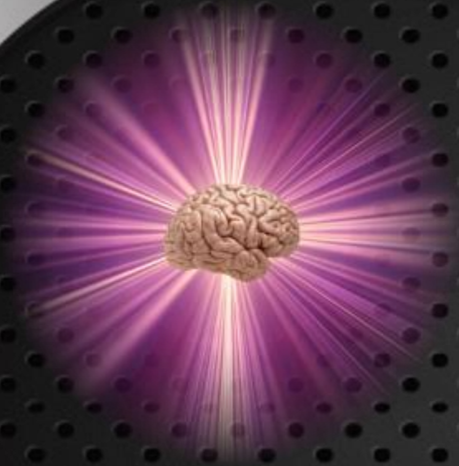
AGH University of Science and Technology
Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering
Department of Biocybernetics and Biomedical Engineering



Computational Intelligence

Associative and Cognitive Neural Systems

Adrian Horzyk
horzyk@agh.edu.pl

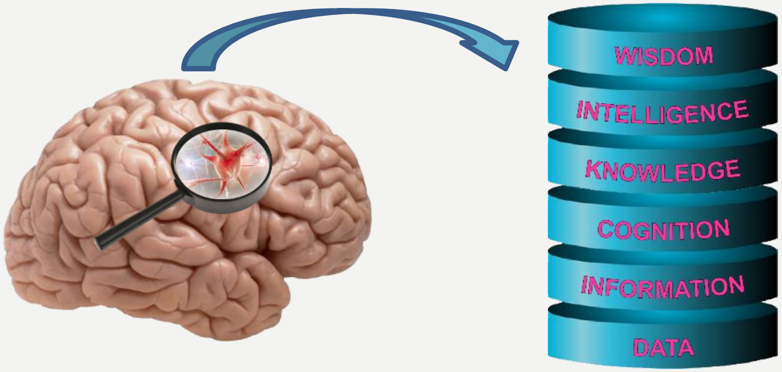
Google: Adrian Horzyk

Data and Relationship Representation in Contemporary Computer Systems

Do we store data and relationships efficiently?

Fundamental Questions

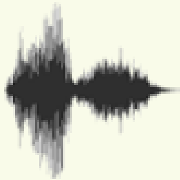
The development of Artificial Intelligence and Computational Intelligence depends on the ability to answer a few fundamental questions like:



- **What is it intelligence?**
- **How does intelligence work inside our brains?**
- **How is knowledge represented in our brains?**
- **How does intelligence use knowledge and how is knowledge extended by intelligence?**
- **How do we learn the world?**
- **How objects and their features and relationships are represented in our brains?**
- **How do biological neurons work and how do they connect?**
- **If biological brains have no hyperparameters, no designers, and no supervision in the learning process, how do they develop connections and adapt weights?**
- **What is undiscovered and unused in CI and AI systems?**
- **How can we use brain structures and processes to develop artificial intelligence and improve algorithms of computational intelligence?**

Data

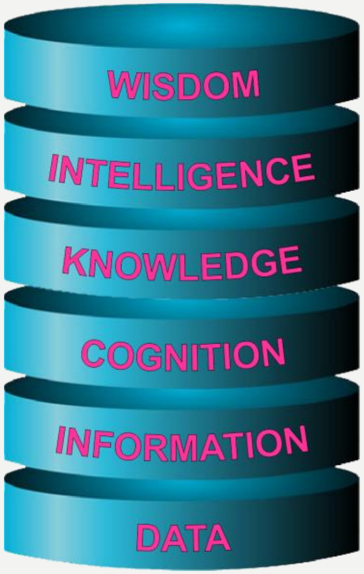
Data is a collections of numbers, signs, symbols, signals, stimuli, physical or empirical measures, and raw entities that describe various objects or actions, e.g.: 36.6ºC, T, $, φ, 25cm, !

爱 

Unrelated data are not useful because data take on the meaning when related.

Data might be raw, inconsistent, unorganized…

They usually describe facts and carries information.



Are Data Tables Convenient?

In computer science, we mostly use tables to store, organize, and manage data,

	ATTRIBUTES				
SAMPLE OBJECTS	SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH	CLASS LABEL
O1	5.4	3.0	4.5	1.5	Versicolor
O2	6.3	3.3	4.7	1.6	Versicolor
O3	6.0	2.7	5.1	1.6	Versicolor
O4	6.7	3.0	5.0	1.7	Versicolor
O5	6.0	2.2	5.0	1.5	Virginica
O6	5.9	3.2	4.8	1.8	Versicolor
O7	6.0	3.0	4.8	1.8	Virginica
O8	5.7	2.5	5.0	2.0	Virginica
O9	6.5	3.2	5.1	2.0	Virginica

but common vertical relations like identity, similarity, neighborhood, minima, maxima, number of duplicates must be searched for if desired or required.

The more data we have, a greater loss of time loss we get when we must **search for** such **relationships**!

Relationships

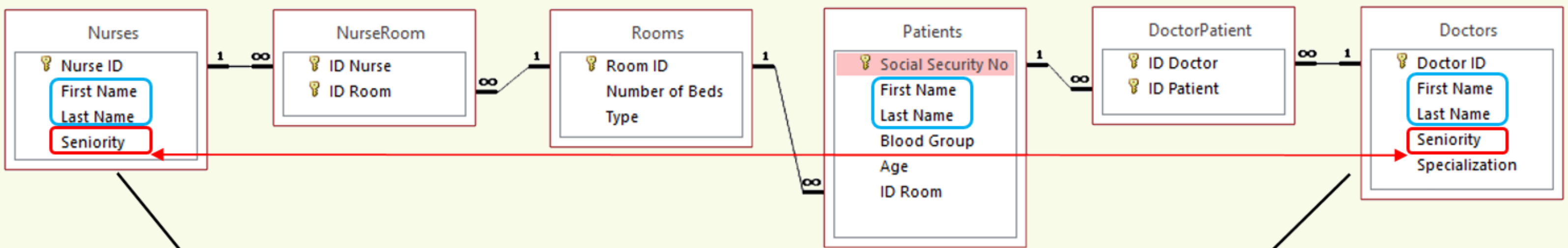
Relationships are various way
in which two or more people or things
(data and objects) are connected.

Relationship (relation) can typically
be of different strengths which usually
come from the frequency of occurrence.

Often **relationships** are a goal of
our reasoning or computations,
so in computer science, especially
in computational intelligence,
we search for various relationships,
e.g., belonging to the classes or
establishing how data are related to
allow desired predictions or
find similar groups of them.

Relational Databases

Relational databases relate stored data only horizontally, not vertically, so we still have to search for duplicates, neighbor or similar values and objects.



Nurses

Nurse ID	First Name	Last Name	Seniority
N1	Amy	Moon	12
N2	Rose	Jolie	18
N3	Kate	Ford	24
N4	Lisa	Brown	9
N5	Sara	Pitt	4
N6	Kate	Lopez	12

Doctors

Doctor ID	First Name	Last Name	Seniority	Specialization
D1	Tom	Hanks	18	orthopedics
D2	Jack	Brown	15	surgery
D3	Lisa	Ford	23	pediatrician
D4	Tom	Trump	35	pediatrician
D5	Kate	Smith	7	surgery
D6	Amy	Hanks	12	surgery

Data are not perfectly related even horizontally, and many duplicates of the same categories occur in various tables which are not related anyhow. In result, we need to lose a lot of resources and computational time to search for necessary data relationships to compute results or make conclusions.

Is it wise to lose the majority of the computational time for searching for data relationships?!

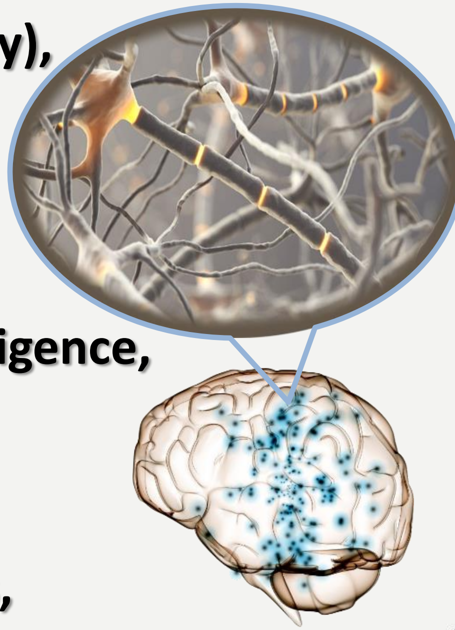
Biological Neurons and Brains

How biology deals with the data and relations?

Biological Neurons and Brains

Biological neurons and brains:

- do not only calculate weighted sums, while the synaptic and neuronal processes are much more complex and temporal,
- do not use non-linear activation functions as artificial neurons,
- do not use gradient descent algorithm to adapt weights,
- work in time parallelly (synchronously and asynchronously),
- process and store representation of data temporarily,
- do not use simple weights, thresholds and other trainable parameters used in computational intelligence,
- don’t use supervision in the sense of computational intelligence,
- automatically represent data and their relationships,
- automatically form knowledge on their basis,
- automatically generalize about the data and relationships,
- enable the development of our intelligence, which is still out of the reach of artificial intelligence and computational intelligence.

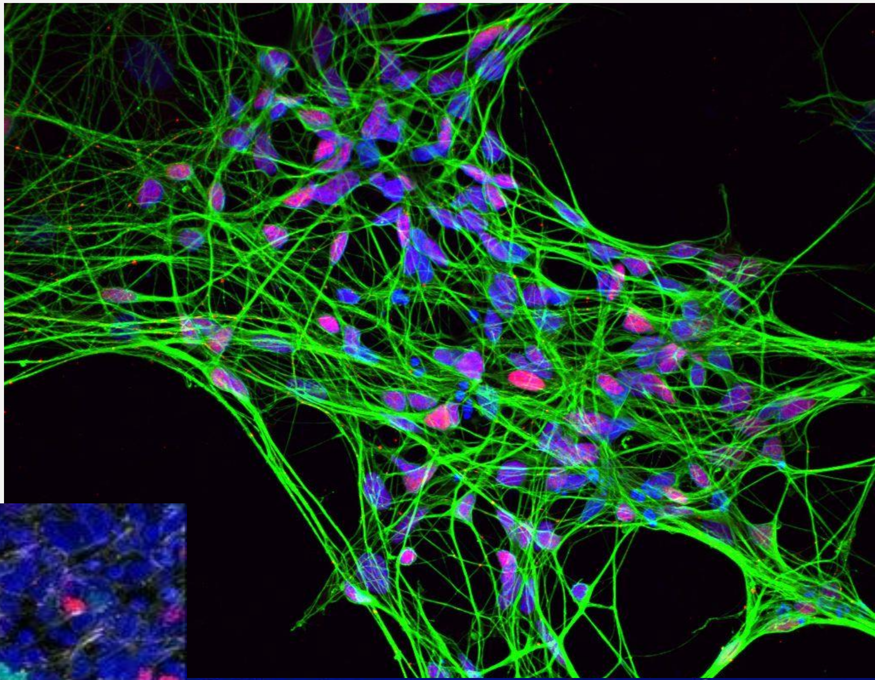


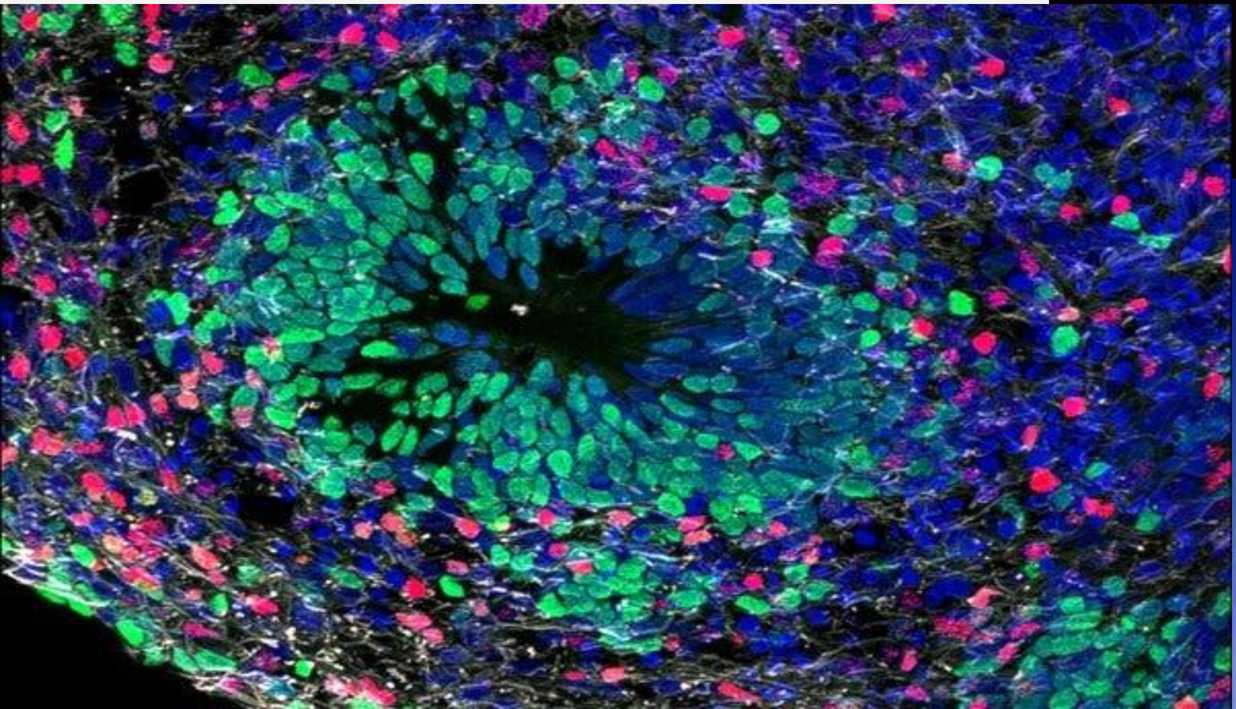
Biological Neural Structures

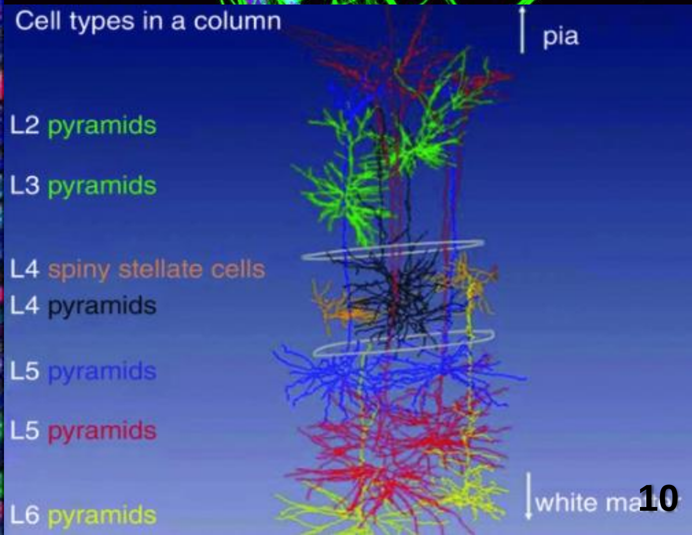
https://physoc.onlinelibrary.wiley.com/doi/10.1113/EP085776
https://www.axolbio.com/page/neural-progenitor-cells-and-neurons
https://www.nytimes.com/2021/02/11/science/neanderthal-brain-organoids.html

Biological neural structures:

- **are not so well structured as artificial (deep) neural networks were everything is put into layers of neurons of different kinds.**
- **are complex graph structures of neurons with sparse connection created dynamically during the training process in time.**





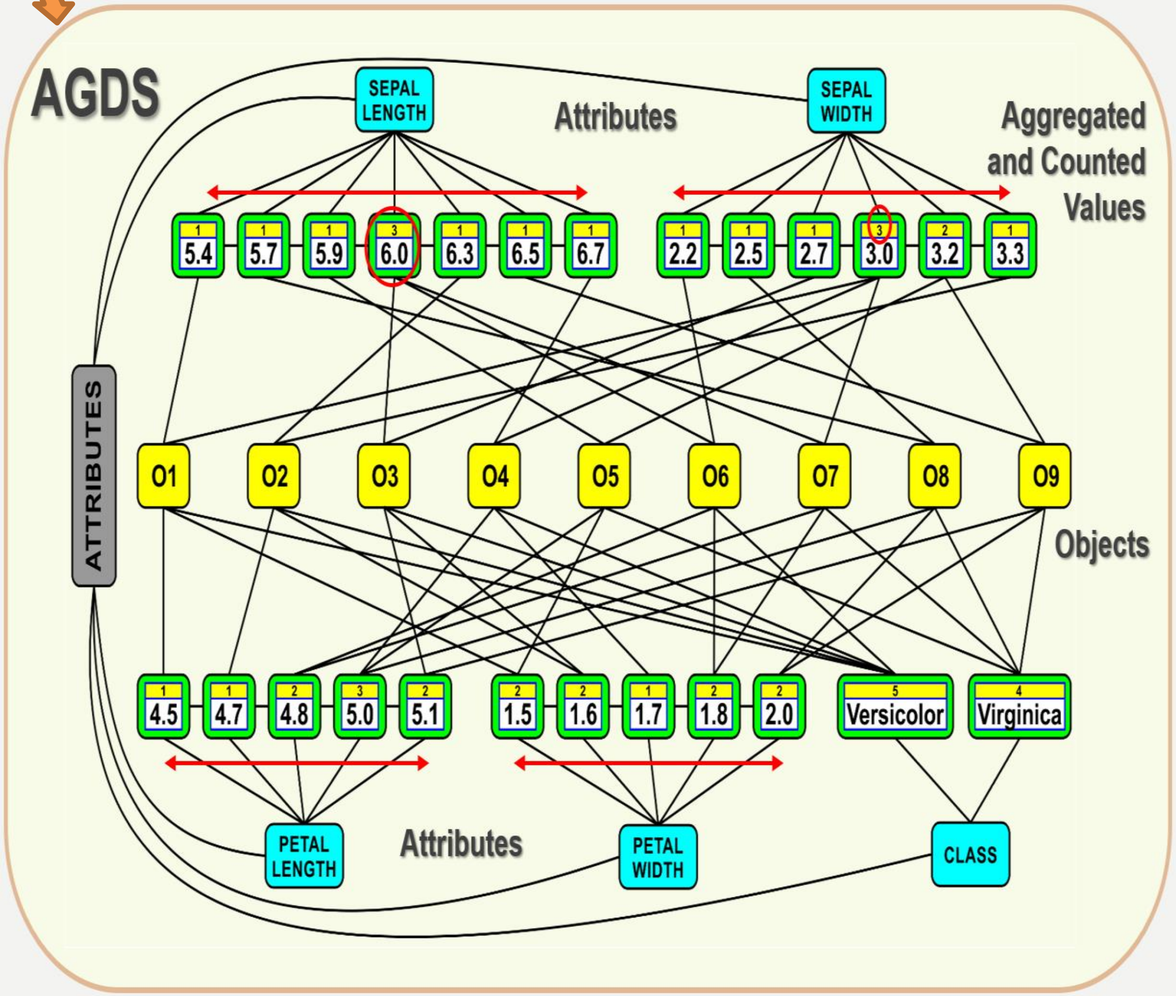


Associative Graph Data Structures (AGDS)

Can we create graph structures which similarly connect neurons and represent objects?

Associative Graph Data Structures (AGDS)

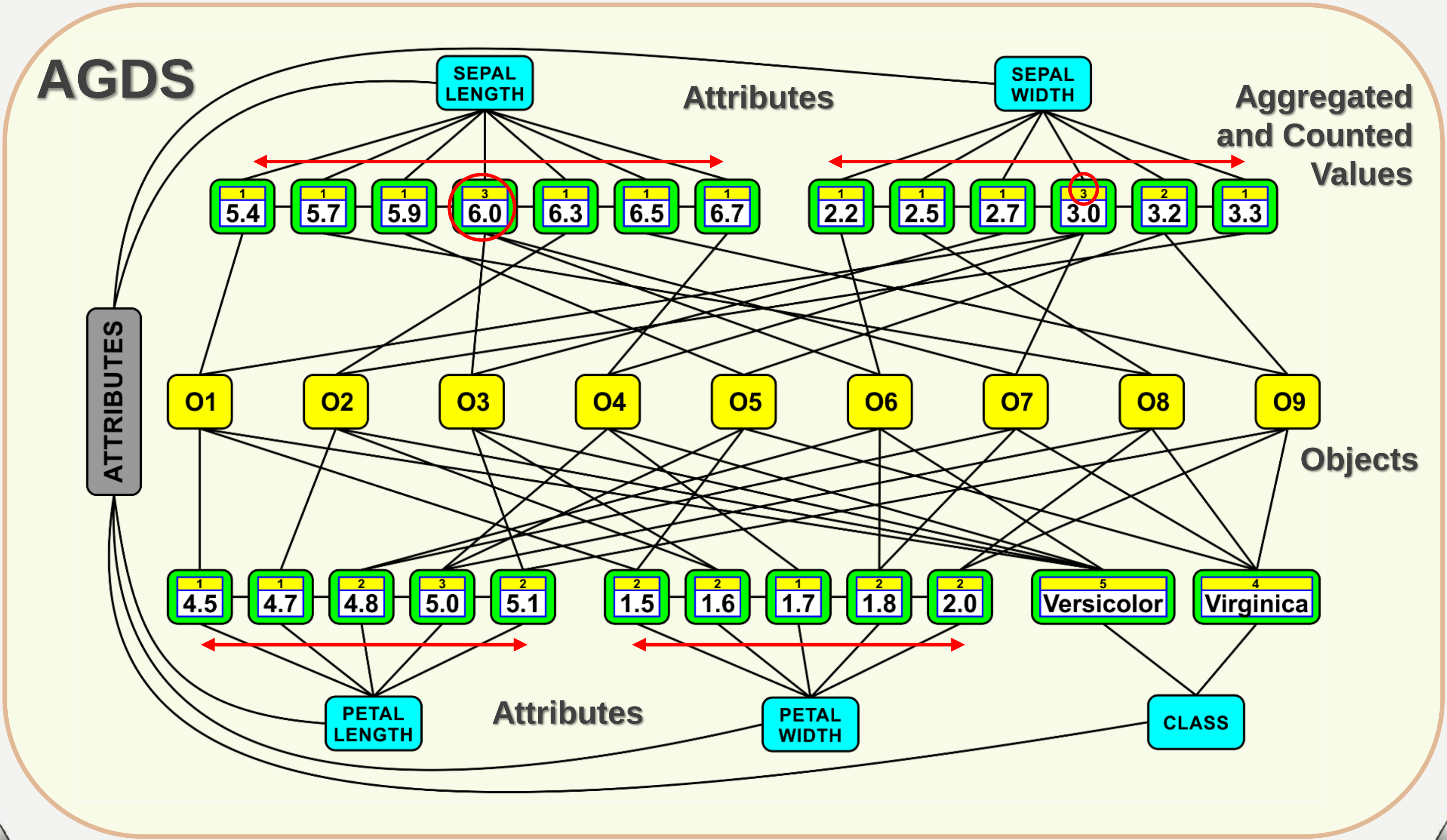
	ATTRIBUTES				
SAMPLE OBJECTS	SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH	CLASS LABEL
O1	5.4	3.0	4.5	1.5	Versicolor
O2	6.3	3.3	4.7	1.6	Versicolor
O3	6.0	2.7	5.1	1.6	Versicolor
O4	6.7	3.0	5.0	1.7	Versicolor
O5	6.0	2.2	5.0	1.5	Virginica
O6	5.9	3.2	4.8	1.8	Versicolor
O7	6.0	3.0	4.8	1.8	Virginica
O8	5.7	2.5	5.0	2.0	Virginica
O9	6.5	3.2	5.1	2.0	Virginica



AGDS allows us to represent any tabular data in the graph of connected nodes representing data, and connections representing relationships.

AGDS Features

Connections can represent various relations between AGDS elements like similarity, proximity, neighborhood, definition etc.



AGDS

Associative Graph Data Structures consist of:

- **Nodes** representing single-value data, ranges, subsets, objects, clusters, classes etc.
- **Edges** representing various relations between nodes like similarity, definition, sequence, neighborhood etc.

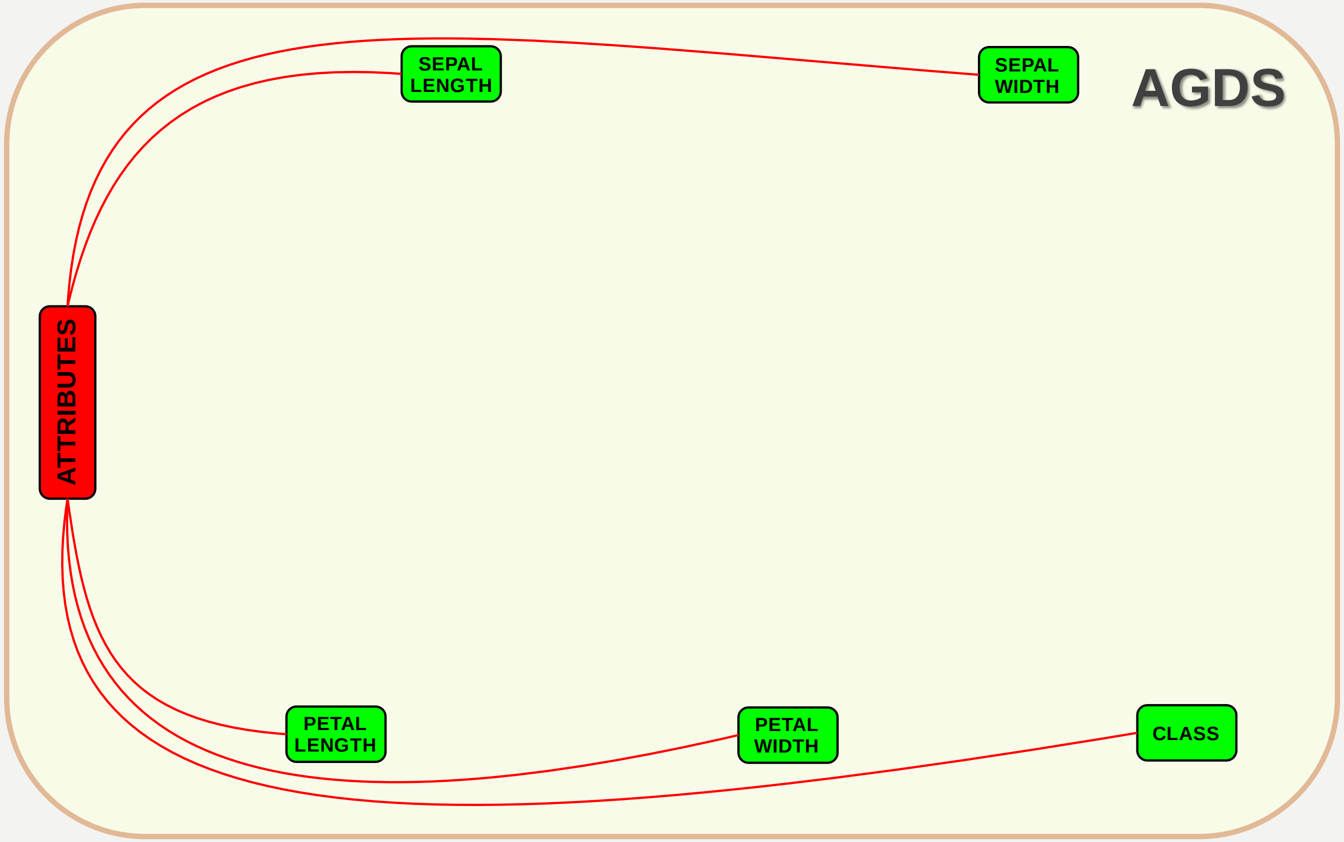
We can use it to represent any tabular data without any information loss, i.e. the transformation of tables into AGDS structure is reversible, so we can always transform the data back to the tabular structure.

This transformation enriches the set of directly represented relationships between data stored in the transformed tables in comparison to the directly represented relationships in tables.

They can be used for classification, clustering, predictions, search for the missing values, outliers search, recommendations, etc.

Associative Transformation of Table
AGDS Construction Process

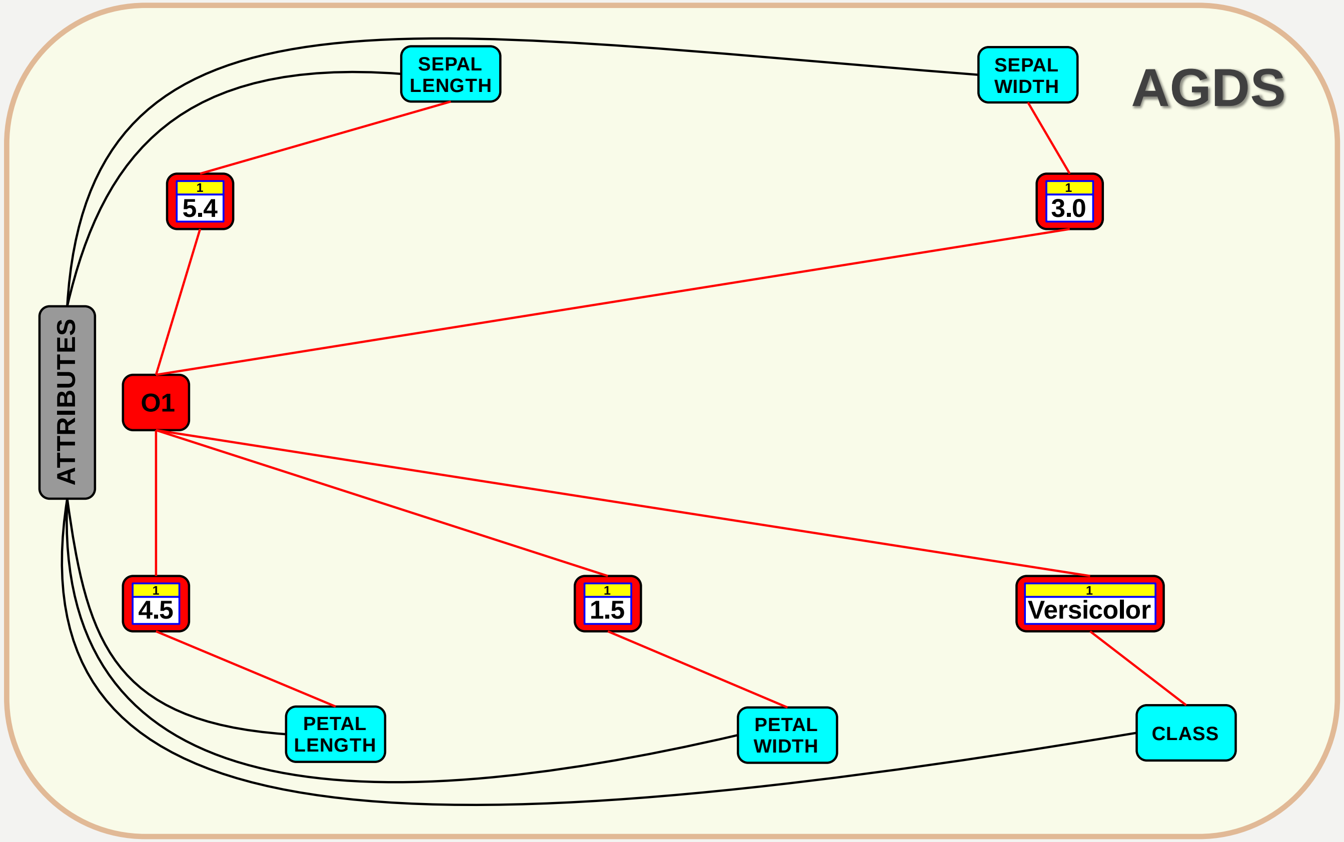
DATASET	ATTRIBUTES				
SAMPLE OBJECTS	SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH	CLASS LABEL
O1	5.4	3.0	4.5	1.5	Versicolor
O2	6.3	3.3	4.7	1.6	Versicolor
O3	6.0	2.7	5.1	1.6	Versicolor
O4	6.7	3.0	5.0	1.7	Versicolor
O5	6.0	2.2	5.0	1.5	Virginica
O6	5.9	3.2	4.8	1.8	Versicolor
O7	6.0	3.0	4.8	1.8	Virginica
O8	5.7	2.5	5.0	2.0	Virginica
O9	6.5	3.2	5.1	2.0	Virginica



The associative transformation process of a table into an AGDS structure starts from the creation of an attributes node and the nodes representing labels of the attributes. Labels of attributes will be linked to the unique attribute values that will be sorted and counted during the insertion of next values.

Associative Transformation of Table
AGDS Construction Process

DATASET	ATTRIBUTES				
SAMPLE OBJECTS	SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH	CLASS LABEL
O1	5.4	3.0	4.5	1.5	Versicolor
O2	6.3	3.3	4.7	1.6	Versicolor
O3	6.0	2.7	5.1	1.6	Versicolor
O4	6.7	3.0	5.0	1.7	Versicolor
O5	6.0	2.2	5.0	1.5	Virginica
O6	5.9	3.2	4.8	1.8	Versicolor
O7	6.0	3.0	4.8	1.8	Virginica
O8	5.7	2.5	5.0	2.0	Virginica
O9	6.5	3.2	5.1	2.0	Virginica

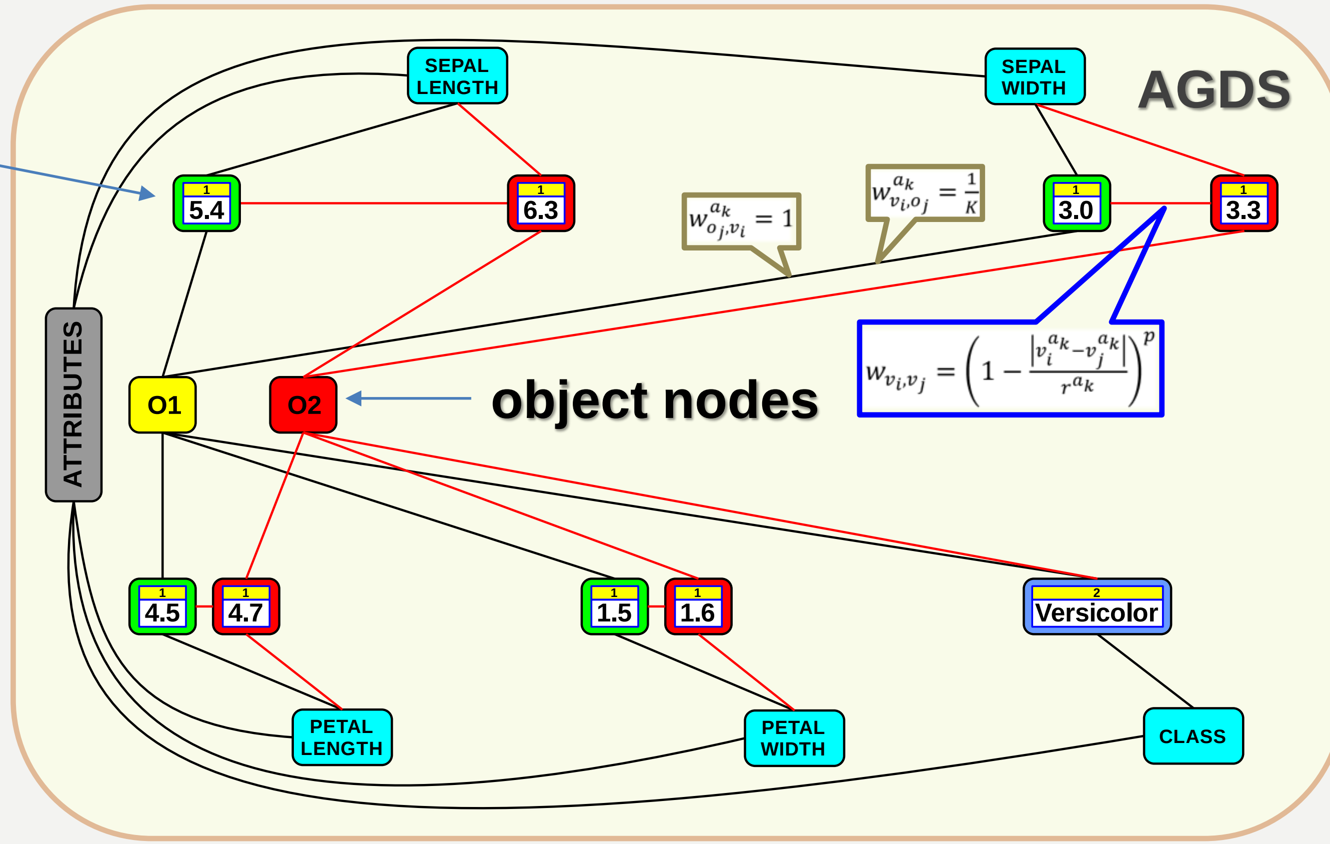


To the previously created backbone structure, the first object (record, entity) O1 is added together with all defining features.

The features and the object are connected mutually and to the label nodes of the attributes.

Associative Transformation of Table
AGDS Construction Process

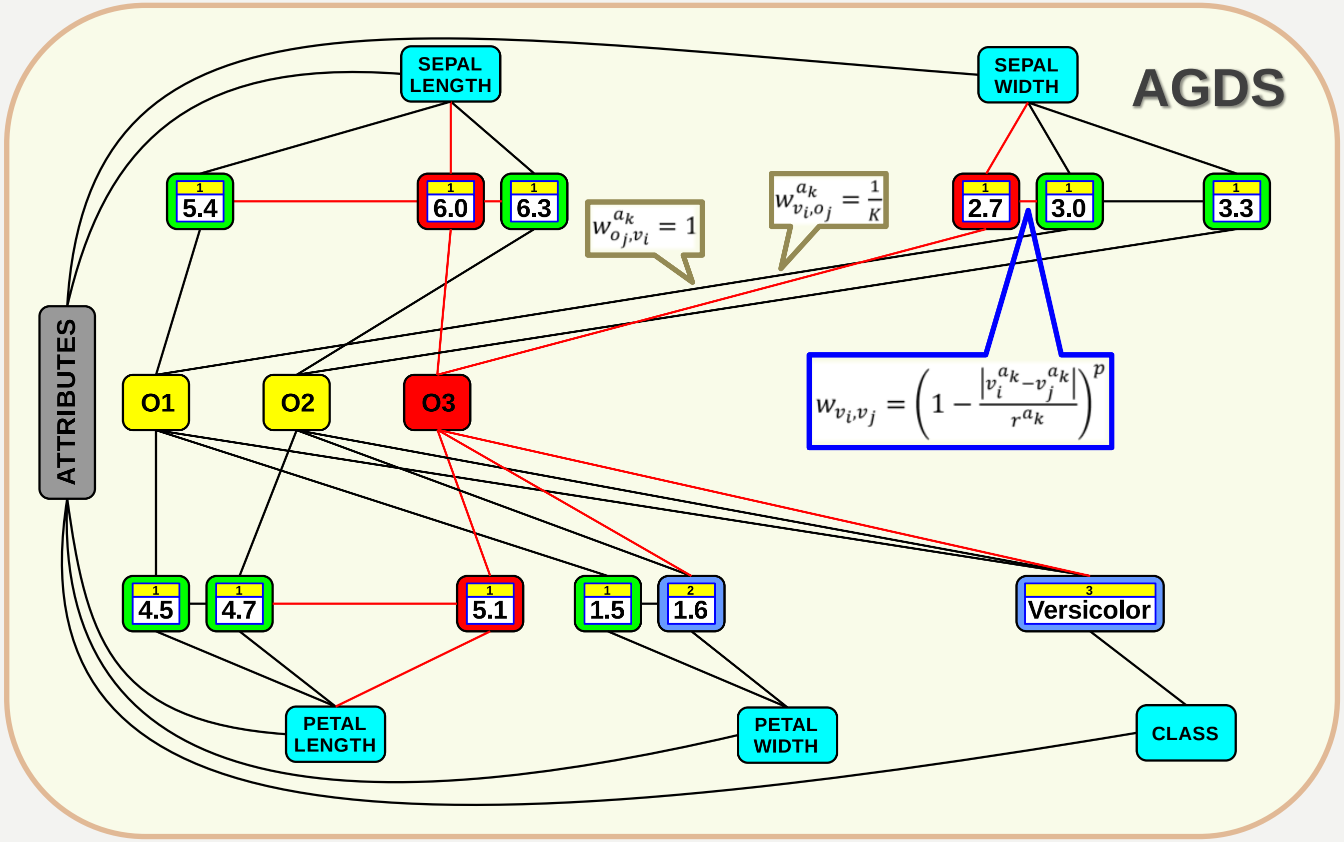
DATASET	ATTRIBUTES				
SAMPLE OBJECTS	SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH	CLASS LABEL
O1	5.4	3.0	4.5	1.5	Versicolor
O2	6.3	3.3	4.7	1.6	Versicolor
O3	6.0	2.7	5.1	1.6	Versicolor
O4	6.7	3.0	5.0	1.7	Versicolor
O5	6.0	2.2	5.0	1.5	Virginica
O6	5.9	3.2	4.8	1.8	Versicolor
O7	6.0	3.0	4.8	1.8	Virginica
O8	5.7	2.5	5.0	2.0	Virginica
O9	6.5	3.2	5.1	2.0	Virginica



The second object is added to the AGDS structure and all its defining features are represented by values nodes that are connected to attribute labels, this new object, and neighbor values nodes which were already in this structure.

Associative Transformation of Table
AGDS Construction Process

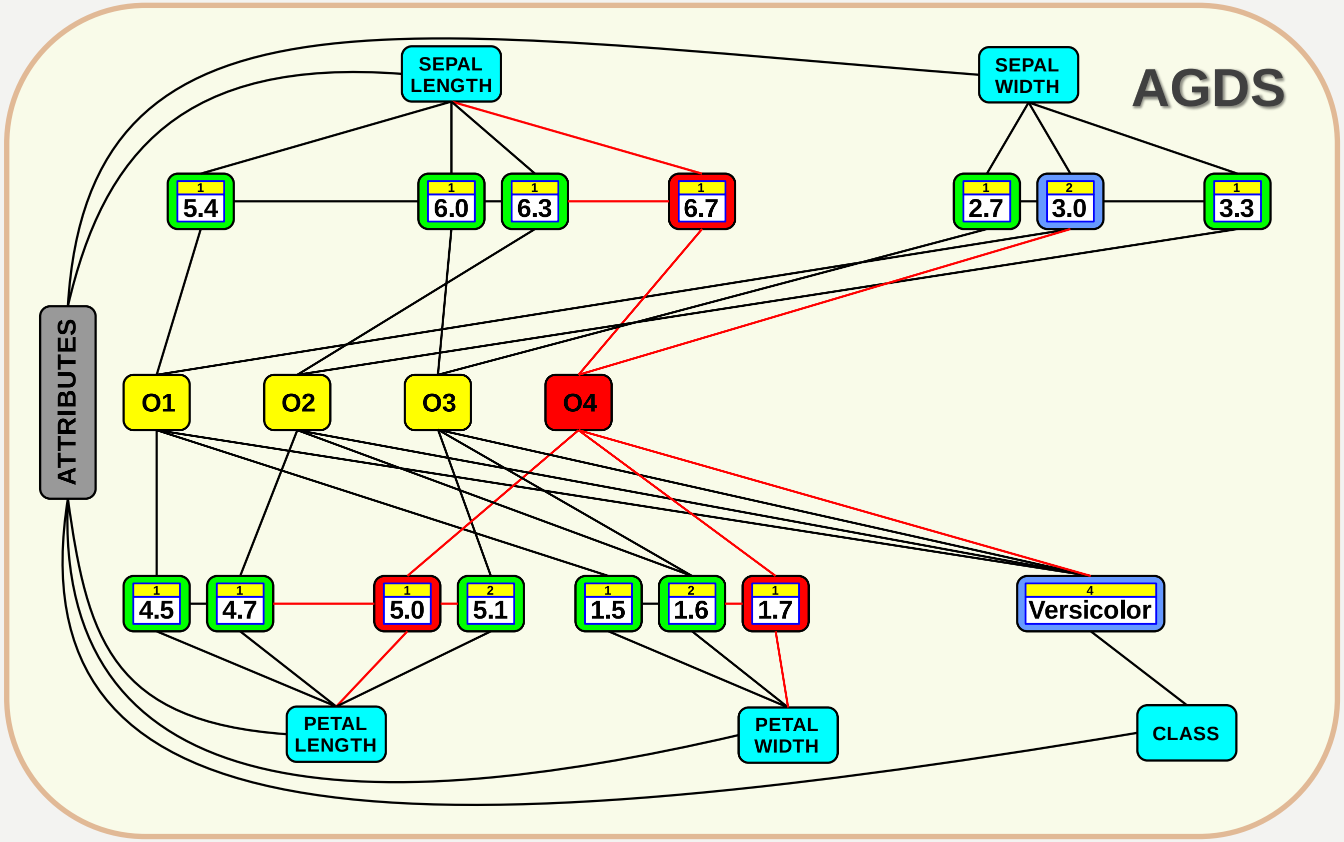
DATASET	ATTRIBUTES				
SAMPLE OBJECTS	SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH	CLASS LABEL
O1	5.4	3.0	4.5	1.5	Versicolor
O2	6.3	3.3	4.7	1.6	Versicolor
O3	6.0	2.7	5.1	1.6	Versicolor
O4	6.7	3.0	5.0	1.7	Versicolor
O5	6.0	2.2	5.0	1.5	Virginica
O6	5.9	3.2	4.8	1.8	Versicolor
O7	6.0	3.0	4.8	1.8	Virginica
O8	5.7	2.5	5.0	2.0	Virginica
O9	6.5	3.2	5.1	2.0	Virginica



During the addition of the next object, we can notice that not all defining features have created new values nodes (e.g. 1.6 of the petal width or Versicolor of a class label) because some values had been already represented in this structure, so the duplicates (in blue) have been aggregated and counted.

Associative Transformation of Table
AGDS Construction Process

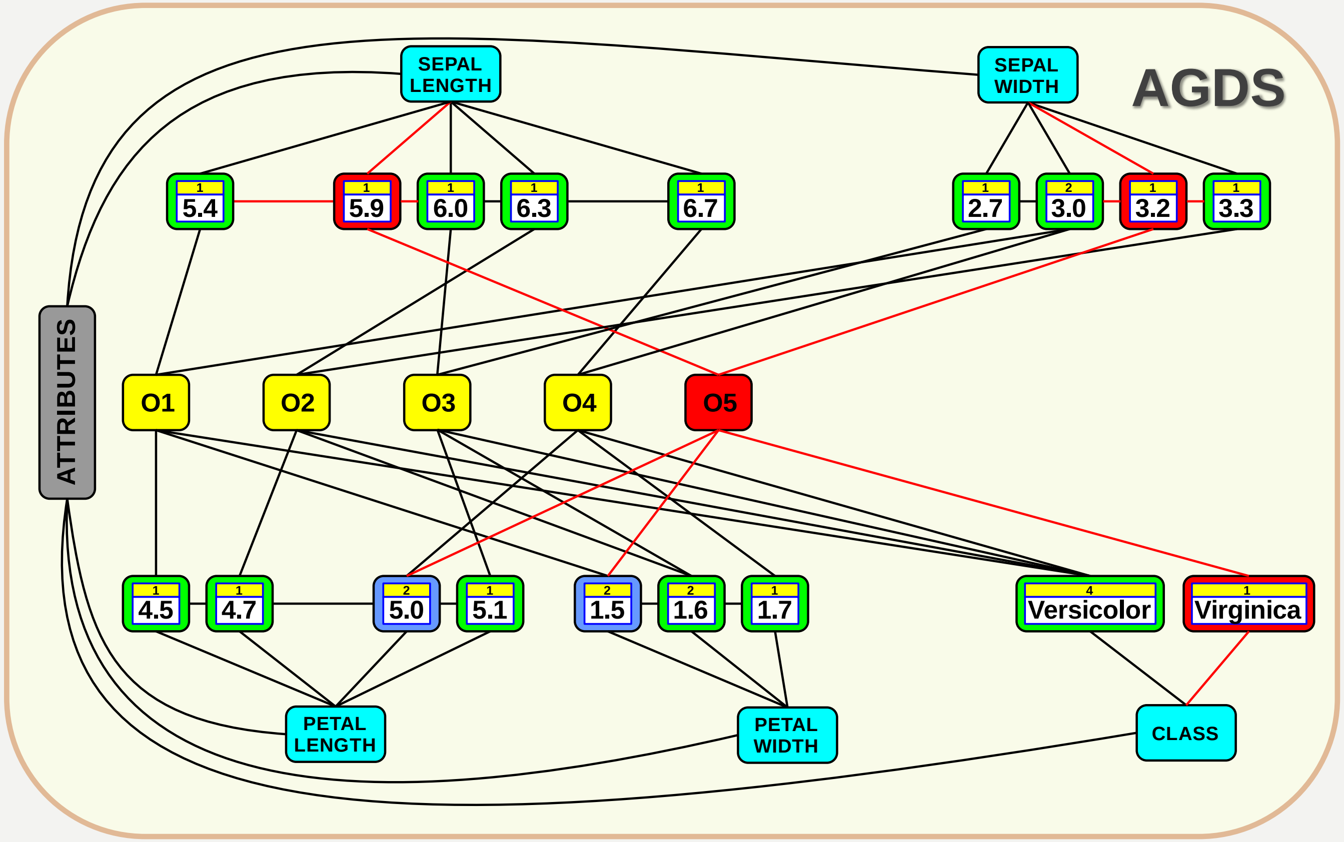
DATASET	ATTRIBUTES				
SAMPLE OBJECTS	SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH	CLASS LABEL
O1	5.4	3.0	4.5	1.5	Versicolor
O2	6.3	3.3	4.7	1.6	Versicolor
O3	6.0	2.7	5.1	1.6	Versicolor
O4	6.7	3.0	5.0	1.7	Versicolor
O5	6.0	2.2	5.0	1.5	Virginica
O6	5.9	3.2	4.8	1.8	Versicolor
O7	6.0	3.0	4.8	1.8	Virginica
O8	5.7	2.5	5.0	2.0	Virginica
O9	6.5	3.2	5.1	2.0	Virginica



The following object creates some new values nodes and uses two of the existing values nodes, incrementing their counters of aggregated duplicates. The aggregation process of duplicates is very important from the knowledge representation point of view because it allows to draw deeper conclusions.

Associative Transformation of Table AGDS Construction Process

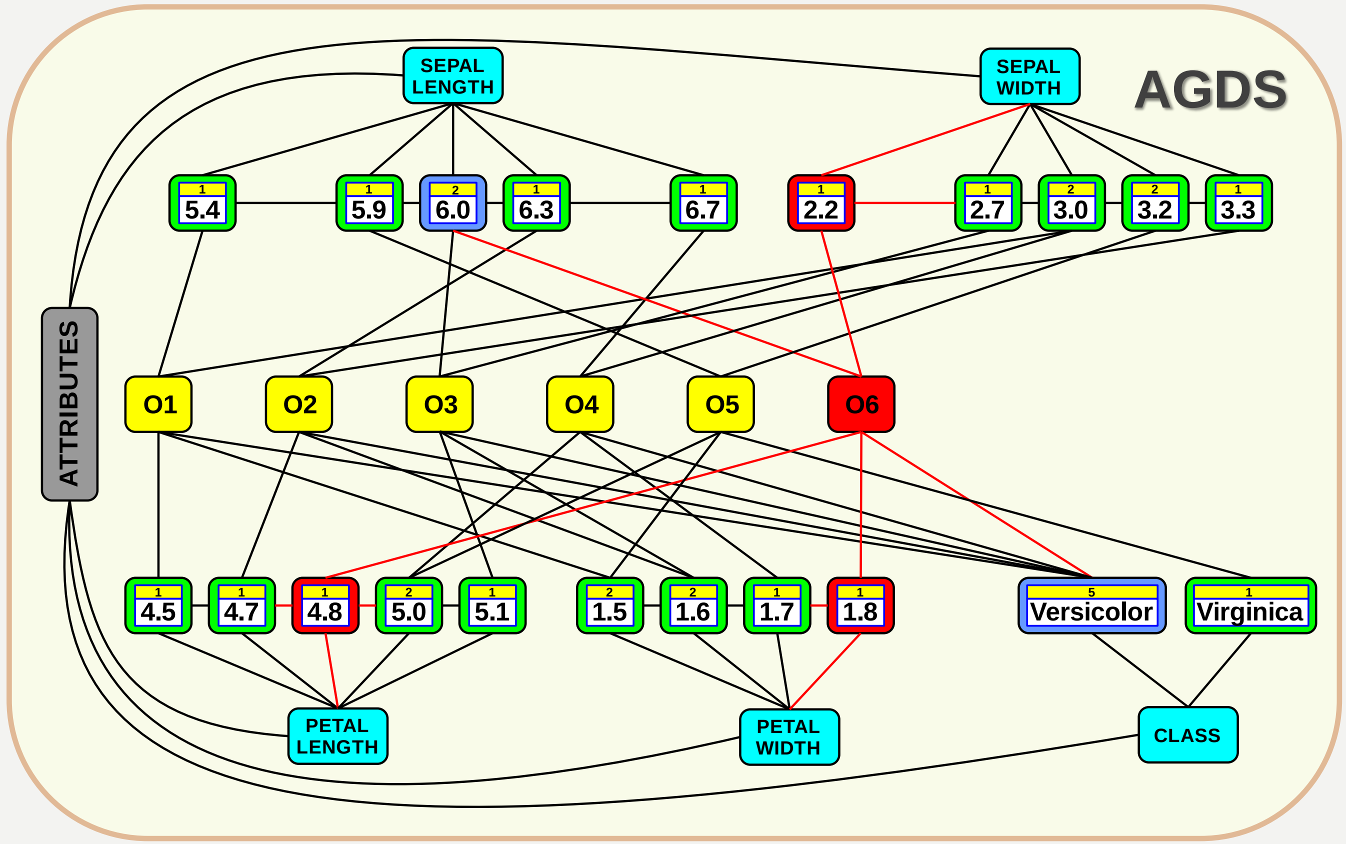
DATASET	ATTRIBUTES				
SAMPLE OBJECTS	SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH	CLASS LABEL
O1	5.4	3.0	4.5	1.5	Versicolor
O2	6.3	3.3	4.7	1.6	Versicolor
O3	6.0	2.7	5.1	1.6	Versicolor
O4	6.7	3.0	5.0	1.7	Versicolor
O5	6.0	2.2	5.0	1.5	Virginica
O6	5.9	3.2	4.8	1.8	Versicolor
O7	6.0	3.0	4.8	1.8	Virginica
O8	5.7	2.5	5.0	2.0	Virginica
O9	6.5	3.2	5.1	2.0	Virginica



Object O5 represents a different (new) class Virginica, so a new node representing this class has been added. Notice, that symbolic (non-numerical) values are not connected as numerical features that are always connected to their neighbors and the connections are weighted.

Associative Transformation of Table
AGDS Construction Process

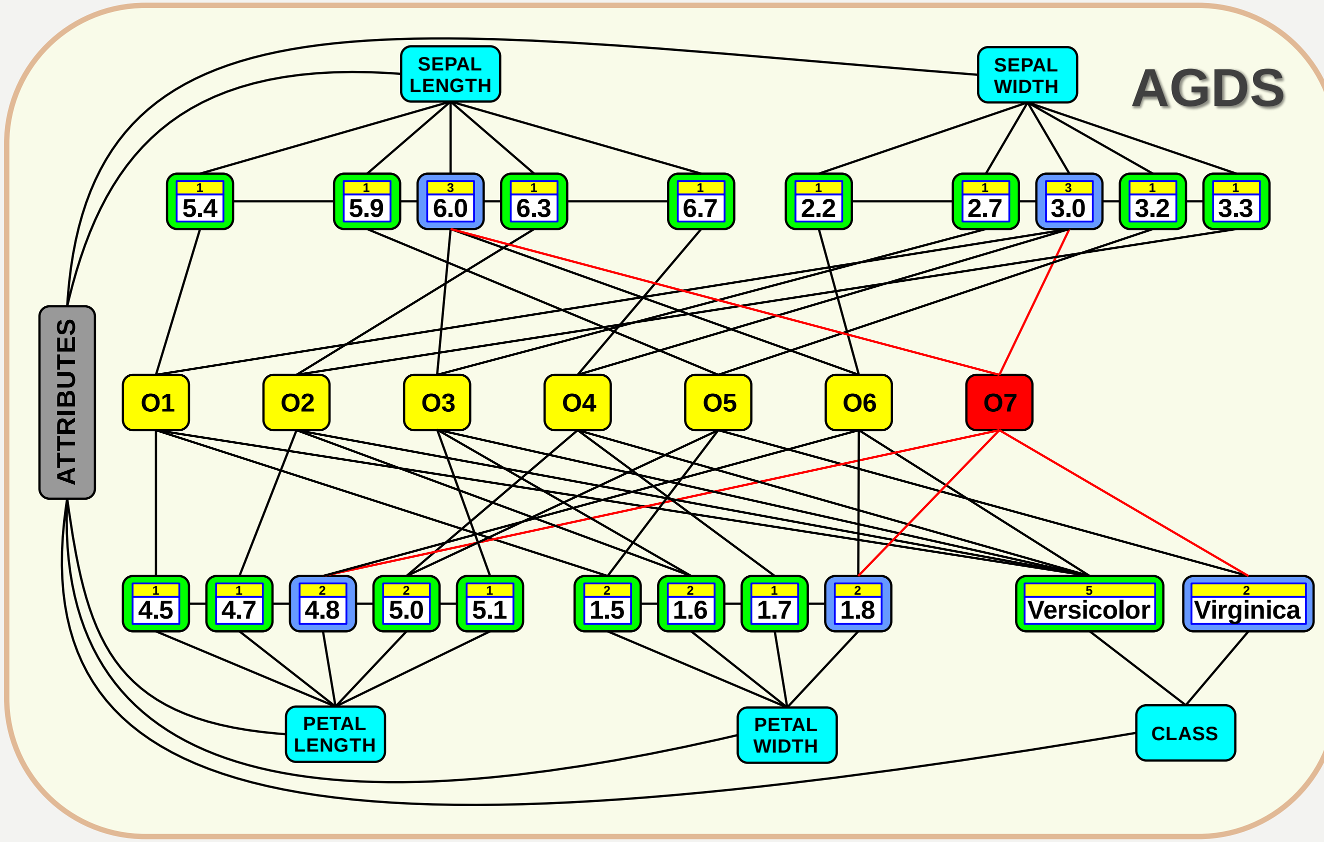
DATASET	ATTRIBUTES				
SAMPLE OBJECTS	SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH	CLASS LABEL
O1	5.4	3.0	4.5	1.5	Versicolor
O2	6.3	3.3	4.7	1.6	Versicolor
O3	6.0	2.7	5.1	1.6	Versicolor
O4	6.7	3.0	5.0	1.7	Versicolor
O5	6.0	2.2	5.0	1.5	Virginica
O6	5.9	3.2	4.8	1.8	Versicolor
O7	6.0	3.0	4.8	1.8	Virginica
O8	5.7	2.5	5.0	2.0	Virginica
O9	6.5	3.2	5.1	2.0	Virginica



The more objects we add to this structure, the less number of new values nodes are added when the transformed table (dataset) contains duplicates.
All object nodes connected to the mutually connected values nodes to other object nodes automatically create indirect associations between such objects.

Associative Transformation of Table
AGDS Construction Process

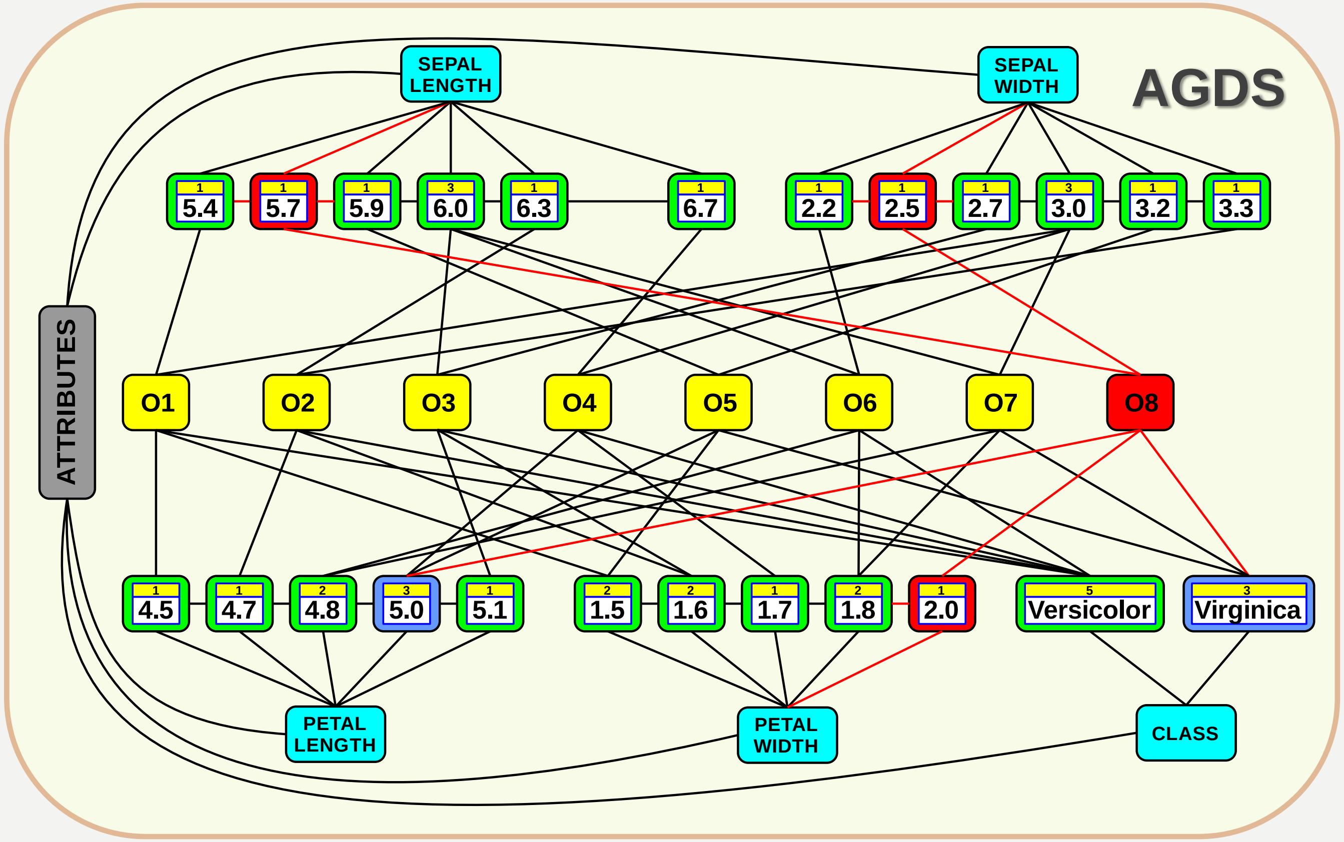
DATASET	ATTRIBUTES				
SAMPLE OBJECTS	SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH	CLASS LABEL
O1	5.4	3.0	4.5	1.5	Versicolor
O2	6.3	3.3	4.7	1.6	Versicolor
O3	6.0	2.7	5.1	1.6	Versicolor
O4	6.7	3.0	5.0	1.7	Versicolor
O5	6.0	2.2	5.0	1.5	Virginica
O6	5.9	3.2	4.8	1.8	Versicolor
O7	6.0	3.0	4.8	1.8	Virginica
O8	5.7	2.5	5.0	2.0	Virginica
O9	6.5	3.2	5.1	2.0	Virginica



In this case, object O7 is added without addition of any new values nodes because all of them have been already added to this structure, so only new connections to the existing nodes are added, and their counters of represented duplicates are incremented. It saves memory when there are many duplicates!

Associative Transformation of Table
AGDS Construction Process

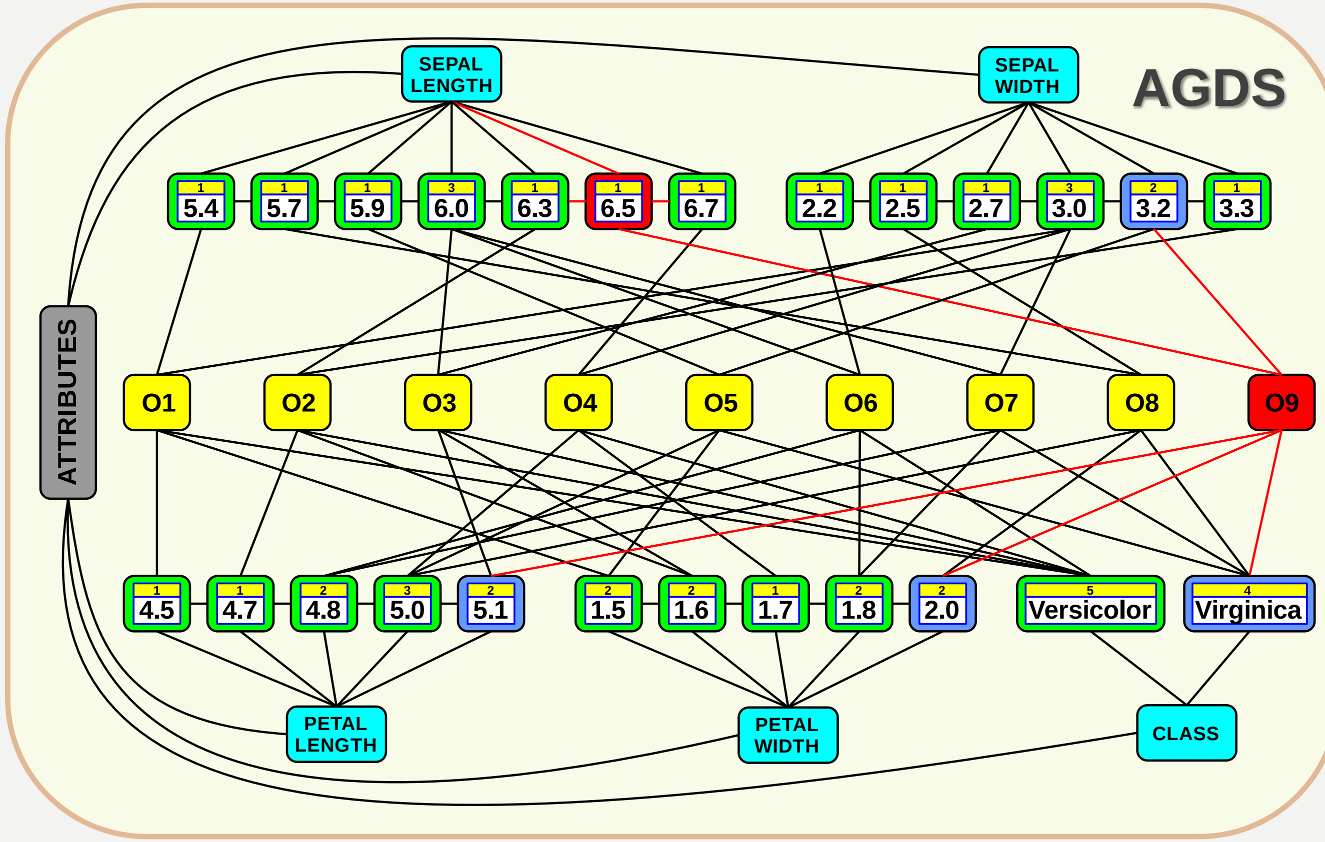
DATASET	ATTRIBUTES				
SAMPLE OBJECTS	SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH	CLASS LABEL
O1	5.4	3.0	4.5	1.5	Versicolor
O2	6.3	3.3	4.7	1.6	Versicolor
O3	6.0	2.7	5.1	1.6	Versicolor
O4	6.7	3.0	5.0	1.7	Versicolor
O5	6.0	2.2	5.0	1.5	Virginica
O6	5.9	3.2	4.8	1.8	Versicolor
O7	6.0	3.0	4.8	1.8	Virginica
O8	5.7	2.5	5.0	2.0	Virginica
O9	6.5	3.2	5.1	2.0	Virginica



Object O8 is also connected to the values node 5.0 which now defines three objects (O4, O5 and O8), so there is a visible similarity between these objects. The similarity between objects O5 and O8 is bigger than between O4 and O8 because there is another shared feature (Virginica) between the first pair!

Associative Transformation of Table
AGDS Construction Process

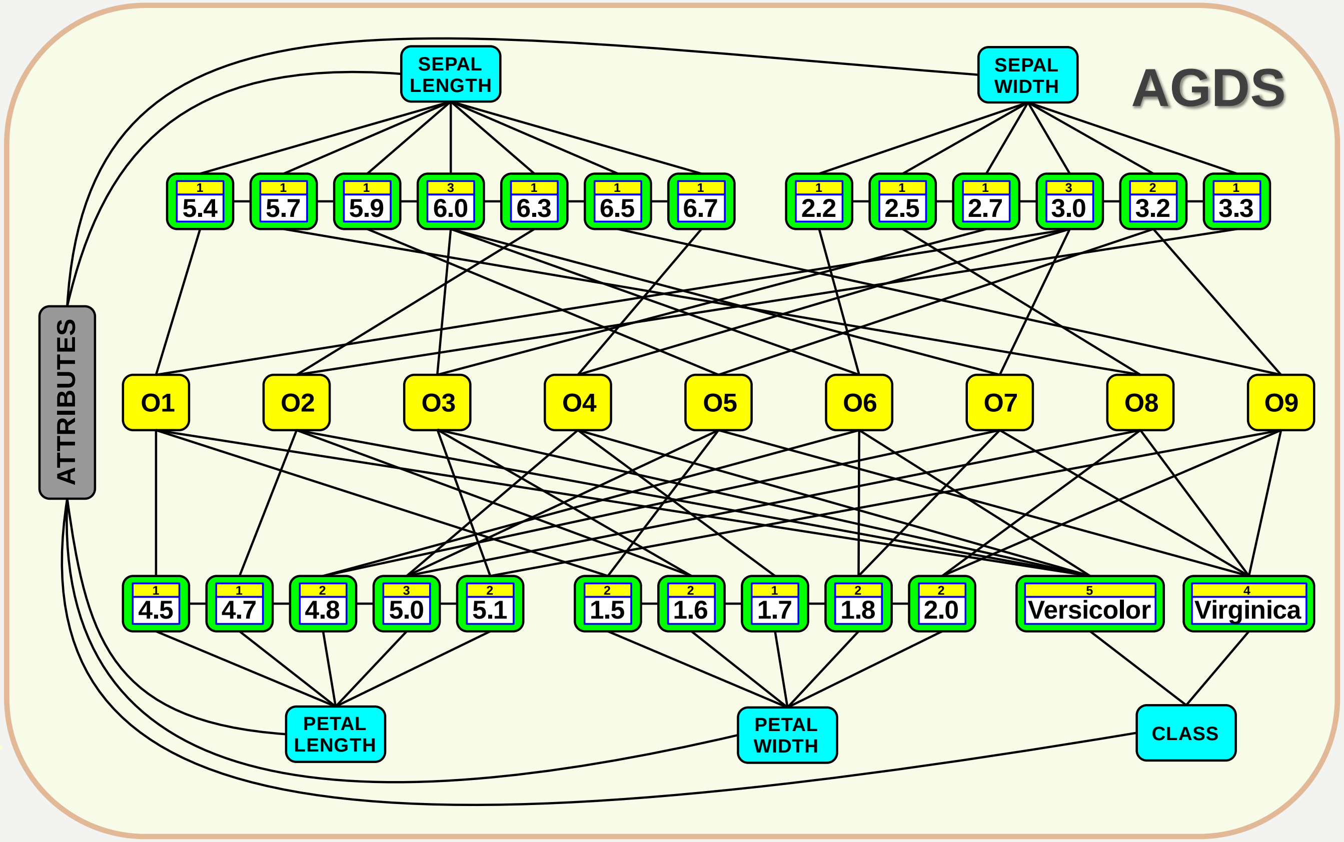
DATASET	ATTRIBUTES				
SAMPLE OBJECTS	SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH	CLASS LABEL
O1	5.4	3.0	4.5	1.5	Versicolor
O2	6.3	3.3	4.7	1.6	Versicolor
O3	6.0	2.7	5.1	1.6	Versicolor
O4	6.7	3.0	5.0	1.7	Versicolor
O5	6.0	2.2	5.0	1.5	Virginica
O6	5.9	3.2	4.8	1.8	Versicolor
O7	6.0	3.0	4.8	1.8	Virginica
O8	5.7	2.5	5.0	2.0	Virginica
O9	6.5	3.2	5.1	2.0	Virginica



Object O9 has added only one new feature to this structure because the other feature values had been already represented.
Now, the transformation process for this small table is already finished,
and we can try to compare these structures and take advantages of this graph!

Associative Transformation of Table AGDS Construction Process

DATASET	ATTRIBUTES				
SAMPLE OBJECTS	SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH	CLASS LABEL
O1	5.4	3.0	4.5	1.5	Versicolor
O2	6.3	3.3	4.7	1.6	Versicolor
O3	6.0	2.7	5.1	1.6	Versicolor
O4	6.7	3.0	5.0	1.7	Versicolor
O5	6.0	2.2	5.0	1.5	Virginica
O6	5.9	3.2	4.8	1.8	Versicolor
O7	6.0	3.0	4.8	1.8	Virginica
O8	5.7	2.5	5.0	2.0	Virginica
O9	6.5	3.2	5.1	2.0	Virginica



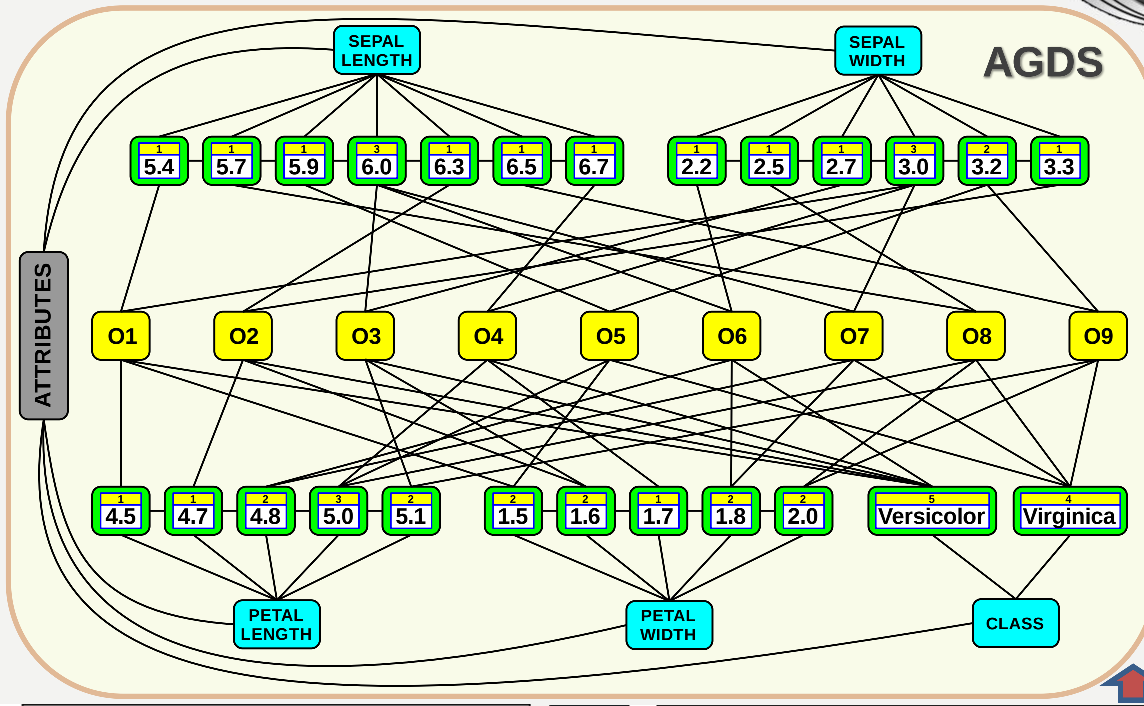
Which structure of the two presented do you like more?

The tabular structure represents data and very basic relations between them.

The AGDS structure additionally represents neighborhood, order, similarity, minima, maxima, counts of duplicates, number of unique values, and ranges of all features. We will not lose time for searching for such relationships!

Associative Transformation of Table
AGDS Construction Process

DATASET	ATTRIBUTES				
SAMPLE OBJECTS	SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH	CLASS LABEL
O1	5.4	3.0	4.5	1.5	Versicolor
O2	6.3	3.3	4.7	1.6	Versicolor
O3	6.0	2.7	5.1	1.6	Versicolor
O4	6.7	3.0	5.0	1.7	Versicolor
O5	6.0	2.2	5.0	1.5	Virginica
O6	5.9	3.2	4.8	1.8	Versicolor
O7	6.0	3.0	4.8	1.8	Virginica
O8	5.7	2.5	5.0	2.0	Virginica
O9	6.5	3.2	5.1	2.0	Virginica



CAT OFF AND SEPARATE DATA FOR EACH ATTRIBUTE SEPARATELY

DATASET	ATTRIBUTES				
SAMPLE OBJECTS	SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH	CLASS LABEL
O1	5.4	3.0	4.5	1.5	Versicolor
O2	6.3	3.3	4.7	1.6	Versicolor
O3	6.0	2.7	5.1	1.6	Versicolor
O4	6.7	3.0	5.0	1.7	Versicolor
O5	6.0	2.2	5.0	1.5	Virginica
O6	5.9	3.2	4.8	1.8	Versicolor
O7	6.0	3.0	4.8	1.8	Virginica
O8	5.7	2.5	5.0	2.0	Virginica
O9	6.5	3.2	5.1	2.0	Virginica

SORT DATA FOR EACH ATTRIBUTE SEPARATELY

DATASET	ATTRIBUTES				
SAMPLE OBJECTS	SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH	CLASS LABEL
O1	5.4	2.2	4.5	1.5	Versicolor
O2	5.7	2.5	4.7	1.5	Versicolor
O3	5.9	2.7	4.8	1.6	Versicolor
O4	6.0	3.0	4.8	1.6	Versicolor
O5	6.0	3.0	5.0	1.7	Versicolor
O6	6.0	3.0	5.0	1.8	Virginica
O7	6.3	3.2	5.0	1.8	Virginica
O8	6.5	3.2	5.1	2.0	Virginica
O9	6.7	3.3	5.1	2.0	Virginica

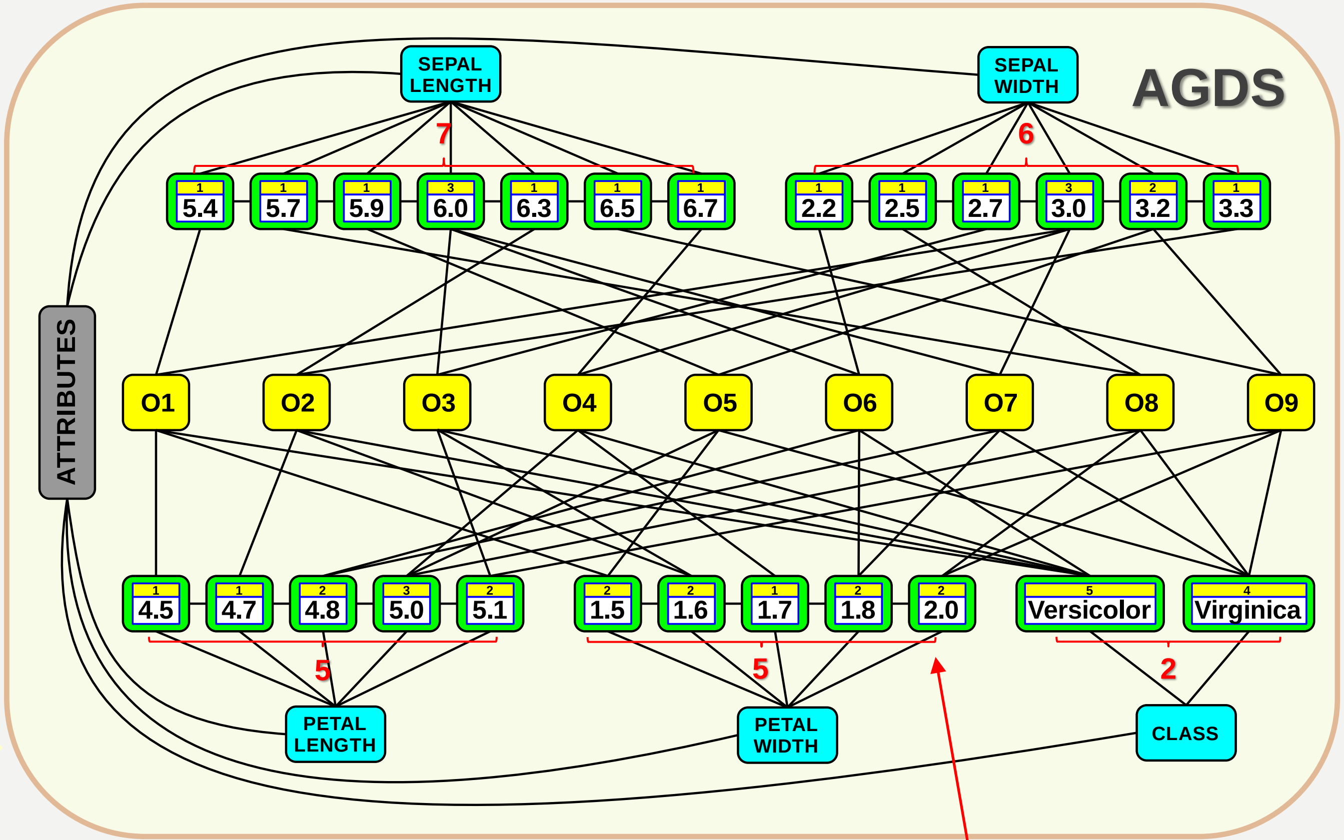
REMOVE DUPLICATES OF ALL ATTRIBUTES SEPARATELY

DATASET	ATTRIBUTES				
SAMPLE OBJECTS	SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH	CLASS LABEL
O1	5.4	2.2	4.5	1.5	Versicolor
O2	5.7	2.5	4.7	1.6	Virginica
O3	5.9	2.7	4.8	1.7	
O4	6.0	3.0	5.0	1.8	
O5	6.3	3.2	5.1	2.0	
O6	6.5	3.3			
O7	6.7				
O8					
O9					

We can create this structure in an alternative way when the dataset (table) is static and does not change in time (no records are added, removed or updated).

Efficiency of Data Access

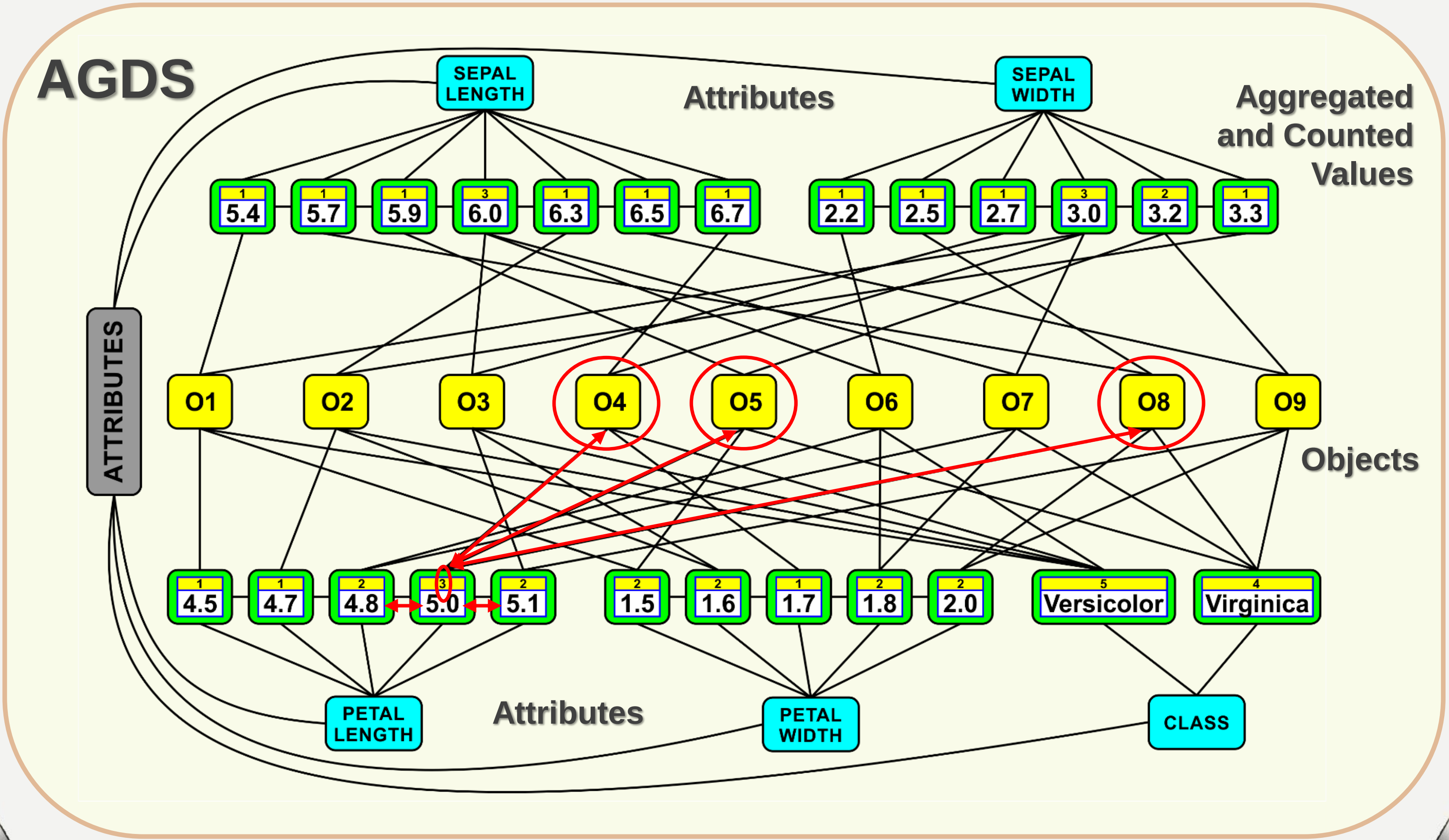
DATASET	ATTRIBUTES				
SAMPLE OBJECTS	SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH	CLASS LABEL
O1	5.4	3.0	4.5	1.5	Versicolor
O2	6.3	3.3	4.7	1.6	Versicolor
O3	6.0	2.7	5.1	1.6	Versicolor
O4	6.7	3.0	5.0	1.7	Versicolor
O5 9	6.0	2.2	5.0	1.5	Virginica
O6	5.9	3.2	4.8	1.8	Versicolor
O7	6.0	3.0	4.8	1.8	Virginica
O8	5.7	2.5	5.0	2.0	Virginica
O9	6.5	3.2	5.1	2.0	Virginica



Features of each attribute can be organized using: sorted tables, sorted lists, hash tables or AVB+trees to provide quick access to them!
Notice, that the number of unique features for each attribute is always less or equal to the number of all features in the dataset (table).

Direct Relationship Representation

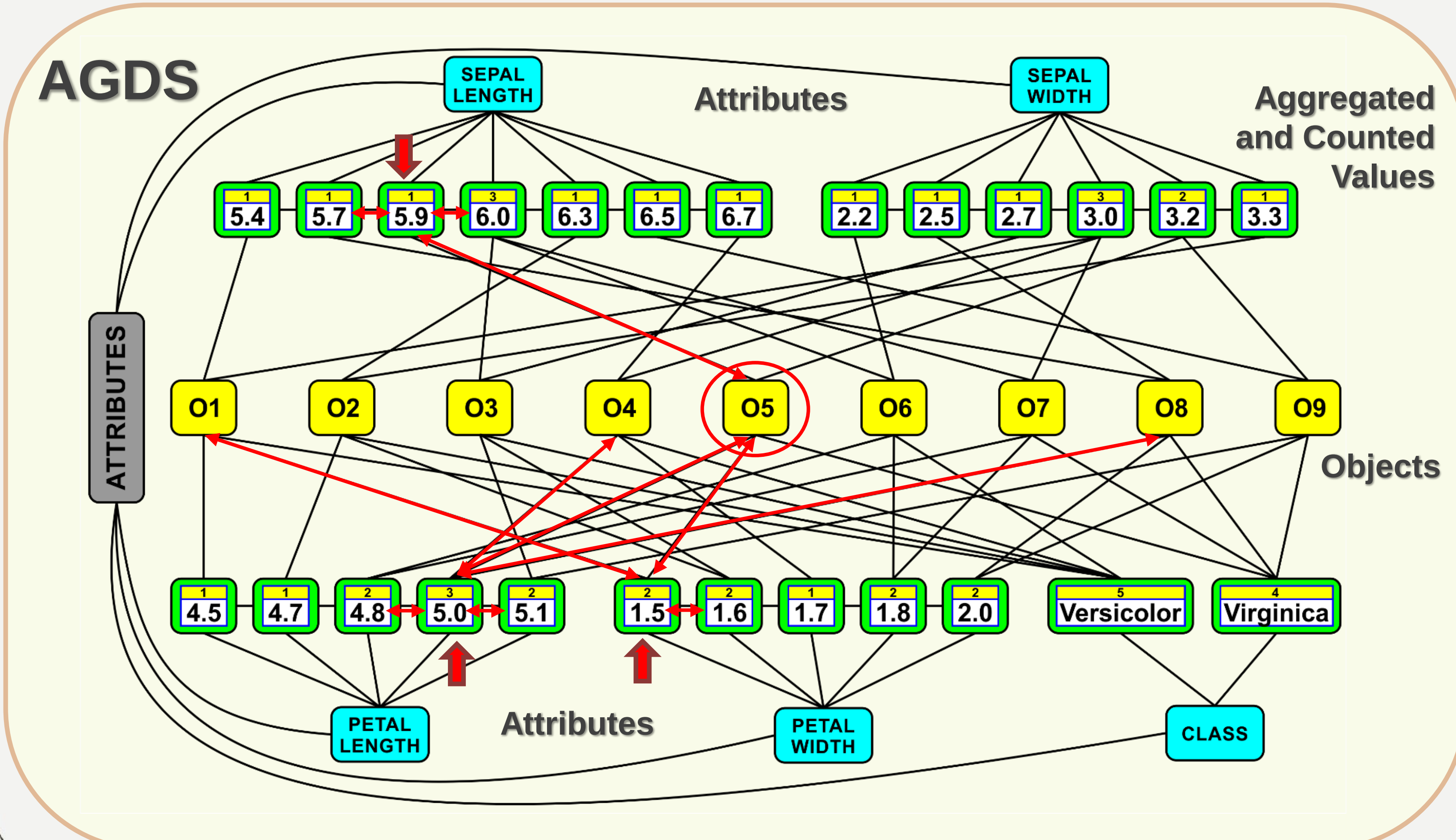
AGDS can directly represent much more useful often used relationships than other structures, and these relationships are immediately available similarly to what happens in our brains.



Searching for the most similar objects

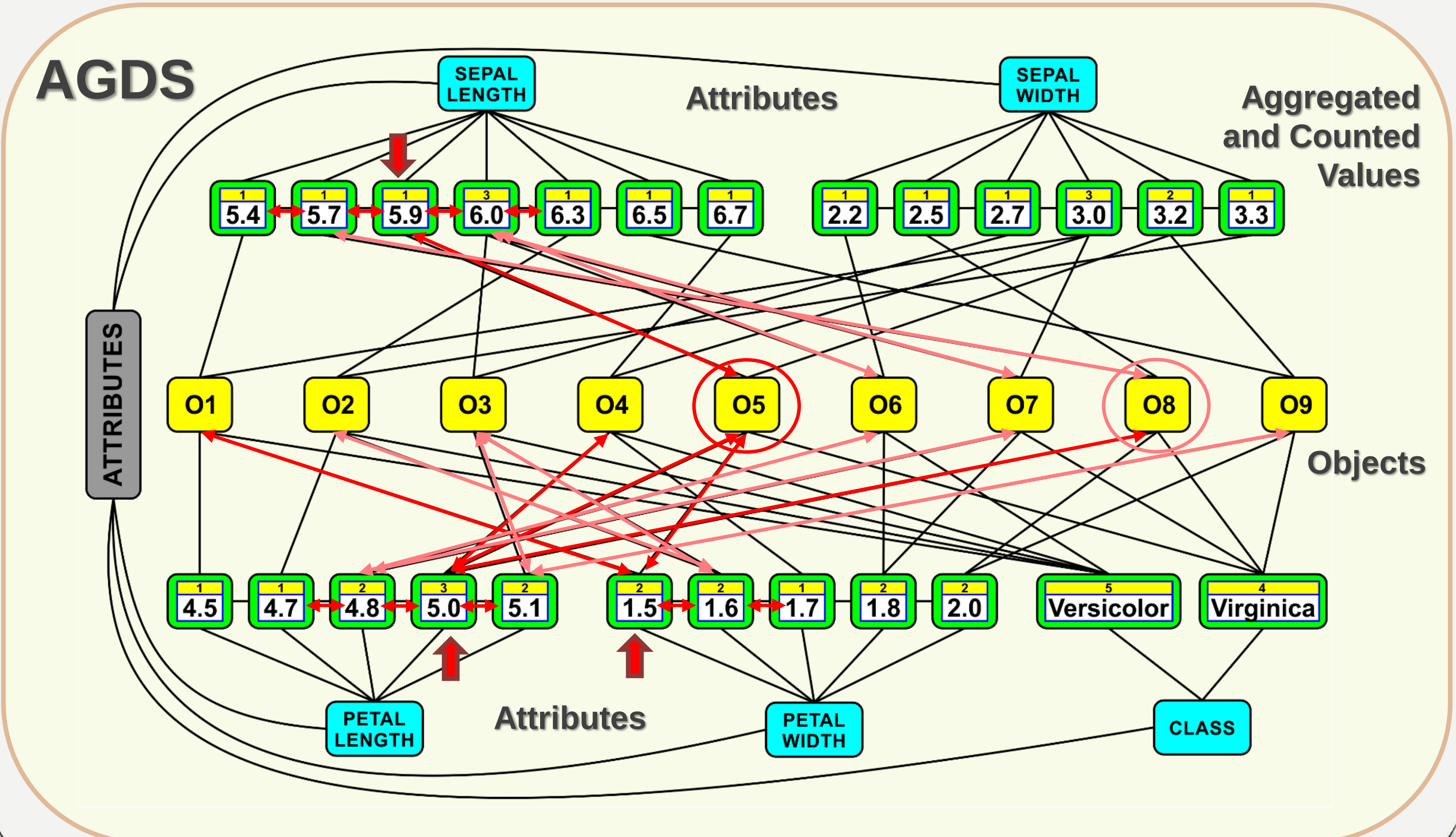
AGDS be used for searching for the most similar objects to any given input data (combination of input features or objects):

In the direct stimulation of inputs, O5 is the strongest associated!



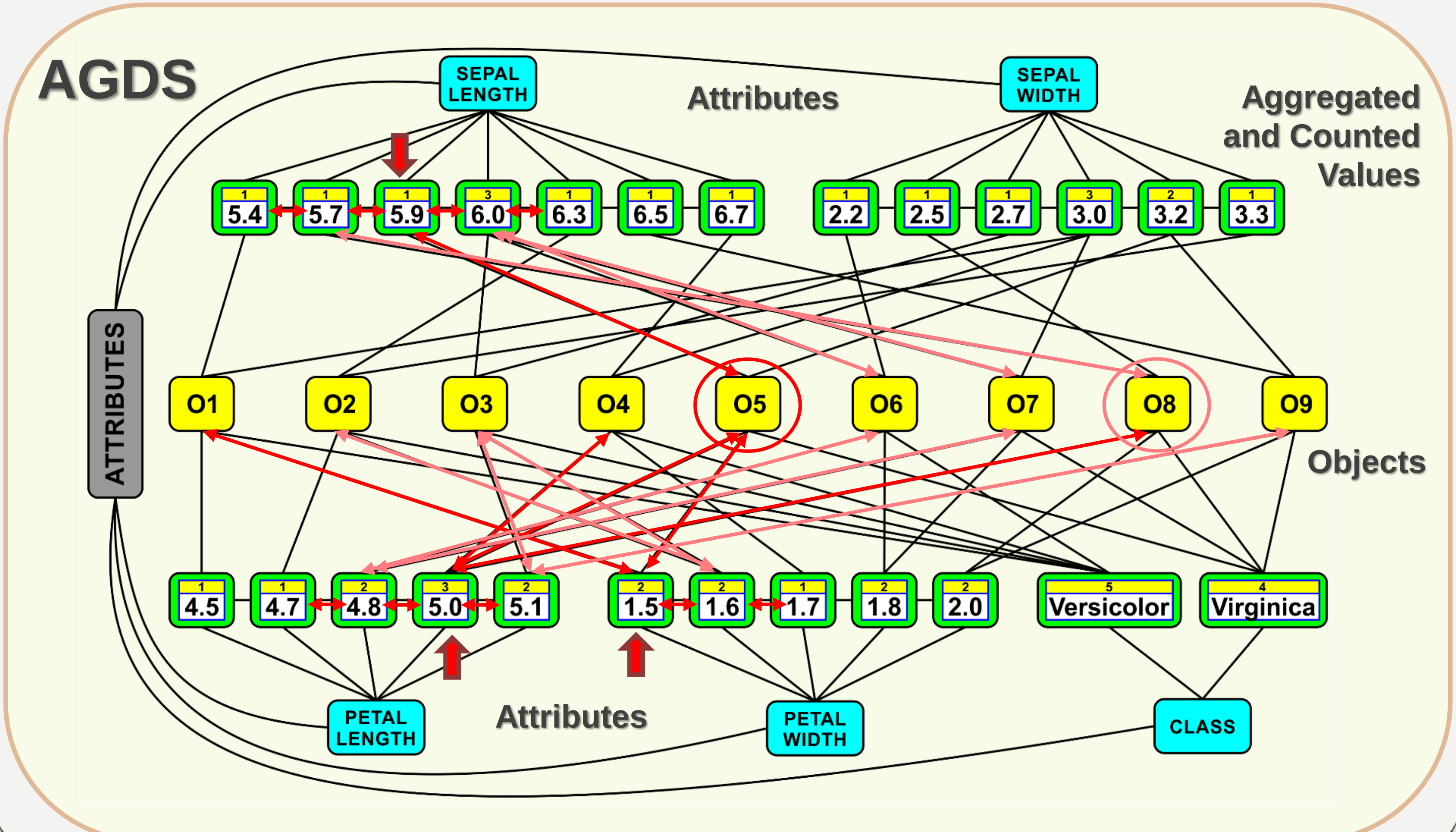
Searching for the most similar objects

In not so clear situations, we can stimulate neighbor value nodes and calculate their states (using connection weights), and stimulate the connected object nodes and value nodes further:



Searching for the most similar objects

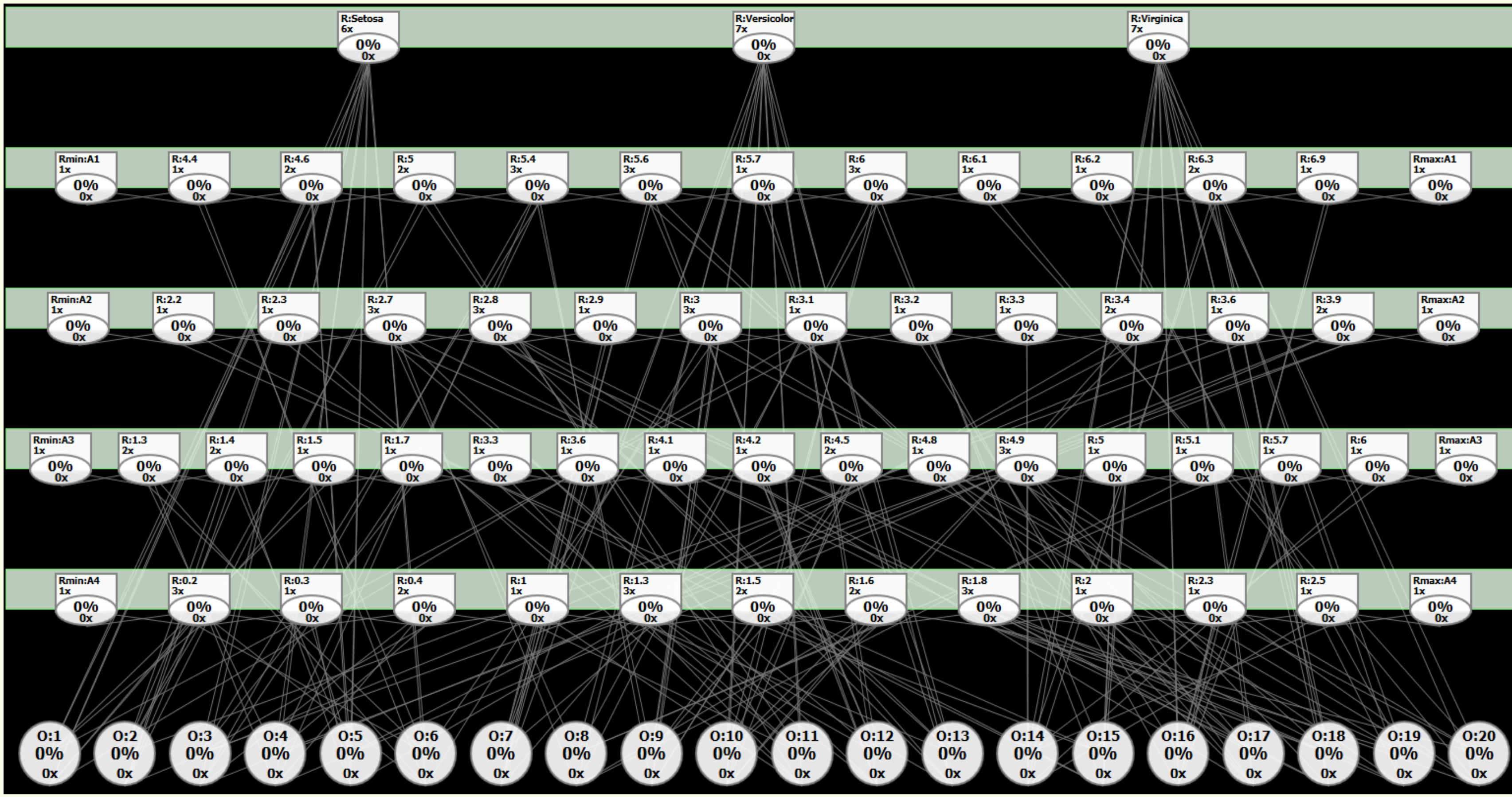
In not so clear situations, we can stimulate neighbor value nodes and calculate their states (using connection weights), and stimulate the connected object nodes and value nodes further:



How it works in practice?

The simulation presents the stimulated nodes and their sequence:

We can also transform nodes into neurons and count up their spikes:

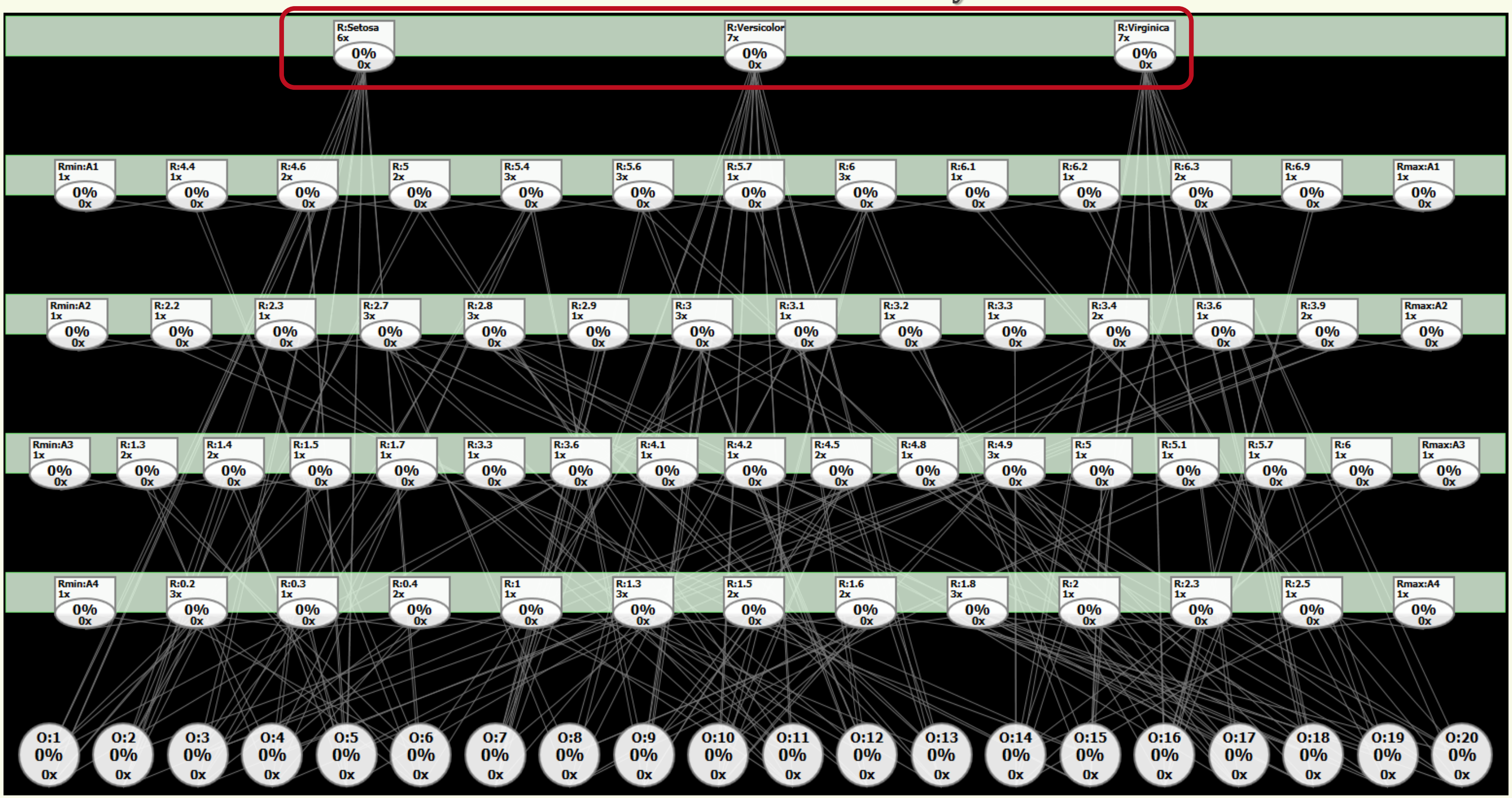


Object nodes represent combinations of input features (value nodes).

Stimulate and get nodes representing the most associated values or objects:

The simulation presents the stimulated nodes and their sequence:

Class Labels are treated and connected in the same way as other Attribute Values.



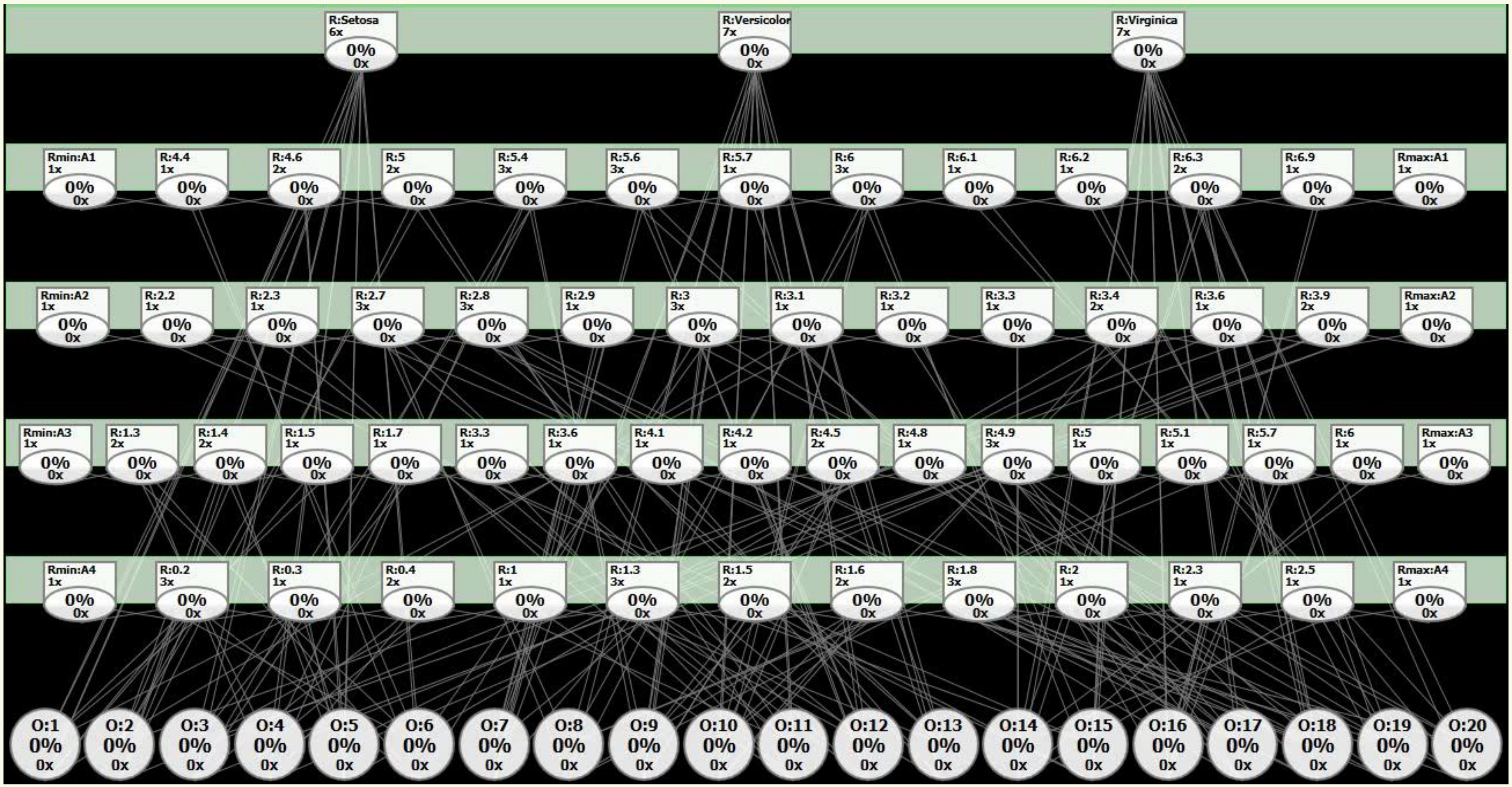
Every combination of nodes can be used as inputs in such a network.

All other nodes can be used as outputs if activated frequently.

Stimulate and get nodes representing the most associated values or objects:

The simulation presents the stimulated nodes and their sequence:

Class Labels are treated and connected in the same way as other Attribute Values.



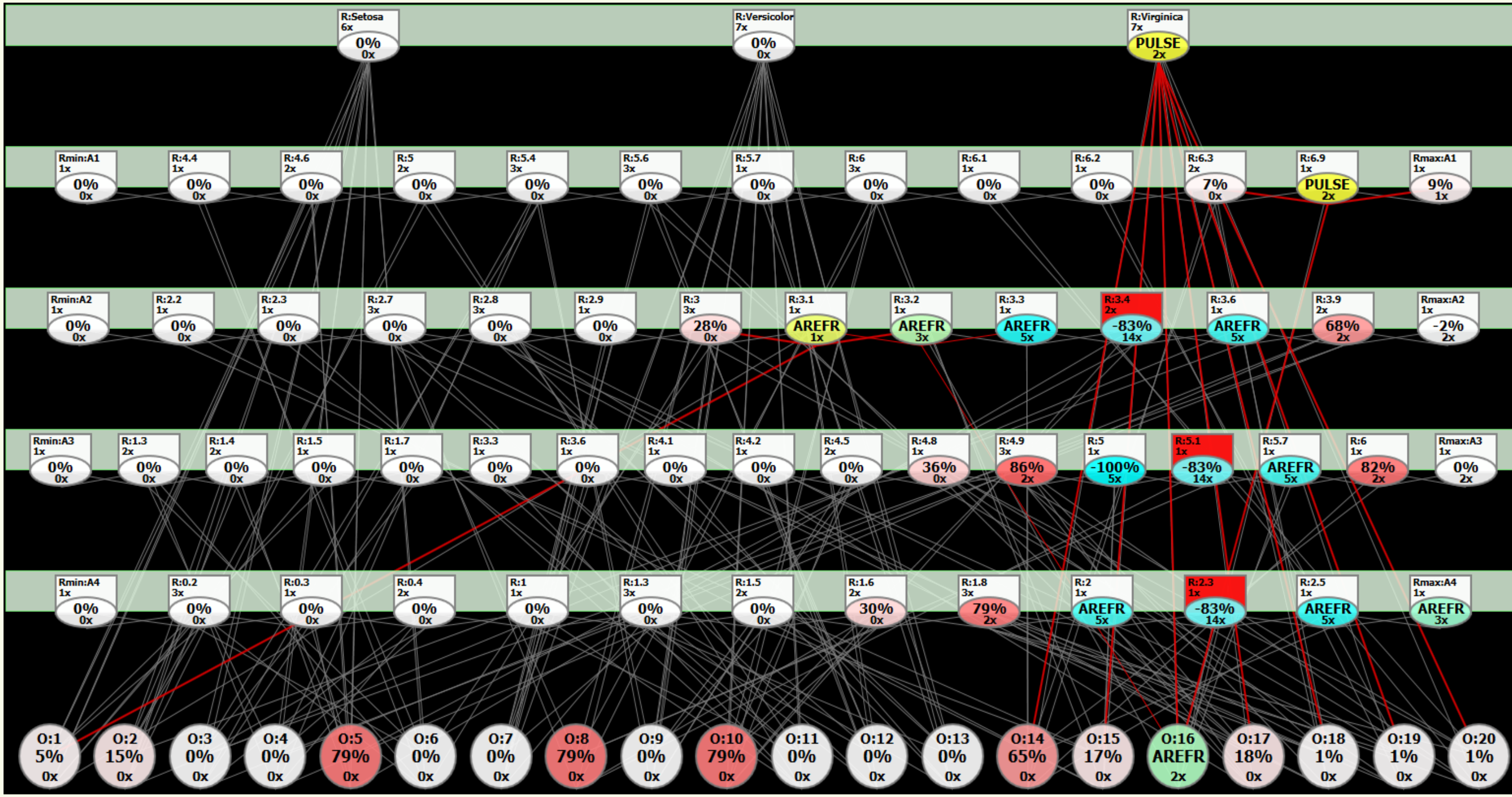
Every combination of nodes can be used as inputs in such a network.

All other nodes can be used as outputs if activated frequently.

Stimulate and get nodes representing the most associated values or objects:

The simulation presents the stimulated nodes and their sequence:

The inputs [?, 3.4, 5.1, 2.3) were classified as Virginica!



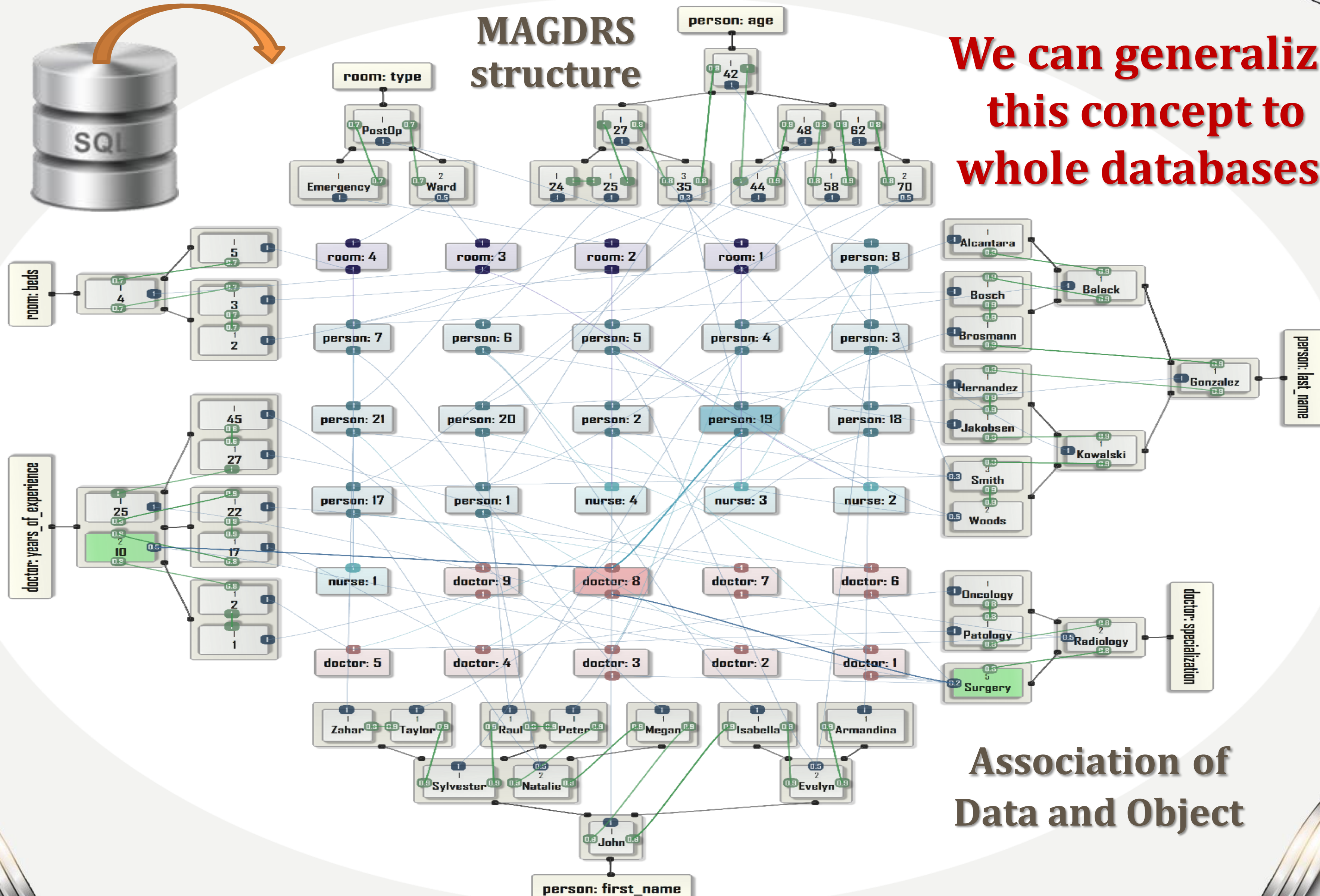
This network recognized training pattern No. 16 as the most associated.

The missing leaf-length attribute on input was identified as 6.9.

Multi-Associative Graph Data Relationship Structures

How to generalize this concept to databases?

Multi-Associative Graph Data Relationship Structures (MAGDRS)



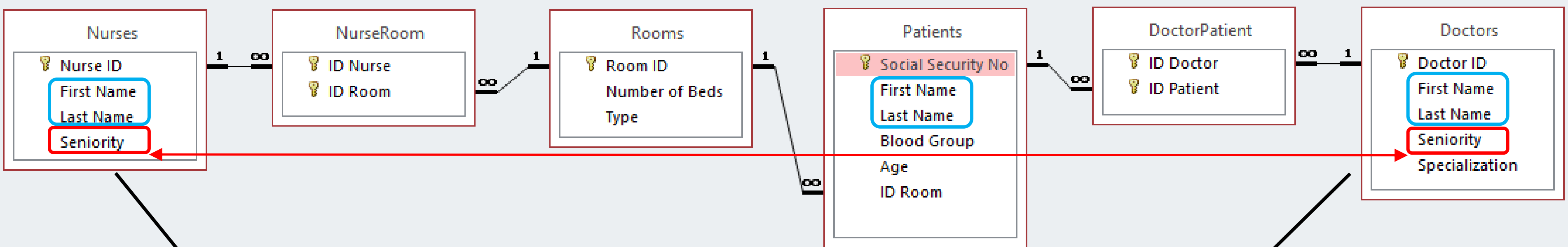
We can generalize this concept to whole databases!

Association of Data and Object

Associative Transformation of Relational Databases

Let's try to implement associative transformation to a small relational database and construct an associative graph:

✓ Representation of horizontal relations between entities (objects)



Nurses

Nurse ID	First Name	Last Name	Seniority
N1	Amy	Moon	12
N2	Rose	Jolie	18
N3	Kate	Ford	24
N4	Lisa	Brown	9
N5	Sara	Pitt	4
N6	Kate	Lopez	12

Doctors

Doctor ID	First Name	Last Name	Seniority	Specialization
D1	Tom	Hanks	18	orthopedics
D2	Jack	Brown	15	surgery
D3	Lisa	Ford	23	pediatrician
D4	Tom	Trump	35	pediatrician
D5	Kate	Smith	7	surgery
D6	Amy	Hanks	12	surgery

to additionally represent vertical relationships, sort data of all attributes simultaneously, aggregate all duplicates and allow faster reasoning and search for related data and relationships.

Associative transformation of DB

Patients

Social Security No	First Name	Last Name	Blood Group	Age	ID Room
17102595834	Jack	Hanks	0	0	R6
17102785603	Nina	Rock	AB	0	R6
73040593020	Tom	Kite	A	44	R4
84021205010	Tom	Ford	AB	33	R1
90503029943	Emy	Cruise	A	27	R2
94051382054	Lisa	White	B	23	R3
94080639502	Paula	Smith	B	23	R2

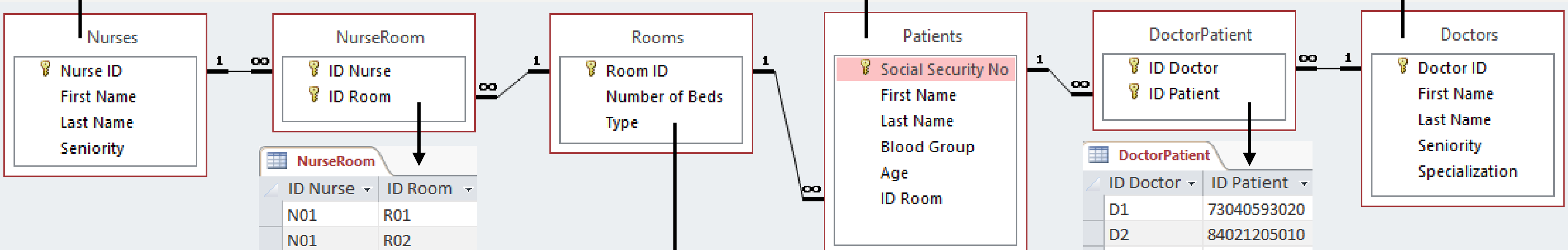
Nurses

Nurse ID	First Name	Last Name	Seniority
N1	Amy	Moon	12
N2	Rose	Jolie	18
N3	Kate	Ford	24
N4	Lisa	Brown	9
N5	Sara	Pitt	4
N6	Kate	Lopez	12

Small hospital database

Doctors

Doctor ID	First Name	Last Name	Seniority	Specialization
D1	Tom	Hanks	18	orthopedics
D2	Jack	Brown	15	surgery
D3	Lisa	Ford	23	pediatrician
D4	Tom	Trump	35	pediatrician
D5	Kate	Smith	7	surgery
D6	Amy	Hanks	12	surgery



NurseRoom

ID Nurse	ID Room
N01	R01
N01	R02
N02	R06
N03	R03
N03	R04
N03	R05
N04	R01
N04	R02
N05	R03
N05	R04
N05	R05
N06	R06

Rooms

Room ID	Number of Beds	Type
R1	2	intensive care
R2	2	intensive care
R3	3	therapeutic
R4	3	therapeutic
R5	3	therapeutic
R6	8	newborns

DoctorPatient

ID Doctor	ID Patient
D1	73040593020
D2	84021205010
D2	90503029943
D3	17102595834
D3	17102785603
D4	17102595834
D4	17102785603
D5	90503029943
D5	94080639502
D6	17102595834
D6	73040593020
D6	94051382054
D6	84021205010
D6	94080639502

Rule of Associative Transformation

Transforms only these tables for which all foreign keys are already represented by the nodes in the associative MAGDRS structure:

Patients

Social Security No	First Name	Last Name	Blood Group	Age	ID Room
17102595834	Jack	Hanks	0	0	R6
17102785603	Nina	Rock	AB	0	R6
73040593020	Tom	Kite	A	44	R4
84021205010	Tom	Ford	AB	33	R1
90503029943	Emy	Cruise	A	27	R2
94051382054	Lisa	White	B	23	R3
94080639502	Paula	Smith	B	23	R2

Nurses

Nurse ID	First Name	Last Name	Seniority
N1	Amy	Moon	12
N2	Rose	Jolie	18
N3	Kate	Ford	24
N4	Lisa	Brown	9
N5	Sara	Pitt	4
N6	Kate	Lopez	12

Associative Transformation Process

Doctors

Doctor ID	First Name	Last Name	Seniority	Specialization
D1	Tom	Hanks	18	orthopedics
D2	Jack	Brown	15	surgery
D3	Lisa	Ford	23	pediatrician
D4	Tom	Trump	35	pediatrician
D5	Kate	Smith	7	surgery
D6	Amy	Hanks	12	surgery

Nurses **1**: Nurse ID, First Name, Last Name, Seniority

NurseRoom **2**: ID Nurse, ID Room

Rooms **1**: Room ID, Number of Beds, Type

Patients **2**: Social Security No, First Name, Last Name, Blood Group, Age, ID Room

DoctorPatient **3**: ID Doctor, ID Patient

Doctors **1**: Doctor ID, First Name, Last Name, Seniority, Specialization

Possible sequence of transformation of tables

NurseRoom

ID Nurse	ID Room
N01	R01
N01	R02
N02	R06
N03	R03
N03	R04
N03	R05
N04	R01
N04	R02
N05	R03
N05	R04
N05	R05
N06	R06

Rooms

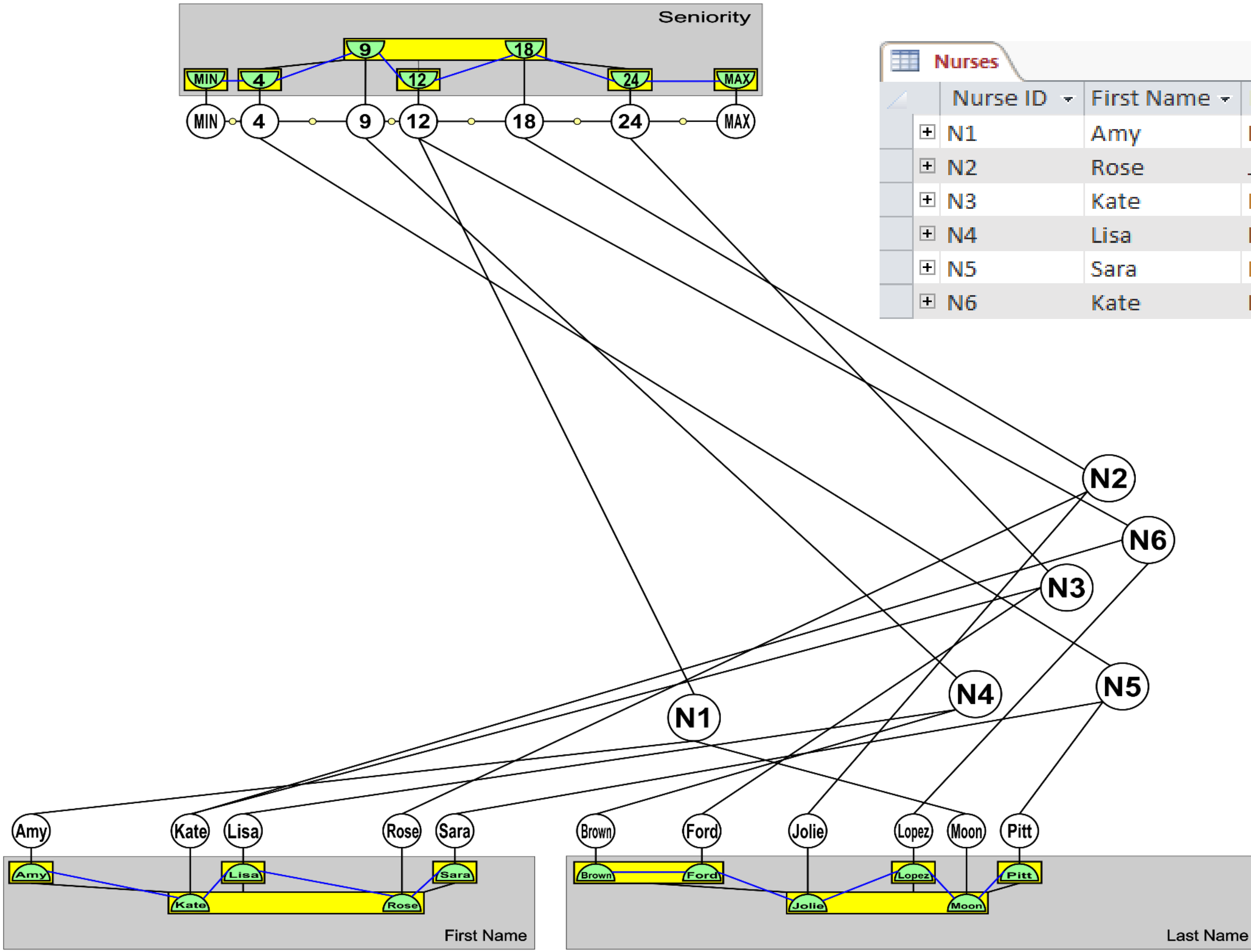
Room ID	Number of Beds	Type
R1	2	intensive care
R2	2	intensive care
R3	3	therapeutic
R4	3	therapeutic
R5	3	therapeutic
R6	8	newborns

DoctorPatient

ID Doctor	ID Patient
D1	73040593020
D2	84021205010
D2	90503029943
D3	17102595834
D3	17102785603
D4	17102595834
D4	17102785603
D5	90503029943
D5	94080639502
D6	17102595834
D6	73040593020
D6	94051382054
D6	84021205010
D6	94080639502

MAGDRS construction for relational DB

Table NURSES is added to the empty network.

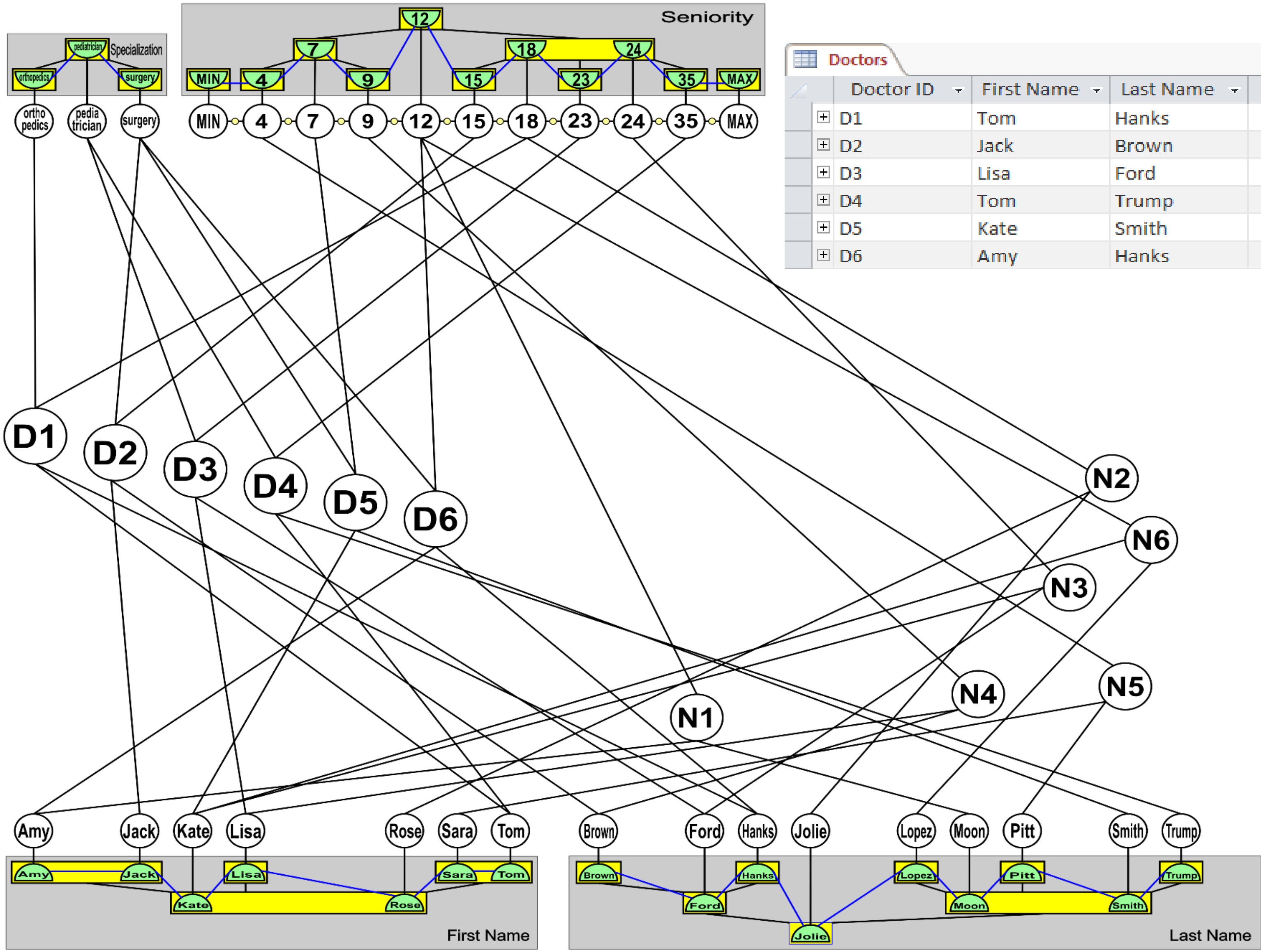


Nurses

Nurse ID	First Name	Last Name	Seniority
N1	Amy	Moon	12
N2	Rose	Jolie	18
N3	Kate	Ford	24
N4	Lisa	Brown	9
N5	Sara	Pitt	4
N6	Kate	Lopez	12

MAGDRS construction for relational DB

Table DOCTORS is added to the network.

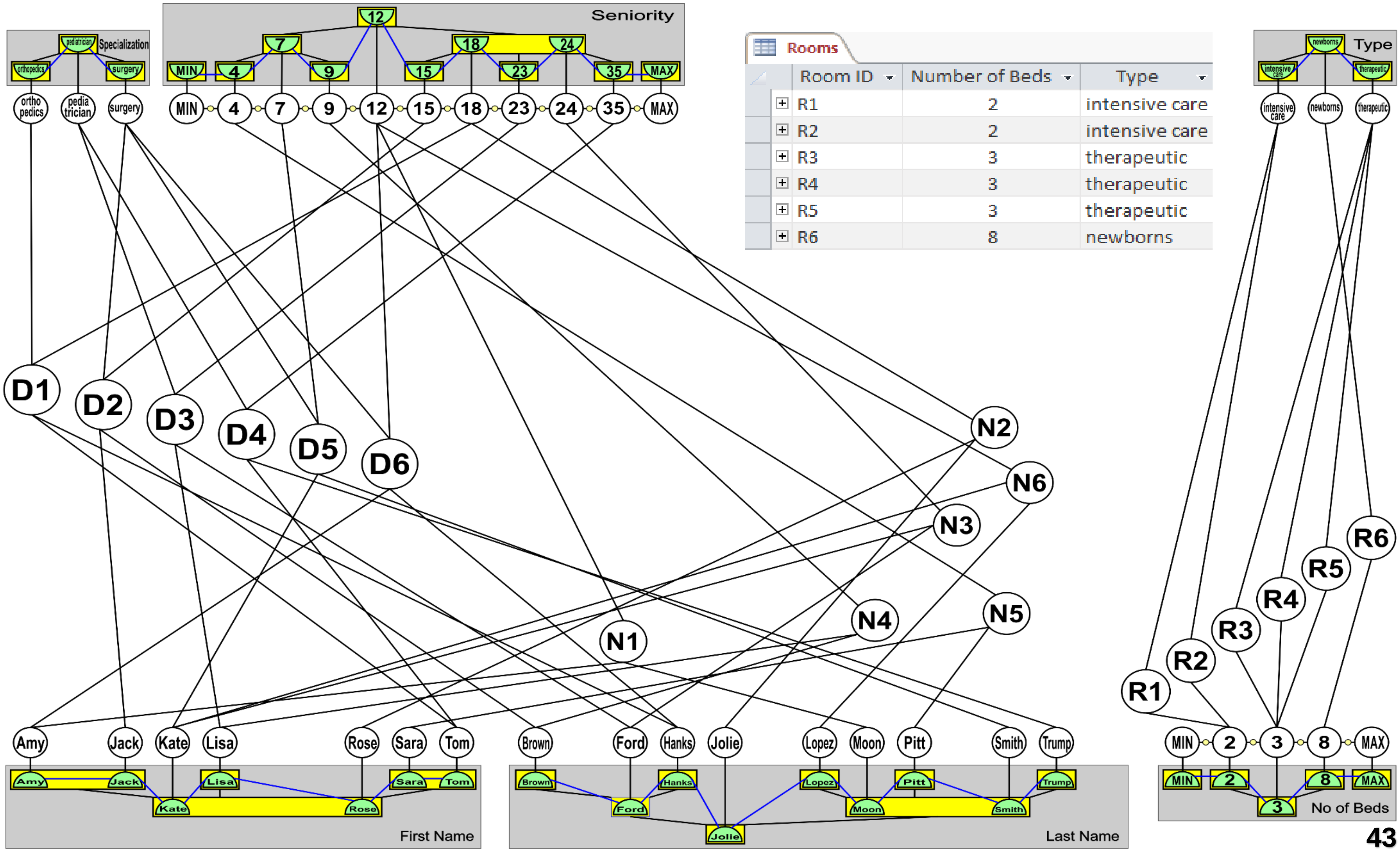


Doctors

Doctor ID	First Name	Last Name	Seniority	Specialization
D1	Tom	Hanks	18	orthopedics
D2	Jack	Brown	15	surgery
D3	Lisa	Ford	23	pediatrician
D4	Tom	Trump	35	pediatrician
D5	Kate	Smith	7	surgery
D6	Amy	Hanks	12	surgery

MAGDRS construction for relational DB

Table ROOMS is added to the network.

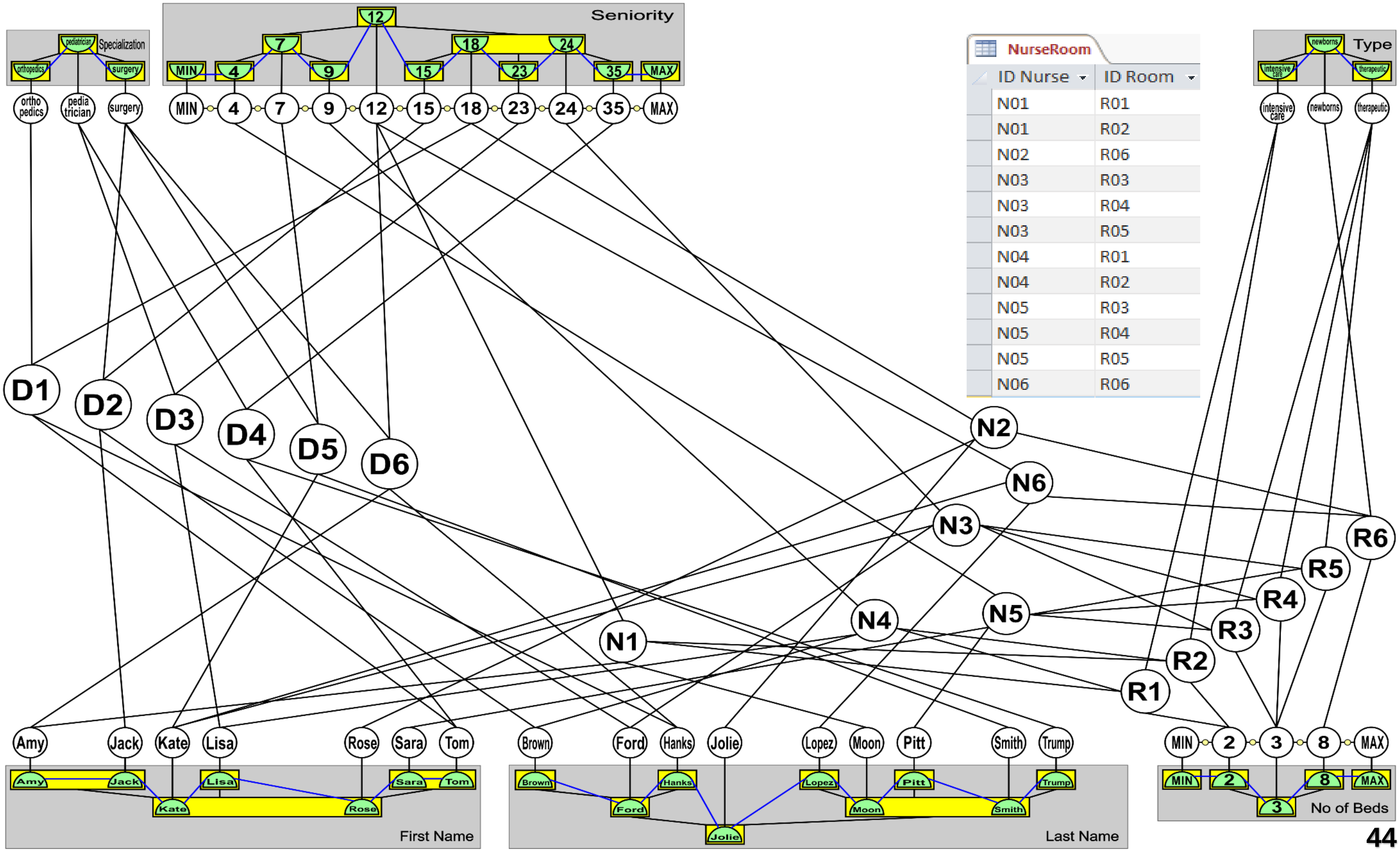


Rooms

Room ID	Number of Beds	Type
R1	2	intensive care
R2	2	intensive care
R3	3	therapeutic
R4	3	therapeutic
R5	3	therapeutic
R6	8	newborns

MAGDRS construction for relational DB

Table NURSEROOM is added to the DASNG network.

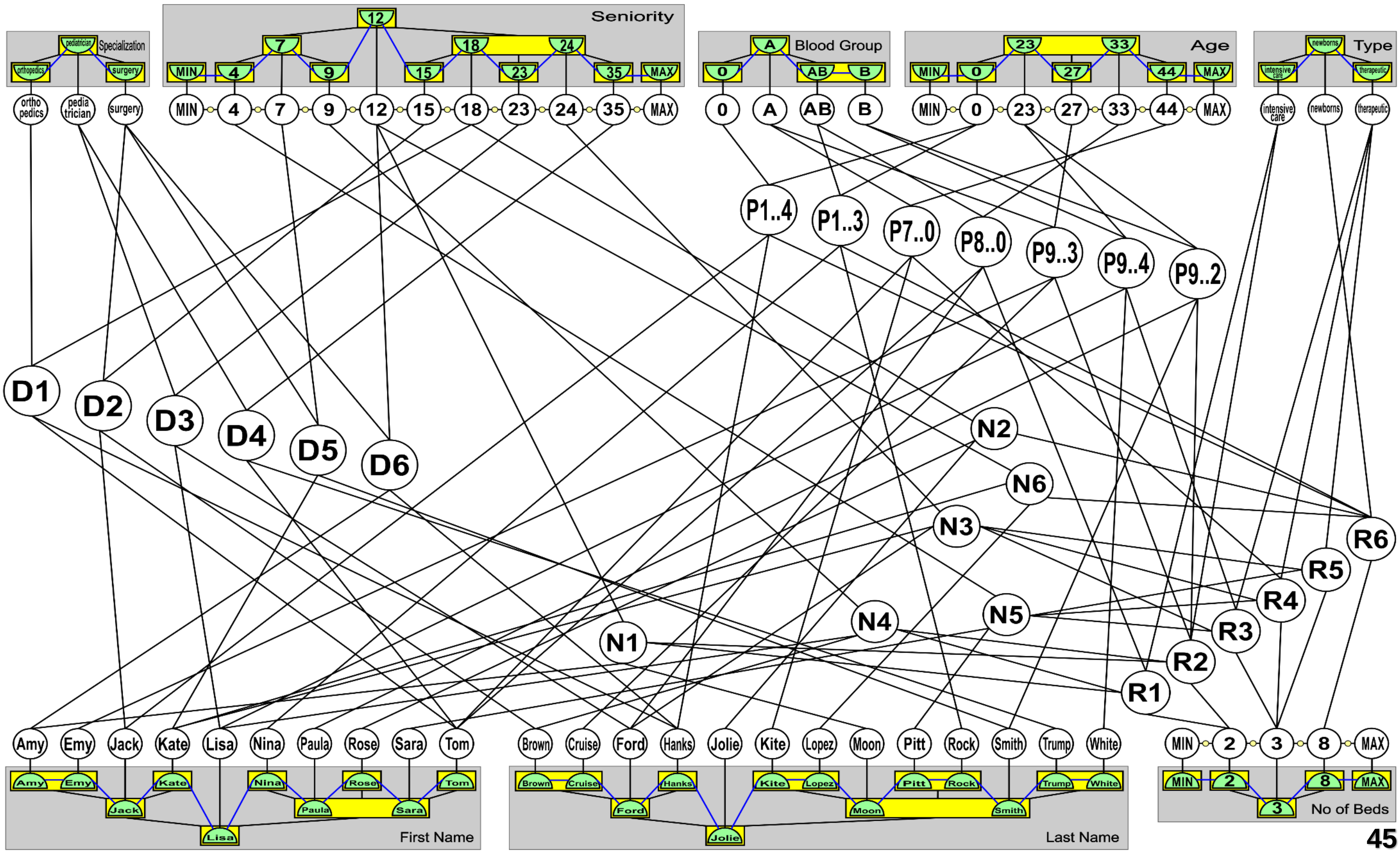


NurseRoom

ID Nurse	ID Room
N01	R01
N01	R02
N02	R06
N03	R03
N03	R04
N03	R05
N04	R01
N04	R02
N05	R03
N05	R04
N05	R05
N06	R06

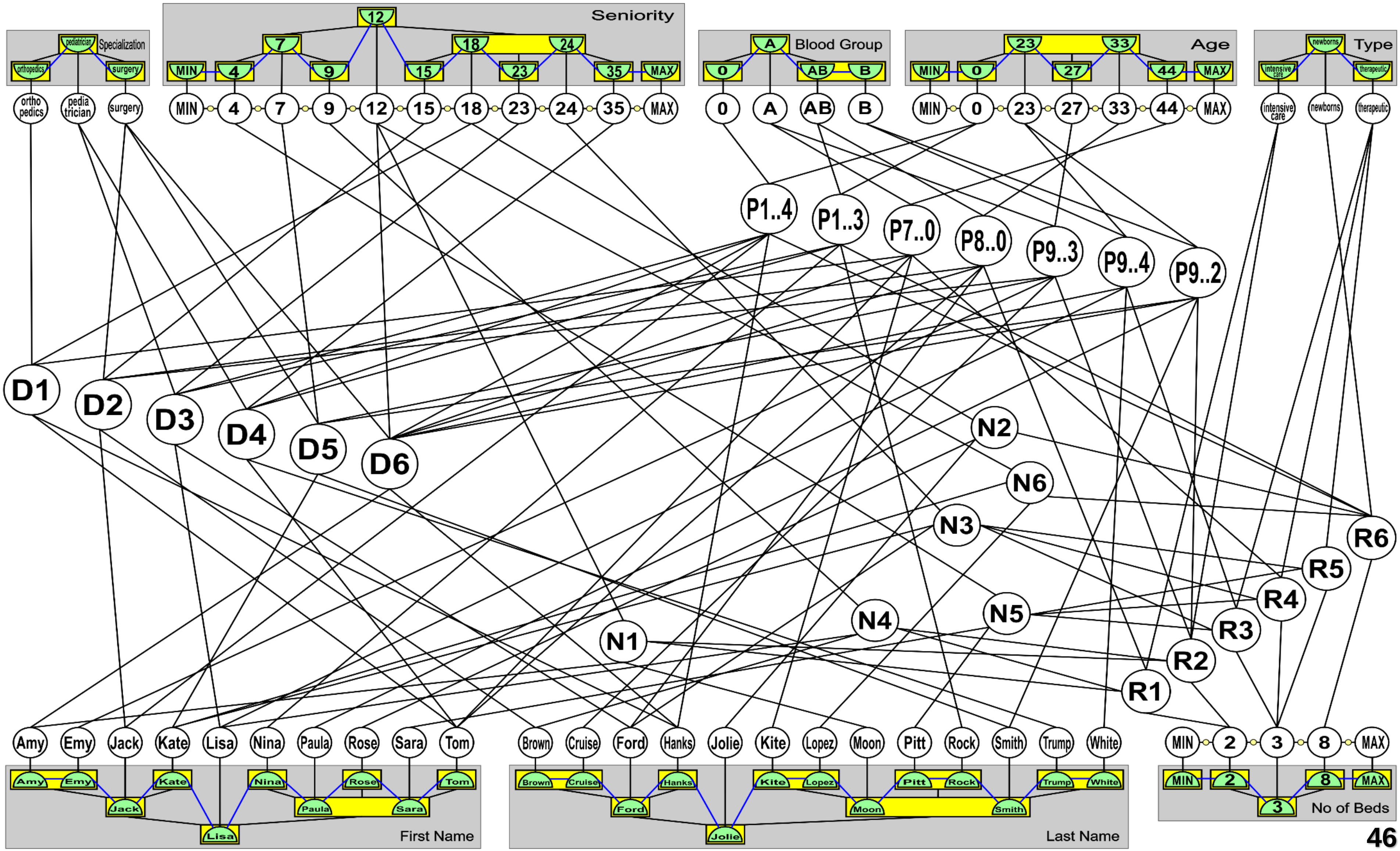
MAGDRS construction for relational DB

Table PATIENTS is added to the network.



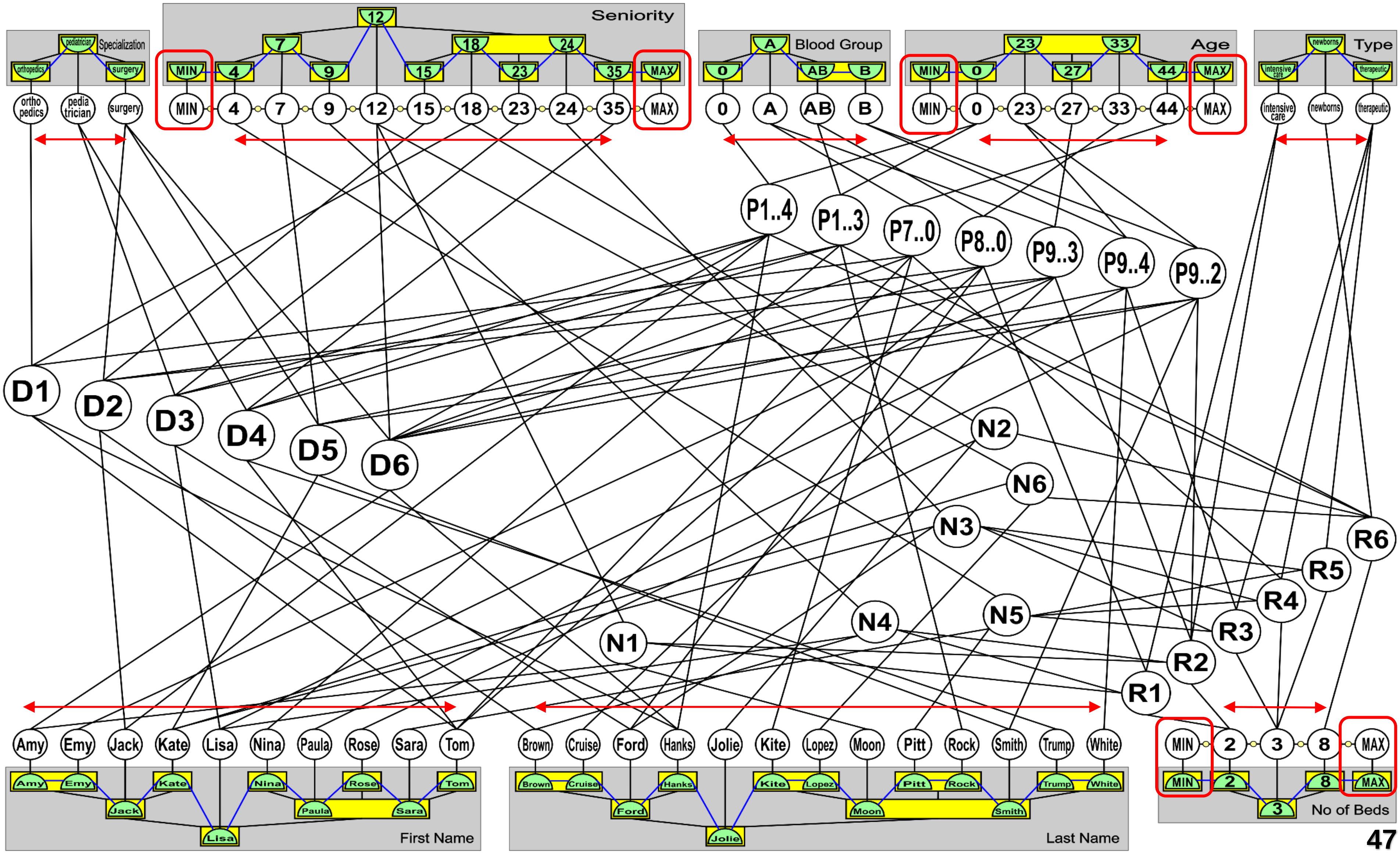
MAGDRS construction for relational DB

Table DOCTORPATIENT is added to the network.



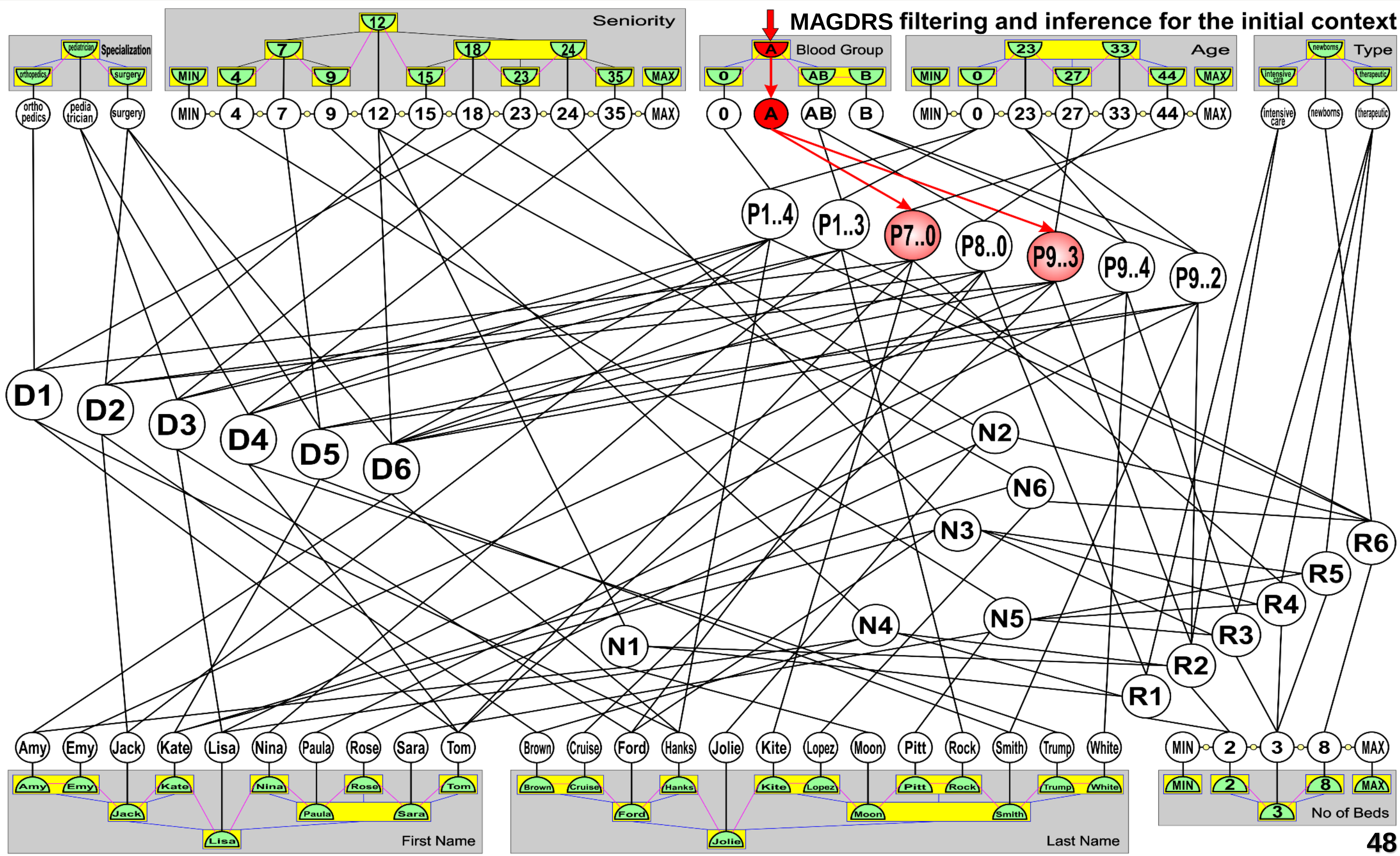
The result of MAGDRS construction

No duplicates and all values are sorted and quickly accessible!



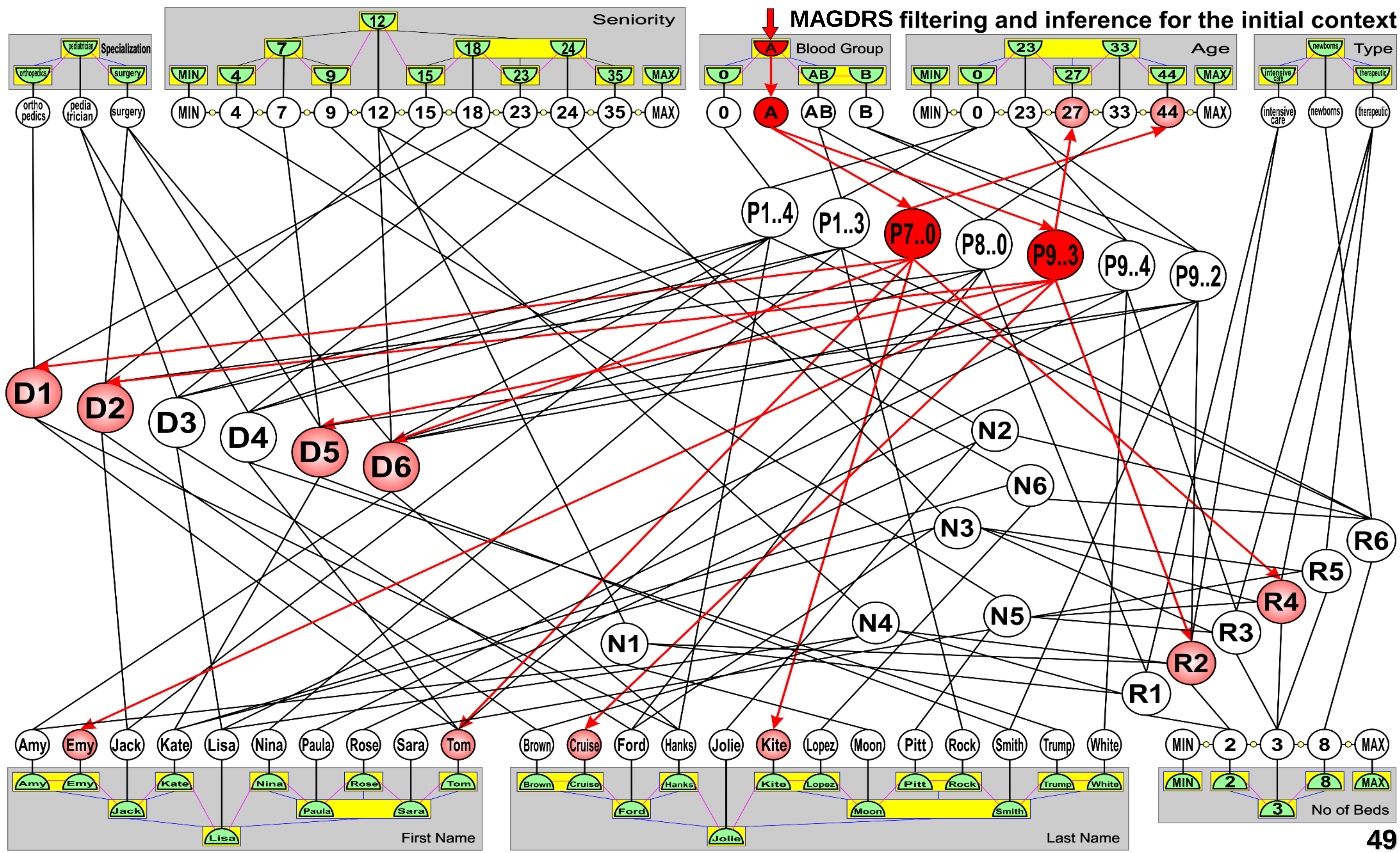
Inferences using MAGDRS network

Retrieving associated pieces of information from the network!



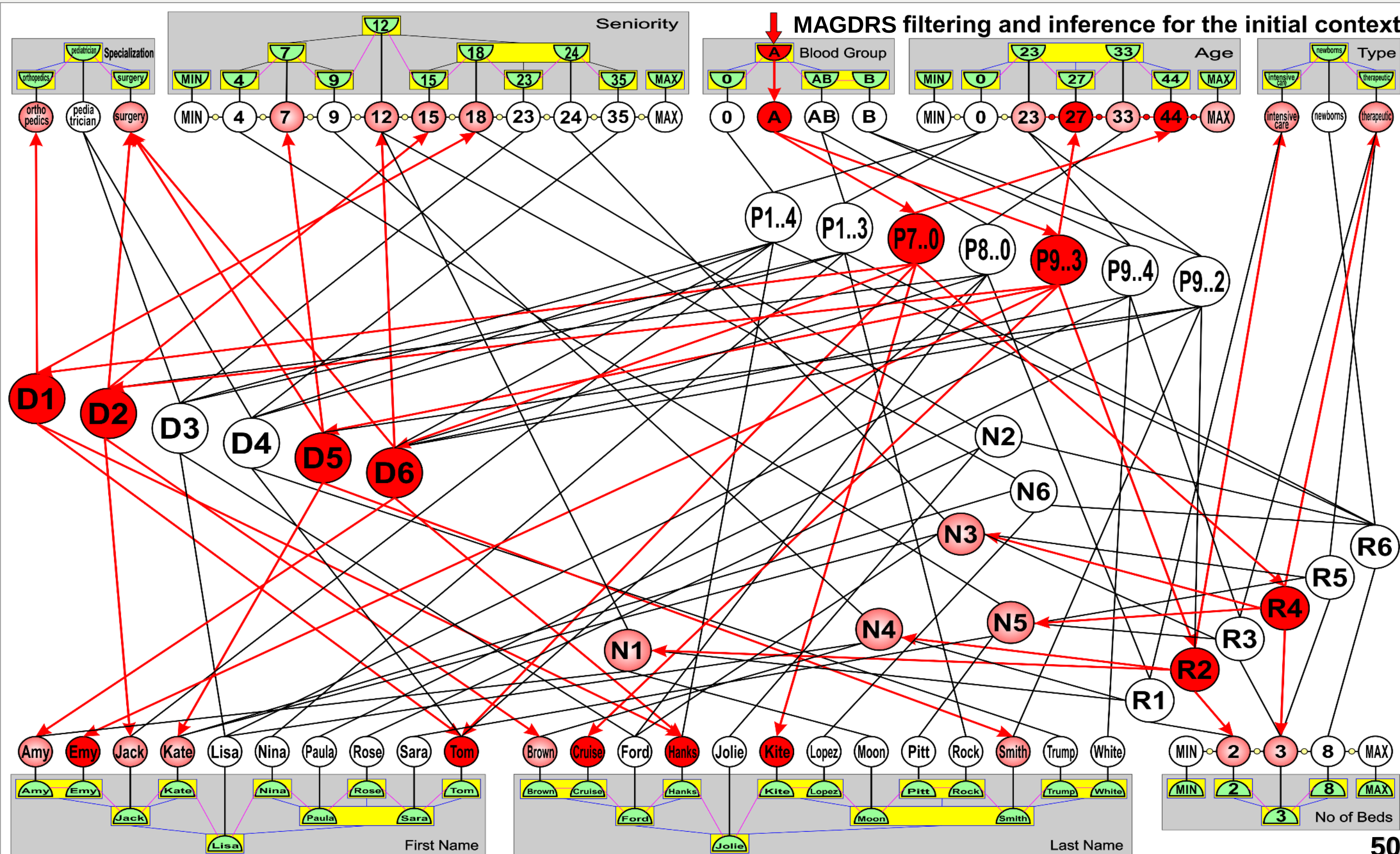
Inferences using MAGDRS network

Retrieving associated pieces of information from the network!

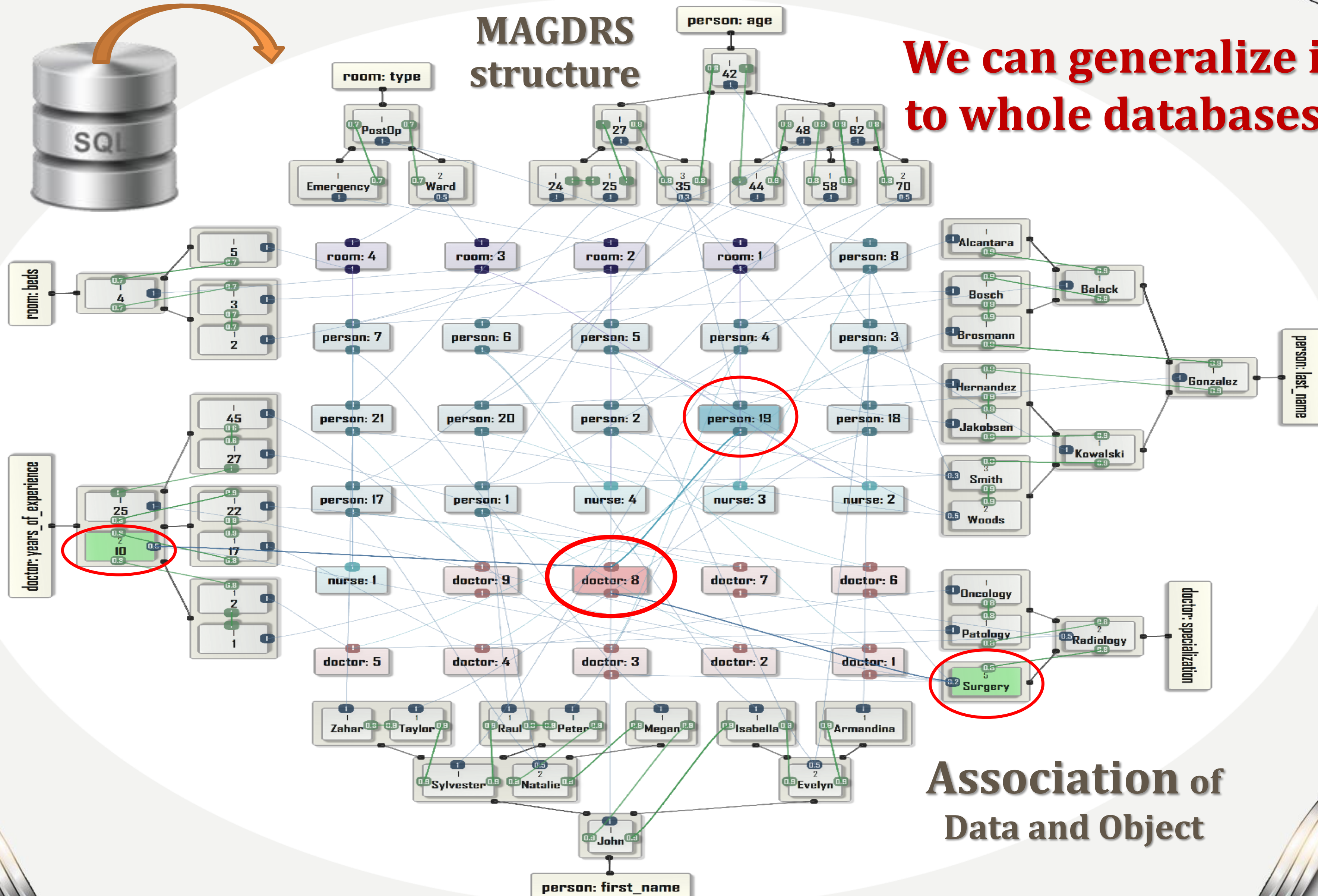


Inferences using MAGDRS network

Retrieving associated pieces of information from the network!



Multi-Associative Graph Data Relationship Structures (MAGDRS)



We can generalize it to whole databases!

Association of Data and Object

Associative Recommendation Systems

How to use associative graphs for recommendation?

Recommendation Systems

Recommendation systems:

- searching through a large volume of dynamically generated information to provide users with personalized content and services;
- should be able to find objects that satisfy input requirements (criteria) or are as close as possible to them;
- are beneficial to both service providers and users, saving time and effort, as well as increase the turnover and positive reactions of the users;
- enhance revenues, for the fact that they are effective means of selling more products;
- have also proved to improve decision-making process and quality.

Associative recommendation systems:

- use stored associations representing various relationships and their strengths to search for the most associated (the associatively closest) data and objects that meet the input criteria;
- use BFS trees to move over the associative graphs to calculate the associations of the nodes until the final nodes of representing the desired objectives are achieved and stimulated to a given extend, reproducing the level of recommendation.

Example Input Data for Candidate Recommendation for a free Position

We want to find the best candidate for the open position “Job Offer”.

We have five candidates! Who is the best one?! Who will you recommend?

	Technical Skills			PersonalSkills			LanguageSkills			Education Level	Sallary	Time Work Type	Field Importance
	Name	Years	Weight	Name	Level	Weight	Name	Level	Weight				
Job Offer	C#	≥ 3	(10/10)	Communication skills	60%-80%	(8/10)	English	≥ C1	(8/10)	Bachelor	7500	FullTime	TechnicalSkills (9/10)
	Entity Framwork	≥ 3	(9/10)	Ability to work under pressure	80%-100%	(10/10)	Polish	≥ Native	(7/10)				PersonalSkills (8/10)
	T-SQL	≥ 2	(8/10)	Decision making	60%-80%	(7/10)							LanguageSkills (5/10)
	Cloud dev	≥ 1	(6/10)										Education (10/10)
													Sallary(10/10)
													TimeWorkType (10/10)
Candidate 1	C#	2	X	Ability to work under pressure	70%	X	Polish	Native	X	Bachelor	7000	HalfTime	X
	T-SQL	3		Communication skills	80%								
	Cloud dev	2											
Candidate 2	Entity Framwork	3	X	Communication skills	70%	X	English	B2	X	Master	6500	FullTime	X
	C#	1		Decision making	90%								
	T-SQL	1		Ability to work under pressure	90%								
Candidate 3	Cloud dev	2	X	Communication skills	80%	X	Polish	Native	X	Master	6500	HalfTime	X
	C#	2		Decision making	90%		English	C1					
Candidate 4	Entity Framwork	1	X	Ability to work under pressure	70%	X	English	Native	X	Bachelor	8000	FullTime	X
	C#	1		Decision making	50%								
	T-SQL	2											
	Cloud dev	2											
Candidate 5	Cloud dev	1	X	Communication skills	70%	X	Polish	C1	X	Bachelor	7000	HalfTime	X
	Entity Framwork	2		Ability to work under pressure	80%								
	T-SQL	3		Decision making	90%								

Sample AGDS Structure

The **charging level x** of the internally stimulated node is defined as a weighted sum:

$$x_n = \sum_{k=1}^{s_n} x_k \cdot w_k$$

Reciprocal edges are created between value nodes $V_i^{a^k}$ and $V_j^{a^k}$ representing similar values $v_i^{a^k}$ and $v_j^{a^k}$ of the same attribute a^k and forward stimuli in both directions with the same **weight**:

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

where

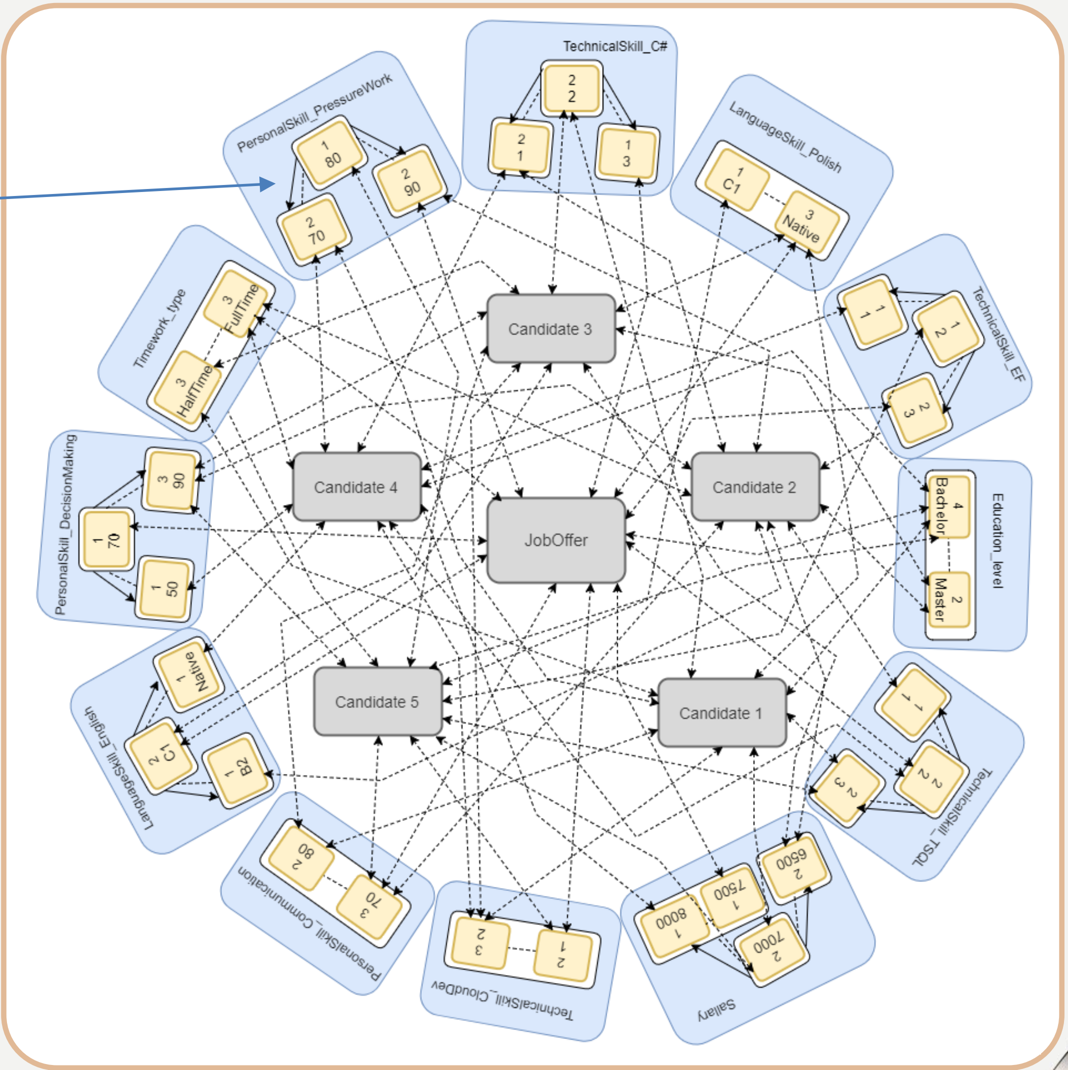
$$r^{a^k} = v_{max}^{a^k} - v_{min}^{a^k}$$

is a variation range of values of the attribute a^k.

The **weight** of the edge that passes the signal from the value node $V_i^{a^k}$ to the object node O_n can be calculated after:

$$w_{O_m, O_n} = \frac{1}{\theta_n}$$

where the **threshold** $\boldsymbol{\theta_n}$ is the number of values and objects that define the object nodes O_n and activate this node.

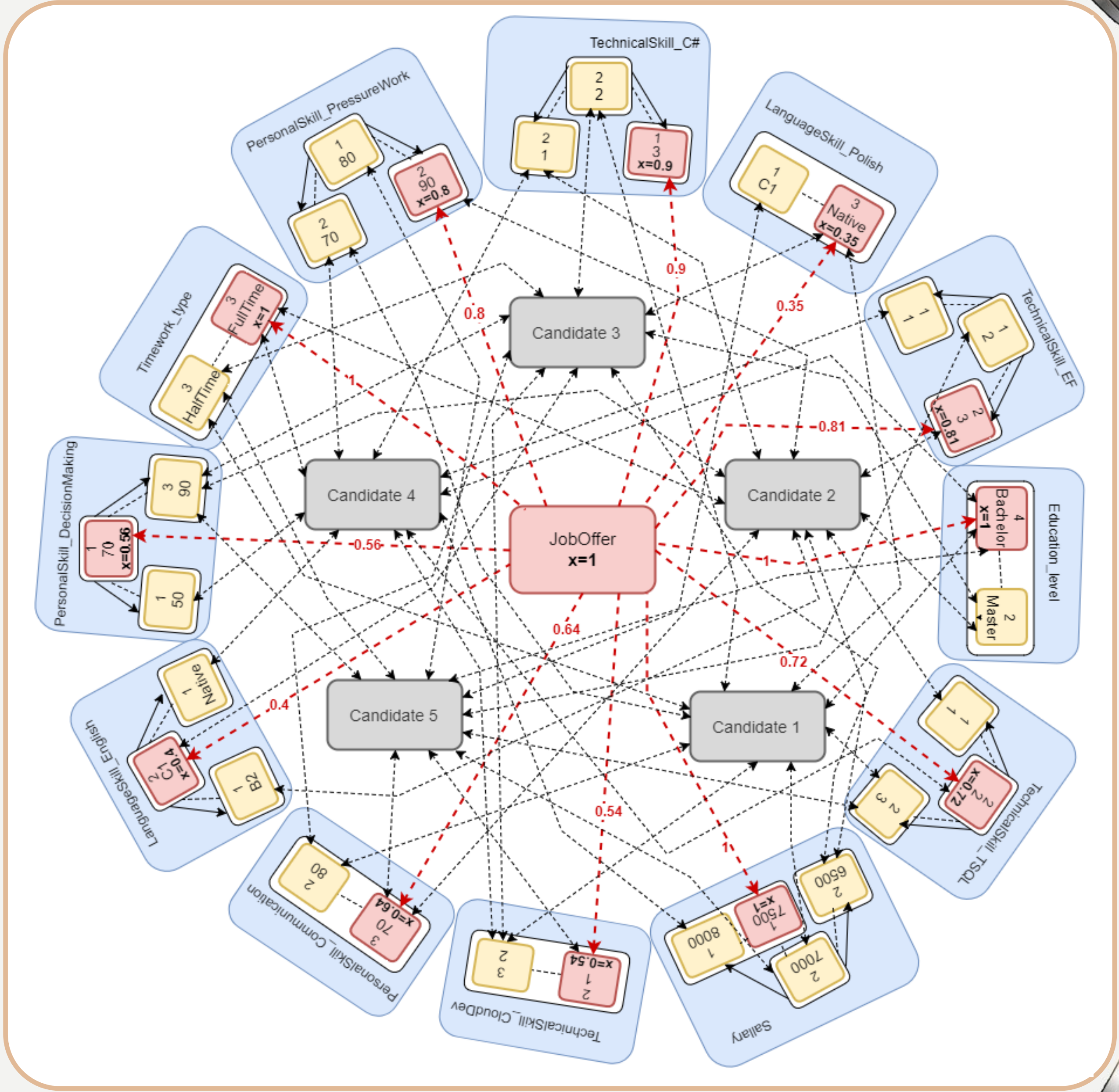


The stimuli passing through the edge in the opposite direction: $w_{O_n, v_i^{a^k}} = 1$, $w_{O_n, O_m} = 1$

AGDS represents knowledge

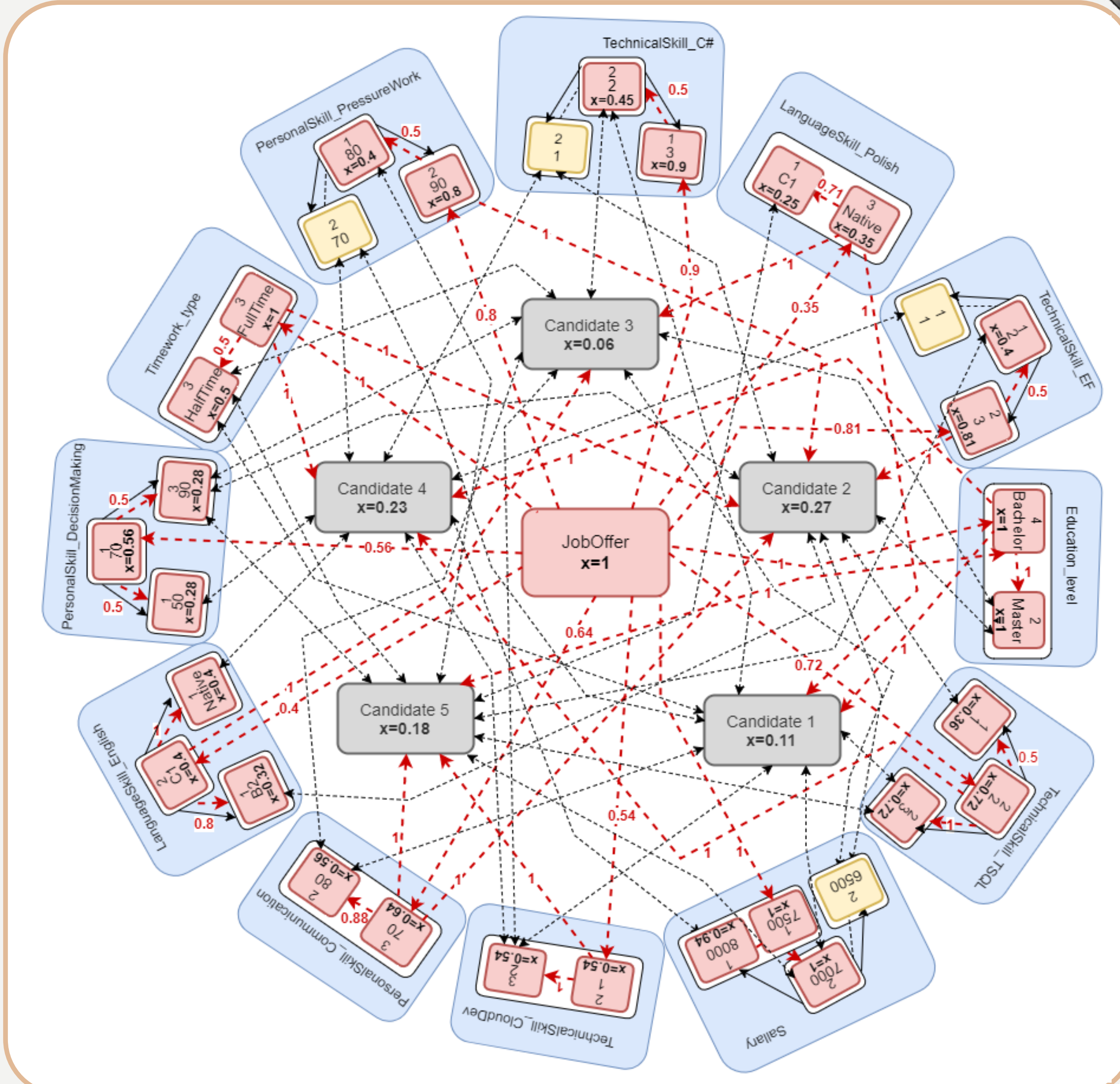
AGDS models and represents knowledge about the collected data and the strength of their relationships.

We will search for the most strongly associated candidate with the job offer in this graph structure.



Associative Inference in AGDS

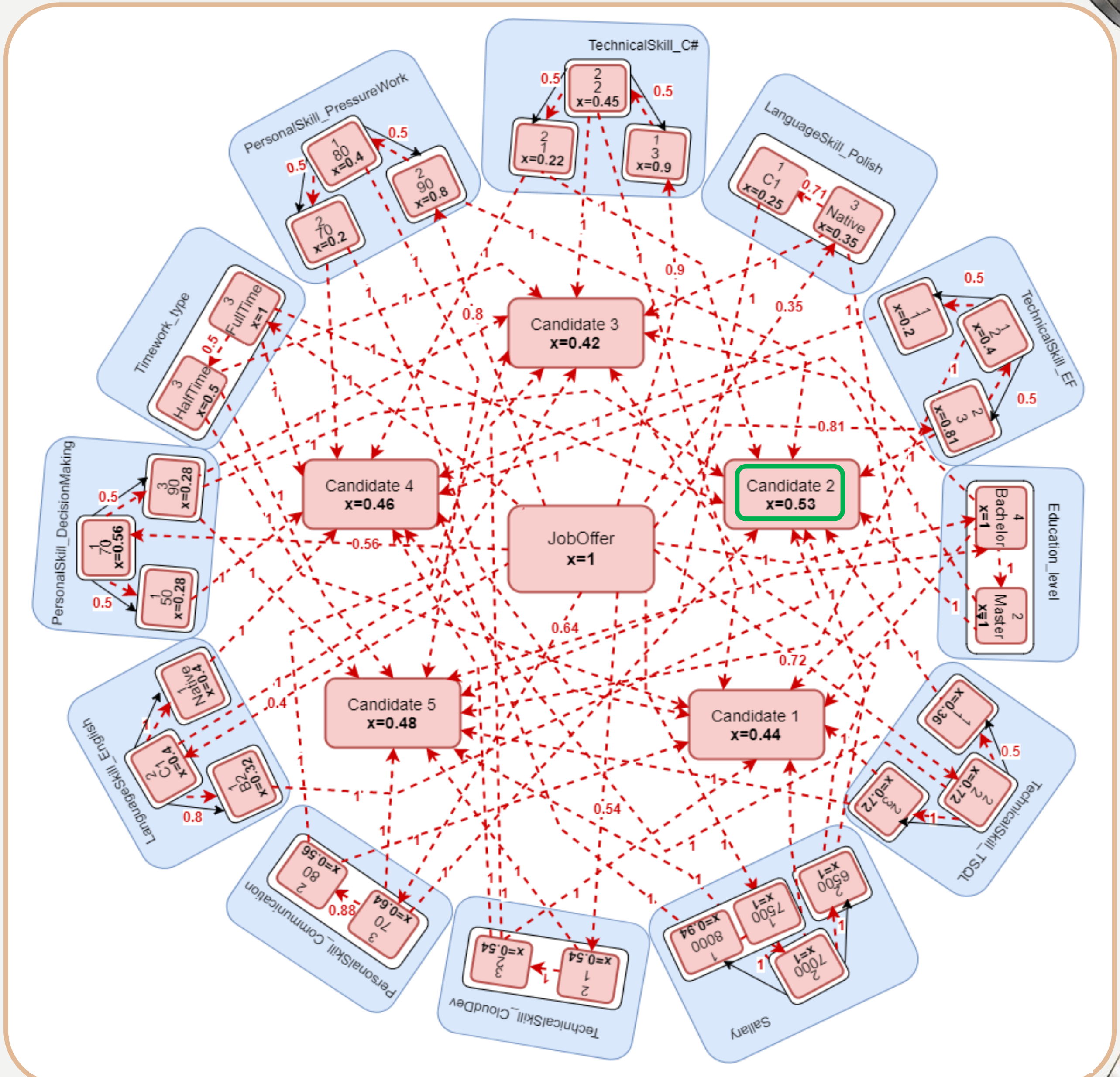
We activate node JobOffer and go along the edges in the BFS order, calculating the states of the connected nodes as a weighted sums of all stimulations, taking into account the connection weights.



Associative Inference in AGDS

We go along the edges in the BFS order, stimulating next nodes and calculating their states until at least one candidate node is stimulated by all connected input nodes.

Candidate 2 is recommended!



Associative Graph Networks for Sequential Data

Can we use associations to model sequences?

MONKEY

is an example of a set of facts describing this monkey:

"I have a monkey. My monkey is very small. It is very lovely. It likes to sit on my head. It can jump very quickly. It is also very clever. It learns quickly. My monkey is lovely. I have also a small dog."

What knowledge we have gained about this monkey on the basis of the above description? Now let's try to answer the following question: What is this monkey like?

KNOWLEDGE GRAPH

Construction of the Associative Neural Graph for the following set of sequential patterns:

1x	S1	I	HAVE	A	MONKEY				
1x	S2	MY	MONKEY	IS	VERY	SMALL			
1x	S3	IT	IS	VERY	LOVELY				
1x	S4	IT	LIKES	TO	SIT	ON	MY	HEAD	
1x	S5	IT	CAN	JUMP	VERY	QUICKLY			
1x	S6	IT	IS	ALSO	VERY	CLEVER			
1x	S7	IT	LEARNS	QUICKLY					
1x	S8	MY	MONKEY	IS	LOVELY				
1x	S9	I	HAVE	ALSO	A	SMALL	DOG		

Open Data
Parameters
Learn
Learn Monkey
Learn Slower
2
Learn Faster
Balance Graph
View -
100
View +
Ask
Ask Monkey
Ask Faster
5
Ask Slower
Weights
Background
Exit
Ask times:
1
INPUT:
(Enter a new sentence)
SIMTIME:
0
OUTPUT:
(Results will be brought back by the ANAKG after asking)
Google: Adrian Horzyk (http://home.agh.edu.pl/~horzyk/index-eng.php) to cooperate in developing associative knowledge-based artificial intelligence solutions: horzyk@agh.edu.pl

Associative Neural Systems

What does it mean associative?

Associative stands for

- **aggregated** – using various kinds of relationships like similarity, sequence, suppression, negation, inversion, …
- **related** – using various kinds of relationships like similarity, sequence, suppression, negation, inversion, …
- **actively** – because they should be triggered automatically if a given input context happens
- **with a defined strength** because relationships maybe differently frequent, may have different strength etc.
- **over time** because the objects and actions might be related nevertheless their real time of occurrences.
- **in a context** of the other objects represented in the neural associative system

BIBLIOGRAPY

1. A. Horzyk, J. A. Starzyk, J. Graham, Integration of Semantic and Episodic Memories, IEEE Transactions on Neural Networks and Learning Systems, 2017, DOI: 10.1109/TNNLS.2017.2728203.
2. A. Horzyk and J.A. Starzyk, Associative Fine-Tuning of Biologically Inspired Active Neuro-Associative Knowledge Graphs, In: 2018 IEEE Symposium Series on Computational Intelligence (SSCI 2018), IEEE Xplore, pp. 2068-2075, 2018.
3. A. Horzyk, Deep Associative Semantic Neural Graphs for Knowledge Representation and Fast Data Exploration, Proc. of KEOD 2017, SCITEPRESS Digital Library, 2017.
4. A. Horzyk, Associative Representation and Processing of Databases Using DASNG and AVB+trees for Efficient Data Access, In: Knowledge Discovery, Knowledge Engineering and Knowledge Management, Eds.: A. Fred, D. Aveiro, J. Dietz, K. Liu, A. C. Salgado, J. Bernardino and J. Filipe, ISBN: 978-3-030-15639-8, Springer Verlag, pp. 242-267, 2019, DOI 978-3-030-15640-4_13.

BIBLIOGRAPY

5. A. Horzyk, Neurons Can Sort Data Efficiently, Proc. of ICAISC 2017, Springer-Verlag, LNAI, 2017, pp. 64-74, ICAISC BEST PAPER AWARD 2017 sponsored by Springer.
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, 39-51.
7. Horzyk, A., How Does Generalization and Creativity Come into Being in Neural Associative Systems and How Does It Form Human-Like Knowledge?, Neurocomputing, 2014.
8. A. Horzyk, Innovative Types and Abilities of Neural Networks Based on Associative Mechanisms and a New Associative Model of Neurons - Invited talk at ICAISC 2015, Springer-Verlag, LNAI 9119, 2015, pp. 26-38, DOI 10.1007/978-3-319-19324-3_3.

BIBLIOGRAPY

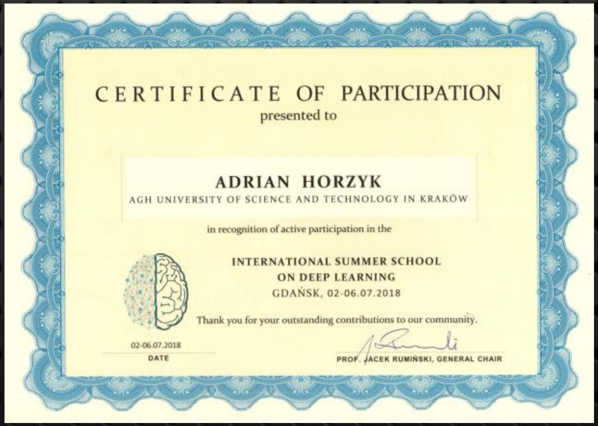
9. Basawaraj, J. A. Starzyk, A. Horzyk, Episodic Memory in Minicolumn Associative Knowledge Graphs, IEEE Transactions on Neural Networks and Learning Systems, Vol. 30, Issue 11, Nov. 2019, pp. 3505-3516, DOI: 10.1109/TNNLS.2019.2927106 (TNNLS-2018-P-9932), IF = 8.793.
10. Basawaraj, Janusz A. Starzyk and A. Horzyk, Lumped Mini-Column Associative Knowledge Graphs, IEEE Xplore, In: 2017 IEEE Symposium Series on Computational Intelligence, 2017
11. 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.

BIBLIOGRAPY

12. A. Horzyk, D. Bulanda, J. A. Starzyk, ASA-graphs for Efficient Data Representation and Processing, International Journal of Applied Mathematics and Computer Science (AMCS), Vol. 30, No. 4, 2020, pp. 717 - 731, DOI 10.34768/amcs-2020-0053, IF.
13. J. A. Starzyk, M. Jaszuk, Ł. Maciura, A. Horzyk, Concurrent Associative Memories with Synaptic Delays, IEEE Transactions on Neural Networks and Learning Systems, DOI 10.1109/TNNLS.2020.3041048, Accepted and Preview Dec. 2020, pp. 1 - 12, Open Access, IF.
14. A. Horzyk, A. Czajkowska, Associative Pattern Matching and Inference Using Associative Graph Data Structures, In: Proc. of 18-th Int. Conf. ICAISC 2019, Rutkowski, L., Scherer, R., Korytkowski, M., Pedrycz, W., Tadeusiewicz, R., Zurada, J.M. (Eds.), Artificial Intelligence and Soft Computing, Springer-Verlag, LNAI 10509, pp. 371-383, 2019, DOI 10.1007/978-3-030-20915-5_34

BIBLIOGRAPY

15. A. Horzyk and J.A. Starzyk, Fast Neural Network Adaptation with Associative Pulsing Neurons, IEEE Xplore, In: 2017 IEEE Symposium Series on Computational Intelligence, 2017.
16. Home page for this course: http://home.agh.edu.pl/~horzyk/lectures/ahdydci.php
17. Nikola K. Kasabov, Time-Space, Spiking Neural Networks and Brain-Inspired Artificial Intelligence, In Springer Series on Bio- and Neurosystems, Vol 7., Springer, 2019.



CERTIFICATE OF PARTICIPATION
presented to

ADRIAN HORZYK
AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY IN KRAKÓW

in recognition of active participation in the

INTERNATIONAL SUMMER SCHOOL
ON DEEP LEARNING
GDAŃSK, 02-06.07.2018

Thank you for your outstanding contributions to our community.

02-06.07.2018
DATE

PROF. JACEK RUMIŃSKI, GENERAL CHAIR