Entities and Surrogates in Knowledge Representation

Michel Chein and Michel Leclère

RR LIRMM-GraphIK January 20, 2011

Abstract

This is a DRAFT version.

1 Introduction

In a universe of discourse (or world), names are privileged accesses to information concerning entities in this universe of discourse. A name is generally a simple expression, e.g., for a person it can consist of a list of first names and family name(s), which is not necessarily sufficient to identify a person. Names, in outside world, are usually ambiguous references and additional information are needed in order to correctly identify a world entity.

The Unique Name Assumption (UNA) does not concern names in the usual meaning but identifiers within a computer system, and a name is simply an attribute of an identifier. UNA states that two different identifiers correspond to two different world entities. UNA is a fundamental property assumed in classical databases and also usually assumed in knowledge representation (e.g., in Description Logics [] or in Conceptual Graphs []). Assuming UNA implies that the designer of a base has to assign identifiers to elements of the world, in a way he has to solve identification problems before constructing its system.

UNA cannot be assumed in some situations, for instance for linking or merging databases independently built, and various identification problems have been considered in computer science since a long time (1959):

- *Entity resolution, Reference reconciliation*: which records represent the same entity, identifying multiple refs to the same object and distinguishing them from mentions of different objects
- *de-duplication*: duplicate data (object) is deleted (Classical problem: creation of mailing lists)
- *merging* records judged to represent the same world entity
- *record linkage*, linking records through refs to same world entities
- *object identification* is used as a generic term to single out, to distinguish, to recognize individual entities (objects) of the world in a computer system.

Most solutions of object identification problems, which are important for many kinds of databases (especially data warehouses) and also for (part of) the web, are based on classification techniques. In this approach, an entity is described by a list of attributes; attribute values are simple data types (e.g., strings or numbers) and approximate similarity measures are assigned for each kind of attributes vector; a similarity measure is built for lists of attributes (often it is a weighted combination of the attribute similarities); finally, a decision procedure allows to state when two lists of attributes represent the same entity. Recently, logical approaches have been developed (e.g., Fatiha Saïs, Nathalie Pernelle and M.-C.

Rousset consider data sources conforming to the same RDFS schema [SPR07, SPR09]), our work also follows a logical approach.

A knowledge base (KB) is composed of logical constructs representing knowledge about a world. We assume that such a world can be modeled by using notions of (nave) set theory, e.g. elements, sets, relations, functions, and thus that knowledge about this set theoretical model are expressed by (a variant or a fragment) of first order logic. In this paper we are interested by identification problems concerning individual entities in world. These individual entities are represented by elements in the model and these elements are represented by special symbols called surrogates in the knowledge base. Surrogates should be in one-to-one correspondence with the set of world entities. Our aim is to define a KB faithful, or unambiguous, with respect to the identification of world entities and to propose mechanisms allowing a user to design, build, maintain such a KB, and also to repair an ambiguous KB. In this paper we consider individual entities, nevertheless similar identification problems occur for other notions such as classes of entities or relationships between entities.

We focus on digital libraries, which are specific knowledge bases, for several reasons. First, all the previous mentioned problems are important in the context of digital libraries. Some problems are induced by the evolution of digital libraries (e.g., adding notices to a base, merging bibliographic bases, maintenance of the different bases), other problems concern the quality of the notice bases (e.g., consistency inside and between bases, relevance of the subject). Secondly, a digital library is composed of several bases of notices: bibliographic notices and authority notices. A bibliographic notice gathers metadata concerning a document, an authority notice gathers metadata concerning an authority. An authority can be an author, a collectivity or a subject. International work is done to standardize these metadata (FRBR, ?) similar to light weight ontologies. Relationships between bibliographic notices and authority notices are represented by attributes in notices (e.g., authorOf, editorOf, ?). Thus, a digital library can be seen as a knowledge base having a graph structure. The problems expressed in our logical framework can be solved using graph techniques (rules, constraints, exact and approximate retrieval procedures cf. [CM09, **?**, **?**]?). Digital libraries allow to assess graph methods on large graphs (several dozen of millions of nodes) and to compare them to classification methods.

2 Logical Framework

In this section we present a first order logical framework aiming at representing object identification problems.

2.1 The language $\mathcal{L}$

The KR language ($\mathcal{L}$) used is a variant of FOL and more precisely a variant of the language $\mathcal{L}$ introduced by Levesque and Lakemeyer [LL00]. Terms are used for representing entities in the application domain. We assume that there are no functional symbols in the KR language.

Definition 1 (Symbols and Vocabulary) *The set of symbols is composed of a set of logical symbols and a set of non logical symbols also called* vocabulary. *The set of* logical symbols *consists of the following distinct sets:*

- X *a countably infinite set of variables,*
- S *a countably infinite set of surrogates, which is partitioned into an infinite set of* literal surrogates *and an infinite set of* individual surrogates*,* $S = L + E$*,*
- *the equality symbol* $=$*,*
- *the connectives* $\exists, \forall, \wedge, \rightarrow, \leftrightarrow$ *and the brackets* (*and*)

The vocabulary *or set of* non-logical symbols *consists of two distinct sets:*

- C *the (ordinary) constants (i.e., function symbols of arity* 0*).*
- P *the predicate symbols (of any arity* > 0*)*

Definition 2 (Terms) *A term is either a variable or a surrogate or a constant.*

Literals surrogates (literal in short whenever there is no ambiguity) are used for representing entities such as integers, strings, dates and so on, e.g., `int 3` represents the integer 3 and `string abc` represents the string "abc". We assume that there is a consensus about the meaning of a literal, i.e., a literal does not need a definition, its meaning is directly given by its syntax. This is not the case for individual surrogates (individual in short whenever there is no ambiguity). An individual has no meaning by itself, it is simply a symbol; individuals are noted by strings beginning by #, e.g., $\#12$. The fundamental property of individuals is that they represent the elements of the universe of discourse. Thus, we only know that $\#12$ and $\#25$ represent different entities in the universe of discourse and if one wants to express what distinguish $\#12$ from $\#25$ one has to represent knowledge concerning the individuals $\#12$ and $\#25$. A constant corresponds to an alias for an individual, e.g., abc or $Charles\_De\_Gaulle$ or $Le\_General$ can be aliases for $\#121443$ which is the individual representing in the system the French well-known personality. Constant are noted by strings disjoint from surrogates and variables. Please note that the $Charles\_De\_Gaulle$ constant is a non-logical symbol different from the literal surrogate `String Charles De Gaulle` which might represent the name of the person represented into the system by the individual surrogate $\#121443$.

Definition 3 (Formulas) *An* atom *is either* $p(t_1, ..., t_k)$*, where p is a predicate and* t_i *is a term, or* $t = t'$ *(equality atom), where* t *and* t' *are terms. Formulas in* $\mathcal{L}$ *are defined from atoms and connectives in the same way as in classical FOL. We mainly use specific formulas defined as follows:*

- *A* conjunct *is a conjunction of atoms. A conjunct can be identified as a set of atoms.*
- *An* $\exists\wedge$*formula is the existential closure of a conjunct without equality atoms.*
- *An* $\exists\wedge =$*formula is the existential closure of a conjunct.*
- *A range restricted rule (rr-rule) is a formula* $\forall X(H \rightarrow A)$ *where* H *is a conjunct without equality atoms,* A *is an atom and* $var(A) \subseteq var(H) \subseteq X$*. Two kinds of rr-rules are distinguished:* P*-rule where* A *is an atom built from a predicate in* P *and* $=$*rule where* A *is an equality atom.*
- *An individual definition for an individual* $e \in E$*, is a formula* $\forall X(H \leftrightarrow (x = e))$ *where* H *is a conjunct without equality atoms,* $x \in X$*.*

In the examples, the conjuncts, $\exists\wedge$formula and $\exists\wedge =$formula will be represented by their set of atoms. A knowledge base (KB) is composed of three sets of formula. More precisely,

Definition 4 (Knowledge Base) *A knowledge base over* $\mathcal{L}$ *is a 3-tuple* $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{C})$ *composed of three finite sets:*

- *a set of* $\exists\wedge =$*formulas* $\mathcal{F}$*, representing facts,*
- *a set of rr-rules* $\mathcal{R}$ *without constants and individuals, representing knowledge which can be applied to facts,*
- *a set of* $\exists\wedge$*formulas* $\mathcal{C}$ *without constants and individuals, representing negative constraints that facts should fulfill.*

Note that, using conjunction, $\mathcal{F}$ can be assumed to be a single $\exists\wedge =$formula.

Notations The set of variables, literals, individuals, surrogates, constants, terms, atoms in a formula A are, respectively, denoted $var(A)$, $lit(A)$, $ind(A)$, $surr(A)$, $const(A)$, $term(A)$ and $atom(A)$. These notations are also used for a set of formulas or for a knowledge base.

Definition 5 (A^*) *For any set of terms T, T^* is the set of all equality between terms in T, (i.e., $T^* = \{t = t' \mid t \in T, t' \in T\}$.*

Let A be an $\exists\wedge =$formula. Let $C_1, C_2, \ldots, C_n$ be the equivalence classes of $term(A)$ (wrt the equality), and for any term $t \in term(A)$, let $C(t)$ be the equivalence class of t.

$A^* = \{p(t'_1, ..., t'_k) \mid p(t_1, ..., t_k) \in atom(A), \forall i = 1, \ldots, k, t'_i \in C(t_i)\} \cup \{C_i^* \mid i = 1, \ldots, n\}$

Definition 6 (Normal form of a formula) *For any equivalence class of terms T, $norm(T)$ is the subset of T defined as follows:*

- *If T only contains variables, then let x be a variable in T, $norm(T) = \{x\}$ (one keeps one variable).*
- *If T contains (at least) a variable and (at least) a constant or surrogate, then $norm(T) = T \setminus X$ (all variables are removed).*
- *If T only contains constants or surrogates $norm(T) = T$.*

Let A be an $\exists\wedge =$formula. Let $C_1, C_2, \ldots, C_n$ be the equivalence classes of $term(A)$ (wrt the equality), and for any term $t \in term(A)$, let $C(t)$ be the equivalence class of t. The normal form of A, noted $norm(A)$ is defined by: $norm(A) = \{p(t'_1, ..., t'_k) \mid p(t_1, ..., t_k) \in atom(A), \forall i = 1, \ldots, k, t'_i \in norm(C(t_i))\} \setminus \{t = t \mid t \in term(A)\}$.

Note that there is no equality atoms containing variables nor valid atoms ($t = t$) in $norm(A)$. Thus, if A is a set of valid atoms, $norm(A)$ is the empty set of atoms. Furthermore, $A \subseteq A^*$, $norm(A) \subseteq A^*$ and if A doesn't contain any equality atoms, then $A = A^* = norm(A)$. $norm(A)$ is a canonical form of A up to a variable renaming.

Comment The last point should be clarified: let A and B be two $\exists\wedge =$formula, if $norm(A) = norm(B)$ then $A \equiv B$ but the converse is not true!

Example Let $A = \{p(x), q(y), q(w), p(z), q(c_2), r(s_3, s_1), x = s_1, c_1 = x, y = z, s_2 = c_2\}$ be a fact (where w, x, y, z are variables, c_1, c_2 are constants and s_1, s_2, s_3 are surrogates), then $A^* = A \cup \{p(s_1), p(c_1), q(z), p(y), q(s_2), r(s_3, c_1), r(s_3, x), s_1 = s_1, c_1 = c_1, x = x, s_1 = x, x = c_1, c_1 = s_1, s_1 = c_1, y = y, z = z, z = y, s_2 = s_2, c_2 = c_2, c_2 = s_2, s_3 = s_3, w = w\}$ and $norm(A) = \{p(s_1), p(c_1), q(y), q(w), p(y), q(s_2), q(c_2), r(s_3, s_1), r(s_3, c_1), c_1 = s_1, s_1 = c_1, c_2 = s_2, s_2 = c_2\}$

2.2 Interpretation and Model of a KB

The interpretation domain is the set of surrogates $S = L + E$.

Definition 7 (Interpretation) *An interpretation of $\mathcal{L}$ is a mapping I from $S + C + P$ to S^* such that:*

- *for all surrogate s in S: $I(s) = s$,*
- *$I(=)$ is the equality on S,*
- *for all constant c in C: $I(c) \in E$,*
- *for all k-ary predicate p in P : $I(p) \subseteq S^k$.*

Two interpretations may only differ by the interpretation of non logical symbols, i.e., elements in the vocabulary. Thus, one can speak of the interpretation of a language or of a vocabulary.

Definition 8 *Let us consider a finite vocabulary $V = (C, P)$ (i.e., C and P are finite). An interpretation I is* finite *if for any p in P, $I(p)$ is finite.*

The semantics of the logical connectives are defined in the same way as in classical FOL. Thus, it is straightforward to define a *model* of a formula A as well as the logical consequence noted $\models$ or the logical equivalence $\equiv$.

Property 1 *Let A be an $\exists\wedge=$formula then $A^* \equiv A$ and $norm(A) \equiv A$.*

Definition 9 (Model of a knowledge base) *An interpretation I is a model of a knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{C})$ if*

- *I is a model of $(\mathcal{F}, \mathcal{R})$,*
- *I is not a model of any C in $\mathcal{C}$.*

2.3 Grounded and Well-grounded KBs

A (satisfiable) KB is grounded if the interpretation of a constant is the same individual in all models of the knowledge base and this means that (in a grounded knowledge base) the constants are aliases of the individual surrogates.

Definition 10 (Grounded knowledge base) *A satisfiable knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{C})$ is* grounded *if: for each constant c in $const(\mathcal{K})$, there is an individual surrogate e in $ind(\mathcal{K})$ such that for any model I of $\mathcal{K}$ $I(c) = e$.*

As said in section 2.1, the two types of surrogates (i.e., literals and individuals) are semantically different. A literal directly refers to an element in the domain, while an individual surrogate means nothing. An identification formula give meaning to an individual, such a formula is a definition of an individual.

Definition 11 (Well-grounded) *An individual surrogate e is* well-grounded *wrt a base $\mathcal{K}$ if there is an identification formula $Id_e = \forall X(NSC(x) \leftrightarrow (x = e))$ such that $\mathcal{K} \models Id_e$. A grounded knowledge base $\mathcal{K}$ is well-grounded if each individual surrogate e in $\mathcal{K}$ is well-grounded.*

Comment e is well-grounded means that any model of $\mathcal{K}$ is a model of $NSC(e)$, and that in any model of $NSC(x)$ the only possible value of x is e. Indeed, $\forall X(NSC_e(x) \leftrightarrow (x = e)) \equiv NSC_e(e) \wedge \forall X(NSC_e(x) \rightarrow (x = e))$.

In such a situation, $\mathcal{K}$ is said to be well-grounded with respect to the set $\mathcal{ID}_{\mathcal{K}} = \{Id_e |\ e \in ind(\mathcal{K})\}$ of identification formulas associated with the individual surrogates of $\mathcal{K}$.

2.4 Checking problems

In this framework, one can define several checking problems.

Problem 1 Let $\mathcal{K}$ be a knowledge base, check if $\mathcal{K}$ is satisfiable.

Problem 2 Let $\mathcal{K}$ be a satisfiable knowledge base, check if $\mathcal{K}$ is grounded.

Problem 3 Let $\mathcal{K}$ be a grounded knowledge base and a set of identification formulas Id, check if $\mathcal{K}$ is well-grounded with respect to Id.

Problem 4 Let $\mathcal{K}$ be a grounded knowledge base, check if $\mathcal{K}$ is well-grounded (and give an Id).

2.5 Reasoning with Homomorphism

In this section, the notions dealing with homomorphism introduced in [BLMS11] for classical FOL are adapted to the previous framework.

2.5.1 Homomorphism

Definition 12 (substitution) *Given a set of variables X and a set of terms T, a substitution σ of X by T is a mapping from X to T. Given a conjunct C, $\sigma(C)$ denotes the conjunct obtained from C by replacing each occurrence of $x \in X \cap var(C)$ by $\sigma(x)$. If a fact A is the existential closure of a conjunct C, then $\sigma(A)$ is the existential closure of $\sigma(C)$.*

Definition 13 (homomorphism) *A homomorphism from a fact A to a (possibly infinite) fact B is a substitution σ of $var(A)$ by $term(B)$ such that $\sigma(A) \subseteq B$.*

Note that there is a homomorphism from a fact A to its equivalent forms A^* and $norm(A)$.

Theorem 1 *Let A and B be two facts. $B \models A$ if and only if there is a homomorphism from $norm(A)$ to $norm(B)$.*

2.5.2 Rule application

Remember that a rule hypothesis H (the same remark holds for the constraints) doesn't contain any equality atoms. Thus, we have $H = H^* = norm(H)$.

Definition 14 (Application of a rr-rule) *Let F be a fact and $R = (H, A)$ be a rr-rule. R is said* applicable *to F if there is a homomorphism, say π, from H to $norm(F)$. In that case, the* application of R to F according to π *produces a fact $\alpha(F, R, \pi) = F \cup \pi(C)$. $\alpha(F, R, \pi)$ is said to be an* immediate derivation *from F. This rule application is said to be* redundant *if $\alpha(F, R, \pi) \equiv F$.*

To check whether $\alpha(F, R, \sigma) \equiv F$, one can check that $norm(\alpha(F, R, \sigma))$ maps to $norm(F)$.

Definition 15 (Derivation) *Let F be a fact and $\mathcal{R}$ be a set of rr-rules. A fact F' is called an* $\mathcal{R}$-derivation *of F if there is a finite sequence (called the* derivation sequence*) $F = F_0, F_1, \ldots, F_k = F'$ such that for all $1 \leq i \leq k$, there is a rule $R = (H, C) \in \mathcal{R}$ and a homomorphism π from H to $norm(F_{i-1})$ with $F_i = \alpha(F_{i-1}, R, \pi)$, i.e., F_i is an immediate derivation from F_{i-1}.*

Definition 16 (Closure wrt $\mathcal{R}$) *Let $\mathcal{R}$ be a set of rr-rules, a fact $\mathcal{F}$ is said* closed *wrt $\mathcal{R}$ iff any application of a rule in $\mathcal{R}$ to $\mathcal{F}$ leads to a fact $\mathcal{F}'$ equivalent to $\mathcal{F}$. Given a fact $\mathcal{F}$ and a set of rules $\mathcal{R}$, a* closure *of $\mathcal{F}$ wrt $\mathcal{R}$ is a closed fact (wrt to $\mathcal{R}$) that is a $\mathcal{R} - derivation$ of F .*

Property 2 *Let $\mathcal{F}$ be a fact and $\mathcal{R}$ be a set of rr-rules: (i) there always exists a closure of $\mathcal{F}$ wrt $\mathcal{R}$, (ii) all closures of $\mathcal{F}$ wrt $\mathcal{R}$ are logically equivalent, and (iii) the normal form of all closures of $\mathcal{F}$ wrt $\mathcal{R}$ are identical.*

Definition 17 ($\mathcal{R}(\mathcal{F})$) *Let $\mathcal{F}$ be a fact and $\mathcal{R}$ be a set of rules, $\mathcal{R}(\mathcal{F})$ denotes the normal form of any closure of F wrt to $\mathcal{R}$.*

Theorem 2 (Saturation) *Let F and F' be two facts and $\mathcal{R}$ be a set of rr-rules. Then $F, \mathcal{R} \models F'$ if and only if there is a homomorphism from $norm(F')$ to $\mathcal{R}(F)$*

2.5.3 Satisfiability, Grounded and well-grounded properties

Property 3 *A knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{C})$ is satisfiable iff:*

(i) there is at most one surrogate in any equivalence class of $\mathcal{R}(\mathcal{F})$ and

(ii) for any C in $\mathcal{C}$ there is no homomorphism from C to $\mathcal{R}(\mathcal{F})$.

Definition 18 (Identity property) *A knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{C})$ has the* identity property *if each equivalence classe of terms in $\mathcal{R}(\mathcal{F})$ that contains a constant also contains an individual surrogate.*

Property 4 *A satisfiable knowledge base is grounded if and only if it fulfills the identity property.*

Proof: Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{C})$ be a satisfiable knowledge base.
($\Leftarrow$) Let I be any model of $\mathcal{K}$. For any constant c in $const(\mathcal{K})$, we note e^c the single individual surrogate belonging to the equivalence class of c. Thus, $I(c) = I(e^c) = e^c$, the interpretation of c is the same for any I.
($\Rightarrow$) Proof by contradiction. Let $\mathcal{K}$ be a satisfiable and grounded knowledge base. Let us assume that a constant c belongs to an equivalence class which does not contain an individual surrogate. Since $\mathcal{K}$ is grounded there is an individual surrogate e which is the interpretation of c in any model I of $\mathcal{K}$. Let I be a model of $\mathcal{K}$, a model I' with $I'(c) \neq I(c)$ can be built as follows:

- choose e' in $E - ind(\mathcal{K})$ different from e;
- for each constant c' equivalent to c, $I'(c') = e'$, and for all other constant c", $I'(c") = I(c")$;
- for all p in P, $I'(p)$ is the union of $I(p)$ and the set of tuples obtained by substituting e' to e in $I(p)$.

I' is a model of $\mathcal{K}$ and since $I'(c) \neq I(c)$, $\mathcal{K}$ is not grounded. □

Property 5 *An individual surrogate e is well-grounded wrt a grounded knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{C})$ iff there is an $\exists\wedge$formula F' having a variable x such that:*

- *there is at least one homomorphism from F' to $\mathcal{R}(\mathcal{F})$ and*
- *every homomorphism from F' to $\mathcal{R}(\mathcal{F})$ maps x to e.*

3 Graphical Framework

In order to solve these problems we use graph representation of formulas and graph homomorphisms.

3.1 Graph notions

Notions defined in this section are adapted from notions defined in [CM09].

Definition 19 (G-vocabulary) *A G-vocabulary is a pair (T, P) where T and P are two pairwise disjoint sets. T is a set of terms and P is a set of predicates of arity $1, ...k$.*

Definition 20 (Graph) *A graph defined over a vocabulary (T, P) is a 4-tuple $G = (C, R, E, l)$ satisfying the following conditions:*

- *(C,R,E) is a bipartite multigraph. C is the concept node set, R is the relation node set and E is the family of edges.*

- *l is a labeling function of nodes and edges that satisfies:*
  1. *A concept node is labeled by a set (possibly empty) of terms in T.*
  2. *A relation node is labeled by a predicate in P.*
  3. *The degree of a relation node is equal to the arity of its label.*
  4. *Edges incident to a relation node labeled r are totally ordered and they are labeled from 1 to k (the arity of r).*

Such a graph looks like a BG-graph on a flat vocabulary where the markers are sets of terms. A concept node labeled by the empty set is called a *generic* concept node

Definition 21 *A* normal graph *is a graph such all labels of concept nodes are pairwise disjoint and without twin relations. The* normal form $norm(G)$ *of a graph G is the normal graph obtained by merging the concept nodes having a non empty intersection of their label (the label of a node resulting from a merging is the union of the labels of the merged nodes) and by deleting twin relations.*

Other notions defined in [CM09] can be easily extended to the graphs defined in this section. The central notion of graph homomorphism can be defined as follows.

Definition 22 (Graph Homomorphism) *Let $G = (C_G, R_G, E_G, l_G)$ and $H = (C_H, R_H, E_H, l_H)$ be two graphs defined over the same vocabulary. A* homomorphism *π from G to H is a mapping from C_G to C_H and from R_G to R_H such that:*

- $\forall (r, i, c) \in G, (\pi(r), i, \pi(c)) \in H$,
- $\forall r \in R_G$, $l_H(\pi(r)) = l_G(r)$,
- $\forall c \in C_G$, $l_H(\pi(c)) \supseteq l_G(c)$

If there is a homomorphism (say π) from G to H, we say that G maps *or* projects *to H (by π).*

Let c and d be two concept nodes of a graph G. The *merging* operation of c and d consists of identifying the two nodes c and d into a new node having the label $l(c) \cup l(d)$. One now defines rules and rule applications and w.l.g. one only considers rules with at most one relation node in the conclusion.

Definition 23 (G-rule) *The two following kinds of graph rules are considered. An* equality rule *is an ordered pair $R = ((c, c'), H)$ where c and c' are distinct (generic) concept nodes in the graph H.*

An atomic rule *is an ordered pair $R = ((c_1, ...c_n)H, p)$ where the c_i are concept nodes (not necessarily distinct) in the graph H and p is a predicate of arity n.*

Application of a rule graph can now be defined.

Definition 24 (Application of an equality rule) *Let G be a graph and $R = ((c, c'), H)$ be an equality rule. R is applicable to G if there is a homomorphism π from H to G. In this case, the result of the* application of R to G according to π *is the graph $G' = (R, \pi)G$ obtained from G by merging $\pi(c)$ and $\pi(c')$.*

Definition 25 (Application of an atomic rule) *Let G be a graph and $R = ((c_1, ...c_n)H, p)$ be an atomic rule. R is applicable to G if there is a homomorphism π from H to G. In this case, the result of the* application of R to G according to π *is the graph $G' = (R, \pi)G$ obtained from G by adding a new relation node labeled p whose i-th neighbor is $\pi(c_i)$ for $i = 1, \ldots, n$.*

3.2 Relationships between formulas and graphs

Simple relationships between logical notions presented in Sect. 2 and graphical notions presented in Sect. 3 allow to solve problems concerning formulas by graph algorithms based on graph homomorphism.

Definition 26 (Graph of a Fact) *The graph of a fact A is the normal graph $graph(A)$ obtained as follows.*

1. *Create a concept node for each equivalence class of terms. Each concept node is labeled by the set of surrogates and constants belonging to the equivalence class and by the empty set if the equivalence class only contains variables.*
2. *From each atom $p(t_1, ...t_k)$, adds a relation node r labeled p with k edges labelled from 1 to k that link r to the k concept nodes resp. associated with the equivalence class of $t_1, ...t_k$.*
3. *Delete twin relation nodes.*

Note that $graph(A) = graph(norm(A))$ and, more generally, for any fact B, such as $norm(B) = norm(A)$, one has $graph(B) = graph(A)$.

Theorem 3 *There is a bijection between the set of normal facts (up to a renaming of the variables) on the vocabulary (C, P) and the normal graphs on the vocabulary $(C \cup S, P)$. The same result holds for rr-rules and G-rules. There is a homomorphism from a normal fact A to a normal fact B iff there is a homomorphism from $graph(A)$ to $graph(B)$. A rr-rule is applicable to a fact A iff the associated G-rule is applicable to $graph(A)$. etc.*

Comment To finish...

References

[BLMS11] J.-F. Baget, M. Leclère, M.-L. Mugnier, and E. Salvat. On rules with existential variables: Walking the decidability line. *Artificial Intelligence Journal*, to appear in 2011.

[CM09] M. Chein and M.-L. Mugnier. *Graph-based Knowledge Representation*. Springer, London, GB, 2009.

[LL00] H. J. Levesque and G. Lakemeyer. *The Logic of Knowledege Bases*. MIT Press, Cambridge, Massachusetts, 2000.

[SPR07] F. Saïs, N. Pernelle, and M.-C. Rousset. L2r: A logical method for reference reconciliation. In *AAAI*, pages 329–334, 2007.

[SPR09] F. Saïs, N. Pernelle, and M.-C. Rousset. Combining a logical and a numerical method for data reconciliation. *J. Data Semantics*, 12:66–94, 2009.