

INTRODUCTION

Background

Querying and analyzing datasets in existing solutions for data exploration and visualization are limited on either flexibility or ease of use. They are commonly also sensitive to structural changes of the datasets.

By providing a new solution based on graph technology, reflective logic and innovative index structures new opportunities are created in this area. This creates possibilities to work with complex cohorts and find previously unknown relationships.

Information Model

The new solution is using an information model consisting of different objects, called Holons, built as a graph.

Each Holon

- contains information that is kept in different fields
- belongs to a Holon type, which covers information of a single concept
- is related directly to at least one other Holon
- can be indirectly related to other Holons by following direct relations through Holons

Ontology

The Capish Information Model is based on an Ontology that sets the rules for the model and describes the definitions of all the entities in it. It defines how the Holons relate to each other, which fields are included in each Holon, the datatypes of the fields, units to be used, etcetera.

REFLECTIVE LOGIC - A NEW METHOD OF QUERYING A GRAPH DATABASE

Reflective Logic

Reflective logic is a new method used to formulate queries with a centrality, called Reflection Point, and to find all information for Holons of that centrality that fulfill the query. The method consists of the following steps:

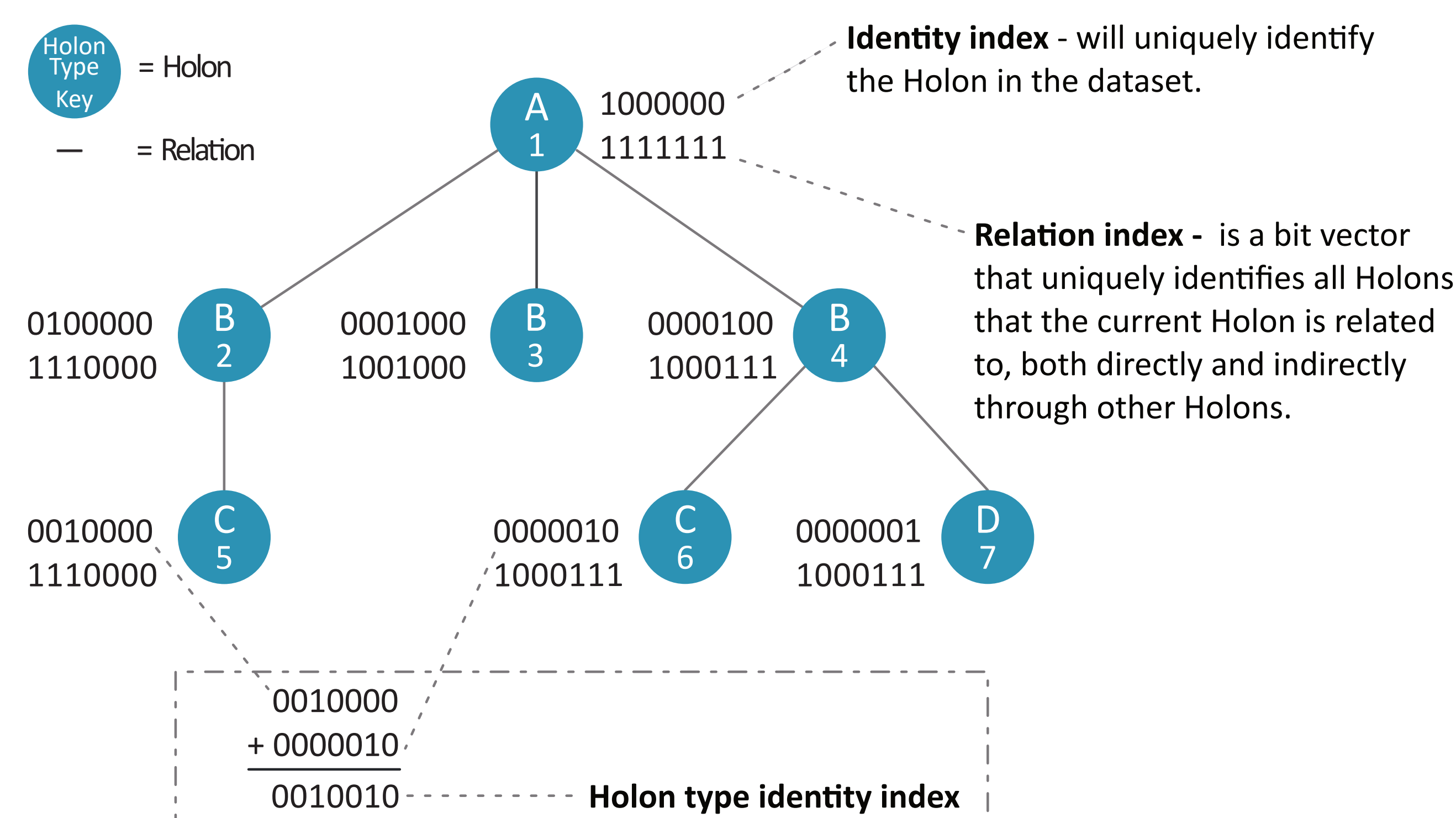
1. Create a subset of data
2. Find all Holons of the specified Reflection Point that are included in or related to the subset of data
3. Create a new subset of data from the complete dataset that contains the Holons identified in the previous step and all Holons related to those Holons

The subset of data from step 1 above is said to be reflected in the Reflection Point, which means that data is shown from a predefined perspective, during step 3. In clinical trial data a typical Reflection Point is the patient, but the study or the site could also be of interest. The result from a search with reflective logic will contain all Holons that are related to all Holons of the Holon type specified as the Reflection Point that match the query criteria.

Index Structures

A couple of different bit map index structures are created in order to handle both common queries and reflective logic.

Each index vector consists of bit vectors, where each bit may be either zero or one and each bit is mapped to a specific meaning. The length is equal to the total number of Holons.



- > **Identity index** - consists of index vectors that each identifies one Holon in the dataset.
- > **Relation index** - consists of the combination of all identity index vectors of the Holons that the current Holon is related to.
- > **Holon type identity index** - consists of one index vector for each type of Holon, where each index vector is created from the combination of the identity index vector of all Holons of that type.

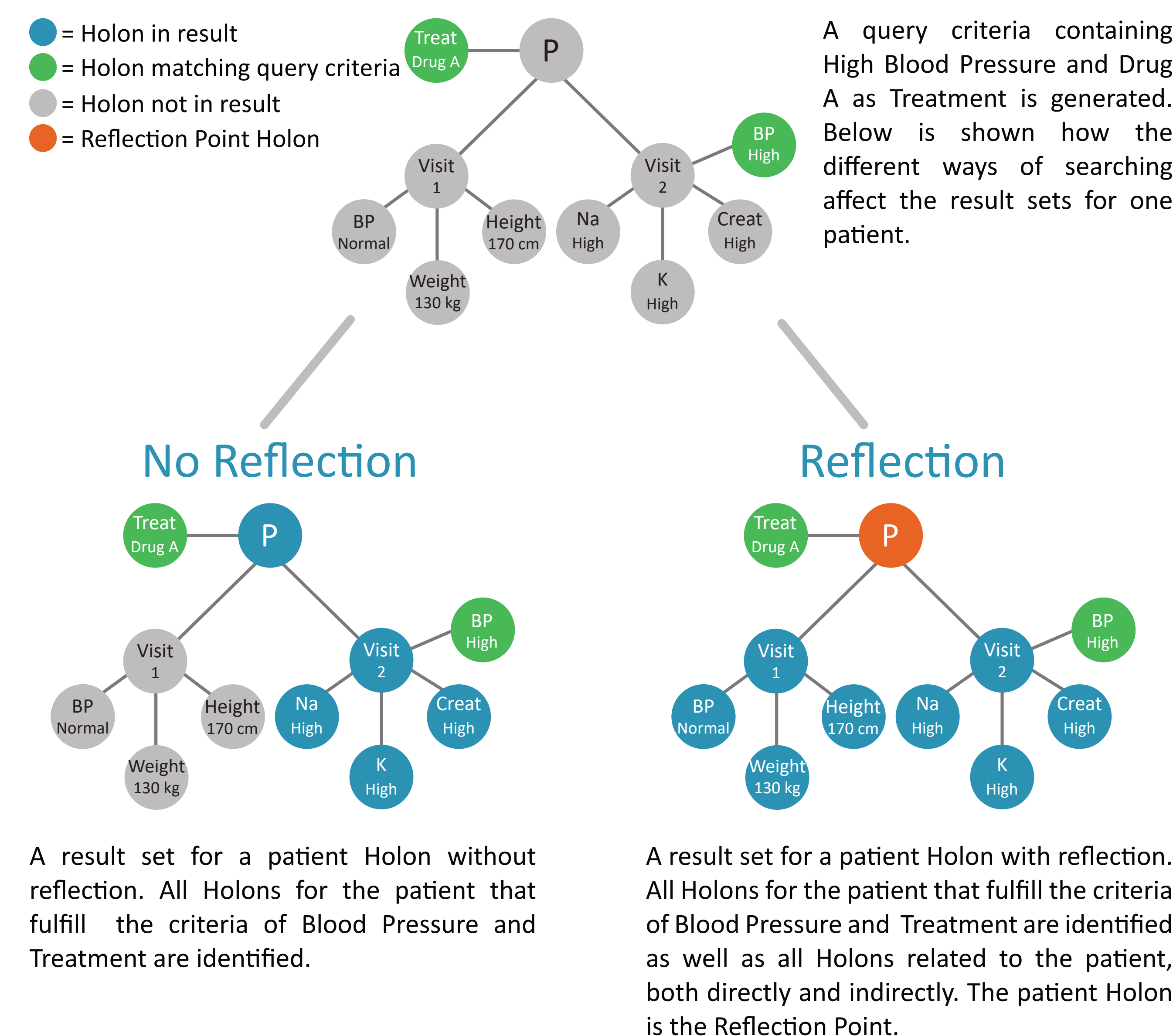
Use of Indices

The database engine utilizes the three index structures in order to answer queries and process relationships between Holons.

The relation index is used to identify all data connected to e.g. a particular patient in a single step. Holon type identity index structures are used to find which Holons of a specific type that are present in a subset of data.

Searching through Indirect Relationships

By utilizing the power of the ontology-based information model, the resulting graph database can be searched in different ways. The illustration below shows the difference between searching with and without reflection.



Use Cases

A search for a value could potentially not only retrieve all the information units containing that value, but could also be configured to retrieve all directly or indirectly related information. Examples of questions:

- > What other adverse events have been reported for patients with elevated liver values who received drug A?
- > Which female patients that were given a dose of drug A have had high blood pressure measurements during episodes of severe headache?

CONCLUSION

Summary

The user can specify what kind of information that is of interest and what criteria that information must fulfill and does not need to know how the datasets are structured.

Queries over complex data, that would normally require deep graph traversal, are quickly answered using the index structures.

By utilizing reflective logic it is possible to control from which vantage point the data should be evaluated, e.g. patient perspective or study/site perspective.

Practical Use

The new paradigm on how to effectively query and retrieve the information from a dataset in a way that exploits the existence of direct and indirect relationships in the Capish Ontology, has been taken advantage of in a new data exploration tool, Reflect. The indexed datasets in Reflect allows the end-user to ask and retrieve answers to complex questions promptly and returns a broader result set depending on the query.

The advantage of this graph retrieval approach grows with the complexity of the data and the questions.

Reflective logic
unlocks knowledge
in datasets