Diagrammatic Description Logic

Definition 1 (Graph).
A graph(V,E) consists of a set V of *vertices* and a set E of *edges*.
Every *edge* in E is a pair of *vertices* from the set V.
Note that our graphs are always directed.
It means that each E element is a pair (v_s,v_d) where v_s is the source vertex of the edge and v_d is the destination vertex of the edge.

Labels. An essential graph equipment is labels. Unlabeled graph are useless abstract graphs. Concrete graph are always labeled. Either only the vertices are labeled, or both vertices and edges are labeled. Edge labels (is they exist) support an information about the connection from the source vertex to the destination vertex.

Example. An utterly abstract graph. Yet, just label each vertex by a character and it becomes a lexical automaton that parses only the English words *race*, *rat*, *ray*, *rice*, *ride*, *role*, *rome*, *rope*, *rose*, *row*, *rude*, *rule*.

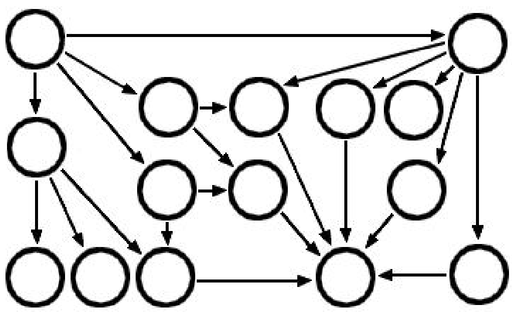

Fig. 1 The abstract graph

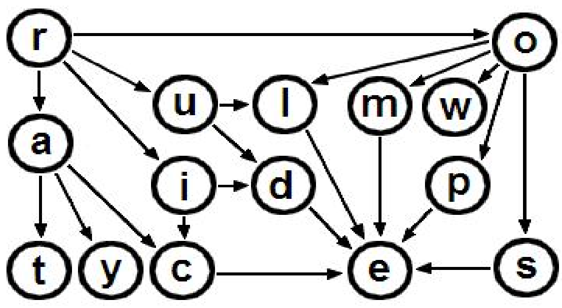

Fig. 2 The same vertex-labeled graph

Definition 2 (Multigraph).
Because they are eventually multiple pertinent connections from the source vertex to the destination vertex, sometimes multiple edges from the same source to the same destination are needed, each supporting one label.
A multigraph(V,E) is a graph(V,E) where E is a multiset.

Example.
Sarah is the spouse and colleage of John.

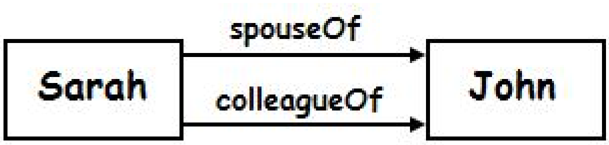

Fig. 3 Finally edges are labeled too

Another example.
Estimated cost but largeur exact revenue.

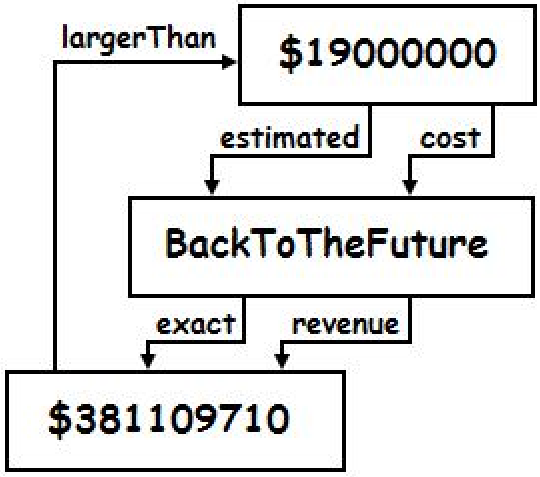

Fig. 4 Great benefits

Definition 3 (Cycle).
A graph path that starts and ends with the exact same vertex is a cycle.

Example.
John and Sarah are husband and wife.

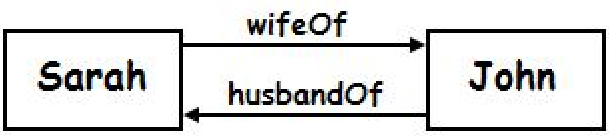

Fig. 5 We go full circle

Definition 4 (DAG).
A graph without a cycle is a directed acyclic graph or DAG.

Definition 5 (Semantic Net).
A semantic net is a multigraph whose vertices are labeled with concepts and whose edges are labeled with inter-concept relations.
A prominent inter-concept relation is the `isA` relation.
The `isA` relation asserts that the source concept is a subtype of the destination concept. As a subtype relation the `isA` relation is transitive and forms a DAG.

Semantic net is the first historically acclaimed, and widely used, graph formalism that allows some powerful knowledge representation (KR). Starting with KL-One, a KR language family finally distinguished itself from the venerable Lisp and Prolog. The `isA` subtype relation leads to the notion of graph-homomorphism which is today recognized as a privileged tool to request and manipulate KR databases.

Example.
Socrates life and intellectual legacy.

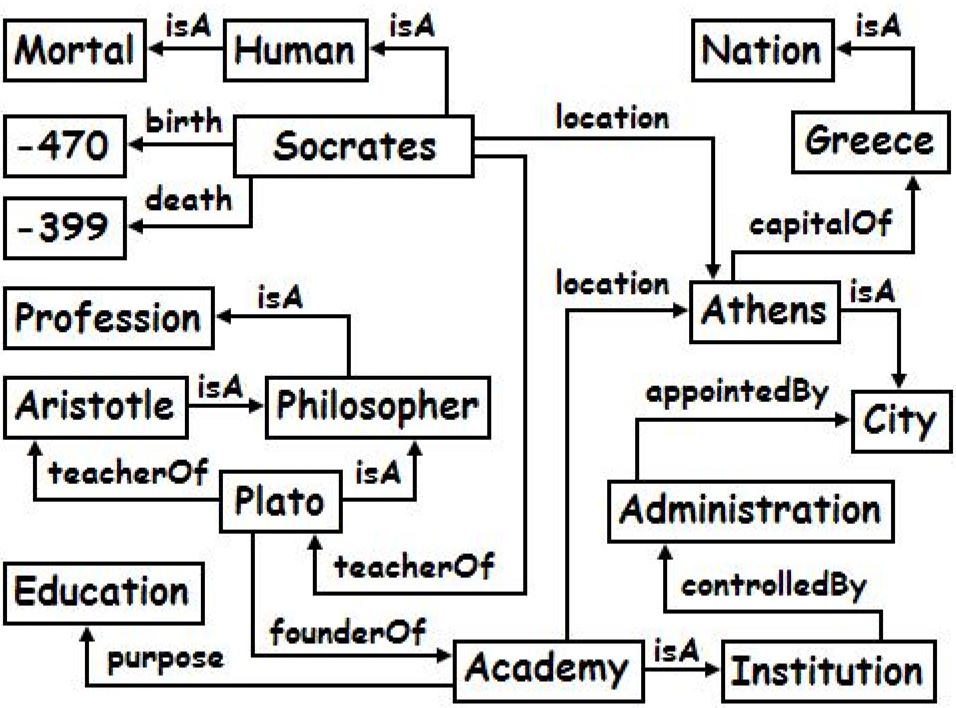

Fig. 6 Socrates

Definition 6 (Graph homomorphism).
A graph homomorphism H is a mapping from a graph G(V,E) to a graph G'(V',E') so that :

- informally : every image (by H) of an edge of G is an edge of G'
- formally : if $(v_s,v_d) \in E$ then $H(v_s,v_d) = (v_s',v_d') \in E'$

Example.
As graph homomorphism is an abstract thing it deserves an abstract example.

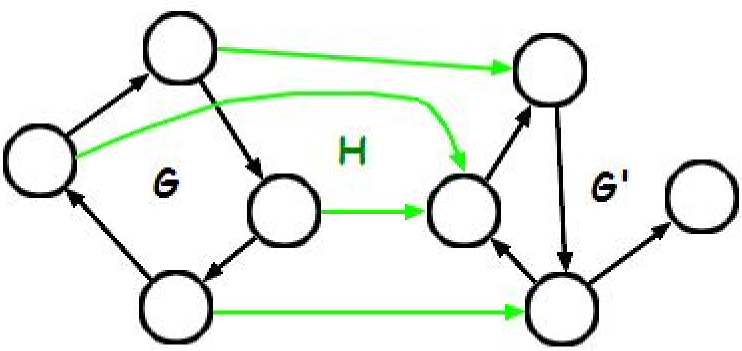

Fig. 7 H is a mapping from the abstract graph G to the abstract graph G’

Definition 7 (Graph subsumption).
A graph G subsumes a graph G' if there exists a graph homomorphism H from G to G’ and H matches the `isA` inverse relation.

Example.
A cat is on a mat.

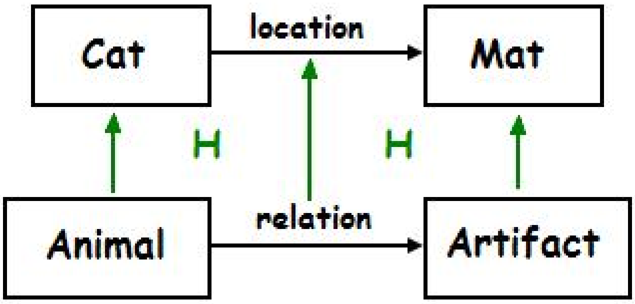

Fig. 8 H is a homomorphism from bottom to top

It is quite straithforward that :
● A cat is a kind of animal.
● To be on something is a kind of relation.
● A mat is a kind of artifact.
● H is a graph homomorphism from bottom to top.
Hence the bottom graph subsumes the top graph.

Definition 8 (Bipartite graph).
A bipartite graph is a graph whose vertices can be divided into two disjoint sets V_1 and V_2 (that is, V_1 and V_2 are each independent sets) such that any edge connects a vertex in V_1 to one in V_2 or a vertex in V_2 to one in V_1.
Graphically.
It is common practice to draw one kind of vertices as square boxes and the other kind of vertices as rounded boxes. Hence, in a bipartite graph drawing, a square box can only connect to rounded boxes, and a rounded box can only connect to square boxes.

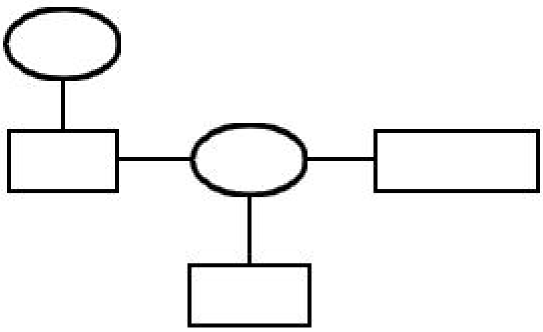

Fig. 9 An abstract bipartite graph

Definition 9 (Conceptual graph).
A conceptual graph (CG) is a bipartite multigraph whose square vertices are labeled with entities and whose rounded vertices are labeled with inter-entity relations.
Apart that, a conceptual graph features an implicit `isA` relation and works much like a semantic net, especially regarding graph subsumption capabilities. The `isA` relations used by a conceptual graph are made explicit by the graph database *ontology*. The graph vocabulary is also stored by the ontology.

Example.
A rich man gives some cash to a beggar.

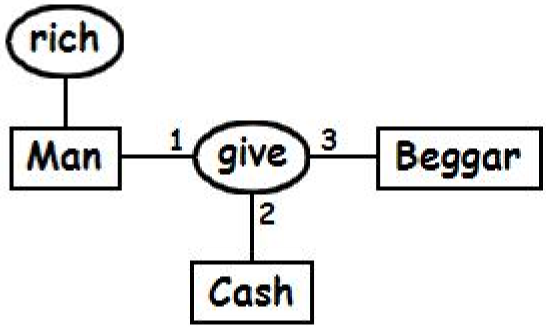

Fig. 10 A simple conceptual graph

The facts that the edge n°1 is the giver, the edge n°2 is the gift and the edge n°3 is the beneficiary are recorded in the graph database ontology.

Definition 10 (Nested conceptual graph).
A nested conceptual graph is a conceptual graph whose square vertices can contain further nested conceptual graph. Eventually, some generalized nested conceptual graph formalism can allow edges to traverse square box boundaries in any manner (from inside to outside, from outside to inside, whatever the nesting level). Nested graphs provide a direct visual way to decorate the information with the associated context.

Discussion.
Semantic nets are now a somewhat deprecated way of KR. On the practical side they have been superceded by the RDF-OWL web standard. On the theoritical side they have been superceded by (nested) conceptual graphs that provide a better basis for everything from formal semantic to formal reasoning assistance.

Example.
Eva dreams about a clown show. A sponsor makes her an offer of a man to act as the clown and the funds to finance the event.

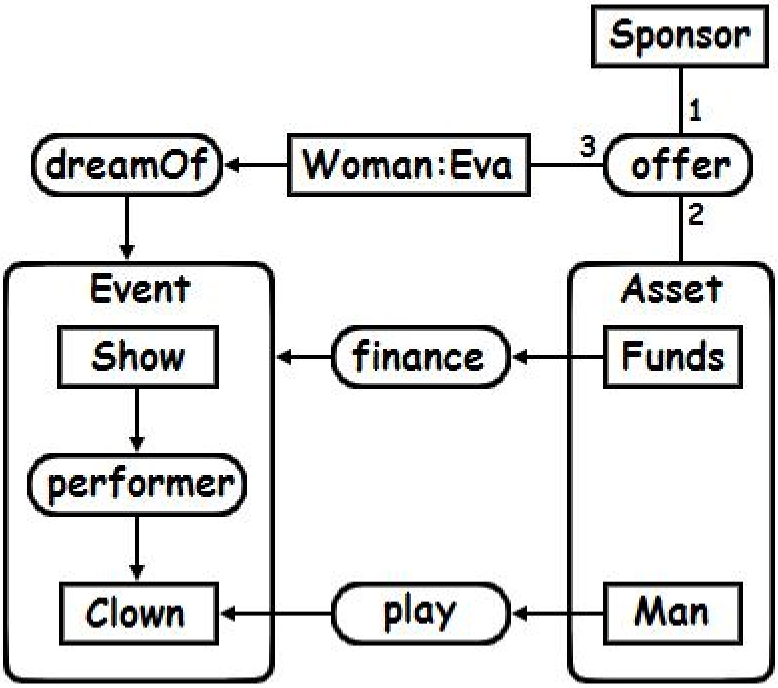

Fig. 11 A generalized nested conceptual graph

Discussion.
Some people argue that generalized nested conceptual graphs are no more graphs but free full diagrams. Anyway, even if so, then generalized nested conceptual graph is the first attempt to formalize free full diagrams.

Definition 11 (Coreference link).
A coreference link is an equality relation connecting 2 entities across 2 contexts.
Graphically. The coreference link is usually drawn as a dotted line connecting two square boxes. Coreference links are transitive.
Motivation. In a nested conceptual graph, a same entity box can appear in multiple contexts. Would this equality not be explicited by coreference link(s), the reader could misbelieve that each separate box is a fresh new entity.

Example.
A woman looks a photograph with her at Venise.

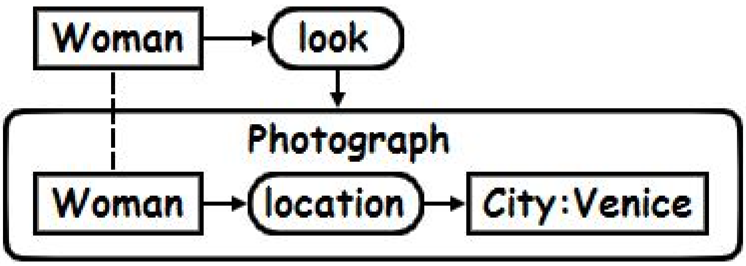

Fig. 12 A nested conceptual graph with one coreference link

Discussion.
One can argue that there is actually no woman on a photograph, but merely a woman portrait. Thus the woman and the woman picture are not really the same entity as suggested by the coreference link. For example the woman portrait has red eyes and it's not a woman property, it's a portrait property. However a coreference link is only equality modulo context, and the pictured woman is the same as the looker woman.

Another example.
The man that wants to kill his dog tells he has rabies.

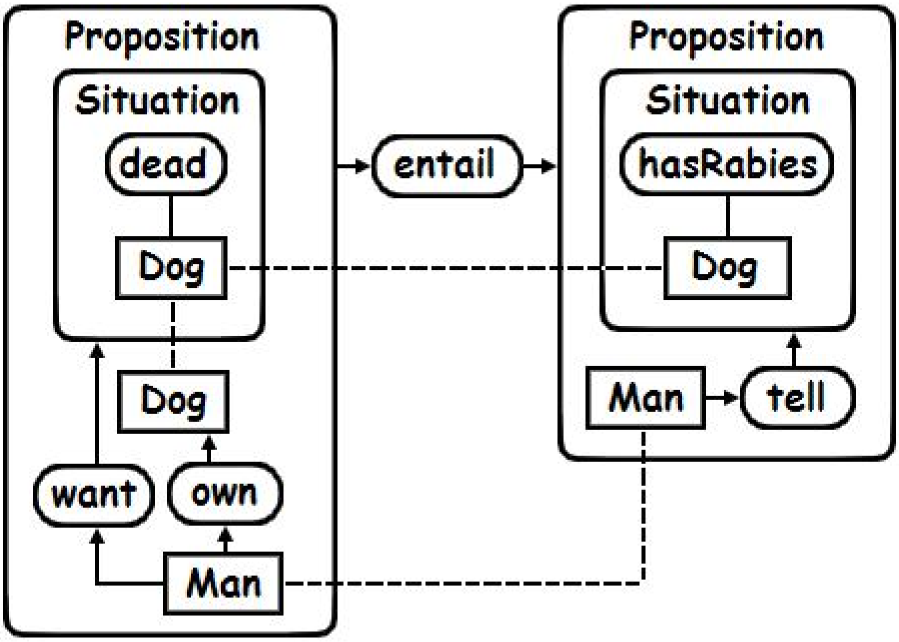

Fig. 13 A nested conceptual graph with 3 coreference links

Definition 12 (Difference link).
A difference link is a relation that denies the equality between two entities.
Graphically. The difference link between two square boxes is usually drawn as a line with a double stroke reminding the ≠ symbol.
Motivation. Difference links are most useful to avoid unwanted subsumption.
Example.

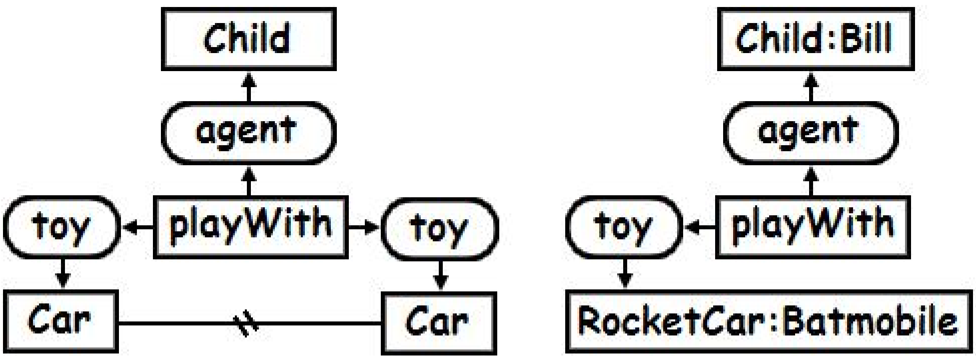

Fig. 14 Without the difference link, the left CG would subsume the right CG

Definition 13 (Negated relation).
A negated relation can be used to deny some relation with one (or more) entity.
Graphically. A relation with square boxe(s) can be negated by the ¬ symbol.
Motivation. Negated relations are most useful to avoid unwanted subsumption.

Example.

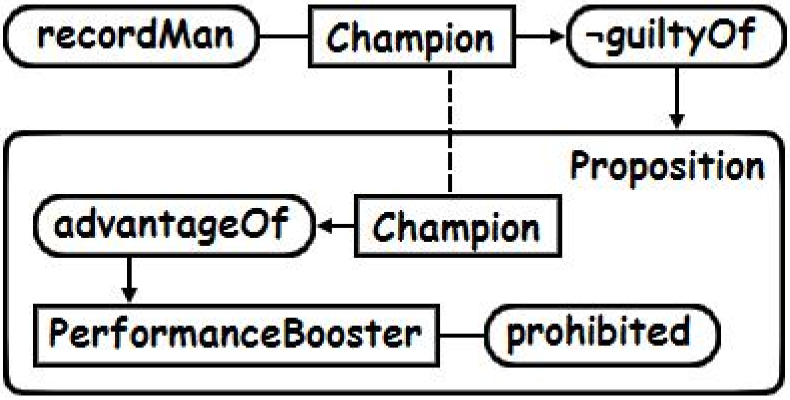

Fig. 15 How a sponsor selects a champion recruit

Definition 14 (Hyper-graph).
An hyper-graph(V,E) is a graph(V,E) whose edges are tuples (v_1,v_2,...,v_n) rather than merely pairs. Thus an *hyper-edge* can connect many vertices. A monadic edge is an edge that decorates 1 vertex. A dyadic edge is an edge that connects 2 vertices. A triadic edge connects 3 vertices.

Example.

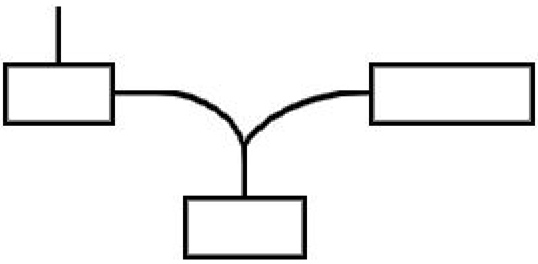

Fig. 16 An abstract hyper-graph

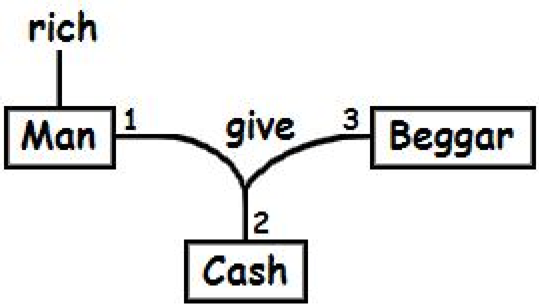

Fig. 17 A labeled hyper-graph

Discussion.
Let's stop a minute. Is the **Fig. 17** the same conceptual graph as the **Fig. 10** ?
The answer is yes, it is. So why is there two underlying graph models ?
The distinction lies in the usage :

- A bipartite graph is easier to draw as a rounded box is easier to label than a simple line.
- An hyper-graph is easier to code as you just have vertices and hyper-edges. You don't have to clutter the code with two kinds of vertices.

A CG-tool user will usually consider CGs as bipartite graphs. Users like objects.
Users ask questions like *should i use a square box or a rounded box ?*
They don't ask questions like *should i use a vertex or an hyper-edge ?*
So hyper-graph is a notion aimed at the CG-tool programmer.