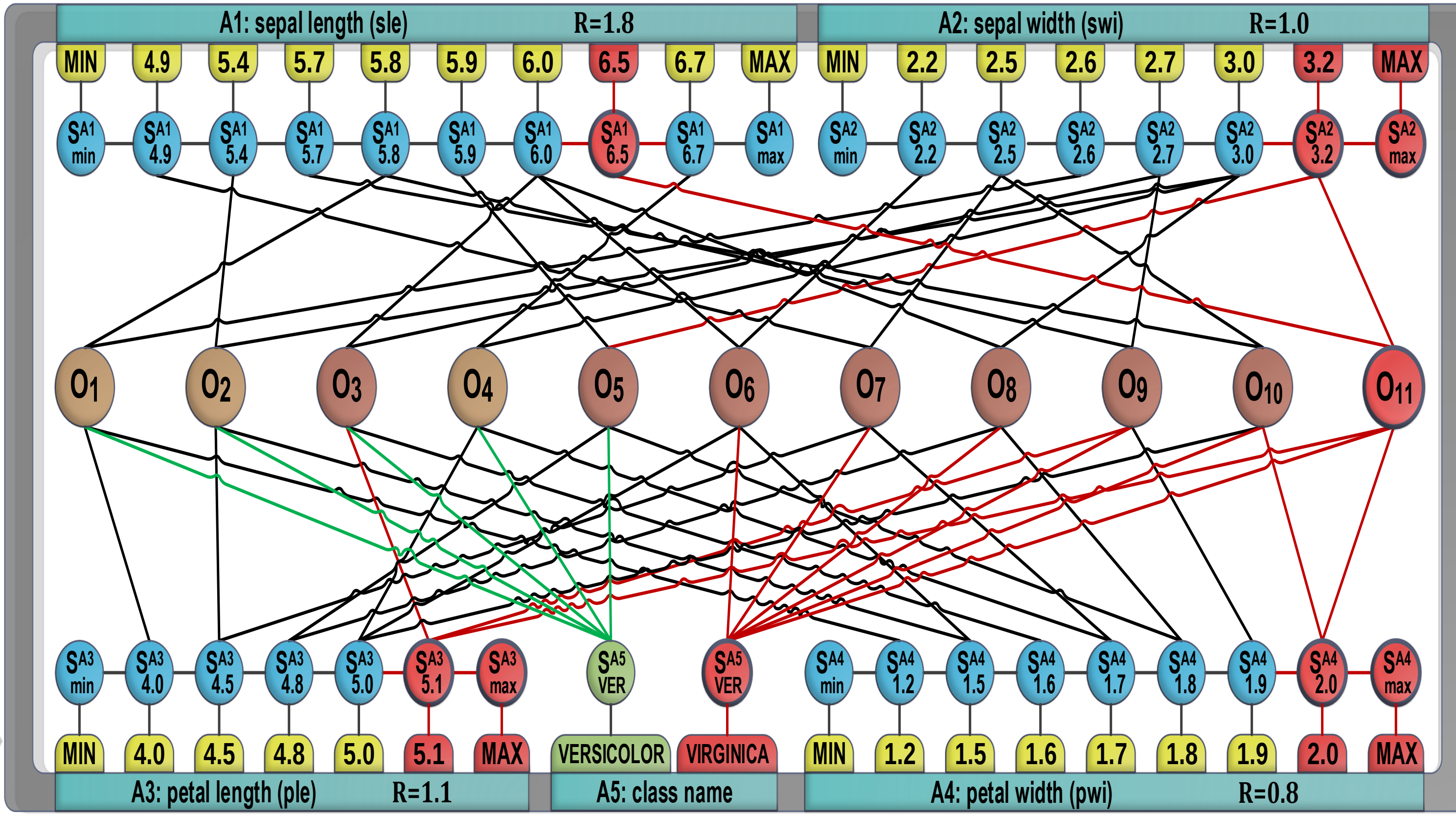
Aggregates can group and associate multiple objects, e.g. 5,0 for the attribute A3 naturally associates the objects R4, R6 and R10. Such associations can be found instantly in constant time.

ASSORT FOR A SELECTED SUBSET OF IRIS DATA

SORTED SUBSET OF IRIS PATTERNS

	Attributes				
	sle	swi	ple	pwi	class name
R1	5.8	2.6	4.0	1.2	VERSICOLOR
R2	5.4	3.0	4.5	1.5	VERSICOLOR
R3	6.0	2.7	5.1	1.6	VERSICOLOR
R4	6.7	3.0	5.0	1.7	VERSICOLOR
R5	5.9	3.2	4.8	1.8	VERSICOLOR
R6	6.0	2.2	5.0	1.5	VIRGINICA
R7	4.9	2.5	4.5	1.7	VIRGINICA
R8	6.0	3.0	4.8	1.8	VIRGINICA
R9	5.8	2.7	5.1	1.9	VIRGINICA
R10	5.7	2.5	5.0	2.0	VIRGINICA
R11	6.5	3.2	5.1	2.0	VIRGINICA

STEP 13



STEP 13. CREATION OF ASSOCIATIVE NEURONAL REPRESENTATION OF THE OBJECT (RECORD) R11

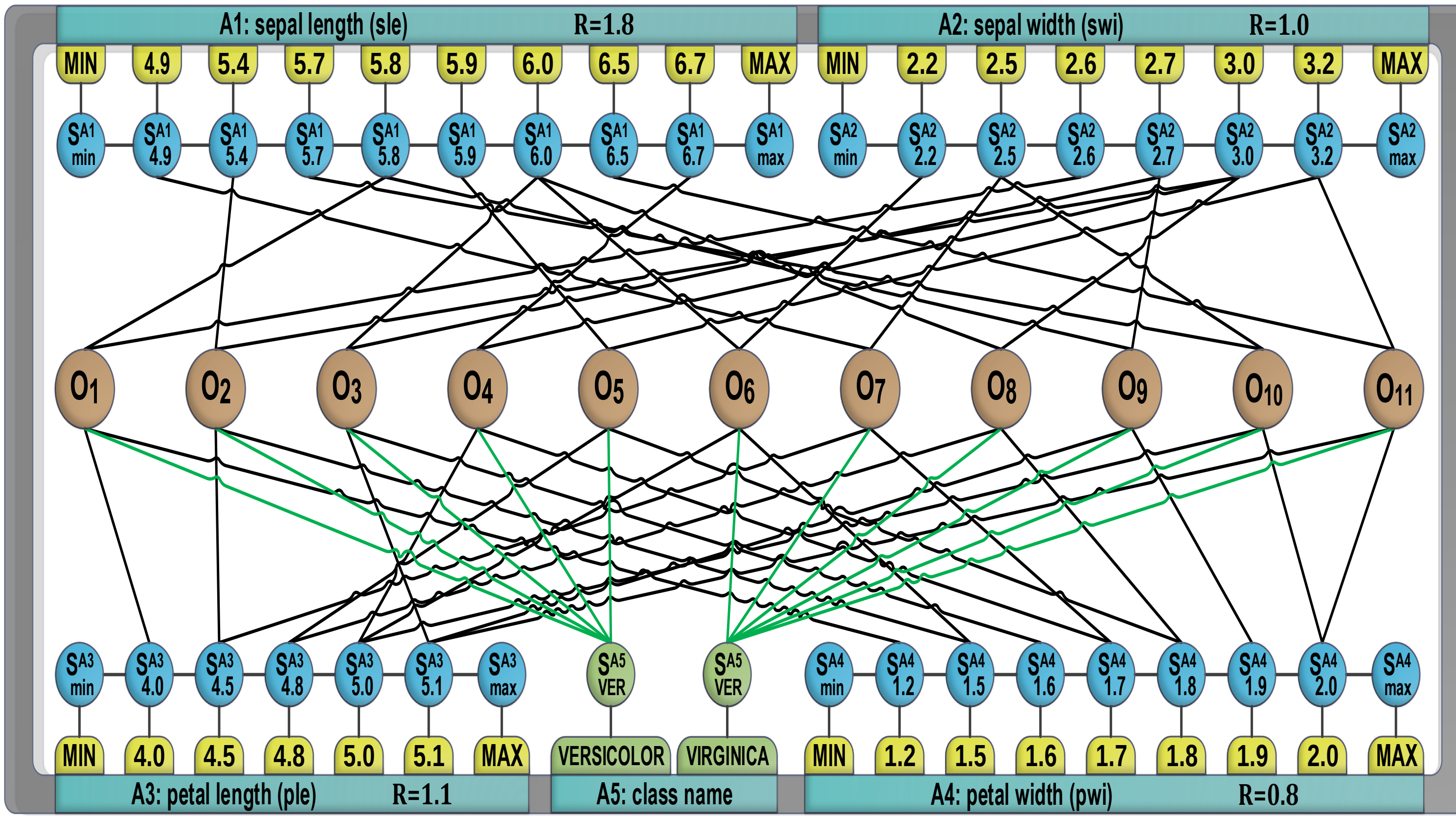
The built-in associative neural graph with the ASSORT algorithm can then serve to quickly and automatically infer the different relationships encoded in this associative structure!

ASSORT FOR A SELECTED SUBSET OF IRIS DATA

SORTED SUBSET OF IRIS PATTERNS

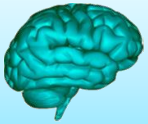
	Attributes				
	sle	swi	ple	pwi	class name
R1	5.8	2.6	4.0	1.2	VERSICOLOR
R2	5.4	3.0	4.5	1.5	VERSICOLOR
R3	6.0	2.7	5.1	1.6	VERSICOLOR
R4	6.7	3.0	5.0	1.7	VERSICOLOR
R5	5.9	3.2	4.8	1.8	VERSICOLOR
R6	6.0	2.2	5.0	1.5	VIRGINICA
R7	4.9	2.5	4.5	1.7	VIRGINICA
R8	6.0	3.0	4.8	1.8	VIRGINICA
R9	5.8	2.7	5.1	1.9	VIRGINICA
R10	5.7	2.5	5.0	2.0	VIRGINICA
R11	6.5	3.2	5.1	2.0	VIRGINICA

STEP 14



THE AANG HAS BEEN CREATED THANKS TO THE ASSORT ALGORITHM

The final associative neuronal graph structure represents all sorted objects simultaneously for all attributes! Hence, we do not need to sort anything again…



SAMPLE INFERENCE USING ASSOCIATIONS

WE HAVE TO STIMULATE SENSORS OR NEURONS IN ORDER TO USE AANG

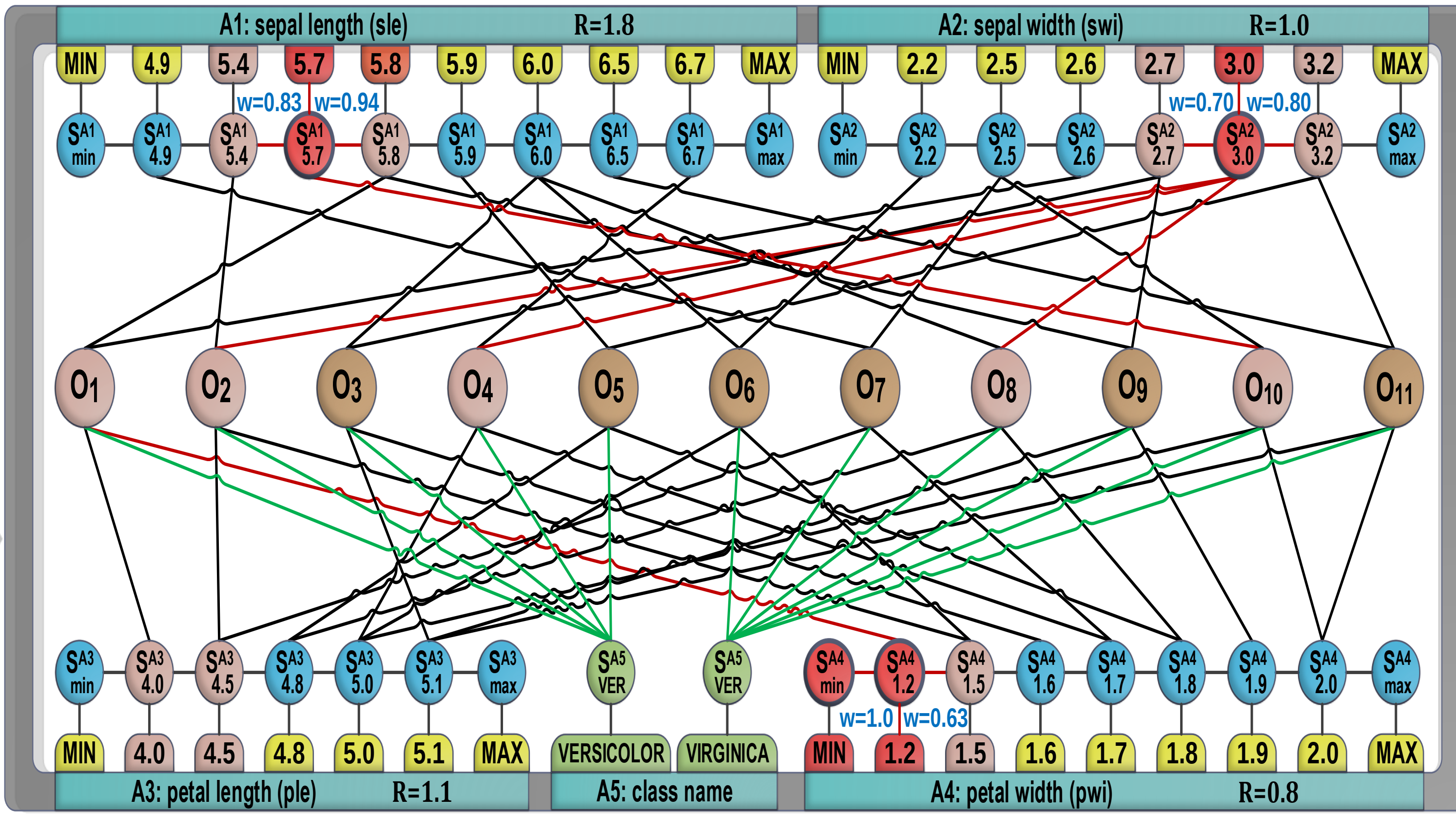
We start to present a new pattern (object) of unknown class to the AANG, stimulating the appropriate sensory fields for a a specified period of time:

Stimularion time
t = 1.0

Externally stimulate input sensory fields to get an answer about input data.

	Attributes				
	sle	swi	ple	pwi	class name
R12	5.7	3.0	4.2	1.2	?

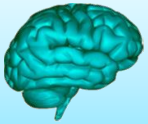
Nothing happend but stimulate further on until the neurons will draw conclusions.



THERE IS NO ACTIVE NEURON REACTION?

So far, the neurons have not yet reacted? Why? Maybe we were too short stimulating the network?

Remember that in artificial associative systems time is a computational factor, so let’s keep on stimulating.



SAMPLE INFERENCE USING ASSOCIATIONS

WE HAVE TO STIMULATE THE NETWORK FURTHER

After a slightly longer period of time, neurons were activated because their charging process was slower.

Stimularion time

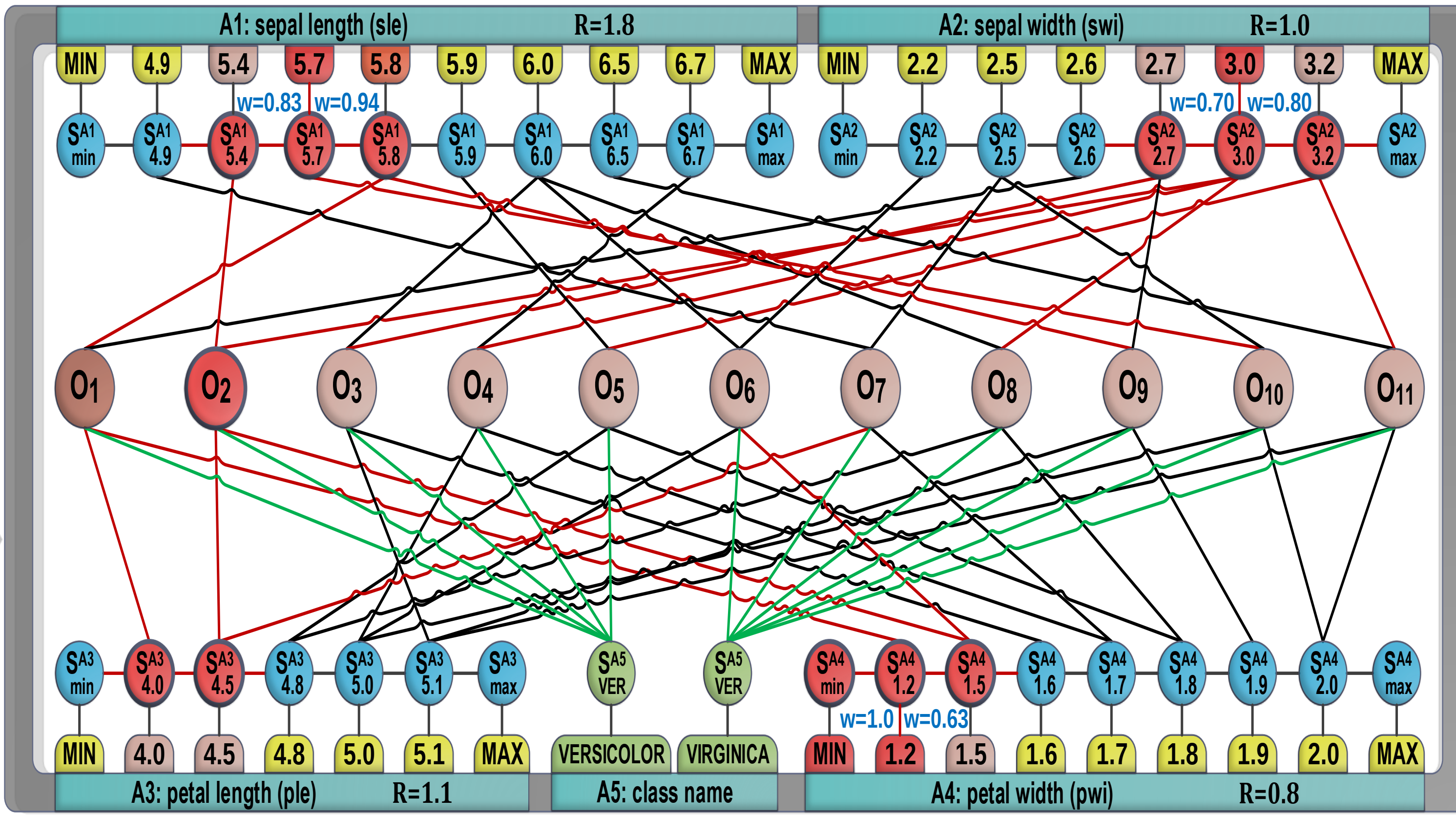
t = 1.25

Further stimulate input sensory fields until the neurons will conclude something about input data.

	Attributes				
	sle	swi	ple	pwi	class name
R12	5.7	3.0	4.2	1.2	?

Neuron O_2 has been activated so the input pattern is the most similar to it:

R2	5.4	3.0	4.5	1.5	VERSICOLOR



THIS TIME THE NEURON O_2 WAS ACTIVATED

The O_2 neuron was activated the fastest, indicating the most similar object to the pattern presented on the input of this network, but we still do not have an answer for its class.

SAMPLE INFERENCE USING ASSOCIATIONS

ACTIVATED NEURONS AUTOMATICALLY STIMULATE THE OTHER CONNECTED NEURONS

The O_2 neuron is connected to the neuron representing the VERSICOLOR class. The weight of this connection is equal to the activation threshold of that neuron, so it produces the correct response.

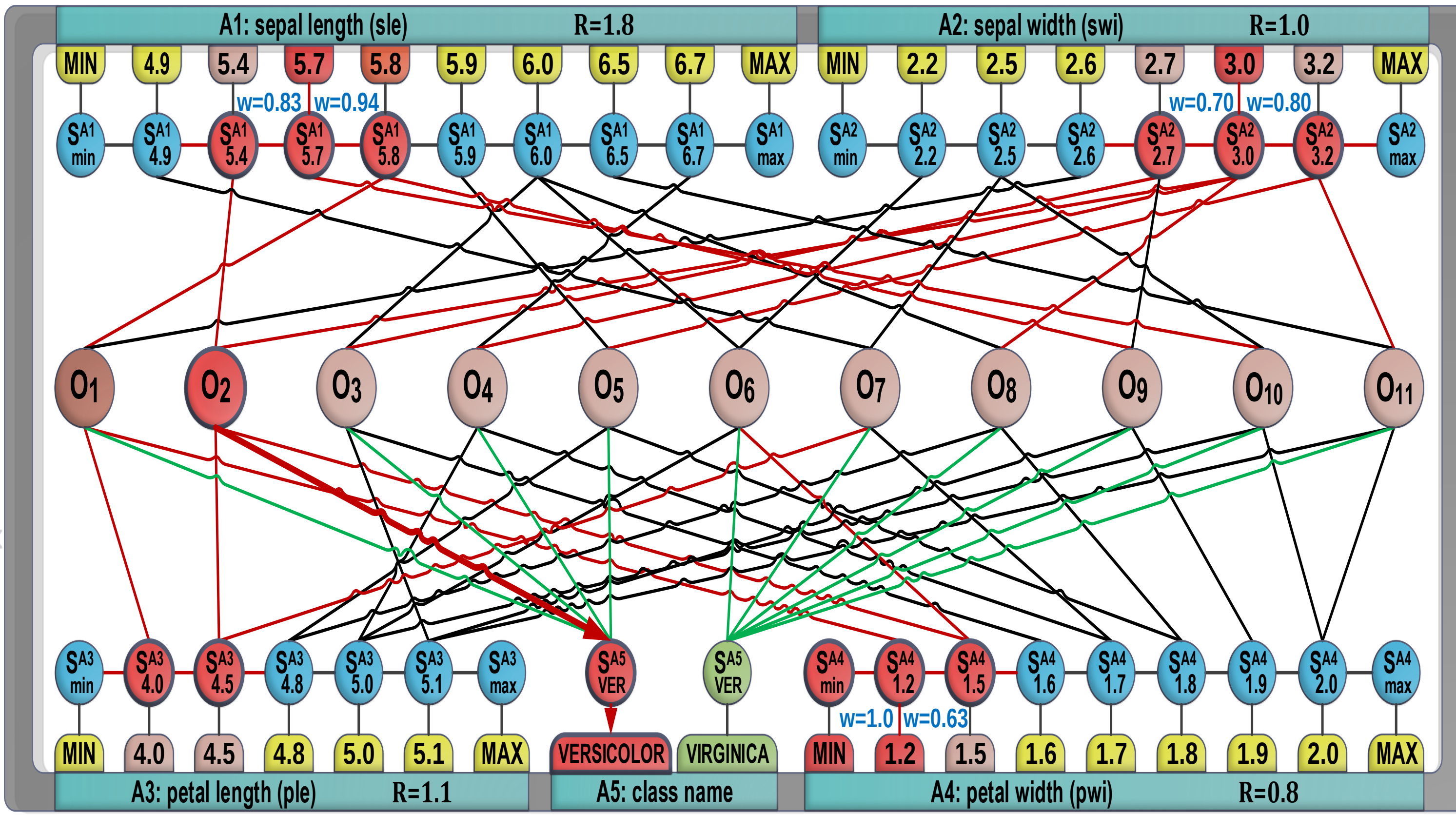
Stimularion time

t = 1.30

Activated neuron O_2 stimulates the connected neuron representing the class Versicolor drawing the correct conclusion about the class of pattern R12.

Attributes					
	sle	swi	ple	pwi	class name
R12	5.7	3.0	4.2	1.2	VERSICOLOR

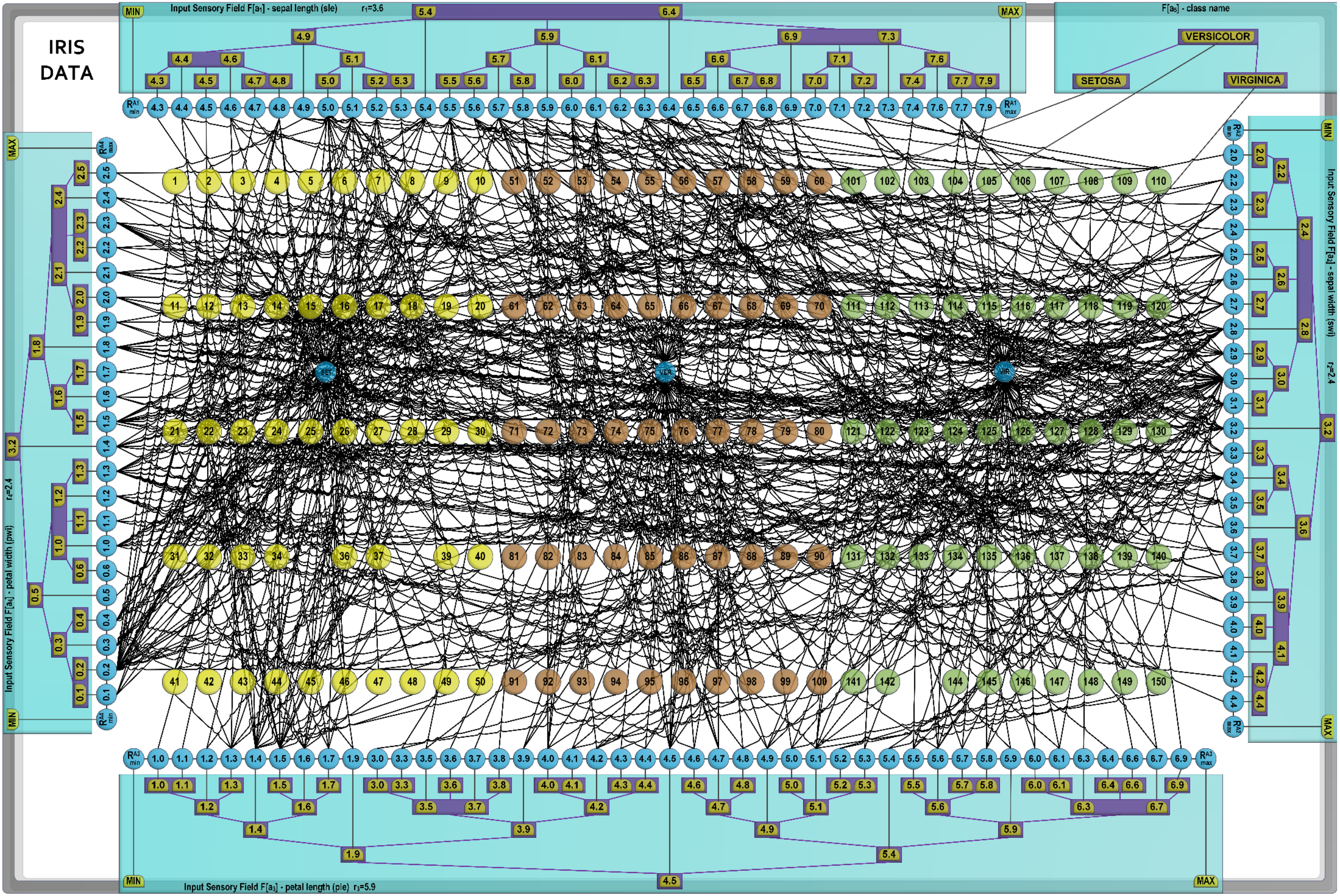
Pattern R12 has been correctly classified as VERSICOLOR.



ACTIVATION OF NEURONS IS THE RESPONSE OF THE AANG NETWORK

In this way, without the use of any inference, search, or comparison algorithm, we have received an answer coming from associations represented by the neurons of this neural network.

THE AANG CREATED FOR ALL IRIS PATTERNS FROM ML REPOSITORY USING AVB-TREES TO REPRESENT ALL FEATURES



WHAT CAN BE ACHIEVED WITH ASSOCIATIONS

Active associative neural graphs AANG enable us to:

- ✓ **Actively interact** data with each other through the use of neurons and a graph structure that integrates data, groups, and sequences that are properly consolidated.
- ✓ **Automatic recall relationships** between data in the table.
- ✓ **Sort objects** using neurons with context-sensitive plasticity triggers for their actions, i.e.:
  - creating new connections,
  - breaking old connections,
  - update of synaptic weights.
- ✓ **Perform local implementation of all** calculations performed during ASSORT sorting without the involvement of external algorithms to iterate on neurons or consolidate their results. The neurons unintentionally do the sorting, which results from their characteristic plasticity operations, known from the biological nervous system.
- ✓ **Sort objects against all attributes simultaneously and simultaneously with linear computational complexity O (n)** and without the need to create additional indices for relational databases to speed up the operations performed on them.
- ✓ **Add new objects in parallel for all attributes** while maintaining order with constant computational complexity O (1) for each of them.
- ✓ **Automatic inference** by stimulating associated data or objects.
- ✓ Automatic construction of **intelligently concluding cognitive associative systems** after the given constraints, conditions, or for the given value or their ranges.

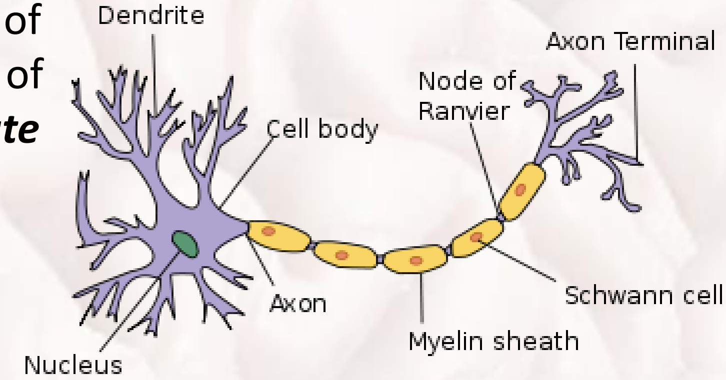
SPIKING NEURONS

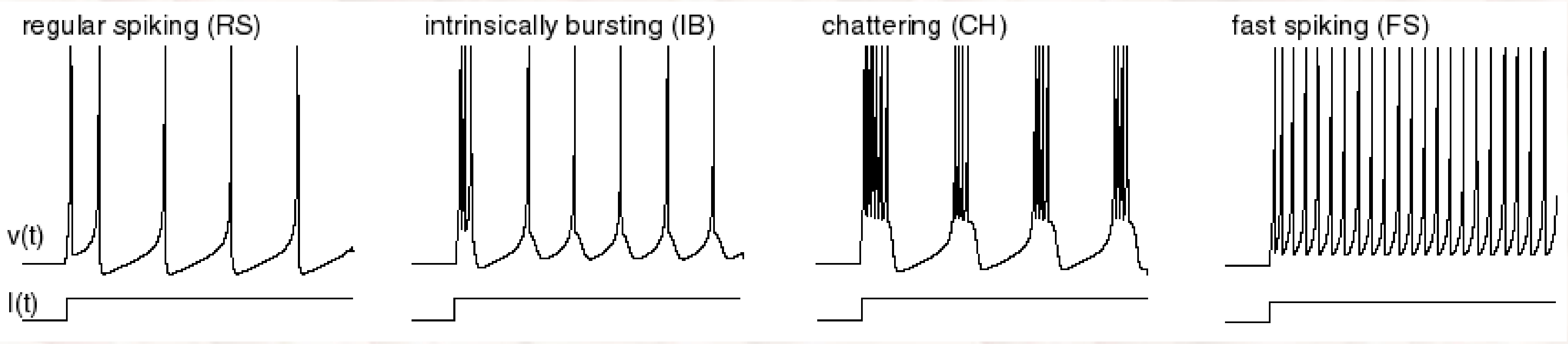
Spiking neurons are models of biological neurons that fall into the third generation of neural network models, increasing the realism in neural simulations.

These models incorporate the concept of **time** into the operating model of the previously used artificial McCulloch-Pitt's model.

Artificial neurons do not fire even if they implement hard-switch activation functions.

The **fundamental problem** is to propose the model that explains how information is encoded and decoded by a series of trains of pulses, i.e. action potentials. Thus, the **fundamental question** of neuroscience is to determine ***if neurons communicate by a rate or temporal code***? Temporal coding suggests that a single spiking neuron can replace hundreds of hidden units on a sigmoidal neural network. Is that true?

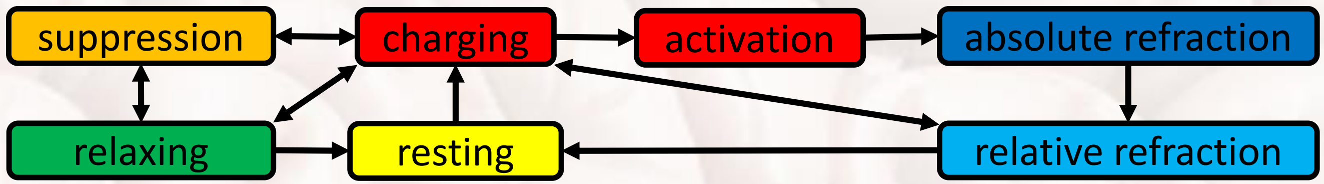


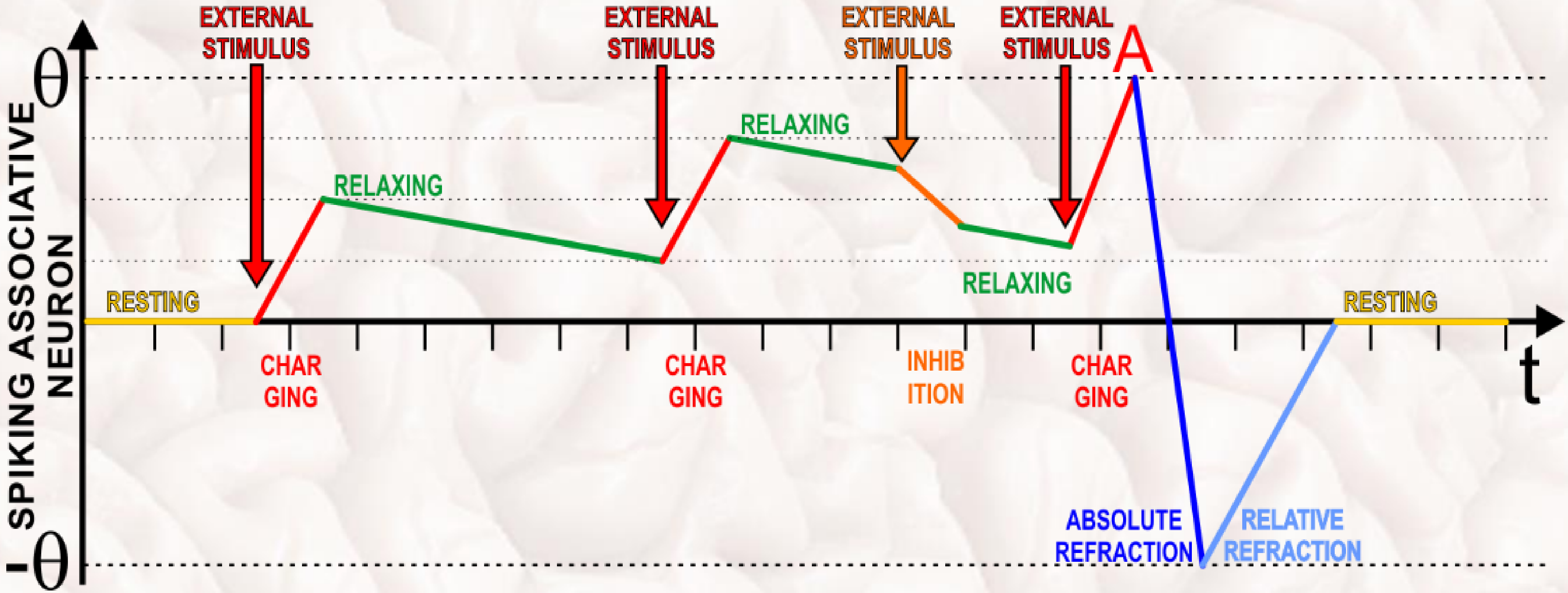


SPIKING ASSOCIATIVE NEURONS

The spiking associative neuron is a functional model of **biological neurons**. This is a simplistic model in comparison to **spiking neurons,** but it does not emphasize or try to model biological platform truly. It focuses on efficient modeling of time-dependent **functional aspects** that are responsible for the information processes that take place in biological neurons and their networks. This model also has some built-in routines that enable it automatically connect to other neurons according to some **plasticity rules**.

This model is **reactive** to external stimulations and internal processes:





HOW DO NEURONS CONCLUDE?

Active Associative Neural Graphs (AANGs) are able to combine **inferences based on similarity and sequences** based on the neuronal graph and its parameters. For now, let's focus on the **conclusions based on similarity**. Let's take a little clipping of this graph, assuming that the chosen neuron representing the object O_j can be externally stimulated for some time in order to find out other neurons of the same kind in this graph, which are most similar to it.

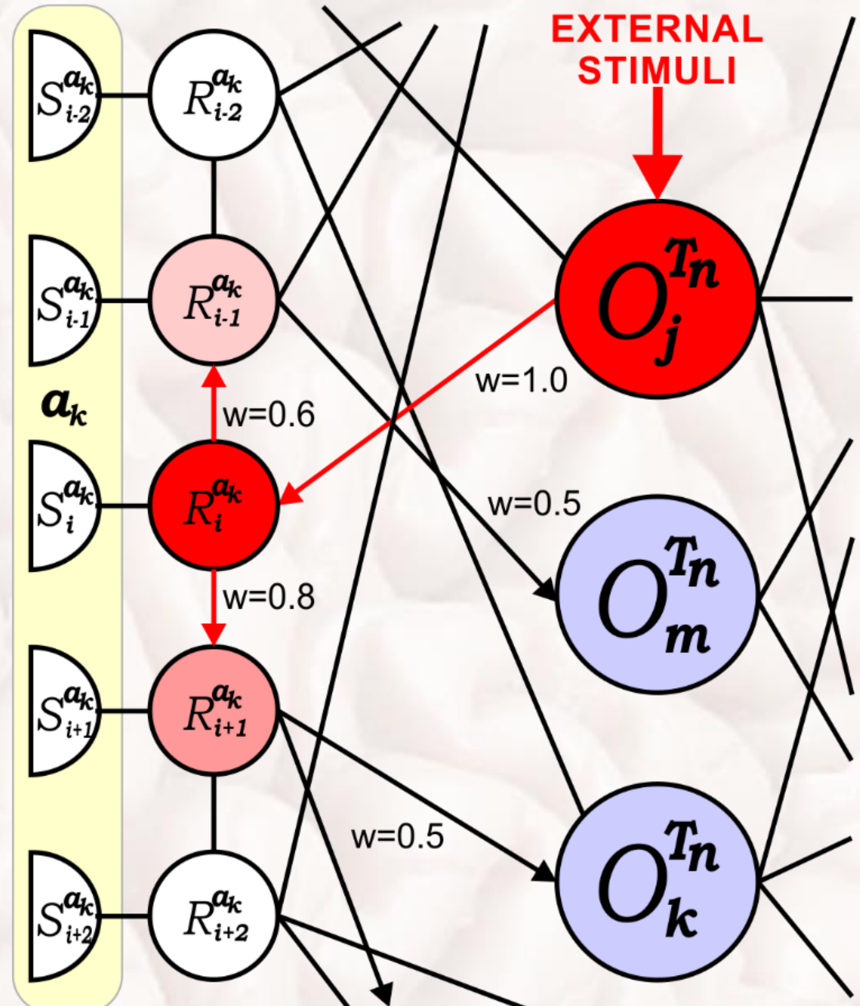
The degree of similarity will be determined by the time after which such neurons will be activated.

For ease of analysis, let us consider only one object neuron O_k, which may potentially be indirectly influenced by another object neuron O_j that is **externally stimulated for some period of time.**

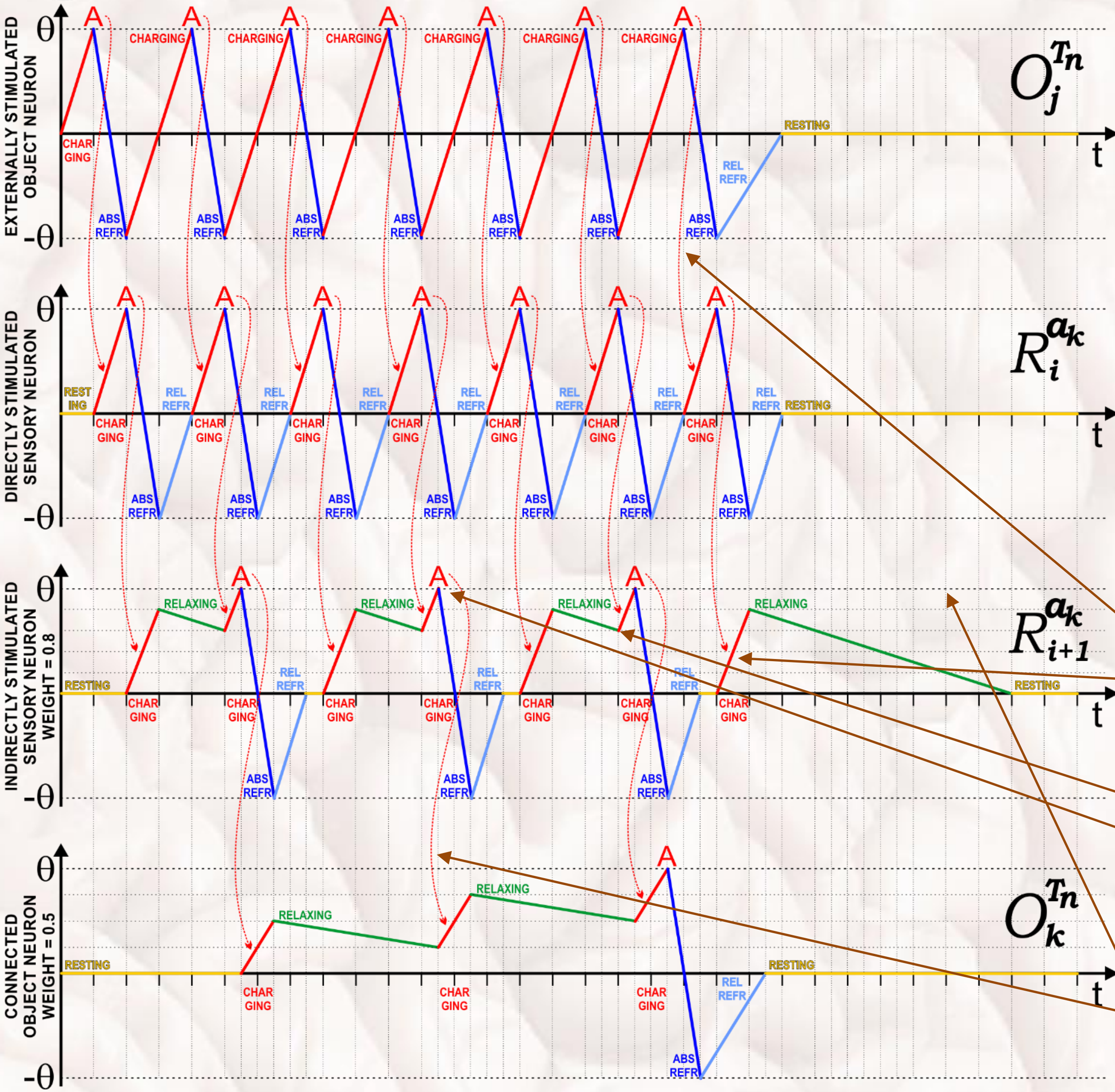
This period can be determined by the user in order to perform analysis of the similarity between these two objects represented by these neurons.

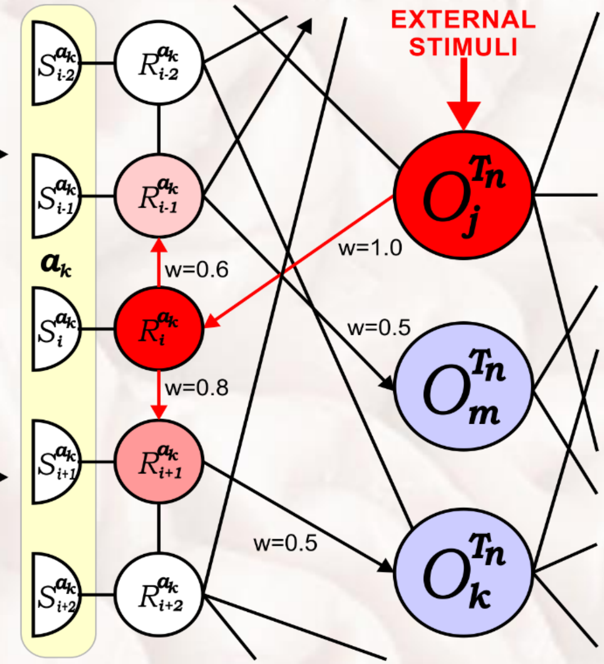
The interaction between neurons is possible thanks to **neuronal affinity associations** fixed in the associative neuronal graph.

These **associations are represented by weighted connections** between sensory neurons R_i and R_{i+1}.



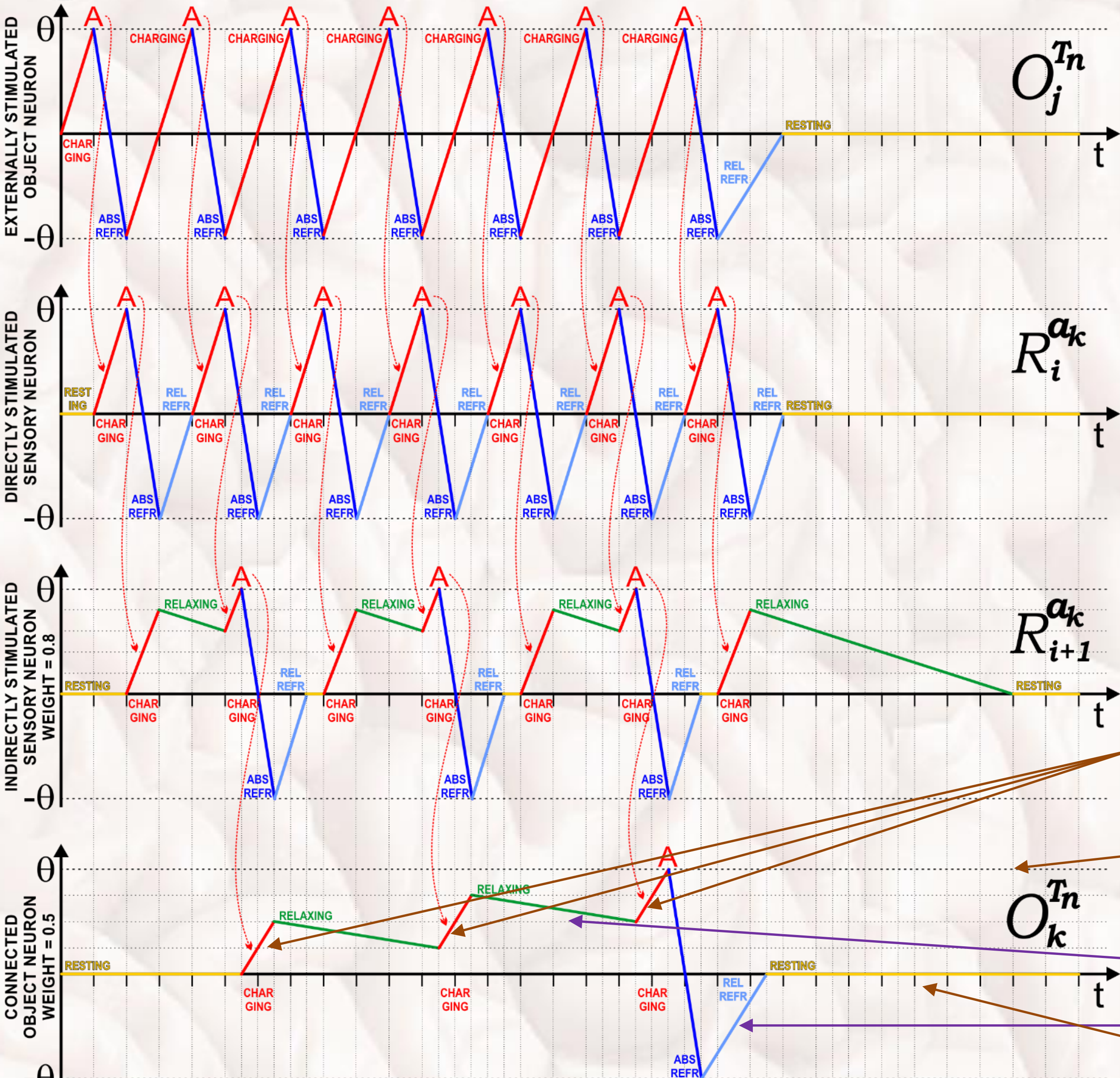
HOW DO NEURONS CONCLUDE?

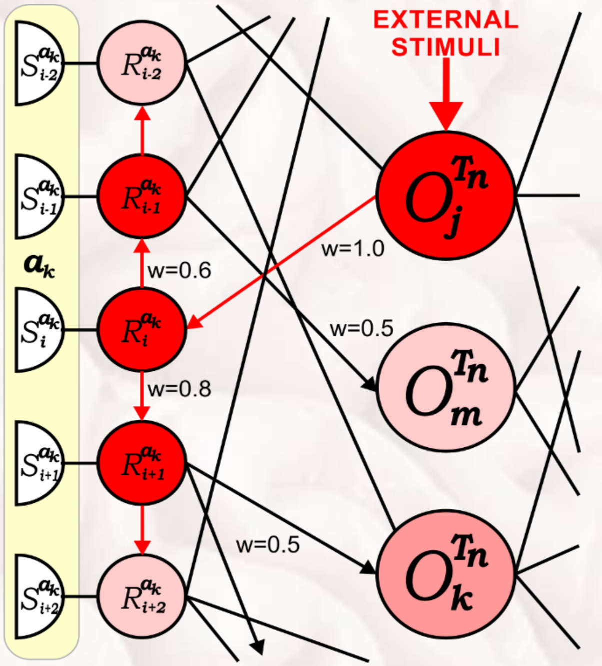




Each activation of the neuron O_j **stimulates and activates** the neuron R_i which **stimulates** the neighboring sensory neurons R_{i+1} and R_{i-1} with the force equal to the weights of these connections, i.e. 0.8 and 0.6, appropriately. Therefore, it is needed to **stimulate these neurons twice**, so that, with regards to **relaxation**, they achieve a total stimulus equal to their **activation thresholds** $\theta = 1$. This will allow them for activation and then to start **stimulation** of the connected neurons, e.g. the neuron O_k.

HOW DO NEURONS CONCLUDE?

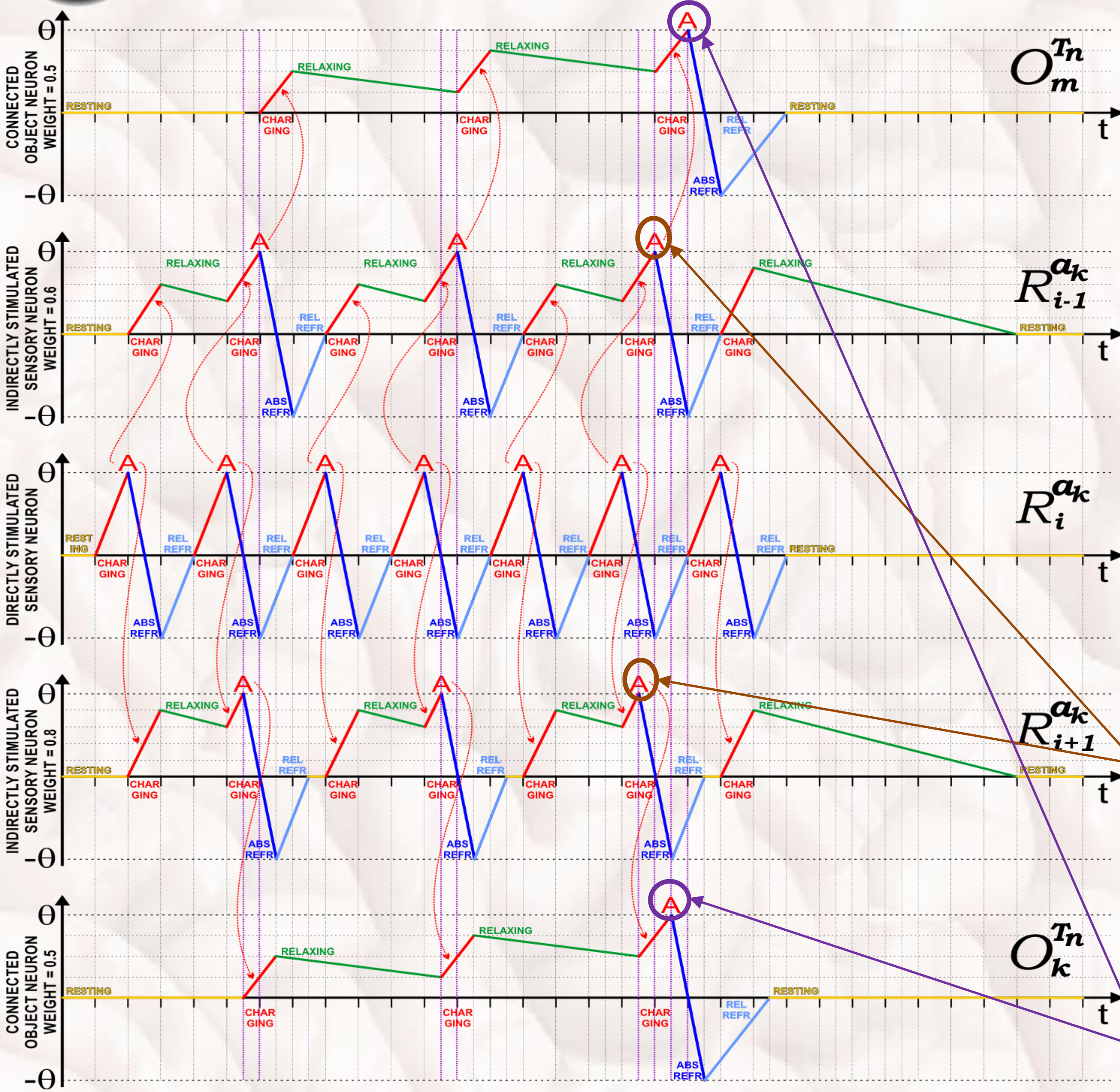


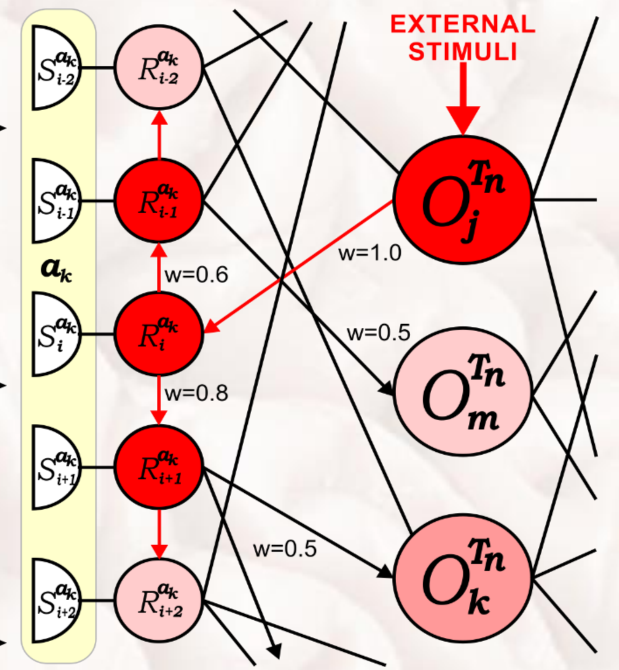


As we can notice, the neuron O_k needs to be **stimulated triple times** through the connection coming from the neuron R_{i+1} and weighted with 0.5 to reach the **activation threshold** θ = 1.0.

When a neuron is not externally stimulated, the **relaxation** and **refraction** processes try to restore the **resting state** in it.

HOW DO NEURONS CONCLUDE?





The sensory neurons R_{i+1} and R_{i-1} are stimulated with different strength according to the weights (0.8 and 0.6) of connections coming from the neuron R_i. It induces different **excitation levels** in them and **different activation moments**. The neuron R_{i+1} achieves this threshold **earlier** than the neuron R_{i-1}, so the neuron R_{i+1} starts **earlier** to stimulate the neuron O_k than the neuron R_{i-1} starts to influence the neuron O_m. Thus, the neuron O_k will be activated **earlier** than the neuron O_m. It implies **greater similarity** of the object represented **by the neuron O_k** than **by the neuron O_m**. This is consistent with **intuition** of the real **similarity**.

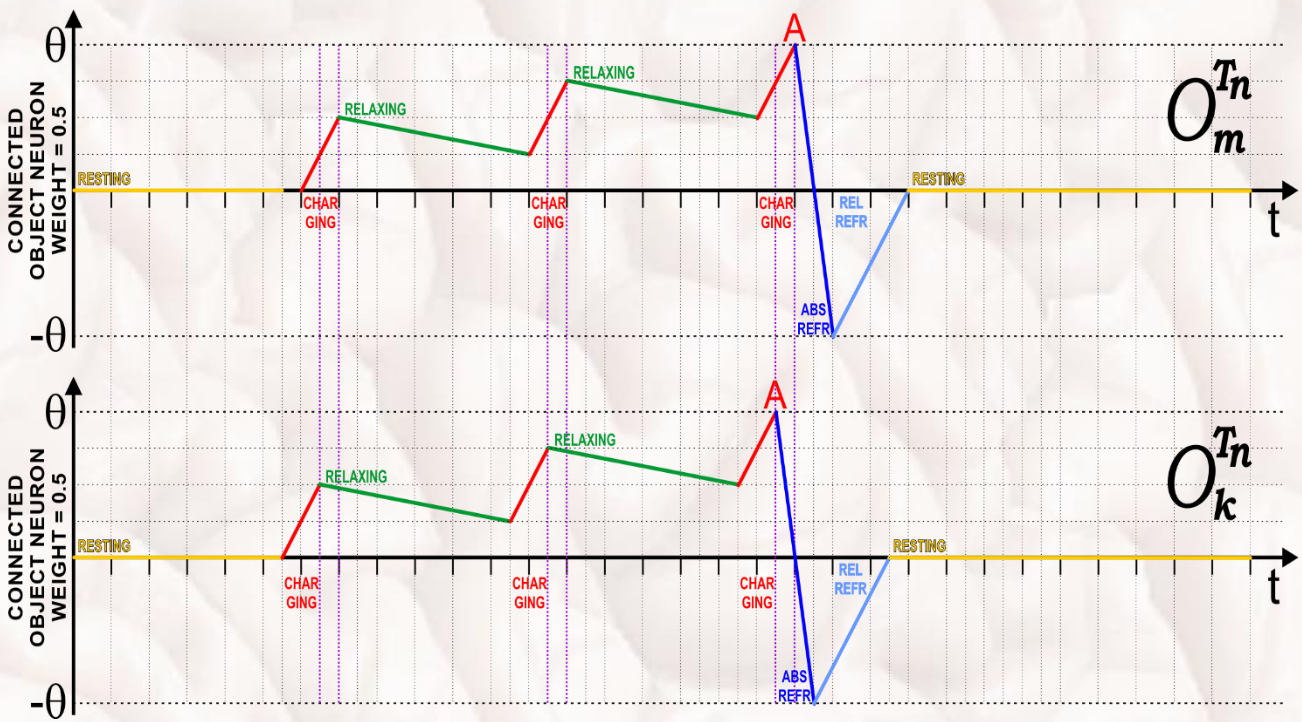
HOW DO NEURONS CONCLUDE?

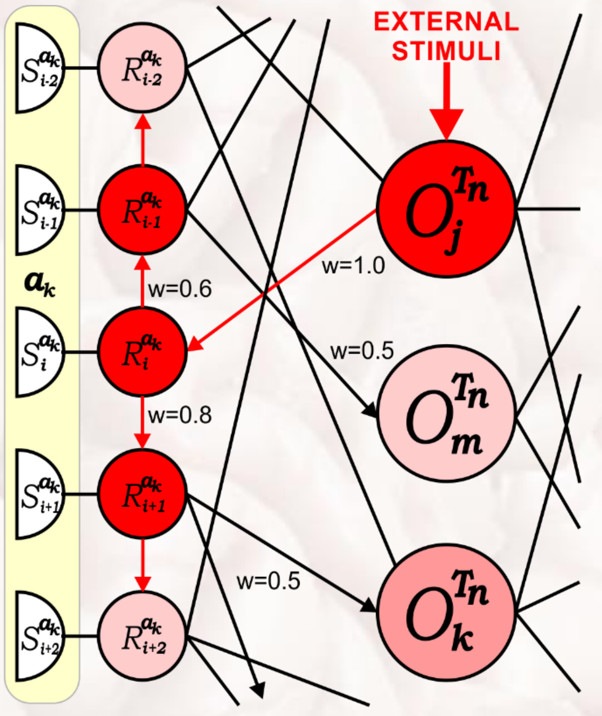
The small shift in activation of the neurons O_k and O_m may seem to be insignificant or negligible, but this phenomenon is crucial for the working way of biological neural networks as well as of the introduced **associative neural graphs (e.g. AANG, DASNG, ANAKG or AAS)**.

The difference in activation time of these neurons representing different objects informs us of **weaker and stronger associations** with these objects, i.e. **less or greater similarity** of them.

In general, these time differences determine **cognitive processes in the human brain** like mental, motor, or sensory responses… influencing the behavior of the whole network.

In this way, **associative neurons** automatically **conclude**, revealing their various relationships with other objects and data represented by other directly or indirectly connected neurons.

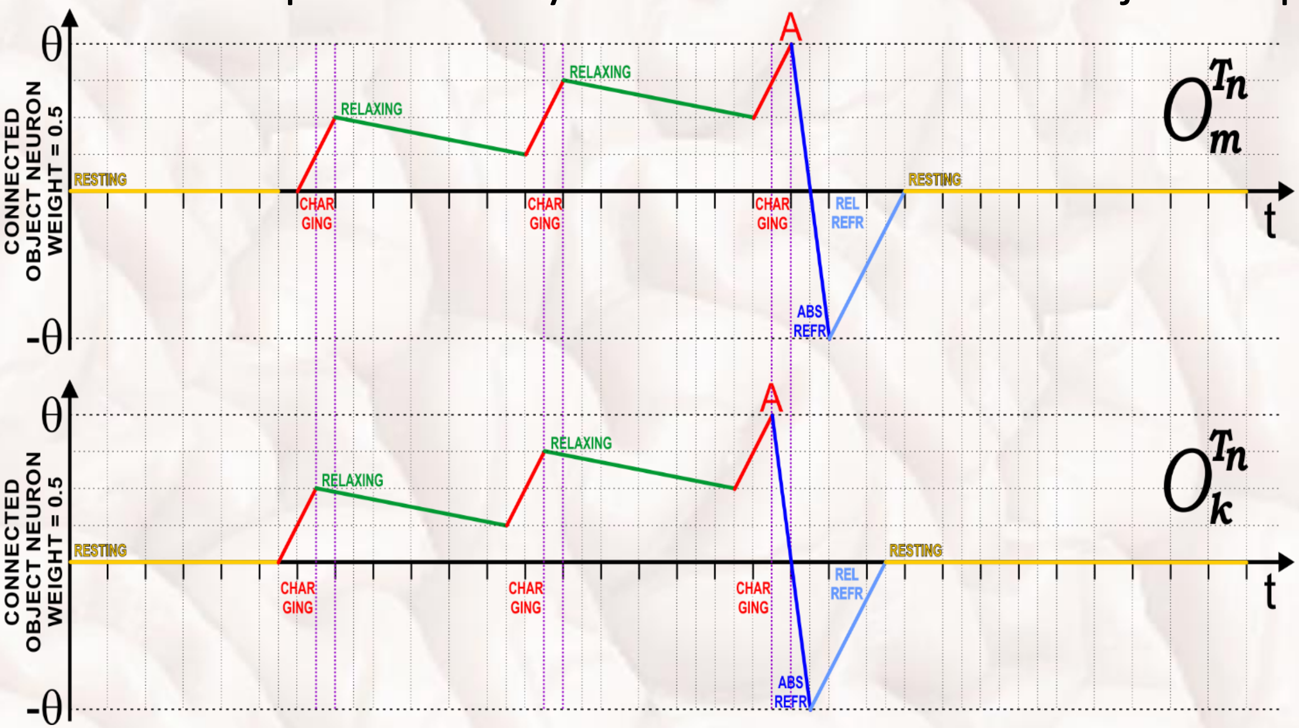


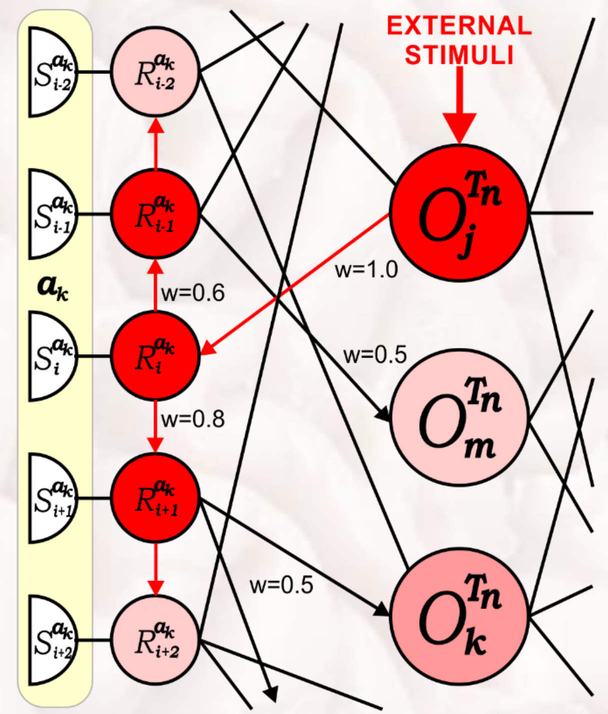


HOW DO NEURONS CONCLUDE?

Looking for answers about similarities, groups of similar patterns to a given pattern, any subset of patterns, any given set of features, or any combination of their ranges, it is enough to stimulate the appropriate neurons or sensors, and just wait for activations of neurons, which **activation moments** determine the network answers.

The chronology of neuronal activity automatically points out similar objects or their groups (**in clusterization problems**), missing features or component objects (**in cognition problems**), or indicates which classes they belong to (**in classification problems**). Neurons can therefore automatically **explore the knowledge** from the relations represented by connections and from objects represented by other neurons.



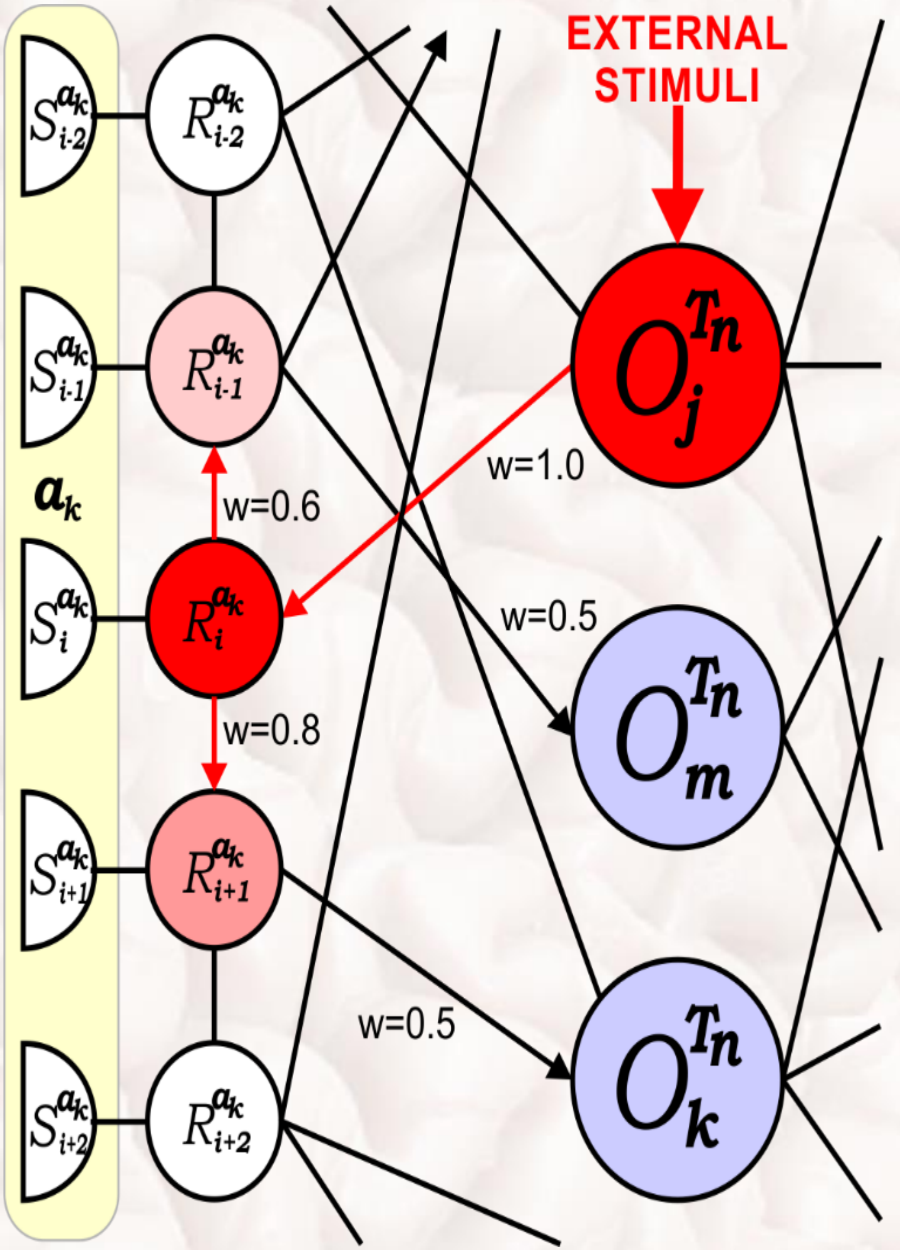


GENERALIZATION OF CONSIDERATIONS

This very simple (trivial) example does not introduce the full scope of the inference, applicability or possible combinations of neuronal activities in the whole network (graph)!

In the whole **associative neural graph**, such an externally stimulated and **activated object neuron** will simultaneously inference with several sensory neurons which will subsequently stimulate various object neurons. **The activated object neuron** can also have direct connections to other object neurons (representing **associative succession** or **defining**) and thereby stimulate them. **It is very tough to precisely, clearly, and sequentially describe all these parallelly running processes, it must be seen!**

All these processes run parallelly in the brain, so the **inference is very fast** in comparison to the **classic inference methods of computational intelligence or knowledge engineering**, where you must repeatedly search through tables, their elements, transaction, join data together etc.



DEEP ASSOCIATIVE NEURAL GRAPHS - DASNG

Using the knowledge of how **associative neurons** behave when representing data and objects, we can try to transform any **relational database** that represents horizontally related objects to the form of a **deep associative neural graph DASNG**.

TABLE C: Subjects

A6:S Subjects	Primary KeyC
business	1
chemistry	2
drama	3
geography	4
history	5
mathematics	6
physics	7

TABLE D: Likes

Foreign KeyC	Foreign KeyA
1	1
1	2
3	1
1	4
2	3
6	12
2	6
4	7
7	5
4	12
6	9
3	12
5	7
7	16
1	12
6	14

TABLE A: Pupils

Primary KeyA	A1:S Last Name	A2:S First Name	Foreign KeyE	Foreign KeyB
1	Brown	Amy	1	1
2	Cruise	Jack	1	2
3	Brown	Jack	2	2
4	Ford	Kate	3	1
5	Hanks	Luke	2	5
6	Jolie	Mary	3	3
7	Ford	Jack	2	2
8	Brown	Kate	1	4
9	Jolie	Rose	2	6
10	Lopez	Tom	1	9
11	Pitt	Tom	3	5
12	Brown	Kate	1	10
13	Ford	Luke	2	7
14	Smith	Luke	2	10
15	Jolie	Mary	2	8
16	Trump	Tom	3	10

TABLE B: Live with/in

Primary KeyB	A3:N No of Siblings	A4:N Floor	A5:S Accommodation
1	0	0	house
2	1	3	appartament
3	0	0	house
4	0	3	appartament
5	2	7	appartament
6	3	0	house
7	2	3	dormitory
8	0	1	house
9	4	2	appartament
10	4	9	dormitory

TABLE E: Profile

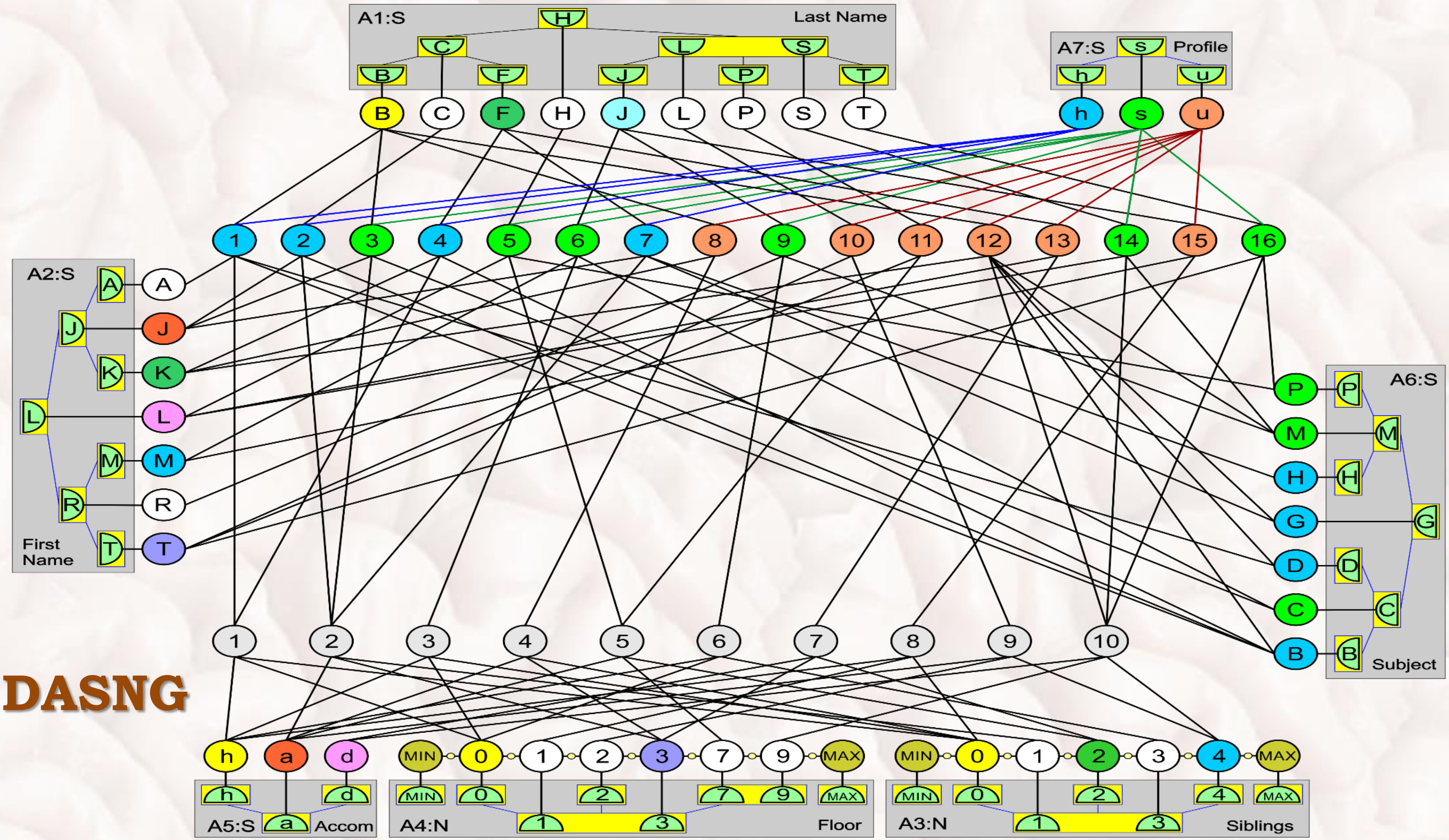
Primary KeyE	A7:S Profile
1	humanistic
2	science
3	unselected

The presented relational database consists of 4 tables containing data and one link table for representing many-to-many relationships (N:M type). Now, we can ask questions:
Which pupils have similar interests? OR ***Which pupils do live in apartments?***

DEEP ASSOCIATIVE NEURAL GRAPHS - DASNG

Deep associative neural graph DASNG represents all **horizontal and vertical relations** which associate data and objects and can be automatically retrieved from the database.

They also **aggregate** the representation of all duplicates, occurring in a relational database:



DASNG

WEIGHTS COMPUTATION FOR DASNG

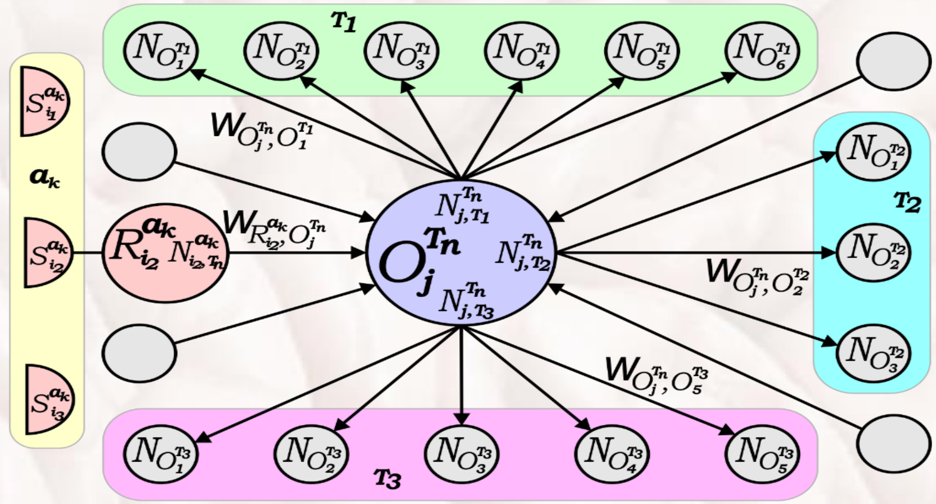
The **weights** of connections between neurons are computed after **very simple formulas**, so **they do not need to be stored or updated, but computed before using them**:

Orderable sensory neurons are connected, the connections are weighed, and the **weights** are:

$$w_{R_{v_i}^{a_k},R_{v_j}^{a_k}} = 1 - \frac{\left|v_i^{a_k} - v_j^{a_k}\right|}{r^{a_k}}$$

The connections between the sensory and object neurons are **weighted** in the following way:

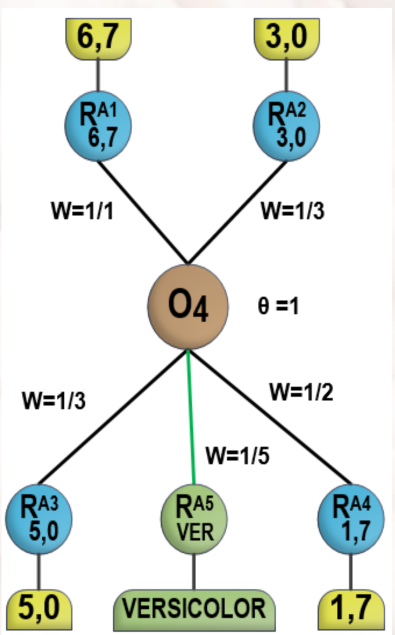
$$w_{R_{v_i}^{a_k},O_j^{T_n}} = \frac{1}{\left\|v_i^{a_k}\right\|} \qquad w_{O_j^{T_n},R_{v_i}^{a_k}} = \theta_{R_{v_i}^{a_k}} = 1$$



The weights of synaptic connections between various object neurons are computed on the basis of the number of objects represented by the object neurons of the considered layer of the DASNG, which represents a single database table. If the given **object neuron of the considered layer** is connected to **M object neurons of another layer**, then the weight is computed in the following way:

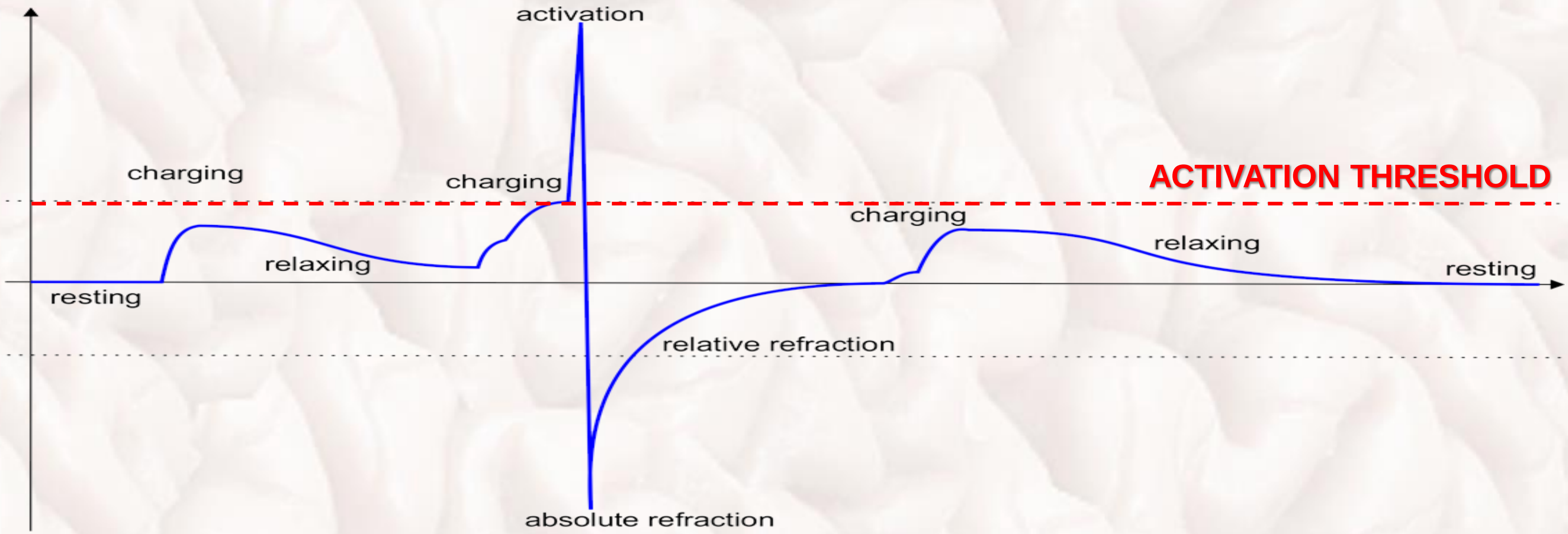
$$w_{O_j^{T_n},O_k^{T_m}} = \frac{1}{N_{j,T_m}^{T_n}} \cong \frac{1}{M} \qquad w_{O_k^{T_m},O_j^{T_n}} = \frac{1}{N_{k,T_n}^{T_m}} \cong \frac{1}{N}$$

where $N_{k,T_n}^{T_m} = N = 1$ for the relations one-to-many (1:M) and the relations many-to-many (N:M). The equation is precise when there are no duplicates of the whole records in the database. We need to create separate lists of connections in each neuron to represent connections to neurons of various layers in order to easily compute the number of objects $N_{j,T_m}^{T_n}$ or the number of connections M.



ACTIVATION THRESHOLDS OF NEURONS

Activation thresholds of neurons play a key role in McCulloch-Pitts' neurons of the first generation, spiking neurons of the third generation and real (biological) neurons while they determine neuronal activity and to which combinations neurons react.



Moreover, activations of the neuron determine which combinations of input stimuli are represented by this neuron.

Therefore, it is very essential to be able to track the state coming from neuronal stimulations with regards to its activation threshold. During simulation, we have no possibility to check all neuronal states constantly, so we need to foresee and compute the predictable time when the neuron achieves its activation threshold.

ACTIVATION THRESHOLDS OF NEURONS

Activation thresholds of sensory neurons are always equal one in this model:

$$\theta_{R_{v_i}^{a_k}} = 1$$

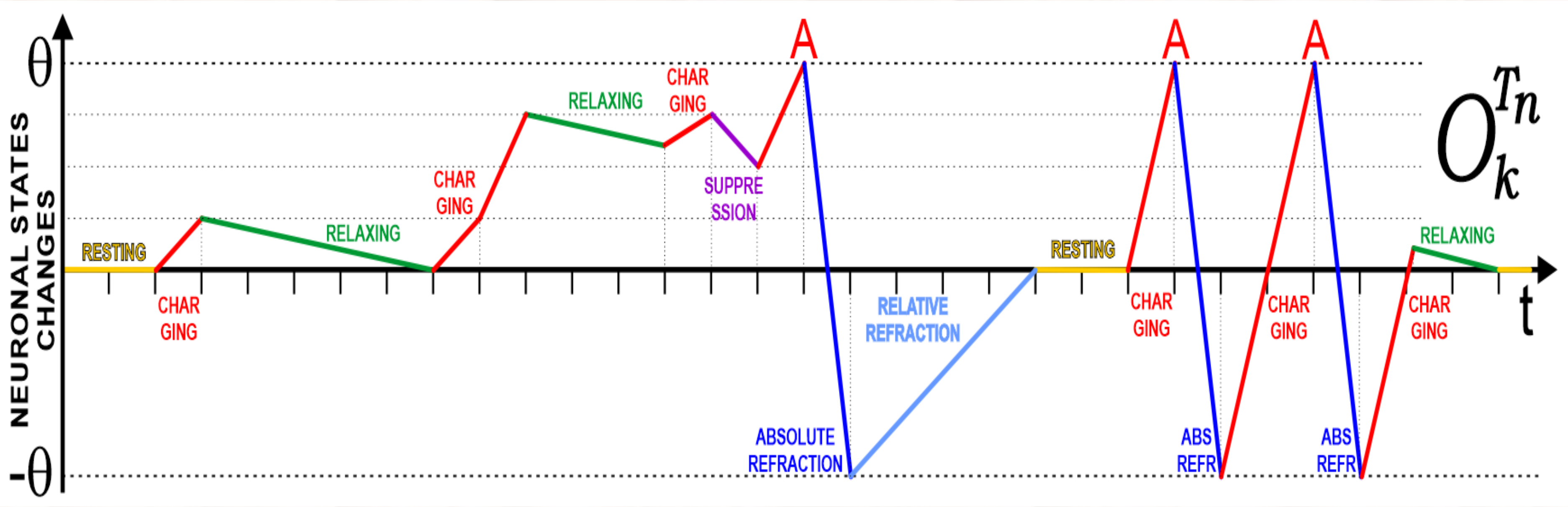
Activation thresholds of object neurons are computed according to the following formula:

$$\theta_{O_j^{T_n}} = \begin{cases} 1 & if \quad \sum_{R_{v_i}^{a_k}} w_{R_{v_i}^{a_k}, O_j^{T_n}} \geq 1 \\ \sum_{R_{v_i}^{a_k}} w_{R_{v_i}^{a_k}, O_j^{T_n}} & if \quad \sum_{R_{v_i}^{a_k}} w_{R_{v_i}^{a_k}, O_j^{T_n}} < 1 \end{cases}$$

This definition of activation thresholds allows for activation of an object neuron whenever it is stimulated by the **whole defining combination** of this neuron, or when it is stimulated by a **sufficiently representative subset of rare or unique features** defining this neuron, e.g. if a feature defines only one object neuron, then it is enough to recognize it when this feature appears.

LINEAR APPROXIMATION

The DASNG model uses a linear approximation of all processes that take place in Associative Spiking Neurons (ASNs) as it greatly simplifies and speeds up calculations:

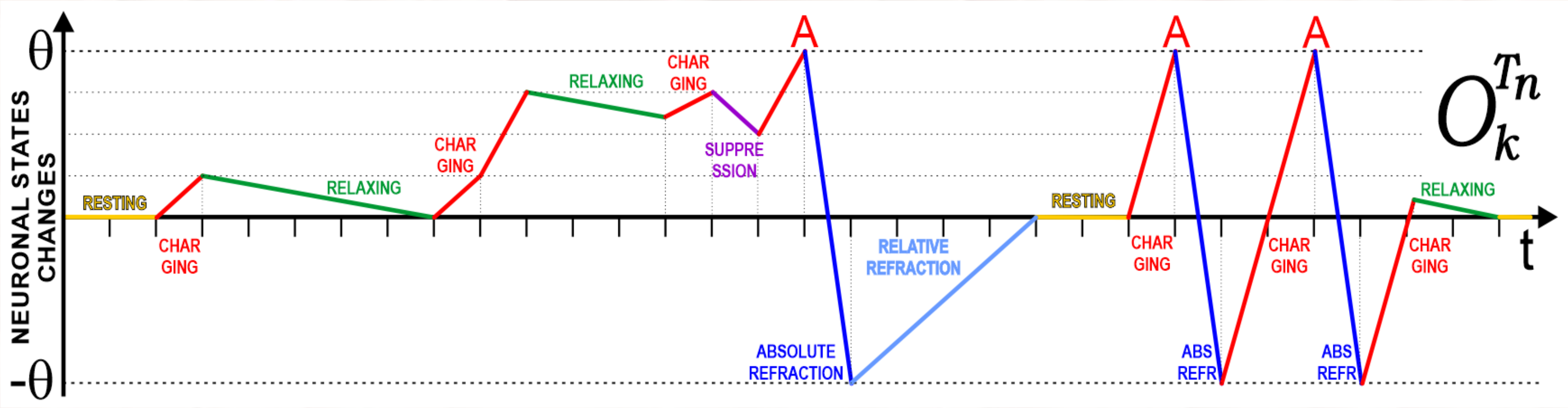


Each neuron creates an **internal neuronal process queue (IPQ)** of successive processes sorted after the time of their beginning. New processes are added to this queue on the basis of stimuli coming from other neurons or a sensor.

This queue can be modified at any time as a result of the appearance of a new **external stimulus**, which are appropriately combined (added) with the **charging or suppression processes** already added to this queue. **They** can also interrupt the **relaxation or relative refraction processes** or the **resting state** of the neuron.

INTERNAL NEURONAL PROCESS QUEUE (IPQ)

The use of the **internal neuronal process queue (IPQ)** is necessary because **associative spiking neurons (ASN)** **<u>operate over time</u>**, so subsequent stimuli and processes must be managed and ordered in time:



Internal neuronal process queue is implemented as a sorter list relative to the start time of the pipelined and ordered processes.

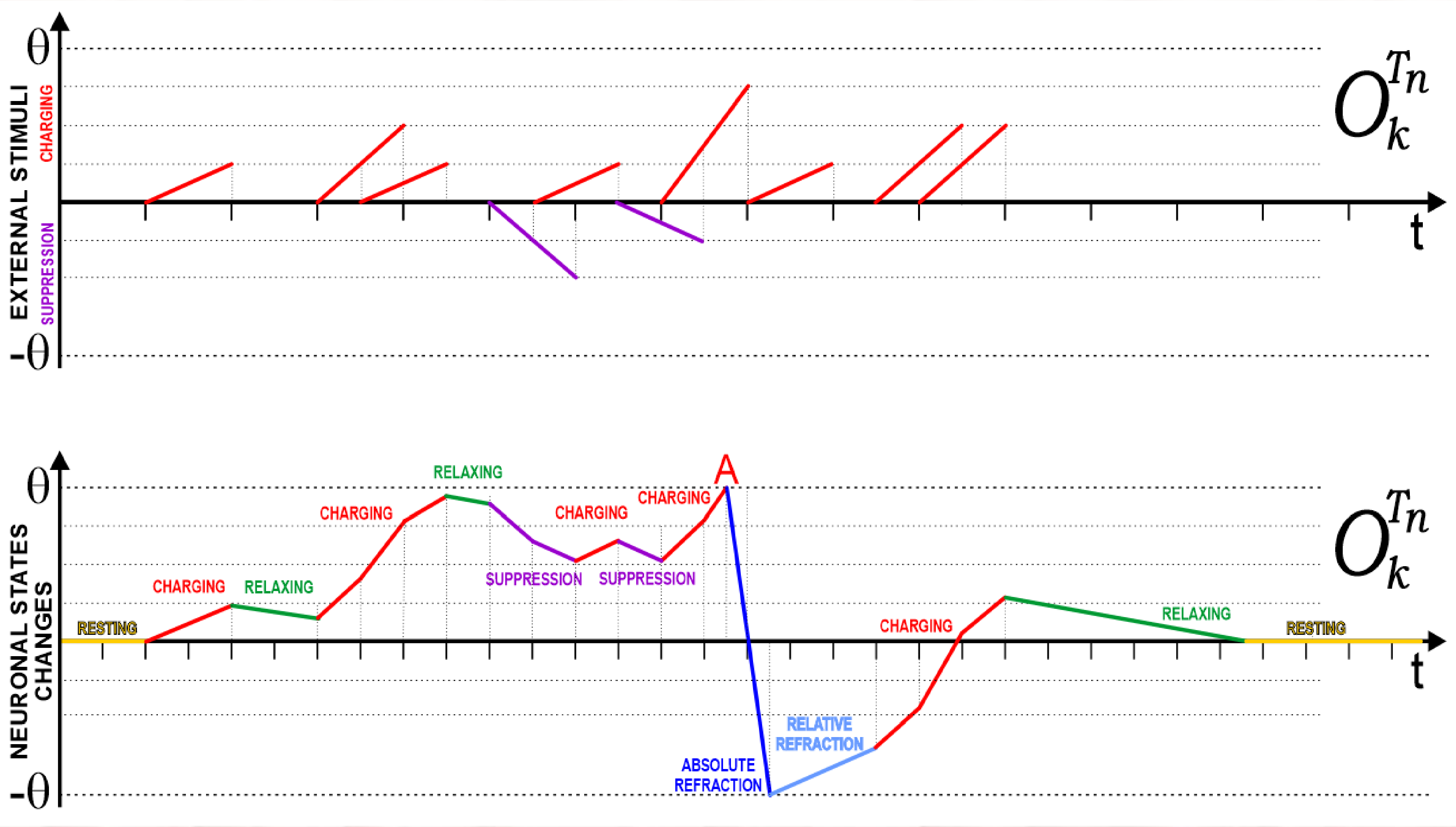
Although neuronal processes may overlap in time (e.g. external stimuli), new processes are added or combined with the existing ones, or they replace them.

As a result, we get a queue of sorter processes that come one after another and do not overlap in time. This way of operating this model significantly simplifies and speeds up all operations, and all results can be updated in the rare discrete moments.

In order to appropriately order parallel processes in time, we use **global event queue (GEQ)** which stores and orders all processes of all IPQs in the DASNG graph.

CREATION OF INTERNAL NEURONAL PROCESS QUEUE

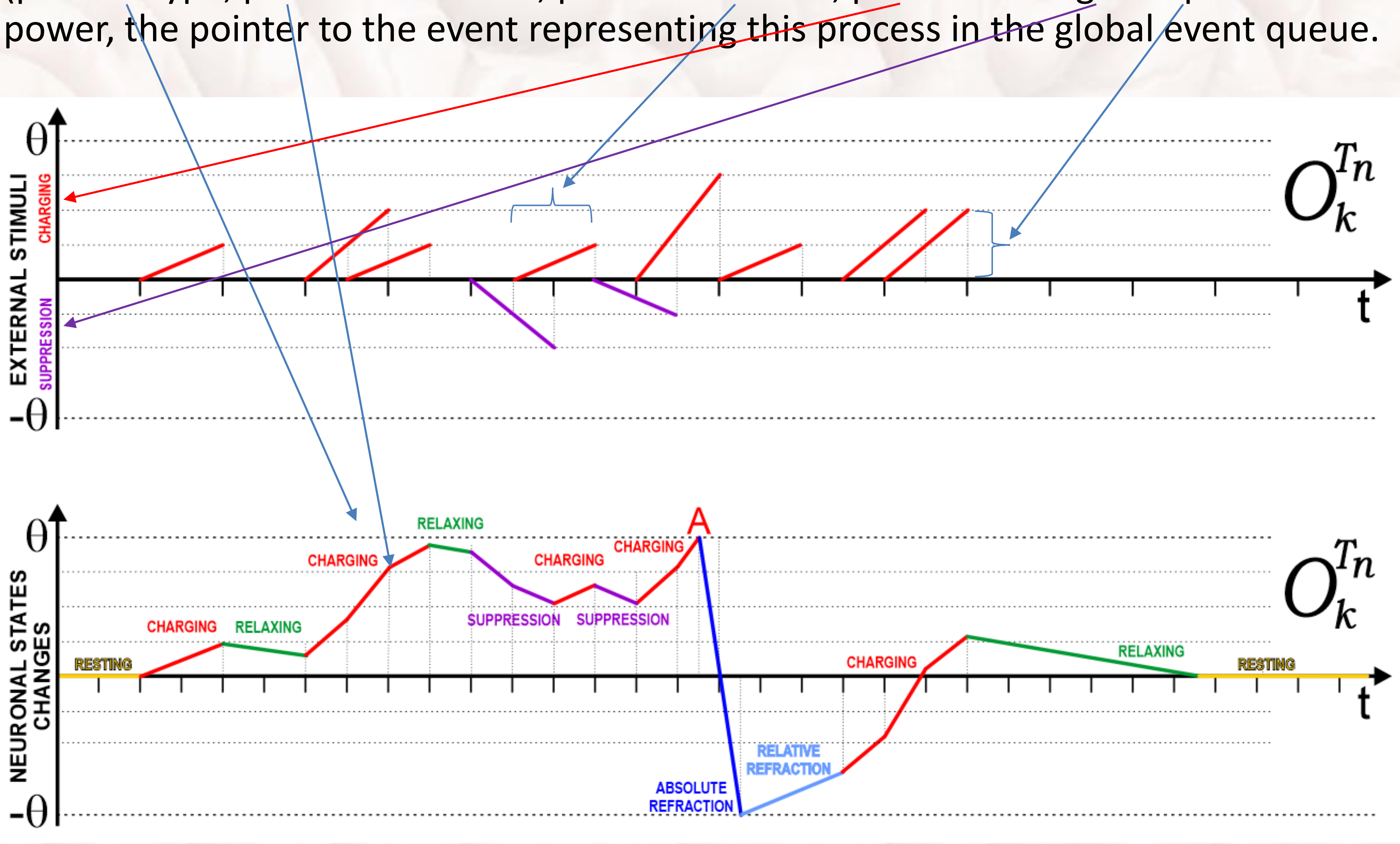
The neuronal process queue is created because of external stimuli that can come to the neuron at different moments and in a varying number depending on the number of neuronal connections and the activity of presynaptic neurons and sensors.



INTERNAL NEURAL PROCESSES

Neuronal processes are thus defined as:

(process type, process start time, process duration, positive or negative process power, the pointer to the event representing this process in the global event queue.



GLOBAL EVENT QUEUE - GEQ

Global Event Queue orders all events related to the neuronal processes in time.

Global Event Queue is responsible for running **updating methods in the nodes of the DASNG graphs and switch between the subsequent processes, e.g.:**

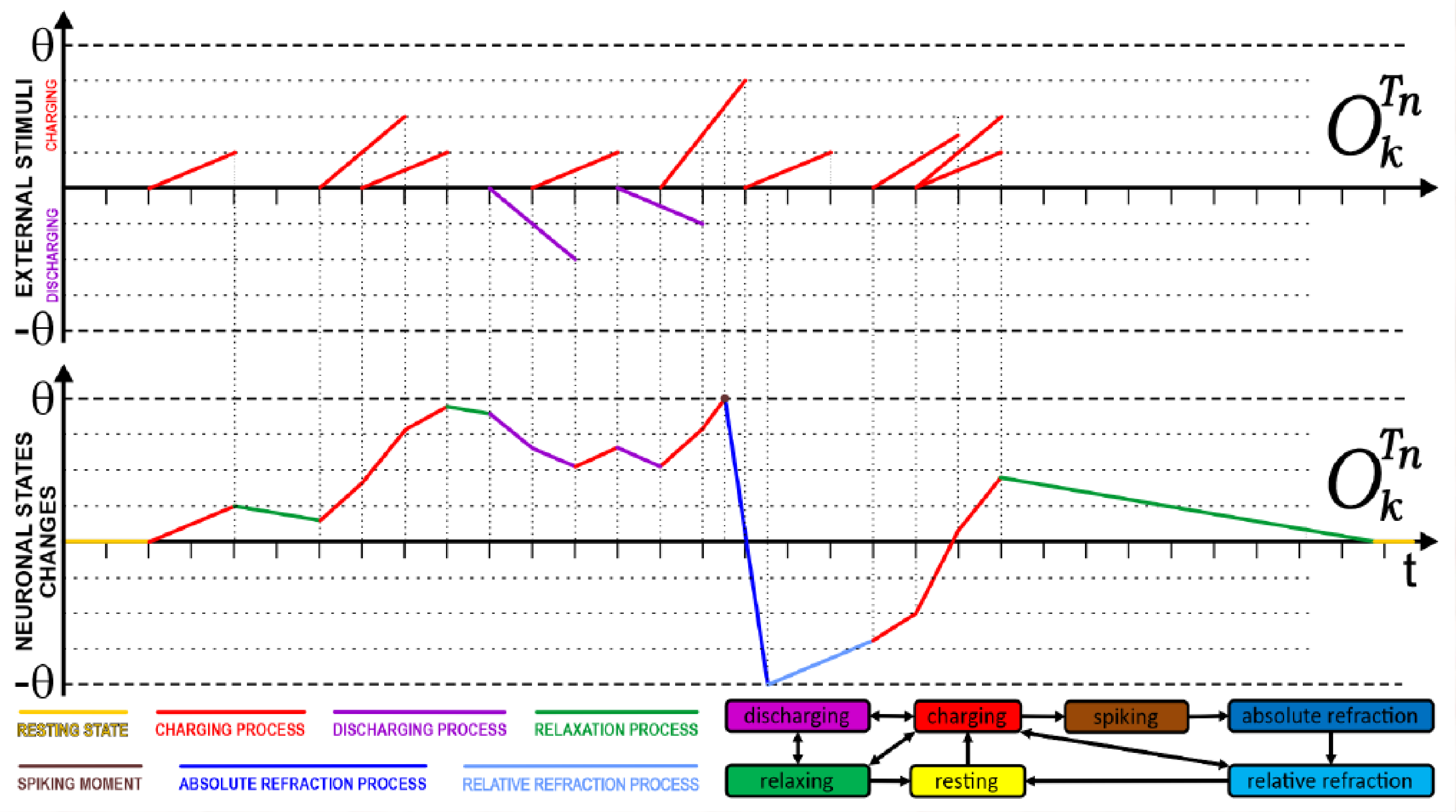
- After charging finishes, relaxation begins if a neuron is in the excitation state.
- After the activation threshold is achieved, this neuron spikes and starts its absolute refraction process.
- After the absolute refraction process finishes, the relative refraction process is automatically started.
- When the relaxation or relative refraction process finishes, neuron switches to its resting state.

The asynchronous parallelism model is based on the global DASNG event queue, which stores information about the predicted end time of the processes started in various neurons which are not in the resting state.

The event in this queue is represented by the pair (end time of the process, pointer), where pointer indicates the DASNG element (e.g. neuron) in which the process is about to terminate. Such a process is the first in the **internal process queue (IPQ)** in this neuron, so we don't need to look for it. Moreover, the IPQs typically consist of a few processes, so the operations on them are very fast!

COMBINING PROCESSES WITH NEW STIMULI

An Internal Process Queue (IPQ) chronologically orders neuronal processes that represent internal changes of a neuron and external stimuli. It avoids collisions of overlapping external stimuli, which are transformed to subsequent processes:



PROCESSES AND EVENTS

A neuronal process is defined as $\boldsymbol{P_k = (r_k, t_k, d_k, s_k, p_k)}$, where:
$\boldsymbol{r_k}$ - specifies the type of the process: charging (CH), discharging (DC), relaxation (RX), relative refraction (RR), or absolute refraction (AR),

$\boldsymbol{t_k}$ - the starting time of the process,

$\boldsymbol{d_k}$ - the duration of the process (a given period of time),

$\boldsymbol{s_k}$ - the strength of the process = relative neuronal change after the finished process,

$\boldsymbol{p_k}$ - a pointer to an **event in the global event queue (GEQ)** that tracks the end of this neuronal process and launches the neuron update.

An event is defined as an ordered pair $\boldsymbol{E_n = (t_n, p_n)}$, where:

$\boldsymbol{t_n = t_k + d_k}$ - is the end time of the process that should be finished and switched to another one or to a resting state, and the indicated neuron appropriately updated,

$\boldsymbol{p_n}$ - a pointer to the updated neuron which the current process should be finished.

All events from the entire neural network triggered by the internal neural processes are chronologically ordered in the **global event queue (GEQ)** after their end time.

Sometimes some events become to be outdated when an internal process queue is updated under the influence of new external stimuli. The outdated events $\boldsymbol{E_n}$ are automatically removed from the GEQ by the processes which indicate them ($\boldsymbol{p_k}$), and usually swapped to new ones watching the ends of the new processes.

ACHIEVEMENT OF THE SPIKING THRESHOLD

Some charging processes can achieve a spiking (activation) threshold or a resting state during their run, so it is necessary to check all charging processes for such an eventuality before addition of a new process event to the GEQ:

If during the neuron charging process the condition $X_{t_s} + s_s > \theta$ is satisfied, then the time $t^{SP} = t_s + d_s \cdot \frac{\theta - x_s}{s_s}$ of achievement of the neuron spiking (activation) threshold must be calculated to correctly set a watching event $E_n = \left(t^{SP}, p_n\right)$ to the GEQ.

If during the neuron discharging process the condition $X_{t_s} + s_s < 0$ is true, then the time of achievement of the neuron resting state must be calculated in the following way $t^{RS} = t_s + d_s \cdot \frac{x_s}{-s_s}$, and the appropriate event $E_n = \left(t^{SP}, p_n\right)$ put into the GEQ.

In the other cases, the state of the neuron is updated at the end of the charging or discharging process after:

$$X_{t_s} = X_{t_0} + s_0 \cdot \frac{t_s - t_0}{d_0}$$

If the neuron achieves it spiking threshold then the IPQ is cleared of all remaining processes and an absolute refraction process is added to the IPQ. During this process the neuron does not react to any further stimuli:

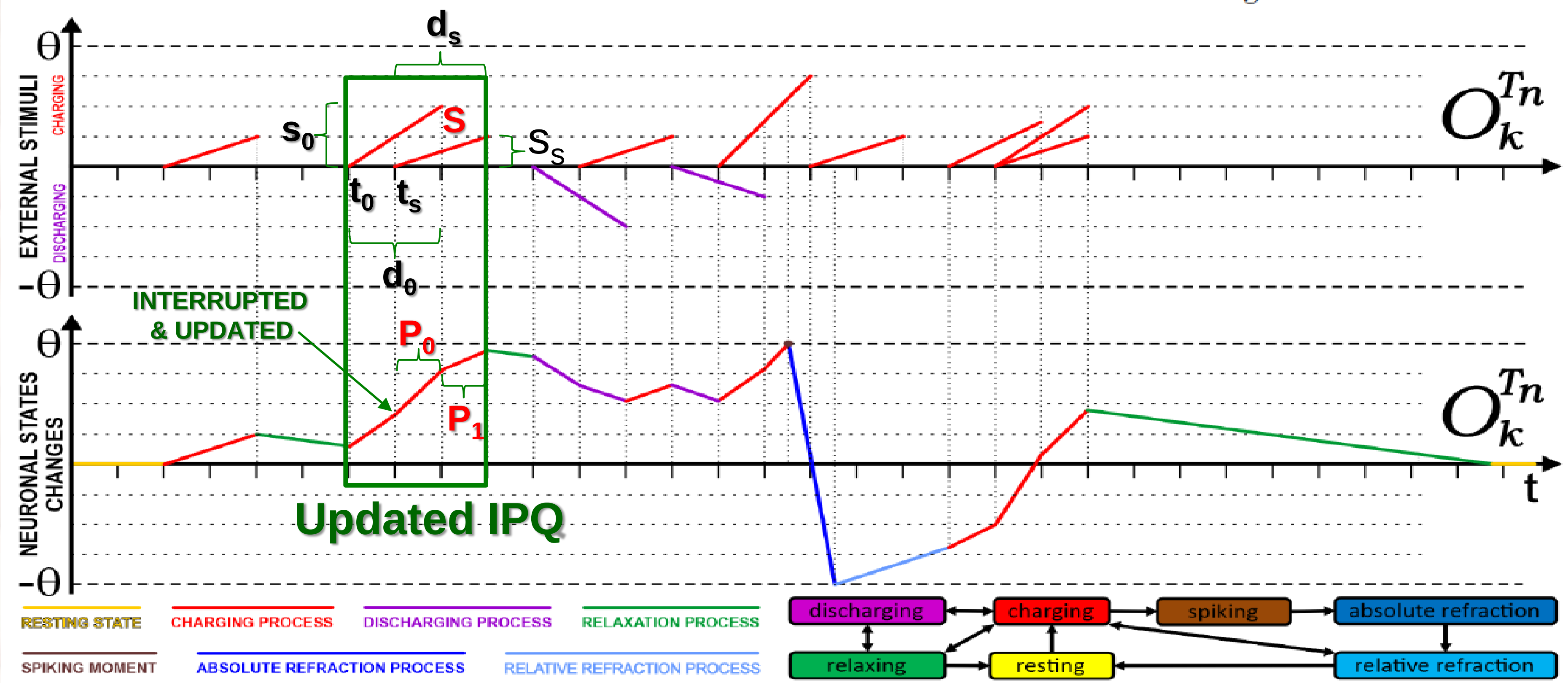
$$P_{AR} = \left(AR, t^{SP}, 1, -2 \cdot \theta, p_{AR}\right)$$

CREATING AND UPDATING THE PROCESS QUEUE AFTER THE EXTERNAL STIMULI

The processes in the IPQ are automatically added or removed for each new stimulation $S = (t_s, d_s, s_s)$ which overlap some processes in the IPQ:

$$\widehat{P}_0 = \left(\widehat{r}_0, t_s, d_0 - (t_s - t_0), s_0 \cdot \frac{d_0 - (t_s - t_0)}{d_0} + s_s \cdot \frac{d_0 - (t_s - t_0)}{d_s}, \widehat{p}_0\right)$$

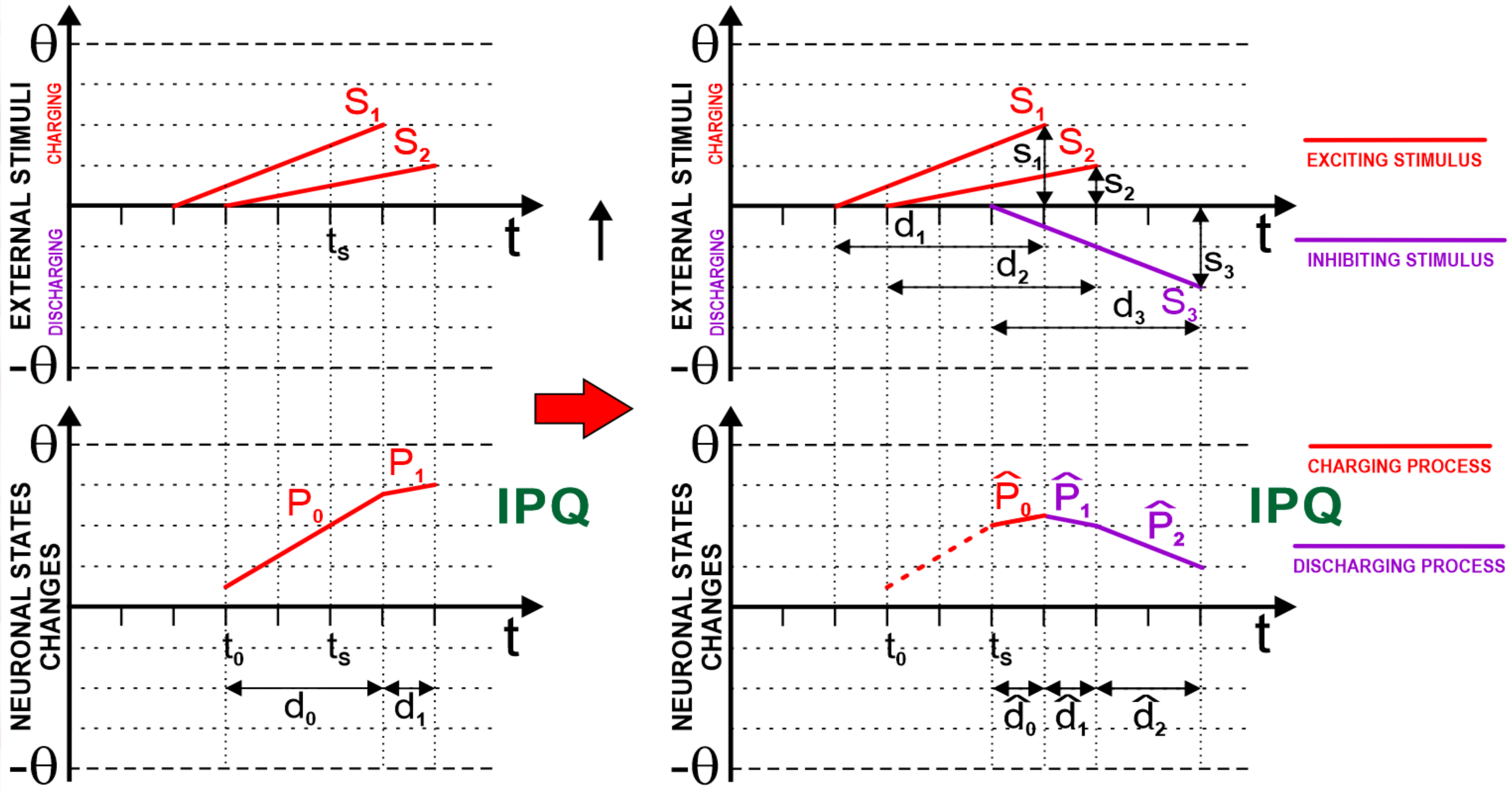
$$\widehat{P}_1 = \left(\widehat{r}_1, t_0 + d_0, d_s - (d_0 - (t_s - t_0)), s_s \cdot \frac{d_s - (d_0 - (t_s - t_0))}{d_s}, \widehat{p}_1\right)$$



UPDATING THE PROCESS QUEUE DUE TO THE NEW EXTERNAL STIMULUS

$$\widehat{P}_0 = \left(\widehat{r}_0, t_s, d_0 - (t_s - t_0), s_0 \cdot \frac{d_0 - (t_s - t_0)}{d_0} + s_s \cdot \frac{d_0 - (t_s - t_0)}{d_s}, \widehat{p}_0\right)$$

$$\widehat{P}_1 = \left(\widehat{r}_1, t_0 + d_0, d_s - (d_0 - (t_s - t_0)), s_s \cdot \frac{d_s - (d_0 - (t_s - t_0))}{d_s}, \widehat{p}_1\right)$$



DURATION OF RELAXATION AND RELATIVE REFRACTION

The duration of the relaxation process d^{RX} depends on the current state of the neuron X, its spiking threshold θ, and on the assumed maximum relaxation period $p^{RX} = 10$:

$$d^{RX} = \frac{p^{RX} \cdot X_{t_0}}{\theta}$$

The duration of the relative refraction process d^{RR} depends on the state of the neuron X, its spiking threshold θ, and on the assumed maximum relative refraction period $p^{RR} = 5$:

$$d^{RR} = -\frac{p^{RR} \cdot X_{t_0}}{\theta}$$

TRANSFORMATION ORDER OF TABLES

During the transformation of the relational database, note that only the tables which all foreign keys are already represented in the DASNG structure can be transformed. On the other hand, they have to wait until all their foreign keys will be transformed during the associative transformation of other tables. So the sequence of transformed tables is important and the tables must be transformed in an appropriate order as we can see in the figure below and our sample tables. This associative transformation of database tables can be performed also to the passive AGDS structures.

1

TABLE C: Subjects	
A6:S Subjects	Primary KeyC
business	1
chemistry	2
drama	3
geography	4
history	5
mathematics	6
physics	7

3

TABLE D: Likes	
Foreign KeyC	Foreign KeyA
1	1
1	2
3	1
1	4
2	3
6	12
2	6
4	7
7	5
4	12
6	9
3	12
5	7
7	16
1	12
6	14

2

TABLE A: Pupils				
Primary KeyA	A1:S Last Name	A2:S First Name	Foreign KeyE	Foreign KeyB
1	Brown	Amy	1	1
2	Cruise	Jack	1	2
3	Brown	Jack	2	2
4	Ford	Kate	3	1
5	Hanks	Luke	2	5
6	Jolie	Mary	3	3
7	Ford	Jack	2	2
8	Brown	Kate	1	4
9	Jolie	Rose	2	6
10	Lopez	Tom	1	9
11	Pitt	Tom	3	5
12	Brown	Kate	1	10
13	Ford	Luke	2	7
14	Smith	Luke	2	10
15	Jolie	Mary	2	8
16	Trump	Tom	3	10

1

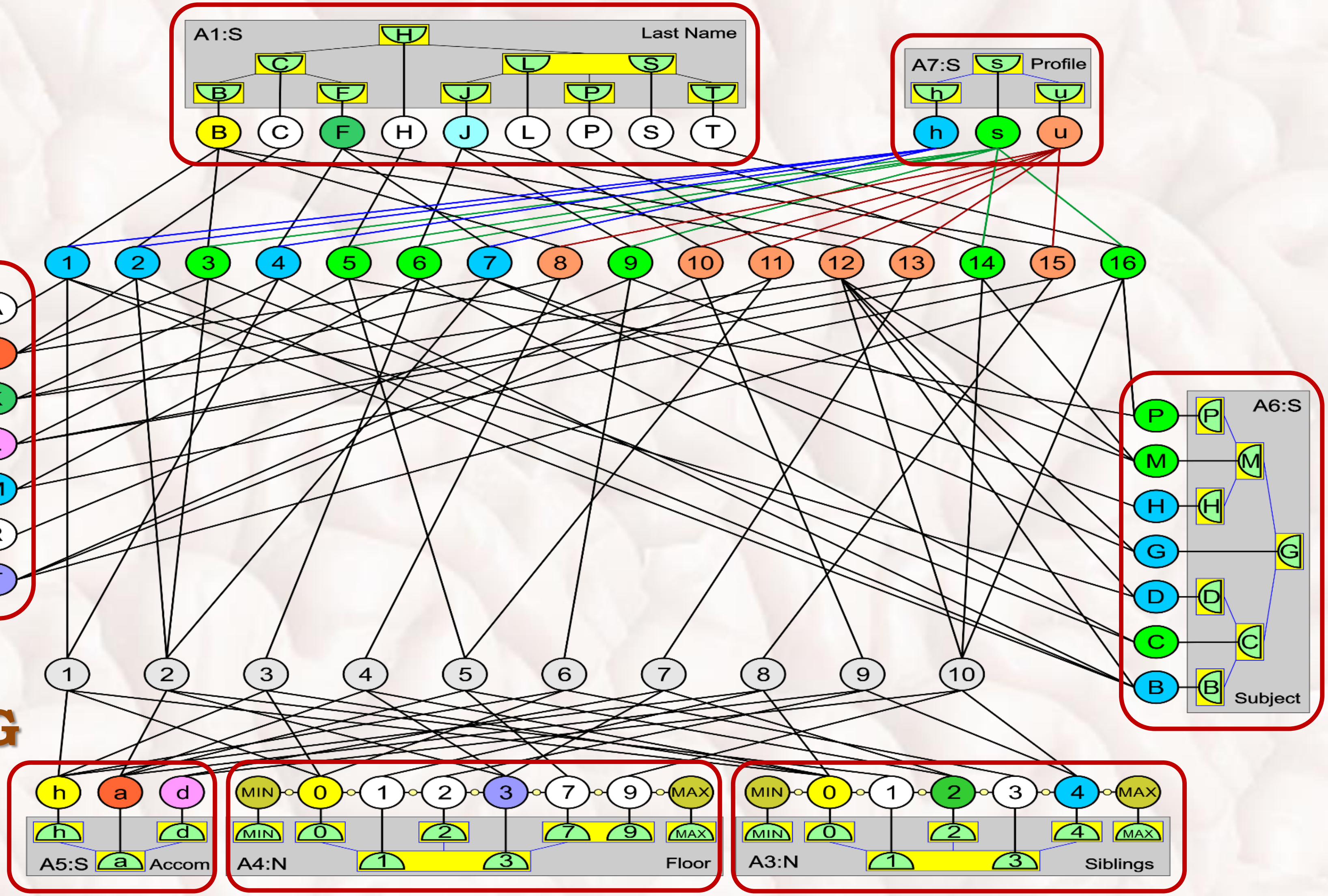
TABLE B: Live with/in			
Primary KeyB	A3:N No of Siblings	A4:N Floor	A5:S Accommodation
1	0	0	house
2	1	3	appartament
3	0	0	house
4	0	3	appartament
5	2	7	appartament
6	3	0	house
7	2	3	dormitory
8	0	1	house
9	4	2	appartament
10	4	9	dormitory

1

TABLE E: Profile	
Primary KeyE	A7:S Profile
1	humanistic
2	science
3	unselected

DEEP ASSOCIATIVE NEURAL GRAPHS - DASNG

For each attribute separately, the **unique attribute values** are represented by sensors and sensory neurons in this **deep associative graph.** Thanks to AVB-trees, we get access to all data usually in constant time.



DASNG

DEEP ASSOCIATIVE NEURAL GRAPHS - DASNG

The **unique records** of each relational database table containing objects defined by several attribute values and/or foreign keys pointing out records of other tables are represented by a separate **layer** in this **deep associative neural graph**. The number of such layers depends on the number of tables in a given database. Here we have two such layers:

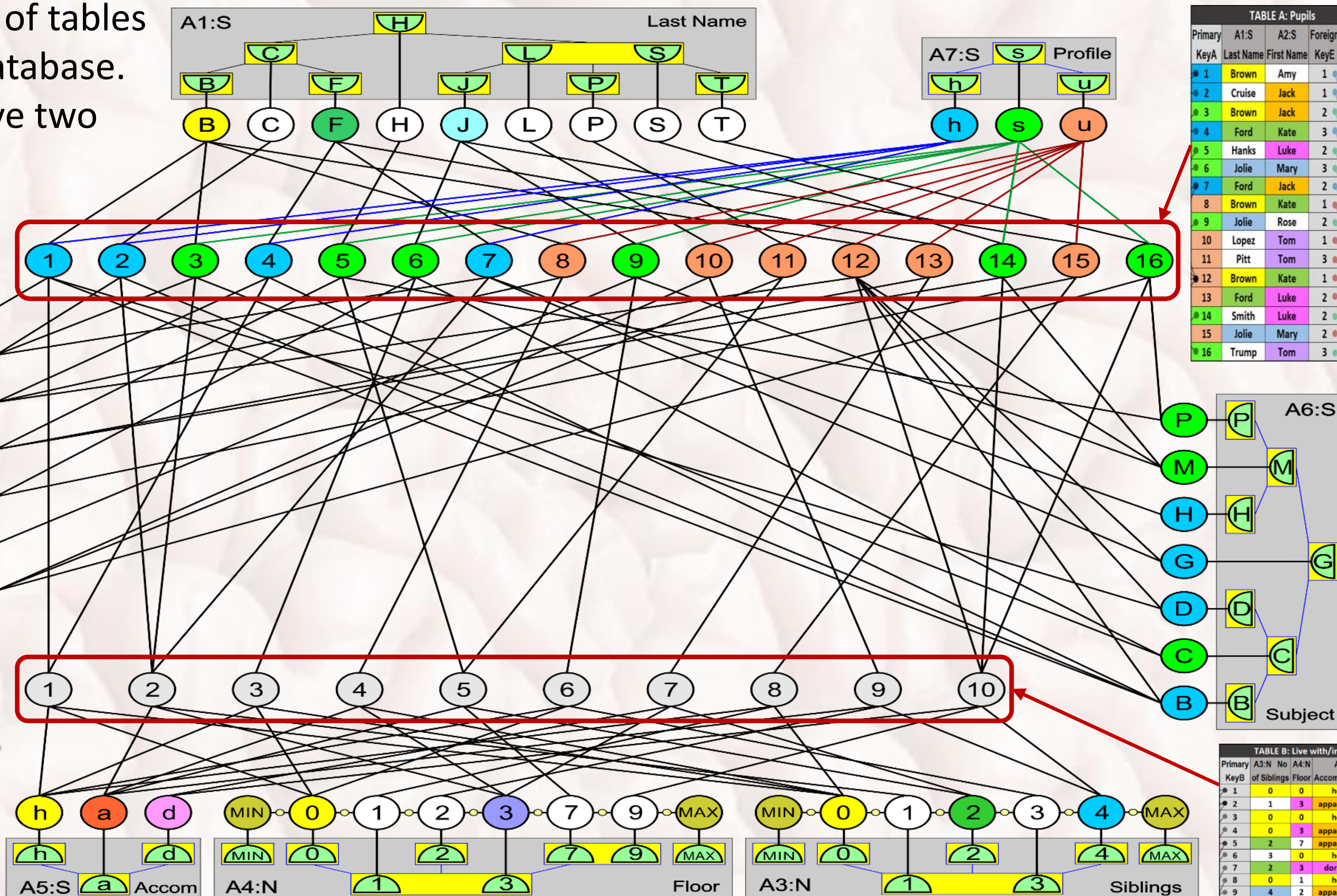


TABLE A: Pupils				
Primary KeyA	A1:S Last Name	A2:S First Name	Foreign KeyE	Foreign KeyB
1	Brown	Amy	1	1
2	Cruise	Jack	1	2
3	Brown	Jack	2	2
4	Ford	Kate	3	1
5	Hanks	Luke	2	5
6	Jolie	Mary	3	3
7	Ford	Jack	2	2
8	Brown	Kate	1	4
9	Jolie	Rose	2	6
10	Lopez	Tom	1	9
11	Pitt	Tom	3	5
12	Brown	Kate	1	10
13	Ford	Luke	2	7
14	Smith	Luke	2	10
15	Jolie	Mary	2	8
16	Trump	Tom	3	10

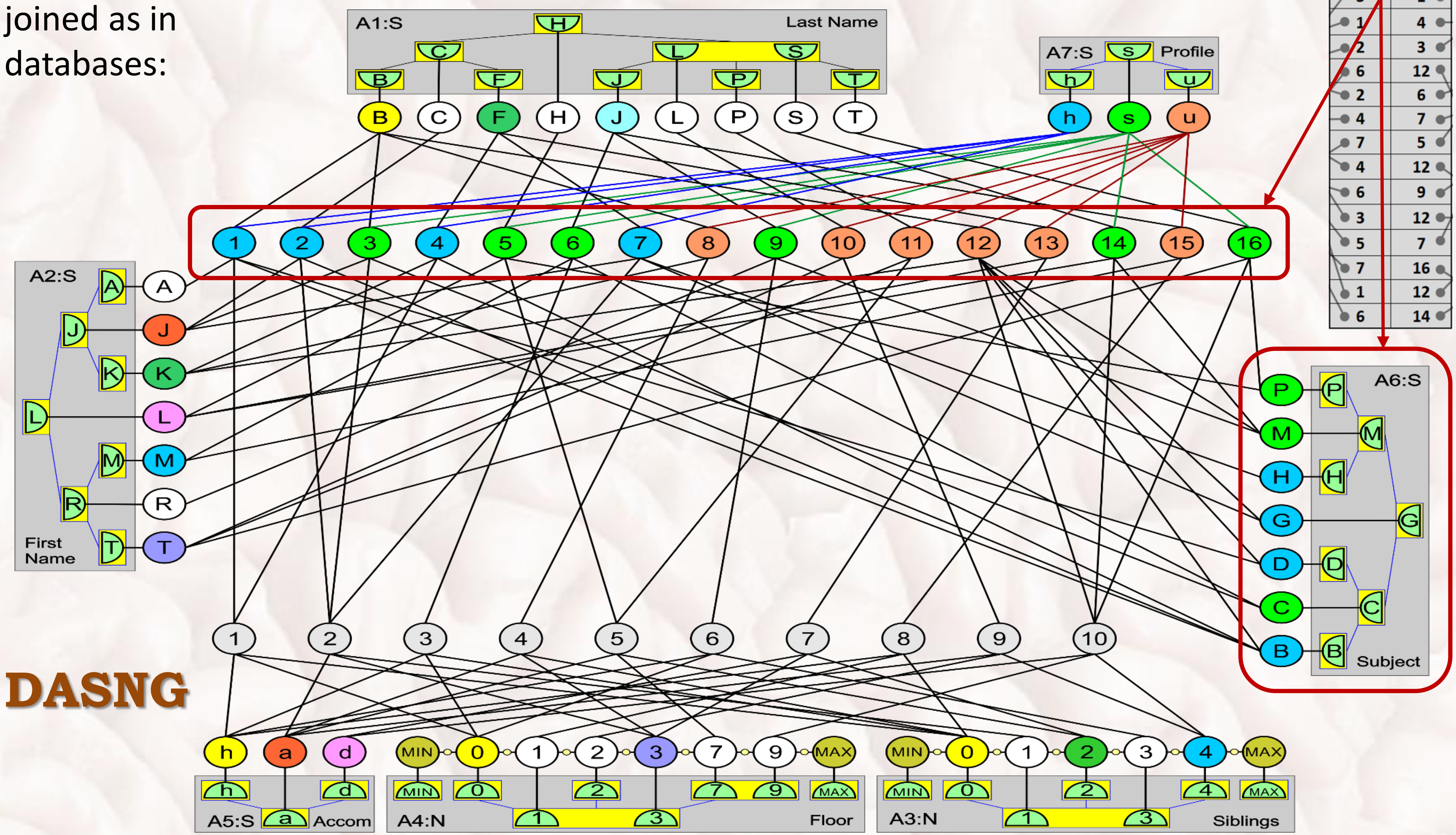
DASNG

TABLE B: Live with/in			
Primary KeyB	A3:N No of Siblings	A4:N Floor	A5:S Accommodation
1	0	0	house
2	1	3	appartament
3	0	0	house
4	0	3	appartament
5	2	7	appartament
6	3	0	house
7	2	3	dormitory
8	0	1	house
9	4	2	appartament
10	4	9	dormitory

DEEP ASSOCIATIVE NEURAL GRAPHS - DASNG

In **deep associative neural graphs**, there is no need to use **link tables** representing **many-to-many relations (N:M)**, because they can be replaced by direct connections between neurons representing related objects. Thus, the records do not need to be joined as in databases:

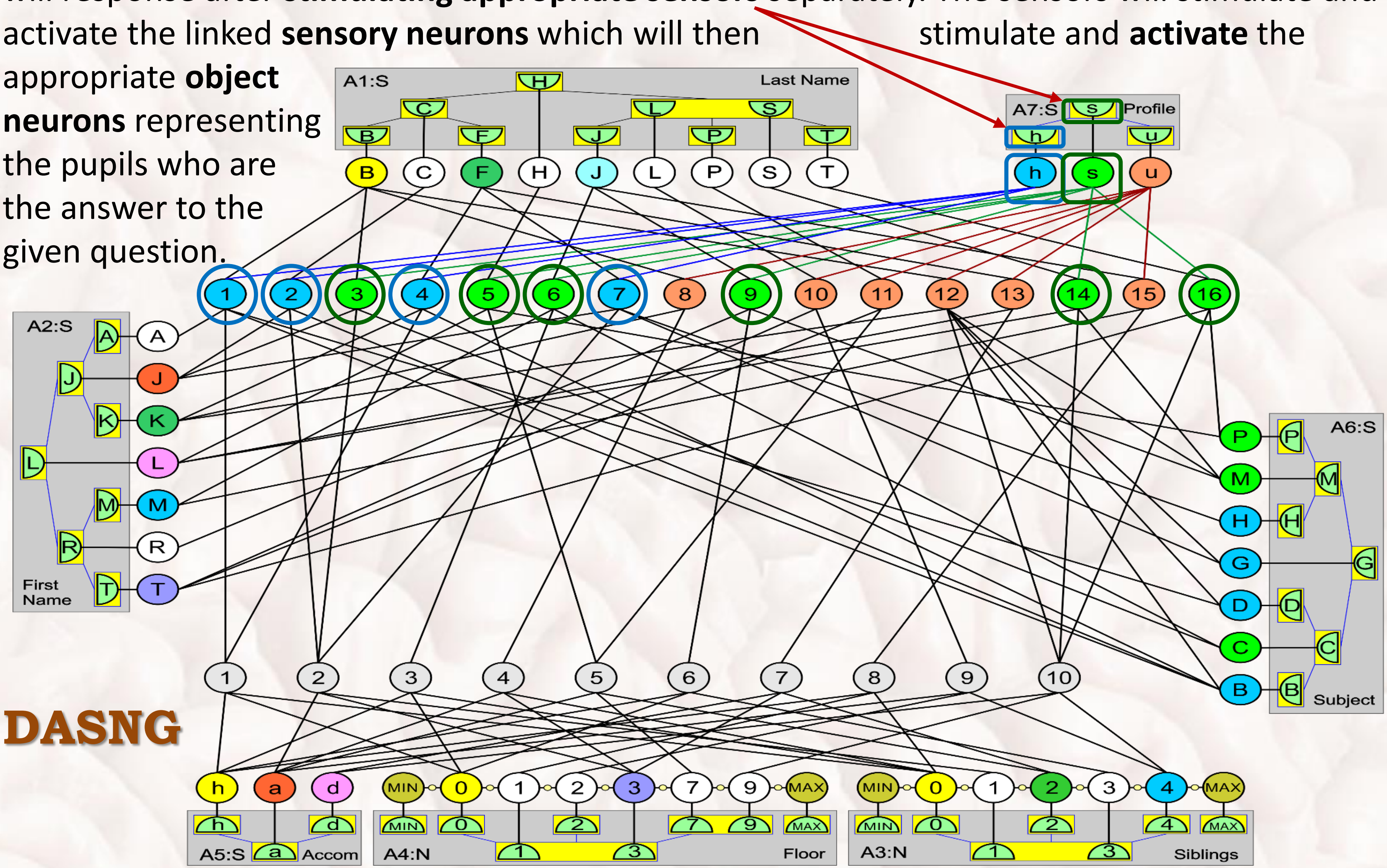
TABLE D: Likes	
Foreign KeyC	Foreign KeyA
1	1
1	2
3	1
1	4
2	3
6	12
2	6
4	7
7	5
4	12
6	9
3	12
5	7
7	16
1	12
6	14



DASNG

DEEP ASSOCIATIVE NEURAL GRAPHS - DASNG

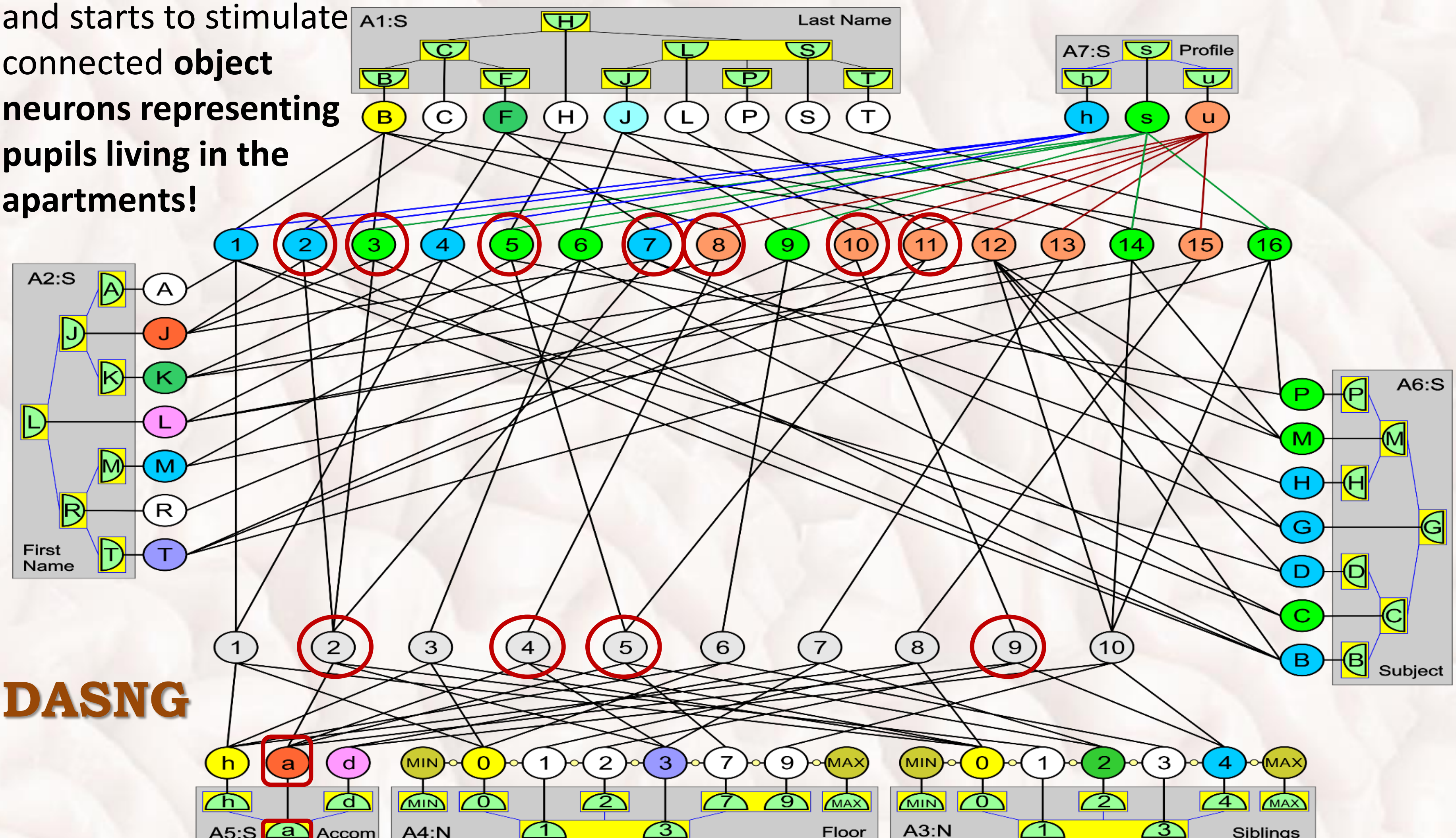
Now, try to answer the question: ***Which pupils have similar interests?*** using the DASNG which will response after **stimulating appropriate sensors** separately. The sensors will stimulate and activate the linked **sensory neurons** which will then stimulate and **activate** the appropriate **object neurons** representing the pupils who are the answer to the given question.



DASNG

DEEP ASSOCIATIVE NEURAL GRAPHS - DASNG

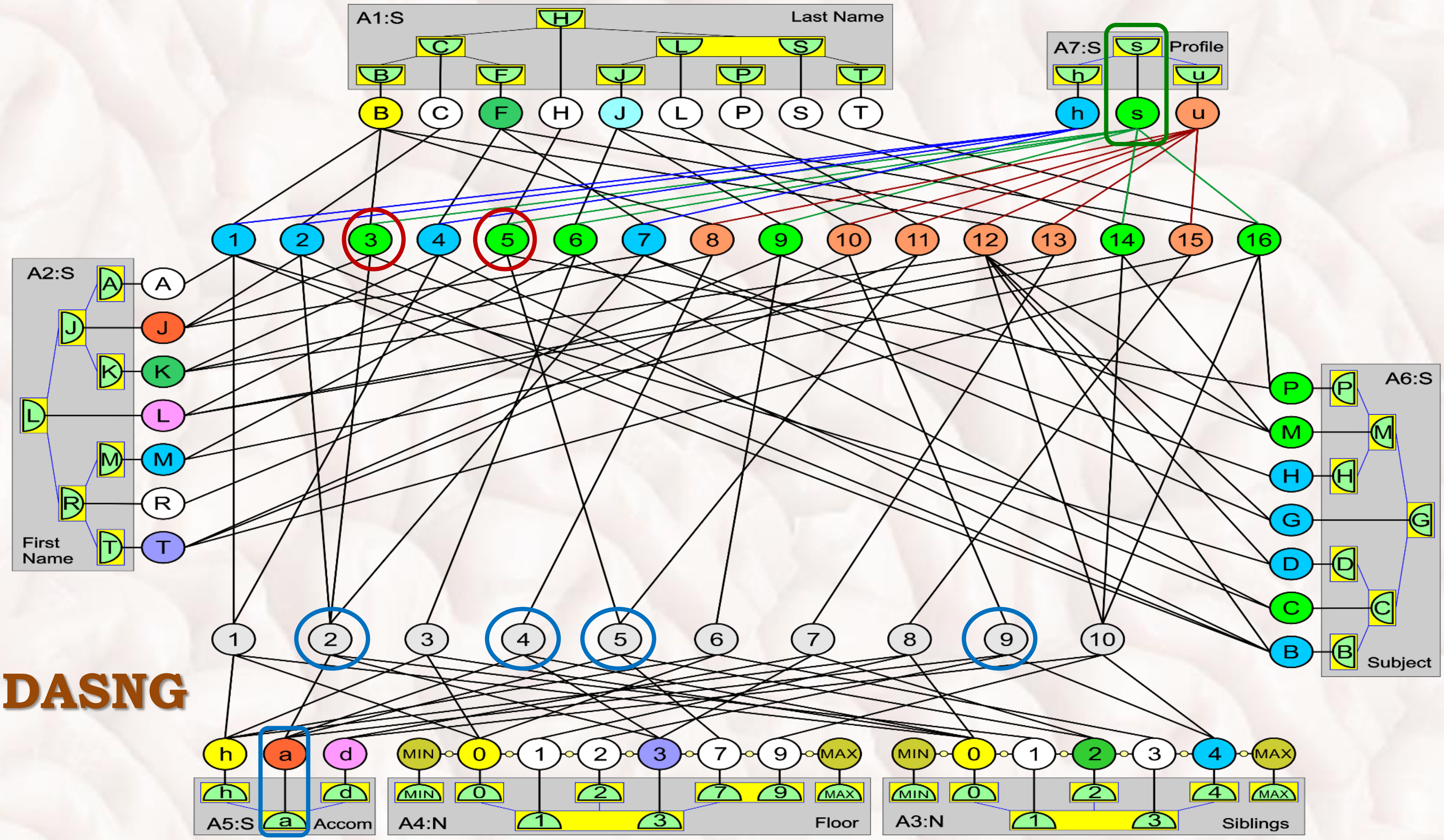
We can also answer the second question: ***Which pupils do live in apartments?*** by stimulating the **sensor „apartment"**, which stimulates and activates its **sensory neuron** that stimulates **the object neurons representing living conditions**, which are activated after some time, and starts to stimulate connected **object neurons representing pupils living in the apartments!**



DASNG

DEEP ASSOCIATIVE NEURAL GRAPHS - DASNG

Note that we can also stimulate various combinations of sensors representing the logical conjunction of selected features, which will result in stimulation and the fastest activation of those neurons which represent **pupils** interested in science AND living in apartments.

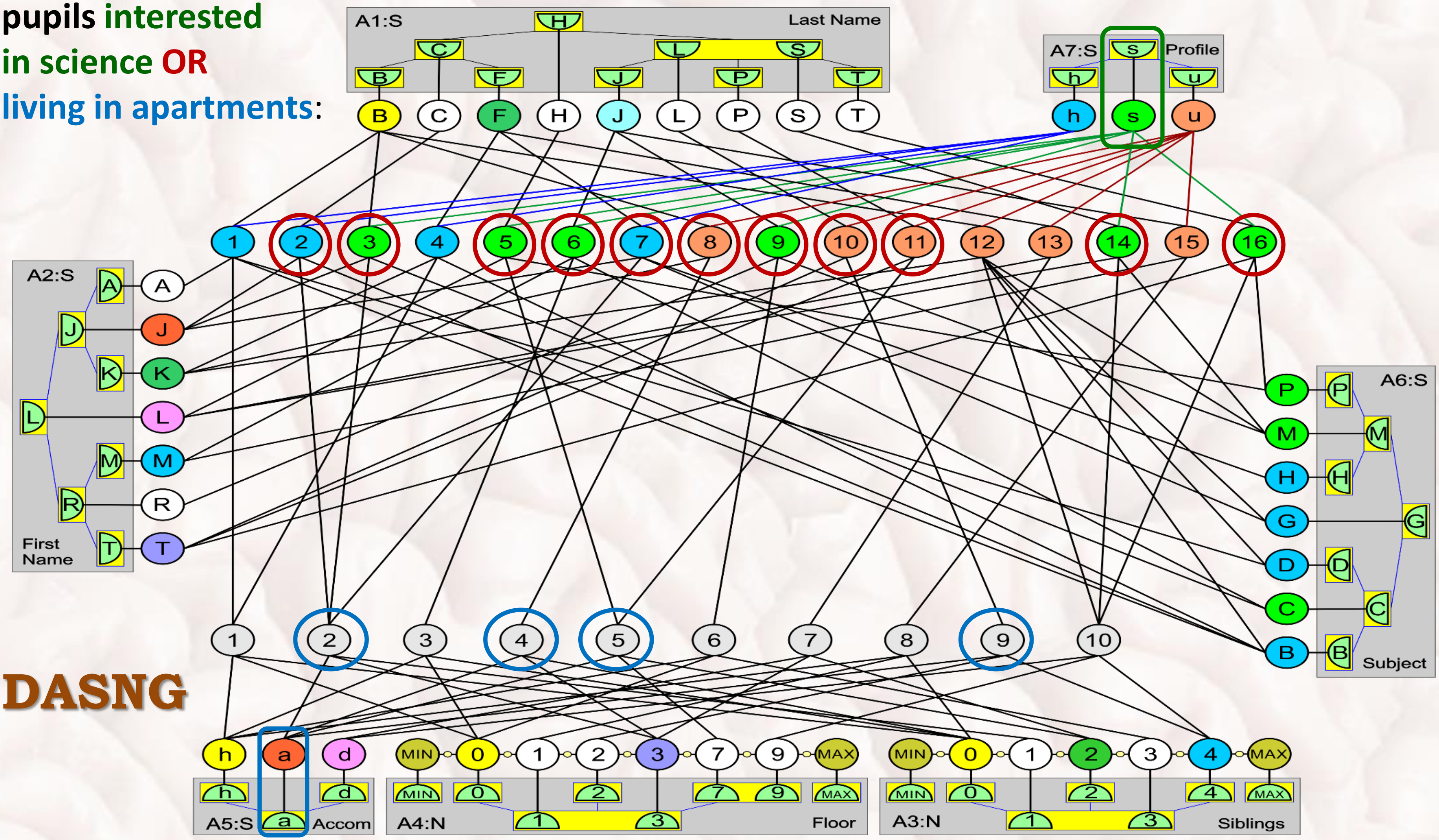


DASNG

DEEP ASSOCIATIVE NEURAL GRAPHS - DASNG

In the case of logical alternative, we wait for the activity of additional pupil neurons longer, where the chronology of activations points out how strong the pupils satisfy the alternative, i.e. the activation moments represent the adaptation degree of pupils to the condition:

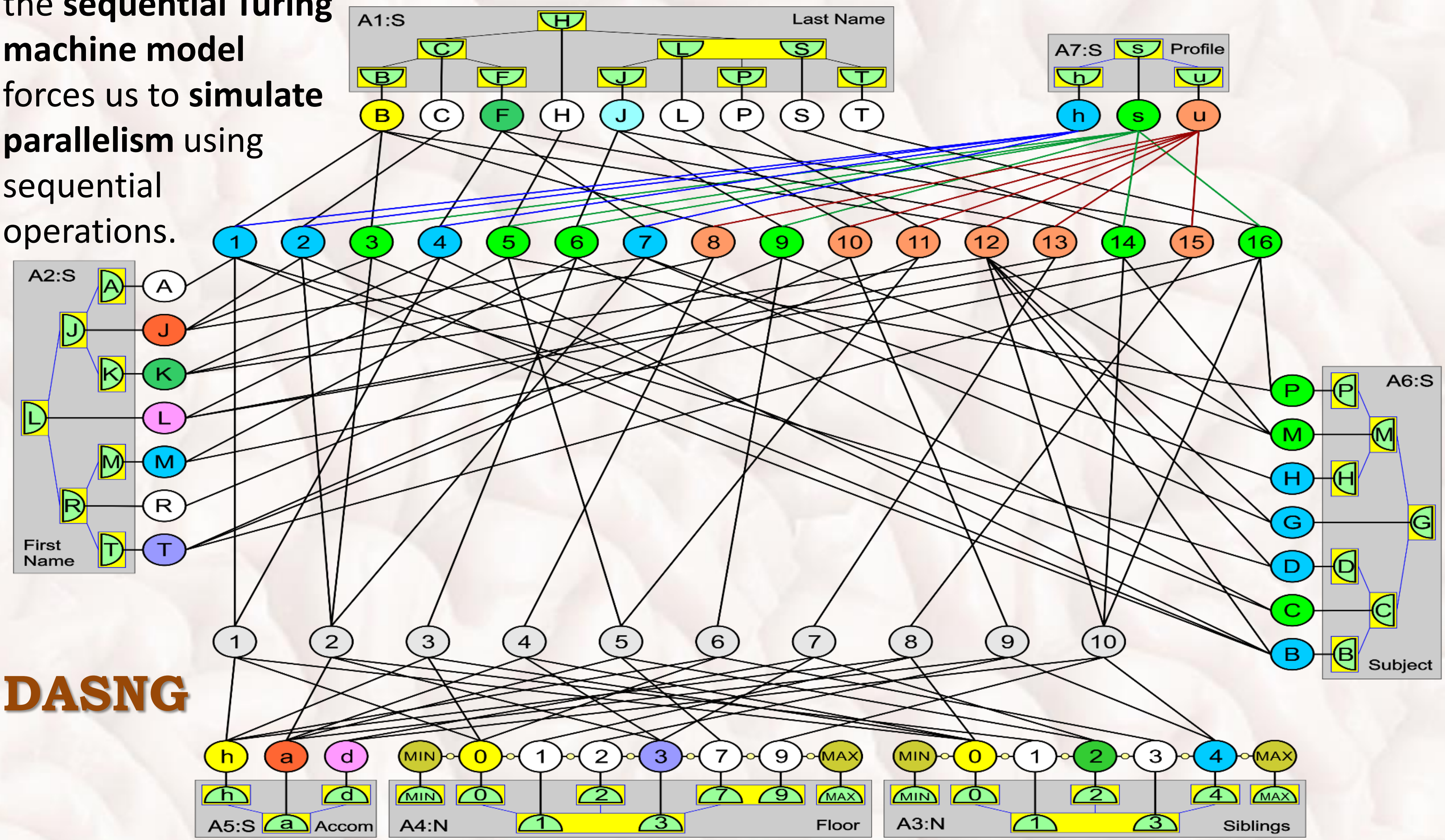
pupils interested in science OR living in apartments:



DASNG

DEEP ASSOCIATIVE NEURAL GRAPHS - DASNG

Sensors and neurons in such associative graphs **can be and should be stimulated in parallel alike in the human brain** in order to achieve responses and answers **in constant time**. Unfortunately, our contemporary computers and computational technology based on the **sequential Turing machine model** forces us to **simulate parallelism** using sequential operations.



DASNG

EXPLORATION OF THE KWNOWLEDGE

We can use tables or deep associative neural graphs for **data mining** or **knowledge exploration**:

SEARCHING THROUGH TABLES

using classic data mining approaches, calculation of frequent patterns, supports, linking and comparing elements, using ECLAT transformation and Apriori algorithm…

X

STIMULATION OF THE DASNG

and just waiting for answers…

TABLE C: Subjects

A6:S Subjects	Primary KeyC
business	1
chemistry	2
drama	3
geography	4
history	5
mathematics	6
physics	7

TABLE D: Likes

Foreign KeyC	Foreign KeyA
1	1
1	2
3	1
1	4
2	3
6	12
2	6
4	7
7	5
4	12
6	9
3	12
5	7
7	16
1	12
6	14

TABLE A: Pupils

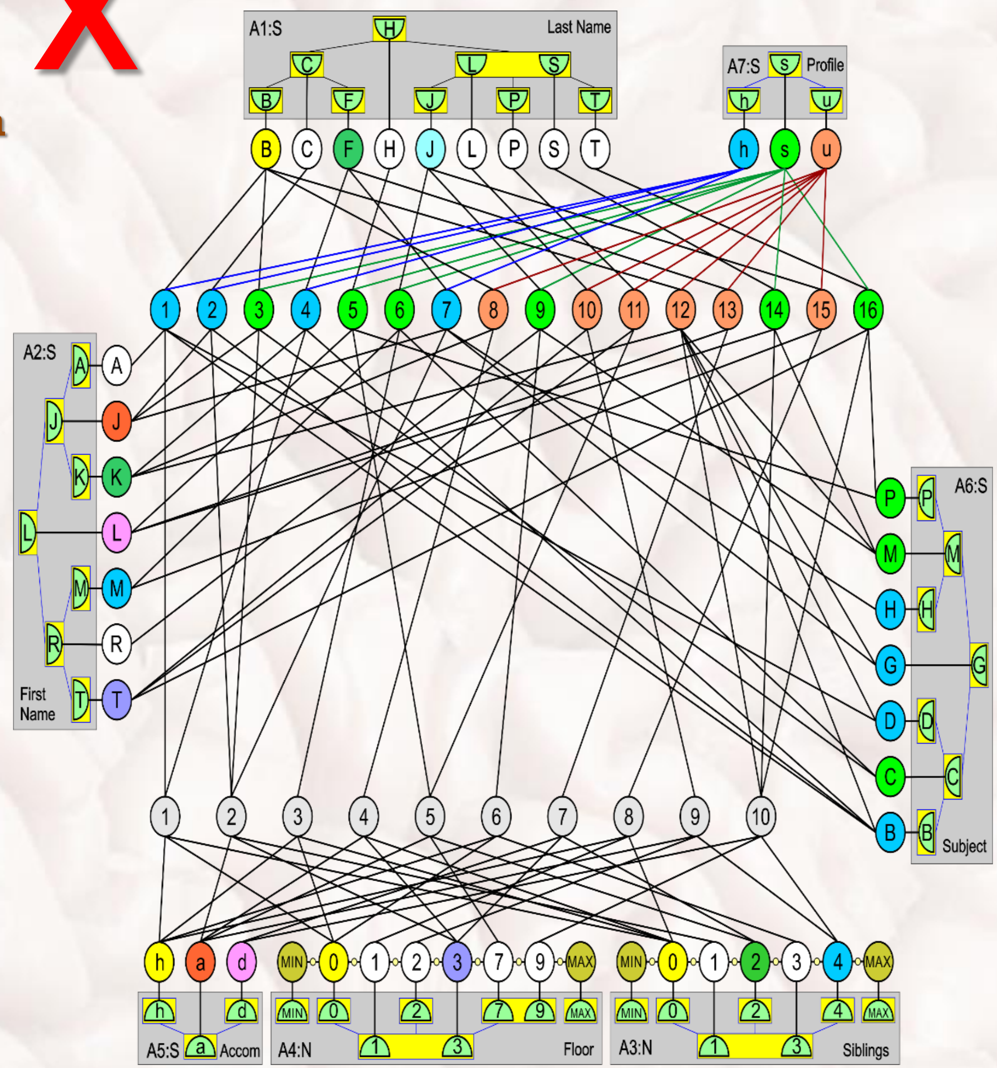
Primary KeyA	A1:S Last Name	A2:S First Name	Foreign KeyE	Foreign KeyB
1	Brown	Amy	1	1
2	Cruise	Jack	1	2
3	Brown	Jack	2	2
4	Ford	Kate	3	1
5	Hanks	Luke	2	5
6	Jolie	Mary	3	3
7	Ford	Jack	2	2
8	Brown	Kate	1	4
9	Jolie	Rose	2	6
10	Lopez	Tom	1	9
11	Pitt	Tom	3	5
12	Brown	Kate	1	10
13	Ford	Luke	2	7
14	Smith	Luke	2	10
15	Jolie	Mary	2	8
16	Trump	Tom	3	10

TABLE B: Live with/in

Primary KeyB	A3:N No of Siblings	A4:N Floor	A5:S Accommodation
1	0	0	house
2	1	3	appartament
3	0	0	house
4	0	3	appartament
5	2	7	appartament
6	3	0	house
7	2	3	dormitory
8	0	1	house
9	4	2	appartament
10	4	9	dormitory

TABLE E: Profile

Primary KeyE	A7:S Profile
1	humanistic
2	science
3	unselected



BIBLIOGRAPHY

1. **A. Horzyk**, J. A. Starzyk, J. Graham, *Integration of Semantic and Episodic Memories*, IEEE Transactions on Neural Networks and Learning Systems, Vol. 28, Issue 12, Dec. 2017, pp. 3084 - 3095, DOI: 10.1109/TNNLS.2017.2728203, **IF = 6.108**.
2. **A. Horzyk**, *Deep Associative Semantic Neural Graphs for Knowledge Representation and Fast Data Exploration*, Proc. of KEOD 2017, SCITEPRESS Digital Library, 2017, pp. 67-79. - presentation, DOI: 10.13140/RG.2.2.30881.92005
3. **A. Horzyk** and J.A. Starzyk, *Fast Neural Network Adaptation with Associative Pulsing Neurons*, IEEE Xplore, In: 2017 IEEE Symposium Series on Computational Intelligence, pp. 339-346, 2017, DOI: 10.1109/SSCI.2017.8285369. - presentation, movie Iris-4, movie Iris-12
4. **A. Horzyk**, Neurons Can Sort Data Efficiently, Proc. of ICAISC 2017, Springer-Verlag, LNAI, 2017.
5. **A. Horzyk**, J. A. Starzyk and Basawaraj, *Emergent creativity in declarative memories*, IEEE Xplore, In: 2016 IEEE Symposium Series on Computational Intelligence, Greece, Athens: Institute of Electrical and Electronics Engineers, Curran Associates, Inc. 57 Morehouse Lane Red Hook, NY 12571 USA, 2016, ISBN 978-1-5090-4239-5, pp. 1-8, DOI: 10.1109/SSCI.2016.7850029.
6. **A. Horzyk**, Human-Like Knowledge Engineering, Generalization and Creativity in Artificial Neural Associative Systems, Springer-Verlag, AISC 11156, ISSN 2194-5357, ISBN 978-3-319-19089-1, ISBN 978-3-319-19090-7 (eBook), DOI 10.1007/978-3-319-19090-7, Springer, Switzerland, 2016, pp. 39-51.
7. Innovative Types and Abilities of Neural Networks Based on Associative Mechanisms and a New Associative Model of Neurons - referat na zaproszenie na międzynarodowej konferencji ICAISC 2015, Springer-Verlag, LNAI 9119, 2015, pp. 26-38, DOI 10.1007/978-3-319-19324-3_3
8. **Horzyk, A.**, *How Does Generalization and Creativity Come into Being in Neural Associative Systems and How Does It Form Human-Like Knowledge?*, **Neurocomputing**, 2014, **IF = 1,634**.
9. **Horzyk, A.**, *Human-Like Knowledge Engineering, Generalization and Creativity in Artificial Neural Associative Systems*, Springer, AISC 11156, 2014.
10. **Horzyk, A.**, *Sztuczne systemy skojarzeniowe i asocjacyjna sztuczna inteligencja*, EXIT, Warszawa, 2013.
11. Tadeusiewicz, R., **Horzyk, A.**, *Man-Machine Interaction Improvement by Means of Automatic Human Personality Identification*, Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen (Eds.), Springer, LNCS 8104, 2013.
12. **Horzyk, A.**, Gadamer, M., *Associative Text Representation and Correction*, Springer Verlag Berlin Heidelberg, LNAI 7894, 2013, pp. 76-87.
13. **Horzyk, A.**, *Information Freedom and Associative Artificial Intelligence*, Springer Verlag Berlin Heidelberg, LNAI 7267, 2012, pp. 81-89.
14. **Horzyk, A.**, *Self-Optimizing Neural Network 3*, L. Franco, D. Elizondo, J.M. Jerez (eds.), Constructive Neural Networks, Springer, Series: Studies in Computational Intelligence, Vol. 258, 2009, pp. 83-101.
15. Dudek-Dyduch, E., Tadeusiewicz, R., **Horzyk, A.**, *Neural Network Adaptation Process Effectiveness Dependent of Constant Training Data Availability*, **Neurocomputing** 72, 2009, pp. 3138-3149, **IF = 1,440**.

horzyk@agh.edu.pl