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

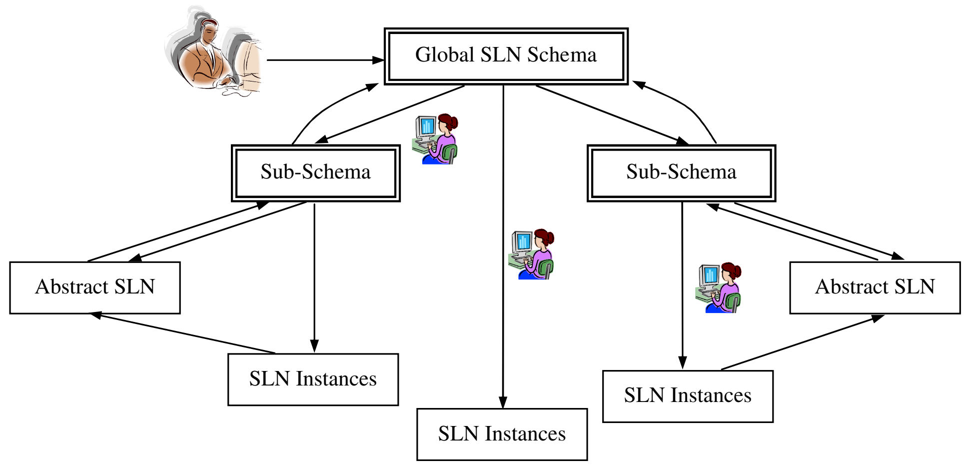

Fig. 1. The role of SLN schema.

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.

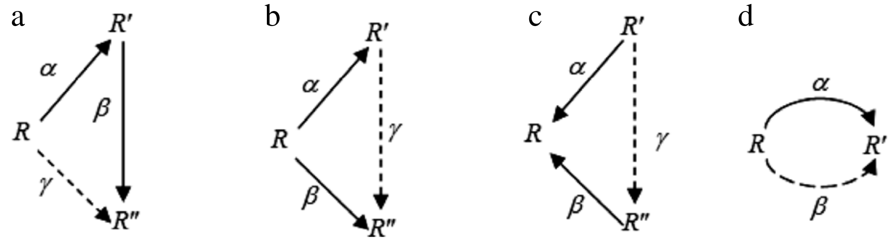

Fig. 2. Basic forms of reasoning rule.

(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.

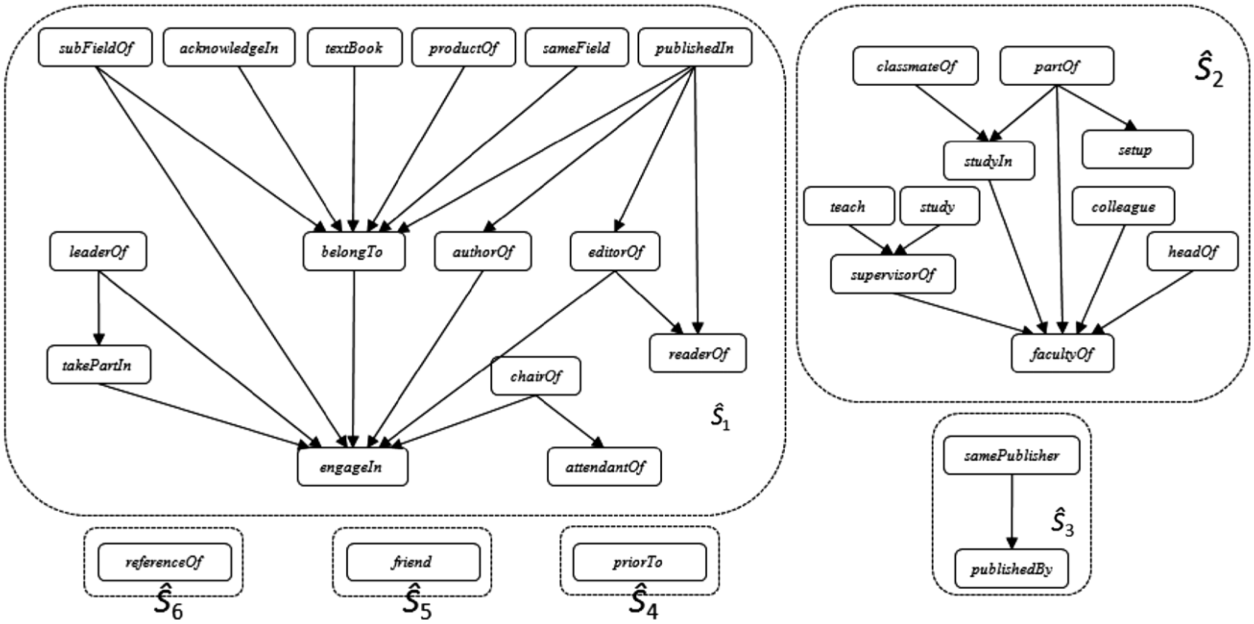

Fig. 3. Semantic link types relative net for RC-NF1 schemas of a research semantic link network.

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 , α_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 $\alpha' (\ne \alpha)$ such that $\alpha \lhd \alpha'$. α is called a bottom semantic link type if there is no semantic link type $\alpha' (\ne \alpha)$ 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 $\alpha' (\ne \alpha)$ 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 $\alpha_{b_1} \ne \alpha_{b_2}$. 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 $E \ne \phi$, 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}.

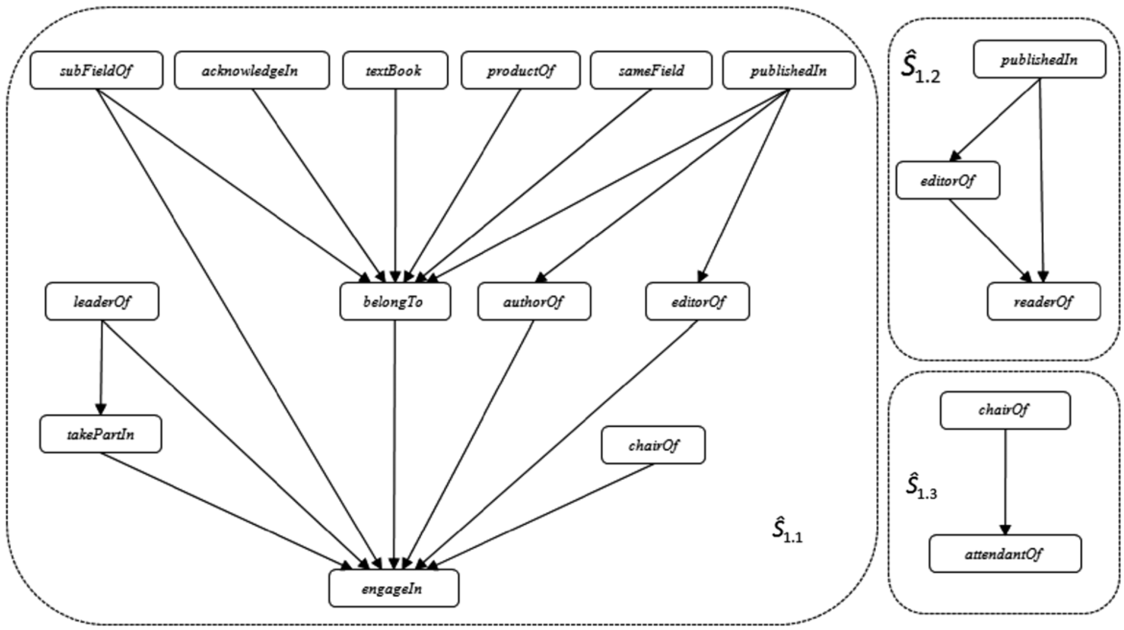

Fig. 4. Semantic link types relative nets for RC-NF2 schemas of sub-schema $\hat{S}_{1.1}$, $\hat{S}_{1.2}$, and $\hat{S}_{1.3}$.

(2) For each α_{b_i} ($1 \leq i \leq m$), 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 $1 \leq i \leq m$.

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 $\hat{S}_1(R_1, L_1, rs_1)$, $\hat{S}_2(R_2, L_2, rs_2)$, ..., 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 $1 \leq i, j \leq m$.

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 ($1 \leq k \leq m$).

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}.

(3) For each α_{b_i} ($1 \le i \le m$), compute its up-closure according to reasoning rule set *rs*, denoted as $C_{up}(\alpha_{b_i})$.
(4) For each pair of $(\alpha_{b_i}, \alpha_{b_j})$, $i \ne j$, let $I_{ij} = C_{up}(\alpha_{b_i}) \cap C_{up}(\alpha_{b_j})$, find all bottom link types of I_{ij}, denoted as $\beta_1, \beta_2, \ldots$, and β_k respectively. Let $L_{b_i} = C_{up}(\alpha_{b_i})$ and $L_{b_j} = (C_{up}(\alpha_{b_j}) - I_{ij}) \cup \{\beta_1, \beta_2, \ldots, \beta_k\}$. Let $reflist = reflist \cup \{(\beta_1, L_{b_j}, L_{b_i}), (\beta_2, L_{b_j}, L_{b_i}), \ldots, (\beta_k, L_{b_j}, L_{b_i})\}$.
(5) We get the sub-schema $\hat{S}_{b_i}(R_{b_i}, L_{b_i}, rs_{b_i})$ according to L_{b_i}, and build reference relations $Ref(\beta_t \in \hat{S}_{b_j}, \beta_t \in \hat{S}_{b_i})$ for each $(\beta_t, L_{b_j}, L_{b_i}) \in reflist$. Clearly, $R_{b_i} \subseteq R$, $L_{b_i} \subseteq L$, and $rs_{b_i} \subseteq rs$. $\hat{S}_{b_i}(R_{b_i}, L_{b_i}, rs_{b_i})$ is a sub-schema of $\hat{S}(R, L, rs)$.

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

Deduction 1. The decomposition by Algorithm 5 is loss-less.

5. Schema maintenance and reasoning

5.1. Schema maintenance

Updating an SLN schema concerns resource types, semantic link types, and reasoning rules. The essence of semantic rule-constraint normal form is to classify the semantic link type set into different parts according to the rule set. So only the updating of the rule set can lead to different sub-schemas. The resource type set and semantic link type set for the sub-schemas will also change. Therefore, we can study the updating issues according to the variation of the rule set.

The following algorithm is to modify decomposition after adding rules to schemas.

Algorithm 6 (*Schema Revision after Extension*). Let $\{\hat{S}_1(R_1, L_1, rs_1), \hat{S}_2(R_2, L_2, rs_2), \ldots, \hat{S}_n(R_n, L_n, rs_n)\}$ be the RC-NF2 schemas decomposed from SLN schema $\hat{S}(R, L, rs)$. A new semantic link type set $\{\alpha'_1, \alpha'_2, \ldots, \alpha'_l\}$ and a new rule set $\{r'_1, r'_2, \ldots, r'_k\}$ are appended to the schema $\hat{S}$. The extension of $\hat{S}$ is denoted as $\hat{S}_E(R_E, L_E, rs_E)$.

(1) Take all new semantic link types as isolated semantic link types firstly, and then construct new sub-schemas as $\hat{S}'_1(R'_1, L'_1, \phi), \hat{S}'_2(R'_2, L'_2, \phi), \ldots, \hat{S}'_l(R'_l, L'_l, \phi)$. Let $\hat{S}_E = \{\hat{S}_1, \hat{S}_2, \ldots, \hat{S}_n, \hat{S}'_1, \hat{S}'_2, \ldots, \hat{S}'_l\}$.
(2) For a rule $\alpha_{i1} \cdot \alpha_{i2} \Rightarrow \alpha_j$ in $rs_E - rs$, we get semantic relatives $\alpha_j \trianglelefteq \alpha_{i1}$ and $\alpha_j \trianglelefteq \alpha_{i2}$. For a rule $\alpha_i \Rightarrow \alpha_j$ in $rs_E - rs$, we get a semantic relative $\alpha_j \trianglelefteq \alpha_i$.
(3) For each semantic relative $\alpha_j \trianglelefteq \alpha_i$ retrieved in step 2, the list of sub-schemas in which α_i involved are $\hat{S}_{i_1}, \hat{S}_{i_2}, \ldots, \hat{S}_{i_{1s}}$, and the list of sub-schemas in which α_j involved are $\hat{S}_{j_1}, \hat{S}_{j_2}, \ldots, \hat{S}_{j_t}$.
   (1) Compute the up-closure $C_{up}(\alpha_i)$ for α_i in $\hat{S}_{i1}$, let $L_{j_p} = L_{j_p} \cup C_{up}(\alpha_i)$, for all $1 \le p \le t$;
   (2) Union all resource types involved in $C_{up}(\alpha_i)$ into R_{j_p}, and union all rules involved in $C_{up}(\alpha_i)$ into rs_{j_p}; and,
   (3) Check if $\hat{S}_{j_p} \subseteq \hat{S}_{j_q}$ or not for each pair (p, q), $1 \le p \le s$, $1 \le q \le t$. If yes, remove $\hat{S}_{j_p}$ from the list of S_E.

The following algorithm is for schema revision after deleting some rules.

Algorithm 7. Let $\{\hat{S}_1(R_1, L_1, rs_1), \hat{S}_2(R_2, L_2, rs_2), \ldots, \hat{S}_n(R_n, L_n, rs_n)\}$ be the RC-NF2 schemas decomposed from SLN schema $\hat{S}(R, L, rs)$. Denote the deleted resource type set as R_D, the semantic link types set as L_D, and the deleted rules set rs_D.

(1) For each semantic sub-schema $\hat{S}_i$, let $R_i = R_i - R_D$, $L_i = L_i - L_D$, and $rs_i = rs_i - D$. Applying Algorithm 1 or 3 on $\hat{S}_i$ to get a decomposition for $\hat{S}_i$ with corresponding semantic constraint normal form. Assume that the decomposition is $\{\hat{S}_{i_1}(R_{i_1}, L_{i_1}, rs_{i_1}), \hat{S}_{i_2}(R_{i_2}, L_{i_2}, rs_{i_2}), \ldots, \hat{S}_{i_k}(R_{i_k}, L_{i_k}, rs_{i_k})\}$.
(2) Unite all sub-schemas retrieved in step 1, denote the new set of sub-schemas as $\{\hat{S}'_1(R'_1, L'_1, rs'_1), \hat{S}'_2(R'_2, L'_2, rs'_2), \ldots, \hat{S}'_m(R'_m, L'_m, rs'_m)\}$.
(3) Check if $\hat{S}'_i \subseteq \hat{S}'_j$ or not for each pair (i, j), $1 \le i, j \le m$. If yes, remove $\hat{S}'_i$.

5.2. Reasoning algorithms

Several kinds of reasoning can be carried out on a semantic link network. The basic reasoning is to obtain the potential semantic relationships between resources. The following algorithm is for deriving potential semantic links.

Algorithm 8. Let R, R', and R'' be resources of a semantic link network instance S of schema $\hat{S}(R, L, rs)$, the set of semantic links from R to R' and from R' to R'' be $A = \{\alpha_1, \alpha_2, \ldots, \alpha_s\}$ and $B = \{\beta_1, \beta_2, \ldots, \beta_t\}$ respectively, the set of known semantic links from R to R'' be C_0, and the types of R, R', and R'' are rt_1, rt_2, and rt_3 respectively.

(1) Let $C = \phi$.
(2) For each pair (α_i, β_j) in $A \times B$, check rs to determine if there is a rule in form of $\alpha_i \cdot \beta_j \Rightarrow \gamma_k$. If yes, let $C = C \cup \gamma_k$, else skip to the next pair.
(3) Let $C = C \cap [rt_1, rt_2]$.
(4) $C = C \cup C_0$ is the solution.

The measure of $A \times B$ is a constant $s \times t$ for $A \subseteq [rt_1, rt_2]$, $B \subseteq [rt_2, rt_3]$. Meanwhile, the size of rule set rs is a constant. So, the complexity of the above algorithm is $O(1)$. For a connected path $R_0 \xrightarrow{\alpha_1} R_1 \xrightarrow{\alpha_2} \cdots \xrightarrow{\alpha_n} R_n$ with length n, the algorithm to compute the semantic relations between R_0 and R_n is to recursively use Algorithm 7. And, the complexity is $O(n)$.

Moreover, we can compute the semantic relations between any two resources. The immediate idea is to find all connected paths between them, compute the semantic relations of each connected path, and then combine all semantic relations between them. For a semantic link network with m resources and n semantic links, under the assumption of equal distribution, there are $d = \frac{n}{m(m-1)}$ links between two resources on average. Therefore, from R_0 to R_n, the number of paths with length k is $P_{m-2}^{k-1}d^k$. Thus, the complexity for computing the semantic relations between any two resources is $O(\sum_{k=1}^{m-1} kP_{m-2}^{k-1}d^k) = O(\sum_{k=1}^{m-1} k(m-2)!n^k/((k-1)!m^k\,(m-1)^k))$. We can see that the complexity depends on the two variables m and n, which could be large. However, it can be reduced by reducing the scale of the semantic link network. The idea of decomposing a semantic link network into RC-NF2 forms can reduce the two variables m and n prominently. And the reasoning in RC-NF2 semantic link networks is closed and independent.

Moreover, we can find some better qualified methods. Indeed, the up-closure can help develop an efficient algorithm to determine semantic links between two resources. The following algorithm is for determining the existence of semantic relation $R \xrightarrow{\alpha} R'$ in a semantic link network.

Algorithm 9. Let rs be the reasoning rule set of semantic link network S and the types of R and R' be rt and rt' respectively.

(1) Determine if $\alpha \in [rt, rt']$. If no, return false; otherwise, continue the next step.

(2) Compute the up-closure of α with respect to rs, denoted as $C_{up}(\alpha)$.
(3) Compute the projection of S with respect to $C_{up}(\alpha)$, denoted as S_0, which may be constituted by several unconnected segments or some isolated resources, where each segment is interconnected.
(4) If R and R' are in two different segments, return false.
(5) If R and R' are in the same segment S, return determination if $R \xrightarrow{\alpha} R'$ is true in S by using Algorithm 8.

This algorithm is efficient because the up-closure for a semantic link type may be much smaller than the original link type set, and the projection may be sparse in light of semantic links. The following algorithm computes semantic links set $\Gamma = \{\alpha | R \xrightarrow{\alpha} R'\}$.

Algorithm 10. Let S be a semantic link network, and the types of R and R' be rt and rt' respectively, then any potential semantic link type between R and R' is in $[rt, rt']$.

(1) Let $\Gamma = \phi$.
(2) For each semantic link type $\alpha \in [rt, rt']$, execute Algorithm 8 to determine if $R \xrightarrow{\alpha} R'$ or not. If yes, let $\Gamma = \Gamma \cup \{\alpha\}$; else skip to the next semantic link type.
(3) Return Γ.

For a given resources R, the following algorithm computes the resource set $A = \{R_i | R \xrightarrow{\alpha} R_i\}$. The set A is to point out all resources which have an α relation with R in a semantic link network.

Algorithm 11. Let S be a semantic link network, and the type of R be rt.

(1) Compute the up-closure of α with respect to rs, denoted as $C_{up}(\alpha)$;
(2) Compute the projection of S with respect to $C_{up}(\alpha)$, denoted as S_0;
(3) Denote $T = \{rt_i | \alpha \in [rt, rt_i]\}$ and $A = \{R' | R' \in S_0$, and the type of R' is in $T\}$;
(4) For each resource $R' \in A$, applying Algorithm 8 to determine if $R \xrightarrow{\alpha} R'$ or not. If no, let $A = A - \{R\}$; else skip to the next resource in A;
(5) Return A.

The algorithm for computing the set of resource pair $\{(R, R') | R \xrightarrow{\alpha} R'\}$ for a given semantic link type α is similar to Algorithm 11. However, the complexity would be much higher because both the start and the end resources need to compute.

6. Case study

6.1. Schema example of science network

We firstly construct a schema example of science network consisting of three components, and then discuss the decomposition of the schema:

(1) a set of resource types {*University*, *Department*, *Professor*, *Student*, *Course*, *Book*, *Project*, *Journal*, *Paper*, *Conference*, *Proceeding*, *Field*, *Publisher*};
(2) semantic link types between these resource types as shown in Table 1; and,
(3) reasoning rules for these semantic link types as shown in Table 2.

6.2. Decompose the schema into RC-NF1

We construct the semantic link type relative net, which consists of six parts as shown in Fig. 3. Therefore, we get the following six RC-NF1 sub-schemas according to Algorithm 1.

Sub-schema $\hat{S}_1$:
R_1 = {*Professor*, *Student*, *Course*, *Book*, *Project*, *Journal*, *Paper*, *Conference*, *Proceeding*, *Field*},
L_1 = {*acknowledgeIn*, *textbook*, *sameField*, *publishedIn*, *subFieldOf*, *belongTo*, *authorOf*, *editorOf*, *leaderOf*, *readerOf*, *takePartIn*, *chairOf*, *engageIn*, *attenantOf*, *productOf*}, and
rs_1 = {*rule* 1–3, *rule* 5–6, *rule* 8, *rule* 12–24, *rule* 32}.

Sub-schema $\hat{S}_2$:
R_2 = {*Professor*, *Student*, *Department*, *University*, *Course*},
L_2 = {*classmateOf*, *partOf*, *teach*, *studyIn*, *setup*, *supervisorOf*, *colleague*, *headOf*, *facultyOf*} and
rs_2 = {*rule* 4, *rule* 7, *rule* 9–11, *rule* 25–27, *rule* 31, *rule* 33}.

Sub-schema $\hat{S}_3$:
R_3 = {*Journal*, *Proceeding*, *Book*, *Publisher*},
L_3 = {*publishedBy*, *samePublisher*} and
rs_3 = {*rule* 28, *rule* 29}.

Sub-schema $\hat{S}_4$:
R_4 = {*Course*},
L_4 = {*priorTo*}, and
rs_4 = {*rule*30}.

Sub-schema $\hat{S}_5$:
R_5 = {*Professor*, *Student*},
L_5 = {*friend*}, and
$rs_5 = \phi$.

Sub-schema $\hat{S}_6$:
R_6 = {*Paper*, *Book*, *Proceeding*, *Journal*},
L_6 = {*referenceOf*}, and
$rs_6 = \phi$.

6.3. Decompose the schema into RC-NF2

Sub-schemas $\hat{S}_4$, $\hat{S}_5$ and $\hat{S}_6$ satisfy RC-NF2 because each has only one rule. Sub-schema $\hat{S}_3$ is also in RC-NF2 for it has only one bottom link type in the set of semantic link types from Fig. 3. However, sub-schemas $\hat{S}_1$ and $\hat{S}_2$ are not in RC-NF2, so they need to be decomposed into RC-NF2 according to Algorithm 3.

We first find the following three bottom semantic link types in sub-schema $\hat{S}_1$: *engageIn*, *readerOf* and *attenantOf*, and then compute the following three up-closures of them, shown in Fig. 4.
$L_{1.1} = C_{up}(engageIn)$ = {*textbook*, *sameField*, *acknowledgeIn*, *productOf*, *publishedIn*, *subFieldOf*, *belongTo*, *authorOf*, *editorOf*, *leaderOf*, *takePartIn*, *chairOf*, *engageIn*},
$L_{1.2} = C_{up}(readerOf)$ = {*readerOf*, *editorOf*, *publishedIn*}, and,
$L_{1.3} = C_{up}(attenantOf)$ = {*chairOf*, *attenantOf*}.
The corresponding set of involved resource types and involved rules for $L_{1.1}$, $L_{1.2}$ and $L_{1.3}$ are listed as follows.

$R_{1.1}$ = {*Professor*, *Student*, *Course*, *Book*, *Project*, *Journal*, *Paper*, *Conference*, *Proceeding*, *Field*},
$R_{1.2}$ = {*Professor*, *Student*, *Book*, *Journal*, *Paper*, *Proceeding*}, and,
$R_{1.3}$ = {*Professor*, *Student*, *Conference*}.

$rs_{1.1}$ = {*rule* 2, *rule* 6, *rule* 8, *rule* 12–24, *rule* 32},
$rs_{1.2}$ = {*rule* 1, *rule* 5, *rule* 13}, and,
$rs_{1.3}$ = {*rule* 3}.

Therefore, we get three RC-NF2 sub-schemas $\hat{S}_{1.1}(R_{1.1}, L_{1.1}, rs_{1.1})$, $\hat{S}_{1.2}(R_{1.2}, L_{1.2}, rs_{1.2})$, and $\hat{S}_{1.3}(R_{1.3}, L_{1.3}, rs_{1.3})$ from sub-schema $\hat{S}_1$. For sub-schema $\hat{S}_2$, we get two bottom semantic link types in

Table 1
Semantic link types in scientific research SLN schema.

Resource type pairs	Semantic link type sets	Explanation
[*Professor, Professor*]	{*colleague, friend*}	A professor may be a colleague or a friend of another professor.
[*Professor, Student*]	{*supervisorOf*}	A professor may be a supervisor of a student.
[*Professor, Course*]	{*teach*}	A professor may teach a course.
[*Professor, field*]	{*engageIn*}	A professor may engage in a research field.
[*Professor, Department*]	{*facultyOf, headOf*}	A professor may be a faculty or the head of a department.
[*Professor, Project*]	{*leaderOf, takePartIn*}	A professor may be a leader of or take part in a project.
[*Professor, Paper*]	{*authorOf, readerOf, editorOf*}	A professor may be an author, a reader, or an editor of a paper.
[*Professor, University*]	{*facultyOf*}	A professor may be a faculty of a university.
[*Professor, Journal*]	{*readerOf, authorOf, editorOf*}	A professor may be an author, a reader, or an editor of a journal.
[*Professor, Conference*]	{*chairOf, attendantOf*}	A professor may be a chair, or an attendant of a conference.
[*Professor, Proceeding*]	{*authorOf, readerOf, editorOf*}	A professor may be an author, a reader, or an editor of a proceeding.
[*Professor, Book*]	{*readerOf, authorOf, editorOf*}	A professor may be an author, a reader, or an editor of a book.
[*Student, Student*]	{*classmateOf*}	Two students may be classmates.
[*Student, Paper*]	{*readerOf, authorOf*}	A student may be an author or a reader of a paper.
[*Student, Journal*]	{*readerOf, authorOf*}	A student may be an author or a reader of a journal.
[*Student, Conference*]	{*attendantOf*}	A professor may be an attendant of a conference.
[*Student, Proceeding*]	{*authorOf, readerOf*}	A student may be an author or a reader of a proceeding.
[*Student, Project*]	{*takePartIn*}	A student may take part in a project.
[*Student, Field*]	{*engageIn*}	A student may engage in a research field.
[*Student, University*]	{*studyIn*}	A student may study in a university.
[*Student, Department*]	{*studyIn*}	A student may study in a department.
[*Student, Course*]	{*study*}	A student may study a course.
[*Student, Book*]	{*readerOf, authorOf*}	A student may be an author or a reader of a book.
[*Course, Course*]	{*priorTo, sameField*}	A course may be prior to another one, and two courses may be in the same field.
[*Course, Field*]	{*belongTo*}	A course may involve in a research field.
[*Book, Book*]	{*referenceOf, sameField, samePublisher*}	A book may reference another one, and two books may be in the same field, or have the same publisher or the same author.
[*Book, Course*]	{*textbook, sameField*}	A book may be a textbook, and a book and a course may be in the same field.
[*Book, Paper*]	{*referenceOf, sameField*}	A book may reference a paper, and a book and a paper may be in the same field or have the same author or have the same publisher.
[*Book, Journal*]	{*referenceOf, sameField, samePublisher*}	A book may be a reference of a journal, and a book and a journal may be in the same field or have the same author or have the same publisher.
[*Book, Proceeding*]	{*referenceOf, sameField, samePublisher*}	A book may be a reference of a proceeding, and a book and a proceeding may be in the same field or have the same author or have the same publisher.
[*Book, Publisher*]	{*publishedBy*}	A book may be published by a publisher.
[*Paper, Paper*]	{*referenceOf, sameField*}	A paper may reference another one, and two papers maybe in the same field or have the same author.
[*Paper, Journal*]	{*publishedIn, sameField*}	A paper may be published in a journal, and a paper and a journal may be in the same field.
[*Paper, Proceeding*]	{*publishedIn, sameField*}	A paper may be published in a proceeding, and a paper and a proceeding may be in the same field.
[*Proceeding, Field*]	{*belongTo*}	A proceeding may belong to a research field.
[*Proceeding, Conference*]	{*productOf*}	A proceeding may be a product of a certain conference.
[*Proceeding, Proceeding*]	{*sameField, samePublisher*}	Two proceedings may be in the same field, or have the same publisher or the same author.
[*Proceeding, Publisher*]	{*publishedBy*}	A proceeding may be published by a publisher.
[*Department, Course*]	{*setup*}	A course may be setup by a department.
[*University, Course*]	{*setup*}	A course may be setup by a university.
[*Department, University*]	{*partOf*}	A department may be a part of a university.
[*Project, Book*]	{*acknowledgeIn*}	Aproject may be acknowledged in a book.
[*Project, Paper*]	{*acknowledgeIn*}	Aproject may be acknowledged in a paper.
[*Project, Field*]	{*belongTo*}	A project may belong to a research field.
[*Conference, Conference*]	{*sameField*}	Two conferences may be in the same field or have the same chair.
[*Conference, Field*]	{*belongTo*}	A conference may belong to a research field.
[*Journal, Journal*]	{*sameField, samePublisher*}	Two journals may be in the same field or have the same publisher.
[*Journal, Proceedings*]	{*sameField, samePublisher*}	A journal and a proceeding may be in the same field or have the same publisher.
[*Journal, Field*]	{*belongTo*}	A journal may belong to a research field.
[*Journal, Publisher*]	{*publishedBy*}	A journal may be published by a publisher.
[*Field, Field*]	{*subFieldOf*}	A research field may be a sub of another one.

facultyOf and *setup*, and then compute the following up-closures of them, shown in Fig. 5.

$L_{2.1} = C_{up}(facultyOf) = \{classmateOf, partOf, teach, studyIn, supervisorOf, colleague, headOf, facultyOf\}$,
$L_{2.2} = C_{up}(setup) = \{setup, partOf\}$.

The corresponding set of involved resource types and involved rules for $L_{2.1}$ and $L_{2.2}$ are listed as follows.
$R_{2.1} = \{Professor, Student, Department, University, Course\}$, and
$R_{2.2} = \{Department, University, Course\}$.
$rs_{2.1} = \{rule\ 4, rule\ 7, rule\ 9\text{–}11, rule\ 26\text{–}27, rule\ 31, rule\ 33\}$, and
$rs_{2.2} = \{rule\ 25\}$.

Then, we get two RC-NF2 sub-schemas $\hat{S}_{2.1}(R_{2.1}, L_{2.1}, rs_{2.1})$ and $\hat{S}_{2.2}(R_{2.2}, L_{2.2}, rs_{2.2})$ from sub-schema $\hat{S}_2$.

To sum up, we get nine RC-NF1 sub-schemas from the original schema: $\hat{S}_{1.1}, \hat{S}_{1.2}, \hat{S}_{1.3}, \hat{S}_{2.1}, \hat{S}_{2.2}, \hat{S}_3, \hat{S}_4, \hat{S}_5$, and $\hat{S}_6$. According to Theorems 2 and 3, the decomposition is loss-less. Clearly, the number of semantic links in a semantic link network influences the reasoning complexity. Dividing a schema into several RC-NF2 sub-schemas can significantly reduce the scale so that the query and reasoning can be executed within a small scale. The soundness and the integrity of the query and reasoning on sub-schemas and up-closure are guaranteed by previous sections.

6.4. *Redundancy-free decomposition*

In most cases, the decomposition to RC-NF2 is enough for the reason that a RC-NF2 schema is the smallest unit in the sense of

Table 2
Rule set in scientific research SLN schema.

No.	Rules	Explanation of the rule
rule 1	*editorOf* ⇒ *readerOf*	An editor of a paper, a journal, a proceeding, or a book is also a reader.
rule 2	*leaderOf* ⇒ *takePartIn*	A professor is a leader of a project means that he also takes part in the project.
rule 3	*chairOf* ⇒ *attendantOf*	A chair of a conference is also an attendant.
rule 4	*headOf* ⇒ *facultyOf*	A head of a department or institute is also a faculty.
rule 5	*readerOf* · *publishedIn* ⇒ *readerOf*	A reader of a paper published in a journal or a proceeding is also a reader of the journal or the proceeding.
rule 6	*authorOf* · *publishedIn* ⇒ *authorOf*	An author of a paper published in a journal or a proceeding is also an author of the journal or the proceeding.
rule 7	*facultyOf* · *partOf* ⇒ *facultyOf*	A faculty of a department which is a part of a university is also a faculty of the university.
rule 8	*editorOf* · *belongTo* ⇒ *engageIn*	An editor of a paper, a journal, a proceeding, or a book engages in the field of the involved field.
rule 9	*colleague* · *facultyOf* ⇒ *facultyOf*	A colleague of a faculty of a department, an institute, or a university is also a faculty.
rule 10	*supervisorOf* · *studyIn* ⇒ *facultyOf*	If a student study in a department or a university, then his supervisor is a faculty of the department or the university.
rule 11	*facultyOf* · *partOf* ⇒ *facultyOf*	If a professor is a faculty of a department, then he is also a faculty of the university.
rule 12	*publishedIn* · *belongTo* ⇒ *belongTo*	If a paper is published in a journal or a proceeding of a field, then the paper is in the same field.
rule 13	*editorOf* · *publishedIn* ⇒ *editorOf*	An editor of a paper is also an editor of the journal or the proceeding which includes the paper.
rule 14	*leaderOf* · *belongTo* ⇒ *engageIn*	If a professor is a leader of a project in a field, then the professor engages in the same field.
rule 15	*takePartIn* · *belongTo* ⇒ *engageIn*	If a professor or a student takes part in a project of a field, then he engages in the same field.
rule 16	*authorOf* · *belongTo* ⇒ *engageIn*	An author engages in the field of his paper or his book.
rule 17	*chairOf* · *belongTo* ⇒ *engageIn*	If a professor is a chair of a conference of a field, then the professor engages in the same field.
rule 18	*editorOf* · *belongTo* ⇒ *engageIn*	An editor of a paper, a journal, a proceeding, or a book of a field engages in the same field.
rule 19	*acknowledgeIn* · *belongTo* ⇒ *belongTo*	If a project is acknowledged in a paper, a proceeding, or a book of a field, then the project is in the same field.
rule 20	*belongTo* · *subFieldOf* ⇒ *belongTo*	If a paper, a journal, a proceeding, or a book is involved in a field, then it is involved in a super field.
rule 21	*engageIn* · *subFieldOf* ⇒ *engageIn*	If a professor or a student engages in a field, then he engages in a super field.
rule 22	*sameField* · *belongTo* ⇒ *belongTo*	If A and B share the same field, and B is involved in the field F, then A is involved in F. A and B are two resources with the type of paper, course, journal, book, or proceeding.
rule 23	*textbook* · *belongTo* ⇒ *belongTo*	If a book B is a textbook of a course C, and C is involved in the field F, then B is involved in F.
rule 24	*productOf* · *belongTo* ⇒ *belongTo*	If a proceeding P is a product of a conference C, and C is involved in the field F, then P is in the field F.
rule 25	$(partOf)^{-1}$·*setup* ⇒ *setup*	A course is setup by a department which is a part of a university means the university setup the course.
rule 26	*classmateOf* · *studyIn* ⇒ *studyIn*	A student and his classmate both study in the same university or the same department.
rule 27	*studyIn* · *partOf* ⇒ *studyIn*	A student studies in a department affiliated to a university means he studies in the university.
rule 28	*samePublisher* · *publishedBy* ⇒ *publishedBy*	If A and B share the same publisher, and B is published by the publisher P, then A is published by P. A and C are resources with type of journal, proceeding, or book.
rule 29	*samePublisher* · *samePublisher*⇒*samePublisher*	If A and B share the same publisher, and B and C share the same publisher, then A and C share the same publisher. A, B and C are resources with type of journal, proceeding, or book.
rule 30	*priorTo* · *priorTo* ⇒ *priorTo*	A course C_1 is prior to C_2, and C_2 is prior to C_3, then C_1 is prior to C_3.
rule 31	*colleague* · *colleague* ⇒ *colleague*	If A is a colleague of B, and B is a colleague of C, then A is a colleague of C.A, B, and C are professors.
rule 32	*subFieldOf* · *subFieldOf* ⇒ *subFieldOf*	If field F_1 is a sub-field of F_2, and F_2 is a sub-field of F_3, then F_1 is a sub-field of F_3.
rule 33	*teach* · $(study)^{-1}$ ⇒*supervisorOf*	If a professor teaches a course and a student studies the course, then the professor is a supervisor of the student.

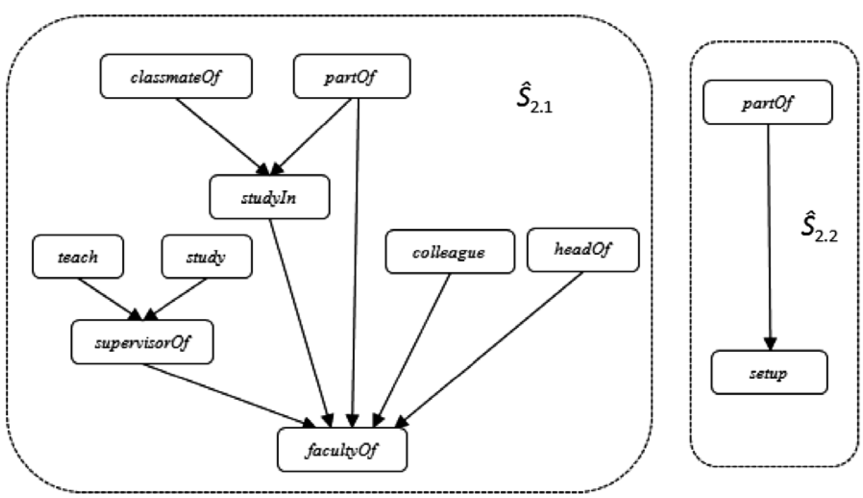

Fig. 5. Semantic link types relative nets for RC-NF2 schemas of sub-schema $\hat{S}_{2.1}$ and $\hat{S}_{2.2}$.

reasoning closed. However, there is some redundancy problem in the above decomposition. In Fig. 4, $\hat{S}_{1.1}$ and $\hat{S}_{1.2}$ share the same semantic relative: *editorOf* $\trianglelefteq$ *publishedIn*. According to Algorithm 4, we get a new decomposition by replacing the sub-schema $\hat{S}_{1.1}$ and $\hat{S}_{1.2}$ with the following three schemas $\hat{S}_{1.1'}$, $\hat{S}_{1.2'}$ and $\hat{S}_{1.4'}$ and the semantic relative net is shown in Fig. 6.

Schema $\hat{S}_{1.1'}$ $(R_{1.1'}, L_{1.1'}, rs_{1.1'})$:
$L_{1.1'} = L_{1.1}$,
$R_{1.1'} = R_{1.1}$, and
$rs_{1.1'} = \{rule\ 2, rule\ 5\text{–}6, rule\ 8, rule\ 12, rule\ 14\text{–}24, rule\ 32\}$;

Schema $\hat{S}_{1.2'}$:
$L_{1.2'} = L_{1.2}$,
$R_{1.2'} = R_{1.2}$, and
$rs_{1.2'} = \{rule\ 1, rule\ 5\}$;

Schema $\hat{S}_{1.4'}$:
$L_{1.4'} = \{publishedIn, editorOf\}$,
$R_{1.4'} = \{Professor, Student, Book, Journal, Paper, Proceeding\}$, and
$rs_{1.4'} = \{rule\ 13\}$.

We can also avoid the redundancy problem by modifying the schema $\hat{S}_{1.2}$ and building a conference to $\hat{S}_{1.1}$ directly rather than

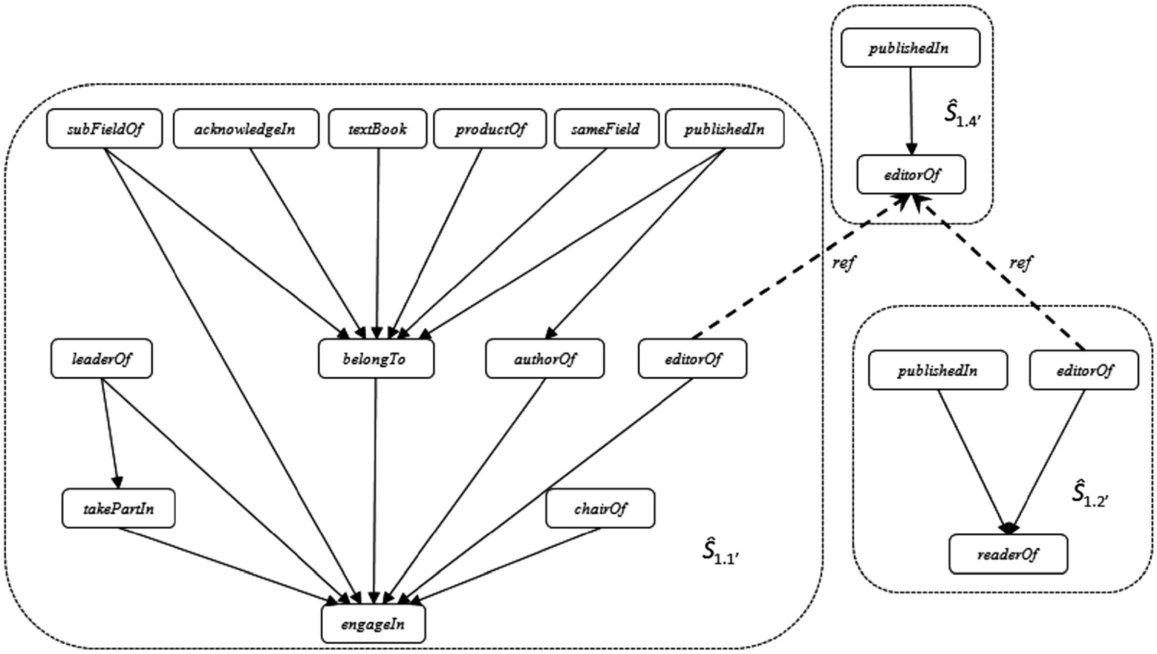

Fig. 6. Semantic link type relative nets for $\hat{S}_{1.1'}$, $\hat{S}_{1.2'}$ and $\hat{S}_{1.4'}$ with redundancy-free RC-NF2.

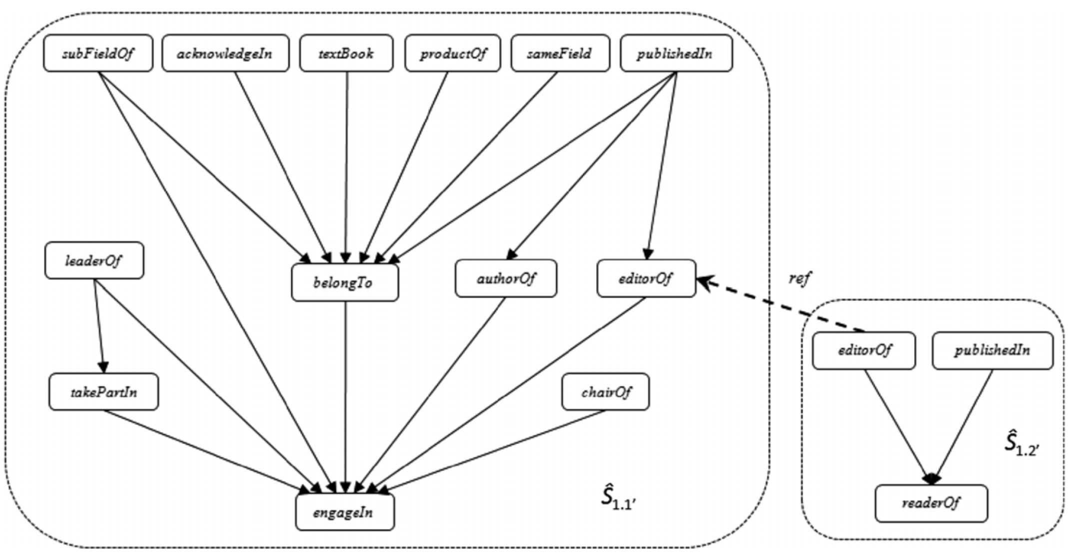

Fig. 7. Semantic link type relative nets for $\hat{S}_{1.1'}$ and $\hat{S}_{1.2'}$ with redundancy-free RC-NF2.

creating a new schema. As shown in Fig. 7, the schema $\hat{S}_{1.1}$ need not change while the schema $\hat{S}_{1.2}$ should be changed into $\hat{S}_{1.2'}$ as follows.

Schema $\hat{S}_{1.2'}$:
$R_{1.2'}$ = {*Professor*, *Student*, *Book*, *Journal*, *Paper*, *Proceeding*},
$L_{1.2'}$ = {*readerOf*, *editorOf*, *publishedIn*} with a reference relation {*ref* (*editorOf* $\in \hat{S}_{1.2'}$, *editorOf* $\in \hat{S}_{1.1}$)},
and $rs_{1.2'}$ ={*rule* 1, *rule* 5}.

Therefore, we have two choices for redundancy-free decomposition of the original schema, and both of them are loss-less.

6.5. Schema maintenance

The following are two examples to show how to maintain the SLN schema while appending some rules to or removing some rules from the rule set according to Algorithms 6 and 7 respectively.

Assume that two semantic link types are inserted into *LinkTypes*: *vistingScholarOf* $\in$ [*Professor*, *Department*] and *vistingScholarOf* $\in$ [*Professor*, *Univesity*], and two new reasoning rules are added to *rs*.

rule 34. *vistingScholarOf* $\cdot$ *partOf* $\Rightarrow$ *vistingScholarOf*
rule 35. *publishedIn* $\Rightarrow$ *sameField*.

According to Algorithm 6, we can modify the sub-schemas as follows.

(1) By *vistingScholarOf* $\cdot$ *partOf* $\Rightarrow$ *vistingScholarOf*, we get that *vistingScholarOf* $\trianglelefteq$ *partOf*. A new sub-schema $\hat{S}_{2.3}$ should be appended, where $\hat{S}_{2.3}$ involves only one semantic relative *vistingScholarOf* $\trianglelefteq$ *partOf*. And its resource types, semantic link types, and the reasoning rule set are as follows.

$R_{2.3}$ = {*Professor*, *Department*, *Univesity*},

$L_{2.3}$ = {*vistingScholarOf*, *partOf*},

$rs_{2.3}$ = {*rule*34}.

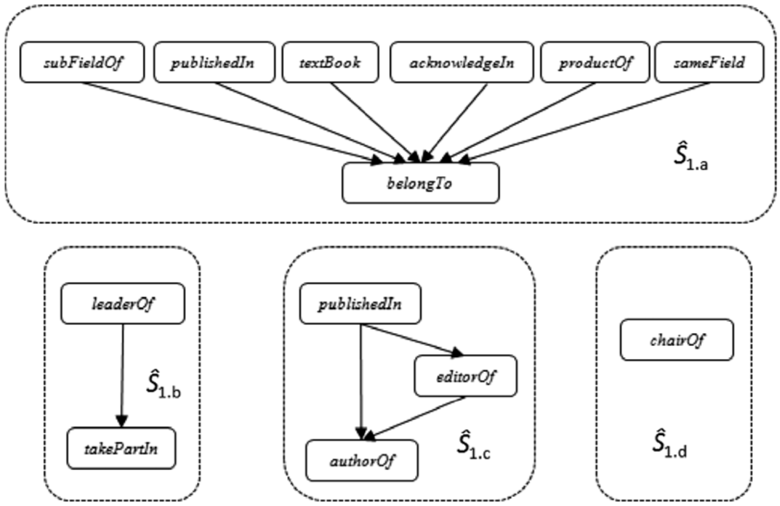

Fig. 8. The semantic link type relative nets after deleting semantic link type *engageIn*.

(2) By the rule *publishedIn* $\Rightarrow$ *sameField*, we get *sameField* $\trianglelefteq$ *publishedIn*. For the two semantic link types occur in the sub-schema $\hat{S}_{1.1}$, we only need to modify it. And, *rule* 35 should be inserted into the rule set.

Assume that the semantic link type *engageIn* is removed from *linkTypes*. According to Algorithm 7, we find the sub-shcema $\hat{S}_{1.1}$ which involve in *engageIn*. Then, we need to delete the semantic link type *engageIn* from the sub-schemas and the rules related with *engageIn*, i.e., *rule* 8, *rule* 14–18 and *rule* 21, from the rule set.

Thus, we get a new semantic relative net as shown in Fig. 8. According the new rule set, the schema $\hat{S}_{1.1}$ can be decomposed into four sub-schemas $\hat{S}_{1.a}$, $\hat{S}_{1.b}$, $\hat{S}_{1.c}$, and $\hat{S}_{1.d}$. The corresponding resource type sets, the semantic link type sets, and the reasoning rule sets can be easily obtained. Finally, we remove the sub-schema $\hat{S}_{1.d}$ for $\hat{S}_{1.d} \subseteq \hat{S}_{1.3}$. And the other sub-schemas such as $\hat{S}_{1.2}$, $\hat{S}_{1.3}$, $\hat{S}_{2.1}$, and $\hat{S}_{2.2}$ need not to be changed.

7. Discussion

As a form of knowledge representation, a traditional semantic network is a directed graph of concepts and semantic relations between concepts [12]. It does not support rule reasoning. The SLN is different from a traditional semantic network in the following aspects.

(1) SLN is a semantic data model for managing various resources.
(2) Nodes in semantic link network can be anything such as web pages, documents, software, images, concepts and even a semantic link network.
(3) Semantic links in semantic link network represent any semantic relations, even implicit relations.
(4) SLN supports rule reasoning.

The schema of relational database is for defining the structure and metadata of relational tables [1,2]. It does not support complex semantics and reasoning. The traditional network database model provides a natural way to specify and manage data, but it is relatively rigid in maintaining data [13,14]. Compared with the previous database models, the SLN model not only provides a natural way for users to create their models but also offers the reasoning ability.

Compared with the XML schema and the RDF schema, which mainly provide syntax for XML and RDF respectively, the SLN schema has the built-in relational reasoning ability.

8. Conclusions

The SLN schema regulates the semantics of semantic link networks so as to manage semantic link network instances. The proposed rule-constraint normal forms can help manage and maintain the SLN schema efficiently. Two algorithms for SLN schema extension and reduction are introduced. Reasoning algorithms for deriving more semantic relations have been proposed. The proposed SLN schema and relevant theory are important parts of the SLN model.

Acknowledgments

We thank all team members of the China Knowledge Grid Group for their help and cooperation (www.knowledgegrid.net). This work was supported by National Basic Research Program of China (2003CB317001), International Cooperation Project of Ministry of Science and Technology of China (2006DFA11970), National Science Foundation of China (60773057), and National High Technology Research and Development Program of China (2007AA12Z220).

References

[1] E.F. Codd, A relational model of data for large shared data banks, Communications of the ACM 13 (6) (1970) 377–387.
[2] J.D. Ullman, Principles of Database and Knowledge-Base Systems, Vol. I, Computer Science Press, 1988.
[3] T. Berners-Lee, J. Hendler, O. Lassila, The semantic web, Scientific American 284 (5) (2001) 34–43.
[4] I. Horrocks, P.F. Patel-Schneider, SWRL: A Semantic Web Rule Language Combining OWL and RuleML. Available at: http://www.w3.org/Submission/SWRL/.
[5] H. Zhuge, Communities and emerging semantics in semantic link network: Discovery and learning, IEEE Transactions on Knowledge and Data Engineering 21 (6) (2009) 785–799.
[6] H. Zhuge, Y. Sun, et al., Algebra model and experiment for semantic link network, International Journal of High Performance Computing and Networking 3 (4) (2005) 227–238.
[7] H. Zhuge, The Knowledge Grid, World Scientific, 2004.
[8] H. Zhuge, Autonomous semantic link networking model for the Knowledge Grid, Concurrency and Computation: Practice and Experience 7 (19) (2007) 1065–1085.
[9] H. Zhuge, et al., Modeling language and tools for the semantic link network, Concurrency and Computation: Practice and Experience 20 (7) (2008) 885–902.
[10] H. Zhuge, Y. Li, Semantic profile-based document logistics for cooperative research, Future Generation Computer Systems 20 (1) (2004) 47–60.
[11] H. Zhuge, Y. Sun, J. Zhang, The schema theory for semantic link network, in: Proceeding of the Fourth International Conference on Semantics, Knowledge and Grid, Beijing, China, 2008, pp. 189–196.

[12] M.R. Quillian, in: M. Minsky (Ed.), Semantic Memory. Semantic Information Processing, MIT press, Cambridge, Mass, 1968.
[13] C.W. Bachman, Data structure diagrams, ACM SIGMIS Database 1 (2) (1969) 4–10.
[14] C.W. Bachman, The programmer as navigator, Communications of the ACM 16 (11) (1973) 653–658.

Hai Zhuge is a professor at the Chinese Academy of Sciences' Institute of Computing Technology. He is also the chief scientist and the former director of the academy's Key Lab of Intelligent Information Processing. He is the chief scientist of the China National Semantic and Knowledge Grid Research Project and the founder of the China Knowledge Grid Research Group. His main research interests include Knowledge Grid, Resource Space Model (RSM), Semantic Link Network (SLN) and Knowledge Flow. He presented over ten keynotes in international conferences. He initiates the International Conference on Semantics, Knowledge and Grid (SKG, www.knowledgegrid.net) and was the chair and program co-chair of several international conferences like CoopIS2009. He is an associate editor of the Future Generation Computer Systems, the associate editor-in-chief of Journal of Computer Science and Technology, and on the editorial board of IEEE Intelligent Systems. He also severs as reviewer of several national foundations such as NSF of Australia, NSF of China, SFI of Ireland, and NSF of USA. He is the author of The Knowledge Grid and The Web Resource Space Model. He was the top scholar in systems and software engineering during 2000 to 2004, according to a Journal of Systems and Software assessment report. His publications appeared in CACM, Computer, IEEE TKDE, IEEE TPDS and ACM TOIT. He received 2007's Innovation Award of China Computer Federation for his fundamental theory of the Knowledge Grid. He is a senior member of IEEE and CCF.

Yunchuan Sun is a Ph.D. student at the Institute of Computing Technology, Chinese Academy of Science, Beijing, China. He also works at the School of Economic and Business Administration, Beijing Normal University, Beijing, China. His current research interest is the theory and application of semantic link network.