Future Generation Computer Systems 26 (2010) 408–420

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

The schema theory for semantic link network

Hai Zhuge [a,*], Yunchuan Sun [a,b,c]

[a] China Knowledge Grid Research Group, Key Lab of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, 100190, Beijing, China

[b] Graduate School of Chinese Academy of Sciences, Beijing, China

[c] School of Economics and Business Administration, Beijing Normal University, 100875, Beijing, China

ARTICLE INFO

Article history:
Received 30 January 2009
Received in revised form
18 May 2009
Accepted 29 August 2009
Available online 6 September 2009

Keywords:
Data models
Normal form
Schema
Semantic link network
Semantic web

ABSTRACT

The Semantic Link Network (SLN) is a loosely coupled semantic data model for managing Web resources. Its nodes can be any type of resource. Its edges can be any semantic relation. Potential semantic links can be derived out according to reasoning rules on semantic relations. This paper proposes the schema theory for the SLN, including the concepts, rule-constraint normal forms, and relevant algorithms. The theory provides the basis for normalized management of semantic link network. A case study demonstrates the proposed theory.

1. Introduction

The schema of a relational database defines the structure of the database. It defines a set of relations with attributes and the dependencies among attributes. The normalized theory of the relational schema is to ensure high consistency, low redundancy and better efficiency [1,2]. A relational data model is limited in representing rich semantic relationships between various resources and supporting reasoning on semantic relations.

The Semantic Web aims at making Web resources machine-understandable by enriching semantics in resources [3]. XML (eXtensible Markup Language) is to describe the structure in Web resources for cross-platform information sharing (www.w3.org/XML). The XML schema defines a set of syntaxes and rules to express the shared vocabularies (www.w3.org/XML/Schema). It provides a means for defining the structure, content and semantics of XML documents. Based on XML, many markup languages have been proposed. RDF (Resource Description Framework, www.w3.org/TR/2004/REC-rdf-mt-20040210) focuses on describing the universal resources on the Web by an object-attribute-value triple. RDF Schema (RDFS, www.w3.org/TR/2004/REC-rdf-schema-20040210) defines a set of syntaxes to store the metadata of resources with XML syntax and provides basic RDF vocabularies for structuring RDF resources. RDFS is still weak in expressing rich semantic relationships and supporting relational reasoning. OWL (Web Ontology Language) is designed to describe the semantics of the resources themselves with ontologies and semantic relationships between resources with roles (www.w3.org/2004/OWL). It can represent the meaning of terms in vocabularies explicitly and the relationships between those terms. Its logical foundation is description logics which has the decidability of ontology consistency. The Rule Markup Language (RuleML) is to express rules in XML for deduction, rewriting, and further inferential-transformational tasks (www.ruleml.org). The Semantic Web Rule Language (SWRL) is based on the combination of OWL with RuleML [4].

The Semantic Link Network (SLN) is a loosely coupled semantic data model for managing Web resources with the following main features of the Web:

(1) Easy to build and easy to use; and,
(2) Any semantic node can semantically link to any other semantic node.

A semantic link network instance is a directed graph, denoted as $S(ResourceSet, LinkSet)$, where S is the name of the semantic link network, $ResourceSet$ is a set of resources, and $LinkSet$ is a set of semantic links in the form of $R \xrightarrow{\alpha} R'$, where $R, R' \in ResourceSet$, and α is a semantic factor representing a semantic relation between R and R'. A set of reasoning rules on semantic links enables a semantic link network to derive out potential semantic links. The

* Corresponding author. Fax: +86 1062562703.
E-mail addresses: zhuge@ict.ac.cn (H. Zhuge), yunch@bnu.edu.cn (Y. Sun).
URL: http://www.knowledgegrid.net/~h.zhuge (H. Zhuge).

0167-739X/$ – see front matter © 2009 Published by Elsevier B.V.
doi:10.1016/j.future.2009.08.012

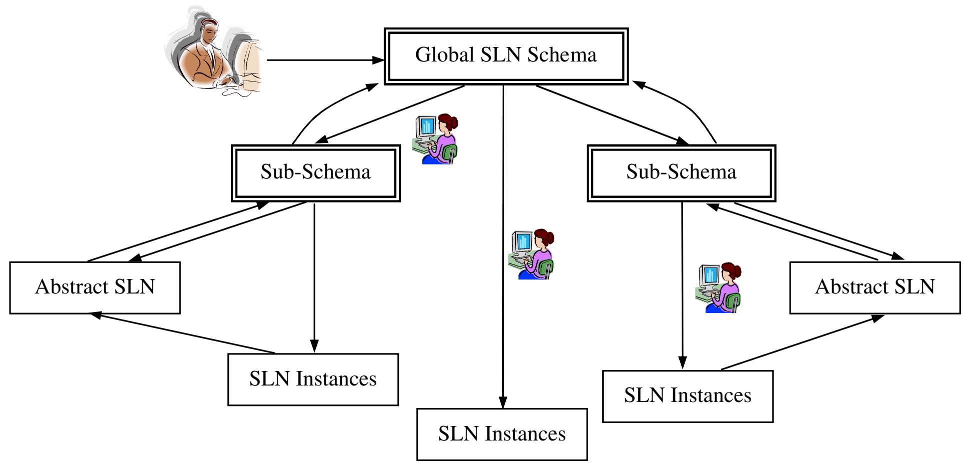

Fig. 1. The role of SLN schema.

basic concept and model of the SLN have been introduced in [5–11]. More references are available at www.knowledgegrid.net/~h.zhuge/SLN.htm.

The motivation of this paper is to construct a schema theory and a rule-constraint normalized theory for SLN construction and resource management.

An SLN schema specifies resource types, semantic link types, and reasoning rules. The resources and semantic links instance are regulated by the resource types and semantic link types. The reasoning on instances is based on the reasoning rules defined by the schema. The SLN schema provides a blueprint to build SLN instances and provides a way to normalize the SLN instances. The global SLN schema reflects a consensus on the basic semantics of the domain. Users can define SLN instances by instantiating the global schema, or define a sub-schema according to the global schema first and then instantiating the sub-schemas.

Fig. 1 shows the role of schema in developing semantic link networks. There are two ways to form the SLN schema:

(1) defined by domain experts; and,
(2) induced from existing abstract semantic link networks or from SLN instances [8].

2. Schema for the semantic link network

The schema of the SLN is a triple S(*ResourceTypes, LinkTypes, Rules*). *ResourceTypes* is a set of resource types denoted as $\{rt_1, rt_2, \ldots, rt_k\}$, where rt_i is defined by its field. *LinkTypes* is a set of semantic link types, where each takes the form of $rt_i \xrightarrow{\alpha} rt_j$, where $rt_i, rt_j \in$ *ResourceTypes*, and α is a semantic relation defined by its field. The *Rules* is a set of reasoning rules on link types. A formal definition of semantic link network is given in [5].

The possible semantic relationship types between two resources are determined by the types of the start resource and the end resource. For example, the possible types of a semantic link between a researcher and a paper are *authorOf*, *editorOf*, and *readerOf*, but not *fatherOf*. So a semantic link instance with semantic factor α from a resource R of type rt_i to another resource R' of type rt_j can be described as $R \xrightarrow{\alpha} R'$. For two resource types rt_i and rt_j, we use $[rt_i, rt_j]$ to denote the set of all semantic link types with the start resource type rt_i and the end resource type rt_j.

For a pair of resource types, relationships between semantic link types can be classified into the following three categories.

(1) *Implication*. A semantic link type α implies semantic link type β, denoted as $\alpha \Rightarrow \beta$, between the same pair of resources as shown in Fig. 2(d).
(2) *Compatible*. Two semantic link types α and β do not affect each other.
(3) *Incompatible*. Two semantic link types α and β cannot co-occur between the same pair of resources.

For a semantic link $R \xrightarrow{\omega} R'$ between two resources R and R', the reversion is a semantic link $R' \xrightarrow{\omega^{-1}} R$, which means that if there is a semantic relationship ω from R to R', then there is a semantic relationship ω^{-1} from R' to R [6].

A reasoning rule takes the following form as shown in Fig. 2(a): $R \xrightarrow{\alpha} R', R' \xrightarrow{\beta} R'' \Rightarrow R \xrightarrow{\gamma} R''$ denoted as $\alpha \cdot \beta \Rightarrow \gamma$ in abbreviation. Fig. 2(b) and (c) show the following two forms of reasoning rule: $R \xrightarrow{\alpha} R', R \xrightarrow{\beta} R'' \Rightarrow R' \xrightarrow{\gamma} R''$ is equivalent to $R' \xrightarrow{\alpha^{-1}} R, R \xrightarrow{\beta} R'' \Rightarrow R' \xrightarrow{\gamma} R''$, i.e., $\alpha^{-1} \cdot \beta \Rightarrow \gamma$. $R' \xrightarrow{\alpha} R, R'' \xrightarrow{\beta} R \Rightarrow R' \xrightarrow{\gamma} R''$ is equivalent to $R' \xrightarrow{\alpha} R, R \xrightarrow{\beta^{-1}} R'' \Rightarrow R' \xrightarrow{\gamma} R''$, i.e., $\alpha \cdot \beta^{-1} \Rightarrow \gamma$.

Lemma 1. *The following two kinds of rules hold:*

(1) $\alpha \cdot \beta \Rightarrow \gamma$ *is equivalent to* $\beta^{-1} \cdot \alpha^{-1} \Rightarrow \gamma^{-1}$, *and,*
(2) $\alpha^{-1} \cdot \beta \Rightarrow \gamma$ *is equivalent to* $\beta^{-1} \cdot \alpha \Rightarrow \gamma^{-1}$.

Fig. 2(d) shows the following reasoning rule: $R \xrightarrow{\alpha} R' \Rightarrow R \xrightarrow{\beta} R'$, in simple $\alpha \Rightarrow \beta$, which means that the semantic relationship α is stronger than the semantic relationship β between two resources.

Proposition 1. *For a rule* $\alpha \Rightarrow \beta$, *if* $\alpha \in [rt_i, rt_j]$, *then* $\beta \in [rt_i, rt_j]$.

Proposition 2. *For a rule* $\alpha \cdot \beta \Rightarrow \gamma$, *if* $\alpha \in [rt_i, rt_j]$, $\beta \in [rt_j, rt_k]$, *then* $\gamma \in [rt_i, rt_k]$.

3. SLN operations

We first define the notion of sub-schema and reference relation between schemas. For the SLN schemas $\hat{S}(R, L, rs)$ and $\hat{S}'(R', L', rs')$, if $R' \subseteq R$ and $L' \subseteq L$, $\hat{S}'$ is called a sub-schema of $\hat{S}$, denoted as $\hat{S}' \subseteq \hat{S}$. If two schemas $\hat{S}(R, L, rs)$ and $\hat{S}'(R', L', rs')$ share a semantic link type, for instance, $rt_i \xrightarrow{\alpha} rt_j$ in L is identical to $rt'_i \xrightarrow{\alpha'} rt'_j$ in L', a reference relation $Ref(\alpha, \alpha')$ can be built from α to α', which means that for two instances S and S' under $\hat{S}$ and $\hat{S}'$ respectively, all semantic links with type of α' in S' can be used in S as semantic links with type of α.

The following are operations on semantic link networks.

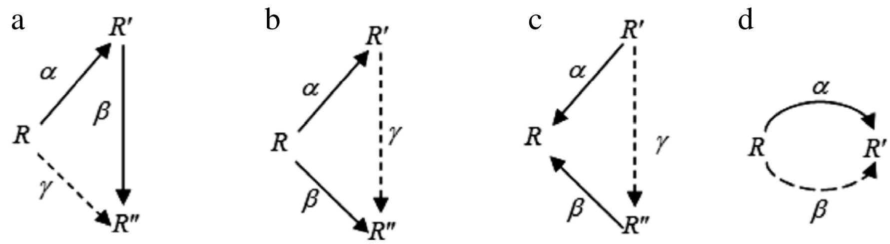

Fig. 2. Basic forms of reasoning rule.

(1) **Union**. The union of two semantic link networks $S(RS, LS)$ and $S'(RS', LS')$ under the same schema $\hat{S}$ is a new semantic link network $S(RS \cup RS', LS \cup LS')$. The union operation does not guarantee to generate a connective semantic link network. The union operation enables user or application to operate on different semantic link networks as a whole.

(2) **Intersection**. The intersection of two semantic link networks $S(RS, LS)$ and $S'(RS', LS')$ under the same schema $\hat{S}$ is a new semantic link network $S(RS \cap RS', LS \cap LS')$. The intersection operation does not guarantee to generate a connective semantic link network. The intersection operation enables user or application to operate on the common part of different semantic link networks.

(3) **Selection**. A selection of a semantic link network S, $\sigma_P(S)$, is a new semantic link network satisfying condition P, a logical expression for selecting a set of resources or a set of semantic links from S. The selection operation enables a user or application to operate on the interested part of a semantic link network.

(4) **Projection**. The projection of a semantic link network S under schema $\hat{S}(R, L, rs)$ on another schema $\hat{S}'(R', L', rs')$, denoted as $S' = \Pi_{\hat{S}'}(S)$, is a new semantic link network derived from S by removing all resources whose types are not in R' and removing all semantic links whose types are not in L'. Reasoning on S' is executed according to the rule set rs'. The projection operation enables a user or application to operate a semantic link network from different schemas, enables a semantic link network to suit different schemas for different applications, and enables reasoning on a semantic link network to be localized on a relevant schema. It is especially useful for large semantic link network.

(5) **Reasoning**. The SLN reasoning is to derive some new semantic links from the semantic link network. An atomic reasoning in a semantic link network is an execution of a rule in rs to get a new semantic link. Reasoning is a series of atomic reasoning connected one another over the semantic link network.

(6) **Join**. The join of two semantic link networks $S(RS, LS)$ and $S'(RS', LS')$ under $\hat{S}(R, L, rs)$ and $\hat{S}'(R', L', rs')$ respectively is a new semantic link network $S \times_P S' = S_J(RS_J, LS_J)$ under a new schema $\hat{S}_J(R_J, L_J, rs_J)$, where P is a set of references $\{Ref(\alpha_1, \alpha_1'), \ldots, Ref(\alpha_m, \alpha_m'), Ref(\beta_1', \beta_1), \ldots, Ref(\beta_n', \beta_n)\}$, $L_J = (L \cup L') - \{\alpha_1', \ldots, \alpha_m', \beta_1', \ldots, \beta_n'\}$, R_J is the set of resources involved in L_J, rs_J is the set $rs \cup rs'$ by replacing α_i' (or β_j') with α_i (or β_j) $(1 \le i \le m, 1 \le j \le n)$, LS_J is the set $LS \cup LS'$ by replacing α_i' (or β_j') with α_i (or β_j), and $RS_J = RS \cup RS'$. It is easy to verify that $S = \Pi_{\hat{S}}(S_J)$ and $S' = \Pi_{\hat{S}'}(S_J)$. The union operation is a special case of the join operation. The join operation enables a user or application to operate on relevant semantic link networks as a whole.

(7) **Decomposition**. A decomposition of $\hat{S}(R, L, rs)$ is a set of sub-schemas $\hat{S}_1(R_1, L_1, rs_1)$, …, and $\hat{S}_m(R_m, L_m, rs_m)$, if there does not exist any pair of $(\hat{S}_i, \hat{S}_j)$ such that S_i is a sub-schema of $\hat{S}_j$; and $R = R_1 \cup R_2 \cup \cdots \cup R_m$, and $L = L_1 \cup L_2 \cup \cdots \cup L_m$. At the instance level, a semantic link network S under $\hat{S}$ has the corresponding decomposition $\{S_1, S_2, \ldots, S_m\}$, where each S_i is under the schema $\hat{S}_i$ respectively. And for all i, we have $S_i = \prod_{\hat{S}_i(S)}$. The decomposition operation enables a user or application to operate on a small schema to raise the efficiency when facing a large schema.

(8) **Query**. A query on semantic link networks is a new semantic link network from a combination of some operations of union, intersection, reasoning, selection, projection, or join.

Two semantic link networks S and S' are called equivalent if and only if the results from S and S' are identical for any query. Clearly, the equivalence among semantic link networks is symmetric, reflexive, and transitive [6].

For a given schema, there are many kinds of decomposition, but some decompositions may not be good. For example, for a schema $\hat{S}(R, L, rs)$, where $R = \{rt_1, rt_2, rt_3\}$, L includes $[rt_1, rt_2] = \{\alpha_1, \alpha_2\}$, $[rt_2, rt_3] = \{\beta_1, \beta_2\}$ and $[rt_1, rt_3] = \{\gamma\}$, and $rs = \{\alpha_1 \cdot \beta_1 \Rightarrow \gamma\}$, the two sub-schemas $S_1(R, \{\alpha_1, \beta_2, \gamma\}, \phi)$ and $S_2(\{rt_1, rt_2\}, \{\alpha_2, \beta_1\}, \phi)$ construct a decomposition. The decomposition is not good since it loses the reasoning rule. A decomposition of semantic link network S is called loss-less if the join of the decomposition is equivalent to S. A decomposition of a schema is called loss-less if, for any instance under the schema, the corresponding instance decomposition is loss-less.

Theorem 1. *A decomposition $\{\hat{S}_1, \hat{S}_2, \ldots, \hat{S}_m\}$ of the schema S is loss-less if*

(1) *$rs = rs_1 \cup rs_2 \cup \ldots \cup rs_m$; and,*
(2) *for any rule $r: \alpha \cdot \beta \Rightarrow \gamma$ in rs_i, for all possible j, if $\gamma \in L_j$ and $r \notin rs_j$, there is a reference relation $Ref(\gamma \in \hat{S}_j, \gamma \in \hat{S}_i)$.*

Proof. We need to verify that for a semantic link network $S(RS, LS)$ under schema $\hat{S}$ is equivalent to the join of the corresponding decomposition $S_1(RS_1, LS_1), S_2(RS_2, LS_2), \ldots, S_m(RS_m, LS_m)$ under schemas $\hat{S}_1, \hat{S}_2, \ldots, \hat{S}_m$ respectively.

Precondition 1 means that for each rule $r : \alpha \cdot \beta \Rightarrow \gamma$ (or $r : \alpha \Rightarrow \beta$) in rs, there is at least one i such that r is in rs_i. According to the instance decomposing algorithm $S_i = \prod_{\hat{S}_i(S)}$, all semantic links with type of α, β and γ in S are included in S_i. For reasoning in S, assume that rules in the sequence: (1) $r_1 : \alpha_1 \cdot \beta_1 \Rightarrow \gamma_1$ (or $\alpha_1 \Rightarrow \gamma_1$), (2) $r_2 : \alpha_2 \cdot \beta_2 \Rightarrow \gamma_2$ (or $\alpha_2 \Rightarrow \gamma_2$), …, (k) $r_k : \alpha_k \cdot \beta_k \Rightarrow \gamma_k$ (or $\alpha_k \Rightarrow \gamma_k$), …, (n) $r_n : \alpha_n \cdot \beta_n \Rightarrow \gamma_n$ (or $\alpha_n \Rightarrow \gamma_n$) are fired one after another to find the γ_n-type semantic link between two resources. Conveniently, we denote semantic links involved in the above reasoning process as l_{α_i}, l_{β_i}, and l_{γ_i} respectively for all $1 \le i \le n$. Then, the reasoning process is executed as a sequence of atomic reasoning: (1) $l_{\alpha_1} \cdot l_{\beta_1} \Rightarrow l_{\gamma_1}$ (or $l_{\alpha_1} \Rightarrow l_{\gamma_1}$), (2) $l_{\alpha_2} \cdot l_{\beta_2} \Rightarrow l_{\gamma_2}$ (or $l_{\alpha_2} \Rightarrow l_{\gamma_2}$), …, (k) $l_{\alpha_k} \cdot l_{\beta_k} \Rightarrow l_{\gamma_k}$ (or $l_{\alpha_k} \Rightarrow l_{\gamma_k}$), …, (n) $l_{\alpha_n} \cdot l_{\beta_n} \Rightarrow l_{\gamma_n}$ (or $l_{\alpha_n} \Rightarrow l_{\gamma_n}$) in the semantic link network S. For each semantic link l_{α_k} involved in the reasoning process as a prerequisite, $l_{\alpha_k} \in LS$ or else the type of l_{α_k} is some l_{γ_t} derived from some previous rule $l_{\alpha_t} \cdot l_{\beta_t} \Rightarrow l_{\gamma_t}$ (or $l_{\alpha_t} \Rightarrow l_{\gamma_t}$) $(t < k)$ in the above sequence.

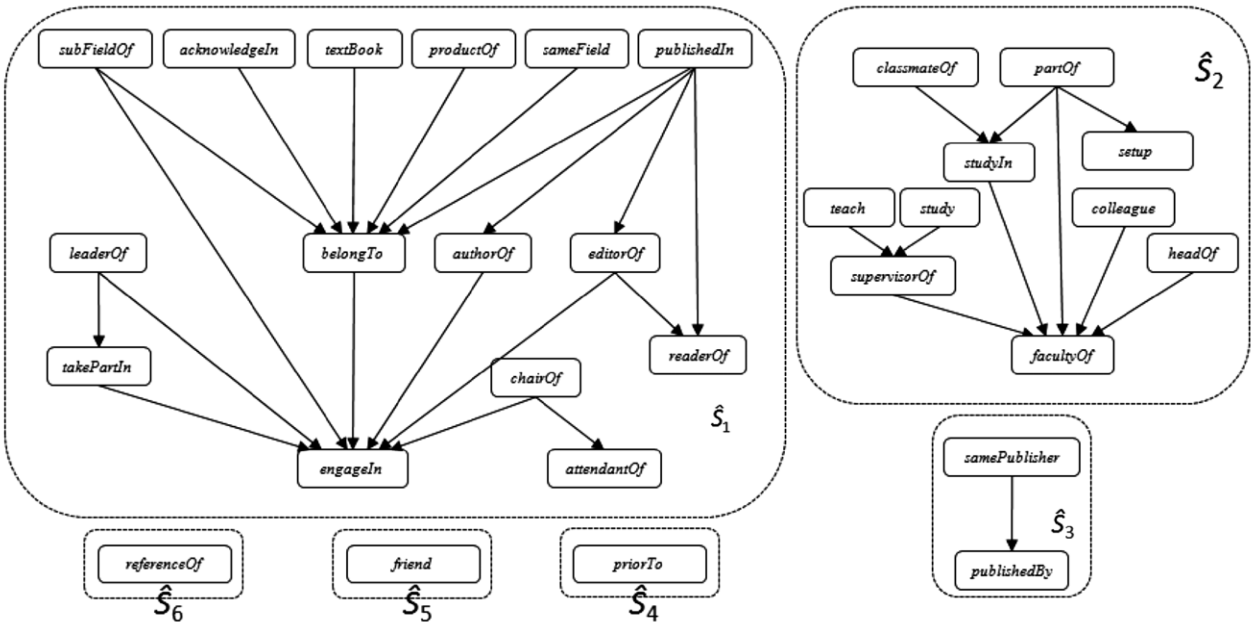

Fig. 3. Semantic link types relative net for RC-NF1 schemas of a research semantic link network.

For the same reasoning, we will verify that the same process can be executed on the join of the corresponding decomposition $S_1, S_2, \ldots$, and S_m under schemas $\hat{S}_1, \hat{S}_2, \ldots$, and $\hat{S}_m$ respectively. Actually, there exists at least one i such that $r_k \in rs_i$ for each rule r_k in the above rule sequence, $\alpha_k, \beta_k, \gamma_k \in L_i$. Thus, S_i under $\hat{S}_i$ consists of all semantic links with types α_k, β_k, and γ_k in S.

(1) If $l_{\alpha_1}, l_{\beta_1} \in LS_i$, then the first atomic reasoning in the above sequence can be executed and the semantic link l_{γ_1} with γ_1 type can be deduced in S_i.

(2) Assume that the first $k-1$ atomic reasoning can be executed and the semantic links $l_{\gamma_1}, l_{\gamma_2}, \ldots, l_{\gamma_{k-1}}$ can be derived in some S_i. If $l_{\alpha_k}, l_{\beta_k} \in LS_j$ for some j, then the kth atomic reasoning can be fired and the semantic link l_{γ_k} with type γ_k can be deduced in S_j. Else, if $l_{\alpha_k} \notin LS_i$, then l_{α_k} is someone semantic link $l_{\gamma_{k'}}$ from the rule $l_{\alpha_{k'}} \cdot l_{\beta_{k'}} \Rightarrow l_{\gamma_{k'}}$ (or $l_{\alpha_{k'}} \Rightarrow l_{\gamma_{k'}}$) according to $r_{k'}$, where $1 < k' < k$.
   (1) If $r_{k'} \in rs_i$, l_{α_k} (i.e., $l_{\gamma_{k'}}$) can be deduced in S_i, then the semantic link l_{γ_k} with type γ_k can be derived.
   (2) If $r_{k'} \notin rs_i$, there exists at least one j s.t. $r_{k'} \in rs_j$, then we can get $l_{\gamma_{k'}}$ in S_j according to the assumption. From the precondition 2, there is a reference relation *Ref* $(\gamma_{k'} \in \hat{S}_i, \gamma_{k'} \in \hat{S}_j)$. Then $l_{\gamma_{k'}}$ can be duplicated into S_i. The rule r_k can be fired and the semantic link l_{α_k} can be deducted in S_i.

According to the law of mathematic induction, the above show that the semantic link l_{α_n} can be derived from the reasoning on the join of the decomposition $\{S_1, S_2, \ldots, S_m\}$. □

The following deduction can be derived from the theorem immediately.

4. Rule-constraint normal forms for the SLN schema

A reasoning rule represents the semantic relevance among semantic link types. For example, the rule *editorOf* $\Rightarrow$ *readerOf* shows that an editor of a paper is certainly a reader of the paper. Similarly, $\alpha \cdot \beta \Rightarrow \gamma$ shows that γ is semantically relevant to α and β.

Definition 1. Let $\hat{S}(R, L, rs)$ be an SLN schema, α_i, $\alpha_j, \alpha_k \in L$, α_i is called direct semantic relative to α_j (denoted as $\alpha_i \trianglelefteq \alpha_j$) if there is a rule in rs with one of the following forms $\alpha_j \Rightarrow \alpha_i$, $\alpha_j \cdot \alpha_k \Rightarrow \alpha_i$, or $\alpha_k \cdot \alpha_j \Rightarrow \alpha_i$.

If there is a sequence of semantic relative $\alpha_1 \trianglelefteq \alpha_2 \trianglelefteq \alpha_3 \trianglelefteq \cdots \trianglelefteq \alpha_m$ (m is an integer), then α_1 is called semantic relative to α_m, denoted as $\alpha_1 \triangleleft \alpha_m$. We can construct a semantic relative net, as shown in Fig. 3, for an SLN schema by drawing an arrow from α_i to α_j for each semantic relative $\alpha_j \trianglelefteq \alpha_i$ retrieved from the rule set.

Definition 2. Let $\hat{S}(R, L, rs)$ be an SLN schema, and $M \subseteq L$ a set of semantic link types. The closure of M, denoted as $C(M)$, under semantic relative is a set of semantic link types derived from the following steps.

(1) Let $C(M) = M$.
(2) Check for each semantic dependence $\alpha_i \triangleleft \alpha_j$, if $\alpha_i \in C(M)$, let $C(M) = C(M) \cup \{\alpha_j\}$; and, if $\alpha_j \in C(M)$, let $C(M) = C(M) \cup \{\alpha_i\}$.
(3) Repeat from step 2 until $C(M)$ remains unchanging.

It is easy to verify that all closures of the single semantic link types construct a classification for the semantic link types of the SLN schema, and we can decompose an SLN schema based on such a classification. Actually, each disjunction part in the semantic link relative net forms a closure.

For an isolated semantic link type α, the closure includes only itself, i.e., $C(\alpha) = \{\alpha\}$. For a schema of semantic link network $\hat{S}(R, L, rs)$, there may be a cycle of semantic relatives $\alpha_1 \trianglelefteq \alpha_2 \trianglelefteq \alpha_3 \trianglelefteq \cdots \trianglelefteq \alpha_m \trianglelefteq \alpha_1, \alpha_1, \alpha_2, \ldots, \alpha_m \in L$, called a *semantic relative cycle*. It is easily to verify the following lemma.

Lemma 2. *Let $\alpha_1 \trianglelefteq \alpha_2 \trianglelefteq \alpha_3 \trianglelefteq \cdots \trianglelefteq \alpha_m \trianglelefteq \alpha_1$ be a semantic relative cycle, where $\alpha_i \in L$ $(1 \le i \le m)$. We have:*

1. *$C(\alpha_i)$ is identical to $C(\alpha_j)$, for all $1 \le i, j \le m$.*
2. *$\alpha_i \in C(\alpha_j)$, for any i, j, $1 \le i, j \le m$.*
3. *For any $\alpha \in L$, if $\alpha \trianglelefteq \alpha_i$ then $\alpha \trianglelefteq \alpha_j$, $1 \le i, j \le m$.*

All semantic link types at one semantic relative cycle construct an equivalent class according to Lemma 2. A cycle $\alpha_1 \trianglelefteq \alpha_2 \trianglelefteq \alpha_3 \trianglelefteq \cdots \trianglelefteq \alpha_m \trianglelefteq \alpha_1$ in the semantic link type relative net can be regarded as a unit, denoted as $U(\alpha_i)$, where α_i is any semantic link type in the cycle. Therefore, a semantic relative cycle is shrunk into a node (all out and in arrows in the cycle will be focused on the node). We can find all cycles in the net by using the classic algorithms which find the cycles in a directed map. In the following discussion, we do not mention the cycles for they are regarded as single semantic link types.

The following definition normalizes the SLN schema.

Definition 3. An SLN schema $\hat{S}(R, L, rs)$ is in Rule-Constraint Normal Form 1 (RC-NF 1), if for any semantic link type $\alpha \in L$, $C(\alpha) = L$ holds.

Lemma 3. *For SLN schema $\hat{S}(R, L, rs)$ with RC-NF1 and $\alpha_1, \alpha_2 \in L$, then $C(\alpha_1) = C(\alpha_2)$.*

Any SLN schema $\hat{S}(R, L, rs)$ can be decomposed into several sub-schemas satisfying RC-NF1 by the following algorithm.

Algorithm 1. Let $\hat{S}(R, L, rs)$ be an SLN schema.

(1) Compute the classification of the semantic link types in S according to the definition of the closure of semantic relative and the rule set rs denoted as $\{L_1, L_2, \ldots, L_k\}$.
(2) For each L_i, $1 \le i \le k$, we can get a resource type set R_i, each of which is attached with the semantic link types in L_i, and a rule set rs_i where each rule is only involved in semantic link types from L_i. Clearly, $R_i \subseteq R$, $L_i \subseteq L$, and $rs_i \subseteq rs$. Thus, $S_i(R_i, L_i, rs_i)$ is a sub-schema of $\hat{S}(R, L, rs)$.
(3) $\hat{S}(R, L, rs)$ is decomposed into k sub-schemas $S_i(R_i, L_i, rs_i)$, where $1 \le i \le k$.

Theorem 2. *The decomposition by Algorithm 1 is loss-less.*

Proof. From the algorithm and the definition of closure, for each rule $r : \alpha \cdot \beta \Rightarrow \gamma$ in rs, α, β, γ are semantic relative. There is some i such that $\alpha, \beta, \gamma \in L_i$ for $\{L_1, L_2, \ldots, L_k\}$ is a classification of L. So, $rs = rs_1 \cup rs_2 \cup \cdots \cup rs_m$. For any rule $r : \alpha \cdot \beta \Rightarrow \gamma$, there is only one i such that $r \in rs_i$ and $\alpha, \beta, \gamma \in L_i$ due to the decomposition construct a classification. So, we do not need any reference relation among schemas. According to Theorem 1, the decomposition is a loss-less one. □

The classification of the schema of semantic link type can be easily found from the semantic relative net. Different unconnected parts determine different sub-schemas. For a RC-NF1 SLN schema, its semantic relative net is a connected graph. That means reasoning on such a semantic link network is closed. However, this does not mean that any two semantic link types are semantically related.

Let $\hat{S}(R, L, rs)$ be an SLN schema, $\alpha \in L$ is called a top semantic link type if there is no semantic link type $\alpha' (\neq \alpha)$ such that $\alpha \lhd \alpha'$. α is called a bottom semantic link type if there is no semantic link type $\alpha' (\neq \alpha)$ such that $\alpha' \lhd \alpha$. Two top semantic link types or two bottom ones are not semantic relative. α is called an isolated semantic link type if there is no semantic link type $\alpha' (\neq \alpha)$ such that $\alpha \lhd \alpha'$ or $\alpha' \lhd \alpha$. An isolated semantic link type is both top and bottom. The following algorithm finds all bottom semantic link types for an SLN schema.

Algorithm 2. For an SLN schema $\hat{S}(R, L, rs)$, find the set B of all bottom semantic link types in $\hat{S}$.

(1) Let $B = \{\alpha | \alpha_i \cdot \alpha_j \Rightarrow \alpha \in rs\}$, where $\alpha_i, \alpha_j \in L$;
(2) Loop for each $\alpha \in B$. Check each rule r in rs, if α occurs in the precondition of r and does not occur in the result of r, remove α from B; else check next rule in rs.

The bottom semantic link types cannot affect other link types in reasoning. We can compute all semantic link types that affect a certain semantic link type.

Definition 4. Let $\hat{S}(R, L, rs)$ be an SLN schema, and $\alpha \in L$. The up-closure of α with respect to the rule set rs, denoted as $C_{up}(\alpha)$, is a set of semantic link types derived from the following steps.

(1) Let $C_{up}(\alpha) = \{\alpha\}$;
(2) Check L, for each semantic relative $\alpha_i \lhd \alpha_j$, if $\alpha_i \in C_{up}(\alpha)$, let $C_{up}(\alpha) = C_{up}(\alpha) \cup \{\alpha_j\}$;
(3) Repeat from step 2 until $C_{up}(\alpha)$ does not change.

Lemma 4. *Let $\alpha_1 \unlhd \alpha_2 \unlhd \alpha_3 \unlhd \cdots \unlhd \alpha_m \unlhd \alpha_1$ be a semantic relative circle, then*

(1) $C_{up}(\alpha_i)$ is identical to $C_{up}(\alpha_j)$ for all $1 \le i, j \le m$.
(2) $\alpha_j \in C_{up}(\alpha_i)$, for any i and j, $1 \le i, j \le m$.

Definition 5. An SLN schema $\hat{S}(R, L, rs)$ is in Rule-Constraint Normal Form 2 (RC-NF2), if it satisfies (1) $\hat{S}(R, L, rs)$ is RC-NF1; and, (2) $\beta \lhd \alpha$ for bottom semantic link type β in L and any other link type α.

Obviously, for a RC-NF2 schema, the up closure of the bottom semantic link type β according to the rule set rs is just L, i.e., $L = C_{up}(\beta)$.

Lemma 5. *For a RC-NF2 SLN schema $\hat{S}(R, L, rs)$, it has a* unique *bottom semantic link type.*

Proof. Assume that there are at least two different bottom semantic link types $\alpha_{b_1} \neq \alpha_{b_2}$. According to Definition 4, $C_{up}(\alpha_{b_1}) = C_{up}(\alpha_{b_2})$. So, we have $\alpha_{b_1} \in C_{up}(\alpha_{b_2})$, $\alpha_{b_2} \unlhd \alpha_{b_1}$. Similarly, $\alpha_{b_1} \unlhd \alpha_{b_2}$. It leads to a contradiction, so the lemma holds. □

Lemma 6. *For an SLN schema $\hat{S}(R, L, rs)$, let $\hat{S}_1(R_1, L_1, rs_1)$ and $\hat{S}_2(R_2, L_2, rs_2)$ be two RC-NF2 sub-schemas. If $\alpha \in \hat{S}_1 \cap \hat{S}_2$, $C^{(1)}_{up}(\alpha)$ in $\hat{S}_1$ is identical to $C^{(2)}_{up}(\alpha)$ in $\hat{S}_2$.*

Proof. For a semantic link type $\alpha_0 \in C^{(1)}_{up}(\alpha)$, $\alpha \unlhd \alpha_0$. Let α_{b_1} be the bottom semantic link types for $\hat{S}_1$ and α_{b_2} for $\hat{S}_2$ according to Lemma 5. Obviously, $\alpha_{b_1} \unlhd \alpha$ and $\alpha_{b_2} \unlhd \alpha$. Thus $\alpha_{b_2} \unlhd \alpha_0$, which means that $\alpha_0 \in C^{(2)}_{up}(\alpha)$. □

Lemma 7. *$\hat{S}_1(R_1, L_1, rs_1)$ and $\hat{S}_2(R_2, L_2, rs_2)$ are two RC-NF2 sub-schemas of $\hat{S}(R, L, rs)$ with bottom semantic link types α_{b_1} and α_{b_2} respectively. Let $E = L_1 \cap L_2$, if $E \neq \phi$, then there does not exist any semantic relative $\alpha \unlhd \beta$ satisfying that $\alpha \in E$ and $\beta \notin E$; and, for each semantic link type $\alpha \in E$, we have $\alpha_{b_1} \lhd \alpha$ and $\alpha_{b_2} \lhd \alpha$.*

Proof. For $\alpha \in E$, and a semantic relative $\alpha \unlhd \beta$. $E = L_1 \cap L_2$, then $\alpha \in L_1$. For L_1 is the closure of a bottom link type according to the semantic relatives, then $\beta \in L_1$. Similarly, we can get $\beta \in L_2$. So $\beta \in E$. The second part of the Lemma is trivial from the definition. □

Reasoning is closed in a RC-NF2 SLN schema. Different semantic link networks based on different RC-NF2 sub-schemas of the same original schema are reasoning closed and independent. And, the reasoning service is more efficient and easier to execute in a sub-schema than in the original one.

For an application, the whole schema may include several sub-schemas with RC-NF2.

We can decompose an RC-NF1 SLN schema into several RC-NF2 sub-schemas according to the following algorithm.

Algorithm 3. Let $\hat{S}(R, L, rs)$ be a RC-NF1 SLN schema.

(1) Find all bottom semantic link types of $\hat{S}$, denoted as α_{b_1}, $\alpha_{b_2}, \ldots$, and α_{b_m}.

Fig. 4. Semantic link types relative nets for RC-NF2 schemas of sub-schema $\hat{S}_{1.1}$, $\hat{S}_{1.2}$, and $\hat{S}_{1.3}$.

(2) For each α_{b_i} $(1 \le i \le m)$, compute its up-closure according to reasoning rule set *rs*, denoted as $C_{up}(\alpha_{b_i})$. Then, we can get a resource type set R_{b_i}, in which each element is involved at least one link type in L_{b_i}, and a rule set rs_{b_i} in which each rule is involved only link types from L_{b_i}. Clearly, $R_{b_i} \subseteq R$, $L_{b_i} \subseteq L$, and $rs_{b_i} \subseteq rs$. Thus, $\hat{S}_{b_i}(R_{b_i}, L_{b_i}, rs_{b_i})$ is a sub-schema of $\hat{S}(R, L, rs)$.

(3) $\hat{S}(R, L, rs)$ can be decomposed into k sub-schemas $\hat{S}_{b_i}(R_{b_i}, L_{b_i}, rs_{b_i})$, where $1 \le i \le m$.

Theorem 3. *The decomposition by Algorithm 3 is loss-less.*

Proof. For a rule r: $\alpha \cdot \beta \Rightarrow \gamma$ in *rs*, suppose $\gamma \in L_{b_i}$ for some *i*. $\gamma \lhd \beta, \gamma \lhd \alpha$, so $\alpha, \beta \in L_{b_i}$ and $r \in rs_{b_i}$. Then, $rs = rs_1 \cup rs_2 \cup \cdots \cup rs_m$. Definition 5 and Algorithm 3 show that the sub-schema $\hat{S}_{b_i}(R_{b_i}, L_{b_i}, rs_{b_i})$ is based on the bottom link type b_i. For any rule r: $\alpha \cdot \beta \Rightarrow \gamma$ in rs_{b_i}, for some *j*, if $\gamma \in L_{b_j}$, then $\alpha, \beta \in L_{b_j}$ and $r \in rs_j$. There is no reference relation needed among sub-schemas. According to Theorem 1, the decomposition from Algorithm 3 is loss-less. □

Definition 6. Let $\hat{S}(R, L, rs)$ be a SLN schema, and $\alpha \in L$ be a semantic link type. The down closure of α with respect to the rule set *rs*, denoted as $C_{down}(\alpha)$, is a set of semantic link types derived from the following steps.

(1) Let $C_{down}(\alpha) = \{\alpha\}$;
(2) Check *L*, for each semantic relative $\alpha_j \lhd \alpha_i$, if $\alpha_i \in C_{down}(\alpha)$, let $C_{down}(\alpha) = C_{down}(\alpha) \cup \{\alpha_j\}$;
(3) Repeat from step 2 until $C_{down}(\alpha)$ does not change.

Intuitively, the down closure of a link type α is the set of all link types which are affected by α. We can easily get the following characters.

Lemma 8. *For a top link type α_t in the SLN schema $\hat{S}(R, L, rs)$, the down closure $C_{down}(\alpha_t)$ constructs a tree with root α_t in the semantic link type relative net.*

Lemma 9. *For two top link types α_1 and α_2 in the SLN schema $\hat{S}(R, L, rs)$, if $C_{down}(\alpha_1) \cap C_{down}(\alpha_2) \neq \phi$, then α_1 and α_2 are in the same RC-NF2 sub-schemas from Algorithm 3.*

In some cases, two different SC-NF2 schemas may share a common segment. Such a common segment leads to redundancy. In Fig. 4, for example, $\hat{S}_{1.1}$ and $\hat{S}_{1.2}$ share a common segment that includes two semantic link types: *publishedIn* and *editorOf*. In fact, we can deal with such a redundancy by decomposing the schemas into advanced forms and building reference relations between schemas.

Definition 7. For an SLN schema $\hat{S}(R, L, rs)$, let $\hat{S}_1(R_1, L_1, rs_1)$, $\hat{S}_2(R_2, L_2, rs_2), \ldots$, and $\hat{S}_k(R_k, L_k, rs_k)$ be a RC-NF2 level decomposition. We call such a decomposition redundancy-free if $\hat{S}_i$ and $\hat{S}_j$ do not share any semantic relative for all $1 \le i, j \le m$.

Algorithm 4. For an SLN schema $\hat{S}(R, L, rs)$, let $\hat{S}_1(R_1, L_1, rs_1)$ and $\hat{S}_2(R_2, L_2, rs_2)$ be two sub-schemas of the RC-NF2 level decomposition from Algorithm 3. If $\hat{S}_1$ and $\hat{S}_2$ share a set of semantic relatives *SR*, let $I = L_1 \cap L_2$.

(1) A new schema $\hat{S}_I(R_I, I, rs_I)$ can be constructed according to *I*.
(2) Decompose the schema into several sub-schemas according Algorithm 2: $\hat{S}_{I_1}, \hat{S}_{I_2}, \ldots,$ and $\hat{S}_{I_m}$.
(3) Denote β_{I_k} as the bottom link type in $\hat{S}_{I_k}$. Build the reference relations $Ref(\beta_{I_k} \in \hat{S}_1, \beta_{I_k} \in \hat{S}_{I_k})$ and $Ref(\beta_{I_k} \in \hat{S}_2, \beta_{I_k} \in \hat{S}_{I_k})$ for each k $(1 \le k \le m)$.

We can use Algorithm 4 to repeatedly deal with redundant problems of the decomposing sub-schemas from Algorithm 3 and the sub-schemas from Algorithm 4 are all in RC-NF2. However, in many cases, such a decomposition may produce some small patch sub-schemas. In fact, we can deal the redundancy by building reference relations directly between the schemas rather than creating new sub-schemas. The following algorithm is an amendment of Algorithm 3. It provides an approach to decompose an RC-NF1 schema into several redundancy-free RC-NF2 sub-schemas.

Algorithm 5. Let $\hat{S}(R, L, rs)$ be a RC-NF1 SLN schema.

(1) Let *reflist* $= \phi$.
(2) Find all bottom semantic link types of $\hat{S}$, denoted as α_{b_1}, $\alpha_{b_2}, \ldots$, and α_{b_m}.