Technical Disclosure Commons

Defensive Publications Series

May 2020

The Atom of Knowledge: Using Info Triples to Build Info Graphs - Short Version

U. Nicolaisen

The Atom of Knowledge: Using Info Triples to Build Info Graphs - Short Version

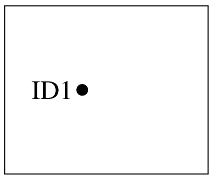
U. Nicolaisen

Abstract—As Artificial Intelligence is developed to be more useful and used, expressing knowledge in a generalised, machine friendly way becomes important. In this article we start with an empty universe and from there set out to find the simplest possible structure to represent knowledge. We claim to find the atom of knowledge and give it the formal name Info Triple. Furthermore we use this structure to build a special kind of graph which we call Info Graph. The Info Graph can be used to model information context. We discuss the maths behind Info Graphs and compare it to existing standards. We compare the intended use of Info Graphs with the way graphs are currently used to model graph databases. Finally we conclude that the use of Info Graphs makes the models more generalised and therefore better suited for AI readers. Thus we believe our findings can serve as a valuable theoretical backbone for future knowledge base implementations.

Keywords—AI, graphs, info graph, info triple, information modelling, knowledge base

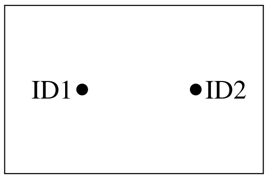
I. THE ATOM OF KNOWLEDGE

Imagine a completely empty universe. Then we introduce an object. This object can be anything, big or small it does not matter. What we solely focus on is that an object exists. We can illustrate this graphically by drawing a single dot representing the object. We can label the dot with the text 'ID1' and use this text as identifier.

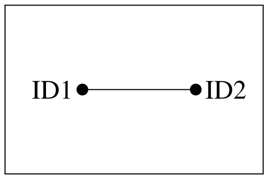


Though our dot in the diagram above can represent any possible object there is, we cannot deduct any information about the object from the diagram other than the object's existence. Even that knowledge is not really knowledge when we don't know what it is that exists. We just have an arbitrary ID that we gave it.

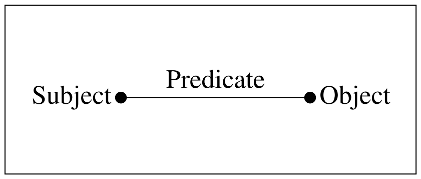
To describe the object, in the simplest possible fashion, we introduce another object which we could call ID2 and we can illustrate it with another dot. Let ID2 be an object that has some kind of relation to ID1 that makes it describe ID1 to some degree.

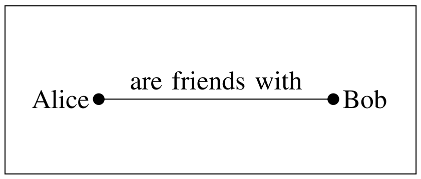


Since this relation between ID1 and ID2 now exists, the relation itself must be a third object in our otherwise empty universe. To show graphically that the relation exists we can draw a line between ID1 and ID2 and hence, our dots become vertices connected by an edge forming a graph.



Now that we have identified a graph as known from discrete mathematics, it could be tempting to continue that path and look at how graphs are normally used to illustrate knowledge, for example for later use in a graph database.





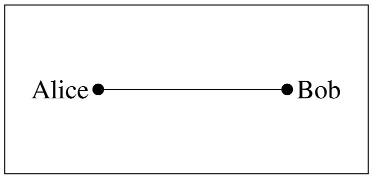
Two existing notable types of graphs used widely for databases are RDF knowledge graph and labeled property graphs.

The RDF knowledge graphs have a directed edge from a subject to an object and the edge is marked with a predicate describing the relationship. The IDs of both vertices and edges are close to human readable, often containing detailed information about the vertex or edge through a long ID in URL format. Some vertices are finite values. That kind of vertex is denoted with a square.

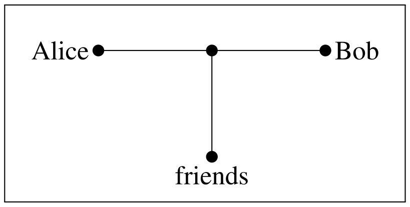
In a labeled property graph the edge is also directed. Here however, the edge and vertex are basically a black boxes where you store all data about the objects as sets of key value pairs. This makes the graph's structure simple and therefore efficient to process. This simplicity is also its downside since the graph structure itself simply isn't that detailed.

None of the graph types above is useful for what we are looking for namely representing the smallest amount of knowledge there is. The RDF graph comes closest, but a simple RDF graph connecting two vertices contains multiple pieces of knowledge at the same time: The existence of the edge itself provides the first information. Besides that the URLS used to name the vertices and edge may each contain one or more pieces of information. Also the fact that the graph is directed gives extra semantic meaning to the edge. That is not what we are looking for.

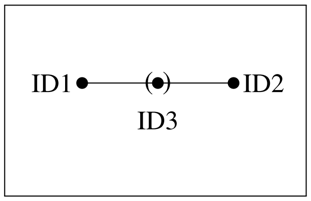
Let's move back to our initial, empty universe. If we were to do the Alice and Bob example then Alice would be object1 and Bob would be object2 and their relationship would be object3. Here object3 contains no information other than stating that there is a relationship between Alice and Bob.



Now we need a way to describe the relationship itself, the edge, in the same atomic fashion as it has been done up until now. And here we make what might be perceived as a rather bold move. Since we have identified the edge as an object with an ID like our vertices have, we perceive the edge as being a subtype of the type vertex. This subtype is enhanced with the IDs of the two vertices it connects. By defining an edge that way, the edge can be treated as a vertex in our diagram and be connected with other vertices through new edges.



To make the graph more readable we introduce parentheses to the graph. In the diagram below the edge is marked with a vertex. We then surround this edge's vertex with parentheses. This is done so the graph can be distinguished from a graph of three vertices connected by two edges.



The diagram above shows a general representation of what we believe to be the atom of knowledge.

II. Info Graphs

This special kind of graph, were edges can be used as vertices we name Info Graph. The smallest Info Graph is simply two vertices connected with an edge. We call this graph an Info Triple because, if we were to represent the graph as a record in a dataset, it would be a triple consisting of the relationship ID, the ID of vertex one and the ID of vertex two.

An Info Graph can be said to be composed of one or more Info Triples. Thereby can an Info Graph be stored as a table of Info Triples were the relationship ID is the key as this ID is unique for each Info Triple. To have consistency in our terminology we call a table of Info Triples for an Info Table.

triple identifier	*identifier1*	*identifier2*
ID3	ID1	ID2
ID5	ID3	ID4
ID7	ID5	ID6

The diagram above shows an example of an Info Table containing three Info Triples.

III. Conclusion

In conclusion we have found what we believe to be the smallest possible and most general structure to store knowledge. That should make Info Tables an attractive basis for a niche of database or knowledge base systems. Furthermore the Info Graph should be useful for modelling information and therefore we have laid the ground for potential graphical user interfaces as well. The purpose of the Info Graph is not to store knowledge in the most search efficient way or the most human readable way, but in the most uniform and simplistic way. Thus by using Info Triples, all knowledge there is can theoretically be stored in a single table with three columns.