We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Chapter

"Set of Strings" Framework for Big Data Modeling

Igor Sheremet

Abstract

The most complicated task for big data modeling in comparison with relational approach is its variety, being a consequence of heterogeneity of sources of data, accumulated in the integrated storage space. "Set of Strings" Framework (SSF) provides unified solution of this task by representation of database as updated finite set of facts, being strings, in which structure is defined by current metadatabase, which is also an updated set of the context-free generating rules. This chapter is dedicated to SSF formal and substantial description.

Keywords: big data, "Set of Strings" framework, string databases, context-free grammars, Post systems

1. Introduction

From the three "V's," traditionally used for description of big data (volume, variety, velocity) [1–7], variety is the most difficult for theoretical modeling. The main reason of such difficulty is heterogeneity of sources of data, accumulated in the integrated storage space. By this, data items, passing to the aforementioned storage, have different structures and formats (more or less formalized texts, multimedia, hyperlinked trees of pages, etc.), which makes practically impossible application to such data of well-known relational-originated approaches to database (DB) description, manipulation, and knowledge extraction/application [8–12]. This obstacle makes hardly achieved the fourth "V" (veracity), which last time is often associated with big data [13–17], as well as with implementation of data mining over such data storages [18–21].

Such background makes necessary the alternative approach to data and knowledge modeling. This chapter contains compact consideration of the so-called "Set of Strings" Framework (SSF), developed in order to integrate on the unified theoretical basis capabilities, already used in the relational-like data representations and associated with them knowledge models, with big data immanent property—its variety.

SSF is a result of an attempt to design the aforementioned basis upon the most general representation of elementary data item, which may be stored, transported, received, processed, and visualized. Such representation is string (no matter, symbol, or bit), and SSF combines the best features of classical string-generating formal grammars, developed by Chomsky [22], with string-operating logical systems, proposed by Post [23].

The second section of this chapter is dedicated to the description of string databases (SDB), while the third, - to their interconnections with relational and non-relational DB. In the fourth section, incomplete information modeling within SSF is considered. The main content of the fifth section are the so-called word equations on context-free languages (WECFL), being key element of the SSF algorithmics.

2. "Set of strings" basic equations

The background of the SSF is representation of a database as a finite set of strings:

$$W_t = \{w_1, ..., w_{m(t)}\} \subset V^*, \tag{1}$$

where W_t means DB at the discrete time moment t and V^* is a set of all strings in the initial (terminal) alphabet V. Such databases will be called lower, if it is necessary to distinguish them from the other, the *set of strings databases*. The structure of DB elements $w_i \in W_t$, named *facts*, is determined by metadatabase (MDB), in which the current state is denoted by D_t.

Couple

$$\Theta_t = < W_t, D_t >, \tag{2}$$

is named data storage (DS). Data storage is in the correct state, if $W_t \in \boldsymbol{W}(D_t)$, where $\boldsymbol{W}(D_t)$ is the set of all correct databases, defined by the MDB.

Access message to DS is triple:

$$\omega_t = < o, c, x >, \tag{3}$$

where o is the operation, which execution is the purpose of the access (insert, delete, update, query), c is the DS component (DB, MDB) which is the objective of the access, and x is the content of the access, i.e., query body, or DB elements (facts), which are inserted or deleted. For simplicity it is supposed that the answer (reply) to the access is obtained by the user at the moment $t + 1$, next to t, and it is denoted A_{t+1}, if $c =$ DB, and A^D_{t+1}, if $c =$ MDB (both sets are finite).

A set of all possible access messages (3) is called data storage manipulation language (DSML).

SSF background is a sequential definition of four interconnected representations of DSML semantics.

Set-theoretical (S)-semantics of DSML is defined by equations on sets, which connect together input data, DB before and after access, and answer (reply) to the access.

Mathematical (M)-semantics follows aforementioned equations but is defined by some well-known and understandable mathematical constructions, being background of DSML.

Operational (O)-semantics is adequate to M-semantics but is represented by algorithms, providing execution of operations on DB.

At last, implementational (I)-semantics is also represented by algorithms, which, in general case, are much more efficient than the previous, in which the main purpose is recognition of algorithmic decidability of answer search (derivation), i.e., possibility of answer generation by finite number of steps.

Let us begin from **S-semantics** of the DSML segment, addressing DB, called lower, as usually, data manipulation language (DML). The equations, defining DML S-semantics, operate the following sets:

1. W_t (database at the moment of user's access to DB).

2. W_{t+1} (database after execution of operation, i.e., at the moment $t+1$, when answer is accepted by the user).

3. I_t (set, expressing user's awareness about some fragment of problem area at the moment of access to DB).

4. A_{t+1} (answer to the access).

Basic equations, defining DML S-semantics, are as follows:

$$W_{t+1} = W_t \cup I_t, \tag{4}$$

$$A_{t+1} = W_{t+1} - W_t, \tag{5}$$

for insertion (speaking more precisely, inclusion),

$$W_{t+1} = W_t - I_t, \tag{6}$$

$$A_{t+1} = W_t - W_{t+1}, \tag{7}$$

for deletion (exclusion),

$$W_{t+1} = W_t, \tag{8}$$

$$A_{t+1} = W_t \cap I_t, \tag{9}$$

and for query (everywhere "−" is subtraction on sets). As seen, Eqs. (4)–(9) fully correspond to the sense of basic operations on DB, inherent to any DML. In Eqs. (6) and (9), set I_t may be infinite.

Example 1. Let database, containing data items from various emergency devices, be as follows:

$$\begin{aligned} W_t = \{&\textit{AREA GREEN VALLEY IS IN NORMAL STATE AT } 15.03, \\ &\textit{AREA BLUE LAKE IS IN NORMAL STATE AT } 15.05, \\ &\textit{AREA LOWER FOREST IS SMOKED AT } 15.20\}, \end{aligned} \tag{10}$$

(due to free use of natural language in facts, it is unnecessary to comment DB content). Equation

$$W_{t+1} = W_t \cup \{\textit{AREA GREEN VALLEY IS SMOKED AT } 15.20\} \tag{11}$$

describes insertion of data item, in which the source is device, mounted at the Green Valley, which was detected as smoked since 15.20. When at this moment $t+1$ user accesses DB with query, in which the purpose is to get information about all smoked areas, the infinite set I_t may be as follows:

$$\begin{aligned} I_{t+1} = \{&\textit{AREA A IS SMOKED AT } 00.00, \ldots, \\ &\textit{AREA A IS SMOKED AT } 23.59, \ldots, \\ &\textit{AREA AA IS SMOKED AT } 00.00, \ldots, \\ &\textit{AREA AA IS SMOKED AT } 23.59, \ldots, \\ &\textit{AREA Z IS SMOKED AT } 00.00, \ldots, \\ &\textit{AREA Z IS SMOKED AT } 23.59, \ldots\}. \end{aligned} \tag{12}$$

The answer to the query is

$$A_{t+2} = W_{t+1} \cap I_{t+1} = \{AREA\ GREEN\ VALLEY\ IS\ SMOKED\ AT\ 15.20,\ AREA\ LOWER\ FOREST\ IS\ SMOKED\ AT\ 15.20\}. \quad (13)$$

In expression (12), names of all areas are strings in the alphabet $V = \{A, ..., Z, 0, ..., 9, .,\}$, so

$$I_{t+1} = \{AREA\} \bullet V^* \bullet \{SMOKED\ AT\} \bullet \{00, ..., 23\} \bullet \{.\} \bullet \{00, ..., 59\}.\blacksquare \quad (14)$$

Note that definitions (4)–(9) are not unique. For example, in the inclusion definition, elements of I_t set, having place in the DB at moment t, may be included to the answer

$$A_{t+1} = W_t \cap I_t, \quad (15)$$

as well as the answer may be defined as

$$A_{t+1} = \{FACT''\} \bullet (W_t \cap I_t) \bullet \{''ALREADY\ PRESENTS\ IN\ DATABASE\} \cup \{FACT''\} \bullet (W_{t+1} - W_t) \bullet \{''IS\ INCLUDED\ TO\ DATABASE\}. \quad (16)$$

So, according to Eq. (16), the answer to the access may be as follows:

$$A_{t+1} = \{FACT\ \text{“}AREA\ GREEN\ VALLEY\ SMOKED\ AT\ 15.20\text{”}ALREADY\ PRESENTS\ IN\ DATABASE,\ FACT\ \text{“}AREA\ LOWER\ FOREST\ IS\ SMOKED\ AT\ 15.20\text{”}\ IS\ INCLUDED\ TO\ DATABASE\}.\blacksquare \quad (17)$$

As may be seen, Eqs. (4)–(9) are based on the closed-world interpretation, which defines that the absence of the fact in the database is equivalent to its absence in the real world (problem area).

DML operations do not touch MDB; thus $D_{t+1} = D_t$.

Let us consider DML **M- and O-semantics** of DML.

The background of M-semantics of the simplest DML is the representation of the MDB D_t as a set of the context-free (CF) generating rules $\alpha \to \beta$, where α is a nonterminal symbol (“nonterminal” for short) and β is a string of both nonterminal and terminal symbols. Every nonterminal symbol, from the substantial point of view, is the name of some substring of fact, entering DB; thus β represents the structure of α. The only nonterminal symbol α_0, which does not enter any string β, is the “axiom” in the terminology of formal grammars and “fact” in the terminology of SSF. So MDB D_t unambiguously defines CF grammar

$$G_t = < V, N_t, \alpha_0, D_t >, \quad (18)$$

where

$$N_t = \{\ \alpha \mid \alpha \to \beta \in D_t\} \quad (19)$$

is the set of nonterminals (“nonterminal alphabet”) of G_t.

Database W_t is named *correct to metadatabase* D_t, if

$$W_t \subseteq L(G_t), \quad (20)$$

i.e., facts, having place in the DB, are words of the CF language $L(G_t)$. In other notation,

$$(\forall w \in W_t)\ \alpha_0 \overset{*}{\underset{G_t}{\Longrightarrow}} w, \tag{21}$$

where $\overset{*}{\underset{G_t}{\Longrightarrow}}$ is used to define that string in alphabet $V \cup N_t$ is generated (or derived) from another one.

Example 2. Let MDB D_t be as follows (nonterminal symbols are framed by metalinguistic brackets):

$<fact> \rightarrow AREA <name\ of\ area> IS <state>$
$\quad AT <time>,$
$<name\ of\ area> \rightarrow <text>,$
$<state> \rightarrow IN\ NORMAL\ STATE,$
$<state> \rightarrow SMOKED,$
$<time> \rightarrow <hours>.<minutes>,$
$<hours> \rightarrow <0\ to\ 1><0\ to\ 9>,$
$<hours> \rightarrow 2<0\ to\ 3>,$
$<0\ to\ 1> \rightarrow 0,$
$<0\ to\ 1> \rightarrow 1,$
$<0\ to\ 9> \rightarrow 0,$
...
$<0\ to\ 9> \rightarrow 9,$
$<0\ to\ 3> \rightarrow 0,$
...
$<0\ to\ 3> \rightarrow 3,$
$<minutes> \rightarrow <0\ to\ 5><0\ to\ 9>,$
$<0\ to\ 5> \rightarrow 0,$
...
$<0\ to\ 5> \rightarrow 5,$
$<text> \rightarrow <symbol>,$
$<text> \rightarrow <symbol><text>,$
$<symbol> \rightarrow A,$
...
$<symbol> \rightarrow Z,$
$<symbol> \rightarrow 0,$
...
$<symbol> \rightarrow 9,$
$<symbol> \rightarrow \text{␣}.$

Database

$$W_t = \{\ AREA\ AW\ IS\ SMOKED\ AT\ 15.10,$$
$$AREA\ E\ IS\ IN\ NORMAL\ STATE\ AT\ 23.59\ \}$$

is correct to this MDB, unlike database
$W_t = \{AREA\ AT\ NORMAL\}$.■

Proposed application of CF grammars differs from the classical, in which the main sense is the description of a set of correct sentences of some language (most frequently, programming language). This description is created by its developers or researchers, is based on syntactic categories referred as nonterminals, and is constant through all life cycle of the language (minor changes may be done by reason of language modification or deeper understanding). In the SSF case, CF generating rules are used for description of the DB element (facts) structure, so nonterminals are more semantic than syntactic objects. From the other side, MDB is updated by DS administration and is a dynamic set, in which changes provide immediate changes of DB in order to keep it in the correct state. Such changes may be defined by the following equations, similar to Eqs. (4)–(9):

$$D_{t+1} = D_t \cup I_t^D, \tag{22}$$

$$A_{t+1}^D = D_{t+1} - D_t, \tag{23}$$

$$W_{t+1} = W_t \tag{24}$$

for insertion (inclusion) of new CF rules to MDB,

$$D_{t+1} = D_t - I_t^D, \tag{25}$$

$$A_{t+1}^D = D_t - D_{t+1}, \tag{26}$$

$$W_{t+1} = W_t \cap L(G_{t+1}) \tag{27}$$

for deletion (exclusion) of CF rules, having place in MDB,

$$D_{t+1} = D_t, \tag{28}$$

$$A_{t+1}^D = D_t \cap I_t^D, \tag{29}$$

$$W_{t+1} = W_t \tag{30}$$

and for query to MDB. Here I_t^D is similar to I_t in Eqs. (4)–(9), being a set of CF rules representing knowledge of DS administration about MDB. As seen, Eqs. (22) and (23) provide extension of MDB; thus

$$L(G_t) \subseteq L(G_{t+1}), \tag{31}$$

and DB remains correct, because

$$W_t \subseteq L(G_t) \subseteq L(G_{t+1}). \tag{32}$$

In Eqs. (25) and (26), where some part (subset) of MDB may be deleted,

$$L(G_{t+1}) \subseteq L(G_t), \tag{33}$$

so some facts $w \in L(G_t)$ may become not satisfying condition (20) of DB correctness to MDB D_{t+1}, because $w \notin L(G_{t+1})$. In Eqs. (25)–(30), it is presumed, that D_t is also SDB, in which MDB defines structure of CF rules, which may be as Example 2.

Let us note that the notion of SDB correctness to MDB is from the substantial point of view weaker than the notion of *data storage correctness*, because in general case

$$\boldsymbol{W}(D_t) \subseteq 2^{L(G_t)}, \tag{34}$$

i.e., set of databases in correct storage is the subset of Boolean of $L(G_t)$, while SDB correct to MDB is such that

$$\boldsymbol{W}(D_t) = 2^{L(G_t)}, \tag{35}$$

i.e., every SDB, containing facts, being words of CF language $L(G_t)$, is correct, which is not true in the reality. DS correctness is the generalization of notion of DB integrity, deeply developed inside relational approach covering the total content of database, i.e., interconnections between its different elements. There are known various tools for integrity criteria declaration and check—first of all, functional dependencies and their multiple modifications [24–32]. Storage correctness, being SDB analog of integrity, is considered inside SSF on the basis of augmented Post systems (APS).

Let us consider now the application of the described segment of the SSF to the representation of the most frequently used data models. We shall call such application by the term "emulation."

3. Emulation of the known data models

We shall demonstrate how relational and non-relational databases may be represented on the described higher background. We shall consider relational data model as a full-scope example of databases with symmetric access (DBSA) [8–12] and a family of asymmetric access (or key-addressed) databases (KADB), which contains, among others, hypertext, page, and WWW- and Twitter-like DB [32–41].

Let us begin from the **relational model of data**.

Consider relational database (RDB), in which the scheme is $\{R_1(A_1^1, ..., A_{m1}^1), ..., R_k(A_1^k, ..., A_{mk}^k)\}$, where R_1,...,R_k are the names of relations and A_1^1, ..., A_j^i, ..., A_{mk}^k are the names of attributes. Every relation R_i at moment t is the set of tuples

$$R_i \subseteq D_j^i \times ... \times D_{mj}^i, \tag{36}$$

where D_j^i is the domain (set of possible values of attribute A_j^i).

We shall define SDB $< W_t, D_t >$, corresponding to this RDB, as follows. We shall include to the MDB D_t rules

$$\begin{gathered} <fact> \rightarrow R_1 : <A_1^1>, ..., <A_{m1}^1>, \\ ... \\ <fact> \rightarrow R_k : <A_1^k>, ..., <A_{mk}^k>, \end{gathered} \tag{37}$$

where "<" and ">" are the dividers (aforementioned metalinguistic brackets in the Backus-Naur notation) and R_i and A_j^i are the strings, being names of relations and attributes, respectively (dividers provide syntactic unambiguity).

Along with Eq. (36), MDB will include rules such that

$$D_j^i = \left\{ b | < A_j^i > \overset{*}{\Rightarrow} b \& b \in \{V - \{,\}\}^* \right\}, \tag{38}$$

i.e., these rules provide generation of sets of words in terminal alphabet V, being domains of the respective attributes. For unambiguity we assume that comma "," does not enter values of attributes $V \in D_j^i$.

By this, every tuple $\left(b_1^i, ..., b_{mi}^i\right)$ of the relation R_i is represented by fact

$$R_i : b_1^i, ..., b_{mi}^i \in W_t. \tag{39}$$

Note that representation of facts in the form (37)–(39) is not unique. As seen from Examples 1 and 2, tuples, entering relations, may be represented as any natural language phrases, described by the corresponding rules.

Let us consider now **key-addressed databases**. Their common feature is that every fact, entering KADB, includes a unique key, which is necessary to select, delete, and update this fact. These DB are associated with NoSQL family of DML [42–45], which in the last years is considered as a practical alternative to SQL-like DML [8–12, 46–48], developed since the introduction of the relational model of data.

We shall represent KADB as set

$$W_t = \{k_1 = d_1, ..., k_m = d_m\}, \tag{40}$$

where symbol "=" inside angle brackets is the divider, $k_i \in (V - \{=\})^*$ is the key, and $d_i \in V^*$ is the data, corresponding to this key (or identified by it). At every moment t, KADB must satisfy the consistency condition: KADB is consistent, if

$$(\forall t)(\forall k \in (V - \{=\})^*) | \{k =\} \cdot V^* \cap W_t | \leq 1, \tag{41}$$

i.e., set W_t would not include two or more elements with one and the same key.

Content of access to KADB must include key k, so $I_t \subseteq \{k =\} \cdot V^*$,

and for inclusion $I_t = \{k = d\}$. S-semantics of insertion to KADB is as follows:

$$W_{t+1} = \begin{cases} W_t \cup \{k = d\}, if \{k =\} \cdot V^* \cap W_t = \{\varnothing\}, \\ W_t - \{k =\} \cdot V^* \cup \{k = d\} otherwise, \end{cases} \tag{42}$$

because postulation of fact $k = d$ at moment t is equivalent to the negation of fact $k = d'$, where $d \neq d'$, which was postulated at some earlier moment $t' < t$. So in the case of KADB update is implemented by insertion, and reply to this access may be defined as follows:

$$A_{t+1} = \begin{cases} k = d, if \{k =\} \cdot V^* \cap W_t = \{\varnothing,\}, \\ (W_t \cap \{k =\} \cdot V^*) \cup \{k = d\} otherwise, \end{cases} \tag{43}$$

thus in the case of update reply contains deleted as well as included fact.

Concerning M-semantics of KADB, we may see, that every known class of such databases is identified by its own structure of keys and techniques of their extraction from the current processed fact.

The simplest approach is implemented in Twitter network, where keys, necessary for access to the descendants of the current element of the hypertext, are bounded by two dividers—"#" from the left and blank from the right.

In the Internet HTTP/WWW service, similar keys are represented as strings of symbols, visualized by other colors in comparison with the rest of the text of the current hypertext page. This is equivalent to splitting terminal alphabet V to two subsets, first including symbols of the ordinary colors and the second symbols of the "key-representing" colors. However, HTTP/WWW hypertexts are organized in a much more complicated manner. First of all, along with the displayed pages, in which the structure is described by hypertext markup language (HTML) or its various later versions (XML et al.), there is another KADB, in which elements contain keys, being the aforementioned strings of another colors, and data are, in fact, unified resource locators (URL), providing direct network access to the subordinated pages. This access is possible, because URL contains string, providing application of the domain name service (DNS) for resolving proper IP address. In fact, HTML is no more than language for the convenient representation of CF rules, which form current metadabase of the WWW KADB.

One of the simplest versions of KADB is the so-called page databases, in which elements are strings of equal length, the first string of the page being key [38, 39]. Thus

$$(\forall t)L(G_t) = V^l\left(V^l\right)^*, \tag{44}$$

where l is the length of the string (in this case divider "=" is redundant). Data may be also string $p : d$, where ":" is the divider and prefix p before the sequence of l-symbol strings defines the name of the program, called for this sequence interpretation (e.g., visualization). In general case d may be the string of bits, not only string of symbols of alphabet V.

Until now we discussed only S-semantics and start point of M-semantics, being representation of metadabase as a set of rules of CF grammar. Second such point in the SSF is the representation of databases with incomplete information.

4. Representation of databases with incomplete information and sentential data manipulation languages

Let D_t be the metadabase. Then database with incomplete information (for short, DBI or, if it is necessary to underline "set of strings" DBI, then SDBI), denoted X_t, is the finite set of the so-called incomplete facts (N-facts) being sentential forms (SF) of CF grammar G_t:

$$X_t = \{x_1, \ldots, x_m\} \subseteq SF(G_t), \tag{45}$$

where

$$SF(G_t) = \left\{x | a_0 \underset{G_t}{\overset{*}{\Rightarrow}} x\right\} \tag{46}$$

is the set of all sentential forms of grammar G_t. Obviously, $L(G_t) \subset SF(G_t)$.

Example 3. Consider MDB from Example 2 and corresponding DBI

$$\begin{aligned} X_t = \{&AREA\ LONELY\ TREES\ IS\ NORMAL\ AT\ 12.31,\\ &AREA\ LONELY\ TREES <state> AT\ 12. <minutes>,\\ &AREA <name\ of\ area> IS\ SMOKED\ AT\ 15.30\}. \end{aligned}$$

The first N-fact of the three, entering this DBI, does not contain nonterminals, so it is fact in the sense of S-semantics of DML. The second N-fact contains nonterminals <*state*> and <*minutes*>, which correspond to the uncertainty of the state of the area Lonely Trees and time moment, when this state occurs; however, the aforementioned moment enters interval from 12.01 to 12.59. The last N-fact contains information about the same area, which was detected as smoked at 15.30. ∎

Before consideration of equations, defining M-semantics of operations on DBI, we shall introduce interpretation of relation $\overset{*}{\underset{G_t}{\Longrightarrow}}$ of the mutual derivability of sentential forms of context-free grammar as relation of mutual informativity of N-facts.

Let G_t be acyclic and unambiguous CF grammar [49]. If so, $\overset{*}{\underset{G_t}{\Longrightarrow}}$ is the relation of partial order on the set $SF(G_t)$ [38, 39].

There is maximal element of the set $SF(G_t)$—it is axiom α_0 ("fact") because for every $x \in SF(G_t)$, $\alpha_0 \overset{*}{\underset{G_t}{\Longrightarrow}} x$. For every subset $X \subseteq SF(G_t)$, there exists set of its upper bounds, e.g., sentential forms ("N-facts") $y \in SF(G_t)$, such that $y \overset{*}{\underset{G_t}{\Longrightarrow}} x$ for all $x \in X$, and minimal (least) upper bound, sup X, such that for every other upper bound y from the mentioned set, the relation $y \overset{*}{\underset{G_t}{\Longrightarrow}} \sup X$ is true. For some $X \subseteq SF(G_t)$, there may exist set of lower bounds, e.g., sentential forms ("N-facts") $y \in SF(G_t)$, such that $x \overset{*}{\underset{G_t}{\Longrightarrow}} y$ for all $x \in X$, and maximal lower bound inf X such that for every other lower bound y, $\inf X \overset{*}{\underset{G_t}{\Longrightarrow}} y$ is true.

Algorithms of sup and inf generation are described in detail in [38, 39].

Example 4. For DBI from Example 3, inf X_t does not exist, but

$$\sup X_t = AREA <name\ of\ area> IS <state> AT\ 1 < 0\ to\ 9>. <minutes>.$$

At the same time,

$$\inf \{AREA\ LONELY\ TREES\ IS <state> AT\ 12.30,$$
$$AREA <name\ of\ area> IS\ SMOKED\ AT\ 12. <minutes>\}$$
$$= AREA\ LONELY\ TREES\ IS\ SMOKED\ AT\ 12.30. ∎$$

Since now we shall use interpretation of $\overset{*}{\underset{G_t}{\Longrightarrow}}$ as of the relation of the *mutual informativity* of incomplete facts. According to this interpretation, $x \overset{*}{\underset{G_t}{\Longrightarrow}} x'$ means that N-fact x' is not less informative in comparison with N-fact x (if $x \overset{+}{\underset{G_t}{\Longrightarrow}} x'$, then x' is more informative and is called *concretization of* x). (This interpretation naturally fits to A. Kolmogorov's algorithmic theory of information basic postulates, i.e., constructive objects mutual complexity [50]).

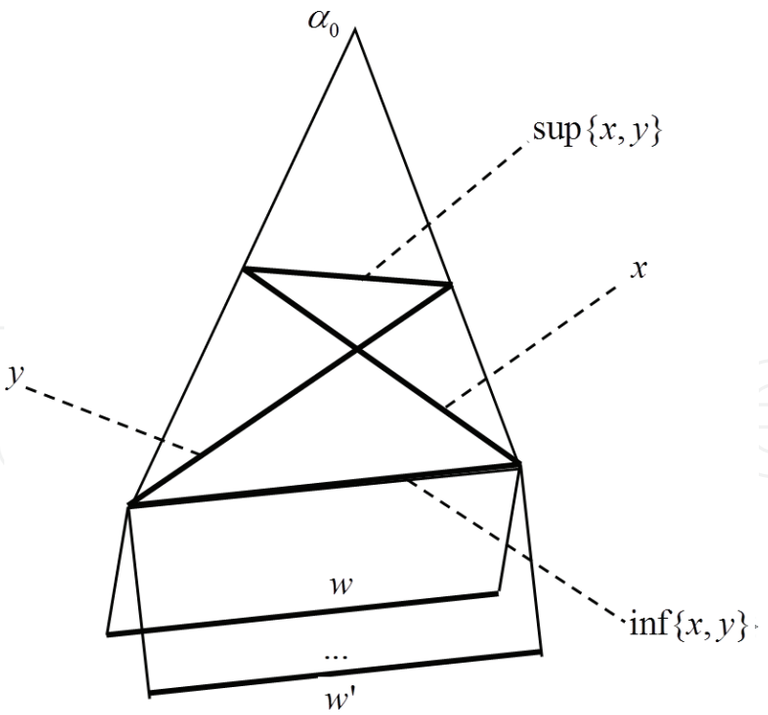


Figure 1.
Graphical illustration of interconnection between $\sup\{x,y\}$ *and* $\inf\{x,y\}$.

Graphical illustration of interconnections between $\sup\{x,y\}$ and $\inf\{x,y\}$ is in **Figure 1**. As seen, $\sup\{x,y\} \overset{*}{\Rightarrow} x$, $\sup\{x,y\} \overset{*}{\Rightarrow} y$, $x \overset{*}{\Rightarrow} \inf\{x,y\}$, $y \overset{*}{\Rightarrow} \inf\{x,y\}$, $\inf\{x,y\} \overset{*}{\Rightarrow} w$, and $\inf\{x,y\} \overset{*}{\Rightarrow} w'$, where $w \in L(G_t)$ and $w' \in L(G_t)$.

Let us consider DBI $X_t = \{x_1, ..., x_{m(t)}\} \subseteq SF(G_t)$. We shall call such DBI nonredundant (NR), if there are no two N-facts x and x' entering X_t, one of which is more informative than the other. (It is obvious that if $x' \underset{G_t}{\overset{*}{\Longrightarrow}} x$, then there is no necessity of storing x', because all information, having place in x', presents in x. So x' is a redundant N-fact, and DBI, containing such N-facts, is also redundant).

Until the contrary is declared, we shall consider only NR DBI lower. By this, when defining M-semantics of update of NR DBI, understood as inclusion of N-fact $x \in SF(G_t)$, it is reasonable to suppose, that it contains maximally informative N-facts, which only may be acquired by the system. In this case inclusion of N-fact $x \in X_t$ to DBI may be defined as follows:

$$X_{t+1} = X_t \bigcup \{x\} - \left\{ y \mid y \in X_t \& \left(x \underset{G_t}{\overset{*}{\Longrightarrow}} y \bigvee y \underset{G_t}{\overset{*}{\Longrightarrow}} x \right) \right\}. \tag{47}$$

According to this definition, N-fact x inclusion to DBI X_t causes extraction from DBI of all N-facts, which are more or less informative than x. Such logics provides maintenance of nonredundancy of the DBI. As seen, all N-facts, having place in X_t and being "compatible" with N-fact x by informativity, are eliminated from this DBI.

It is reasonable to define reply to the inclusion of N-fact x as set of N-facts, eliminated from DBI:

$$A_{t+1} = X_t - X_{t+1}. \tag{48}$$

Example 5. If N-fact $x =$ *AREA GREEN VALLEY IS SMOKED AT* 15.30 is included to DBI X_t from Example 3, then

$$X_{t+1} = \{AREA\ LONELY\ TREES\ IS\ NORMAL\ AT\ 12.31,$$
$$AREA\ LONELY\ TREES < state > AT\ 12. < minutes >,$$
$$AREA\ GREEN\ VALLEY\ IS\ SMOKED\ AT\ 15.30\},$$

because

$$AREA < name\ of\ area > SMOKED\ AT\ 15.30 \overset{*}{\underset{G_t}{\Longrightarrow}}$$
$$AREA\ GREEN\ VALLEY\ IS\ SMOKED\ AT\ 15.30.\blacksquare$$

As to M-semantics of queries, there may by two different versions, which run out of the new DBI features in comparison with DB.

The *first version* is obvious:

$$A_{t+1}^{y} = \left\{x \mid x \in X_t \& y \overset{*}{\Rightarrow} x\right\}, \tag{49}$$

where A_{t+1}^{y} is the answer to the query with content y, and all N-facts from DBI X_t, which are no less informative than x, are included to the answer.

As seen, by this we postulate query language to databases (or DB with complete information): it is the set of sentential forms of CF grammar G_t, and in the case of SF y is the content of query ω_t,

$$I_t = \left\{w | y \overset{*}{\underset{G_t}{\Rightarrow}} w \& w \in V^*\right\}, \tag{50}$$

i.e., it is the set of facts, more informative than N-fact y, having place in the query. So combining Eqs. (9) and (49), we obtain

$$A_{t+1} = W_t \cap I_t = \left\{w | y \overset{*}{\underset{G_t}{\Rightarrow}} w \& w \in W_t\right\}, \tag{51}$$

i.e., the result of access is the subset of database W_t, containing all facts, more informative than y.

The background of the *second* version is the interpretation of the query as an action, which aim is to check if there are such possible facts $w \in L(G_t)$, which are more informative than N-fact x, entering X_t, and N-fact y (query content):

$$(\exists w \in L(G_t))\ x \overset{*}{\underset{G_t}{\Longrightarrow}} w \& y \overset{*}{\underset{G_t}{\Longrightarrow}} w, \tag{52}$$

so, while x is not concretization of y, it is sensible to include x to the answer, because there may be facts $w \in L(G_t)$, which are both x and y concretizations. The set of such facts is the intersection $W_x \cap W_y$, where

$$W_x = \left\{w \mid x \overset{*}{\underset{G_t}{\Longrightarrow}} w\right\}, \tag{53}$$
$$W_y = \left\{w \mid y \overset{*}{\underset{G_t}{\Longrightarrow}} w\right\}.$$

Finite representation of the aforementioned intersection is, obviously, maximal lower bound of the set $\{x,y\}$. For this reason, answer to the query with content y may be set

$$\overline{A}^{y}_{t+1} = \{x \mid x \in X_t \ \ \& \ \exists \inf\{x,y\}\}, \tag{54}$$

and, as an alternative,

$$\overline{\overline{A}}^{y}_{t+1} = \{\inf\{x,y\} \mid x \in X_t \ \& \ \exists \inf\{x,y\}\}. \tag{55}$$

Example 6. Consider DBI X_t from Example 3 and the query with content $y = AREA <name\ of\ area> IS\ SMOKED\ AT <time>$. The purpose of this query is to get information about all areas smoked. According to Eqs. (49), (54), and (55)

$$A^{y}_{t+1} = \{AREA <name\ of\ area> IS\ SMOKED\ AT\ 14.30\},$$

$$\overline{A}^{y}_{t+1} = \{AREA\ LONELY\ TREES <state> AT\ 13. <minutes>,$$

$$AREA <name\ of\ area> IS\ SMOKED\ AT\ 14.30\},$$

$$\overline{\overline{A}}^{y}_{t+1} = \{AREA\ LONELY\ TREES\ IS\ SMOKED\ AT\ 13. <minutes>,$$

$$AREA <name\ of\ area> IS\ SMOKED\ AT\ 14.30\}.\blacksquare$$

Returning to DB, which DML S-semantics was defined by Eqs. (4)–(9), we may now write its M-semantics equations not only for query but also for insertion and deletion. Namely, if string w is the content of the insertion access, then

$$W_{t+1} = \begin{cases} W_t \cup \{w\}, if\, w \in L(G_t), \\ W_t \quad otherwise. \end{cases} \tag{56}$$

Similarly, if string y, containing terminal and nonterminal symbols of CF grammar G_t, is the content of the delete access, then

$$W_{t+1} = W_t - \left\{ w \middle| y \underset{G_t}{\overset{*}{\Rightarrow}} w \& w \in L(G_t) \right\}. \tag{57}$$

Data manipulation languages, described in this section, will be called sentential (SDML), because content of any access to DB/DBI, specified with the help of such DML, is a sentential form of CF grammar, which set of rules is current MDB.

Concerning KADB, capabilities of the sentential DML are compatible with the aforementioned NoSQL languages, which provide selection of DB elements, in which keys are specified in the queries.

Of course, it is not difficult to extend SDML by features, providing construction of more complicated selection criteria (including, e.g., number intervals) [38, 39]. But in comparison with SQL and similar relational languages, providing symmetric access to DB, SDML are rather poor. To achieve capabilities of the relationally complete query languages, it is necessary to extend SDML by features, providing comparison of values, having place in different facts. Such features are critically needed also for knowledge representation, extraction, and processing.

To achieve the formulated purpose, we shall use another tool, differing from CF grammars, namely, Post systems (PS), which also operate strings but, due to variables in their basic constructions (productions), have basic capabilities for aforementioned functions. The result of integration of the described "set of strings"

databases with PS are augmented Post systems. The intermediate layer between SDB and APS is formed by the so-called word equations on context-free languages, considered in the next section.

5. Word equations on context-free languages

Word equation is a well-known object of discrete mathematics, defined as follows [51–56].

Word equation is written as

$$s = s', \tag{58}$$

where s and s' are the so-called terms. Term is a non-empty sequence of symbols of alphabet, which we shall call terminal, presuming it is the same set V, as higher, and variables, which universum is denoted Γ. So $s \in (V \cup \Gamma)^+$, $s' \in (V \cup \Gamma)^+$. *Domain* (set of values) of every variable $\gamma \in \Gamma$, having place in any term, is V^*. Term without any variables is, obviously, word in alphabet V. At least one variable must present in WECFL or just the same in term ss' (or $s's$):

$$ss' \in (V \cup \Gamma)^+ - V^+. \tag{59}$$

Set

$$d = \{\gamma_1 \to u_1, ..., \gamma_n \to u_n\}, \tag{60}$$

where γ_1, ..., γ_n are the variables, u_1, ..., u_n are the strings in alphabet V, and $\to$ is the divider (which is not occasionally the same as higher in the generating rules $\alpha \to \beta$, entering metadabases), is called *substitution*.

Term $s[d]$ is the result of application of substitution d to term s and is defined as follows. If

$$s = \overline{u_1}\gamma_{i1}\overline{u_2}...\overline{u_m}\gamma_{im}\overline{u}_{m+1}, \tag{61}$$

where $\overline{u_i} \in V^*$ and $i = 1, ..., m+1$, , then

$$s[\delta] = \overline{u_1}\,\overline{\gamma_{i1}}\,\overline{u_2}\,...\overline{u_m}\,\overline{\gamma_{im}}\,\overline{u}_{m+1}, \tag{62}$$

where

$$\overline{\gamma_{ij}} = \begin{cases} u_{ij}, \text{if } \gamma_{ij} \to u_{ij} \in d \\ \gamma_{ij} \quad otherwise. \end{cases} \tag{63}$$

Definitions (62) and (63) cover general case, when some of the variables, entering term s, do not enter the substitution (60).

Substitution d is called *terminal substitution* to term $s \in (V \cup \Gamma)^+ - V^+$, if

$$s[d] \in V^+, \tag{64}$$

i.e., result of its application to term is word in the alphabet V. In this case, obviously,

$$\{\gamma_{i1}, ..., \gamma_{in}\} \subseteq \{\gamma_i, ..., \gamma_n\}. \tag{65}$$

Terminal substitution to terms s and s' is called *solution of word equation* (58), if

$$s[d] \equiv s\left[d'\right], \tag{66}$$

i.e., result of application of d to terms s and s' is one and the same word (here $''\equiv''$ is identity sign).

Returning to SDB and M-semantics of their DML, we may see that *set of terms may be the simplest query language to SDB*. If term

$$s = \overline{u_1}\gamma_{i1}\overline{u_2}...\overline{u_m}\gamma_{im}\overline{u}_{m+1}, \tag{67}$$

is query to DB W_t, then

$$I_t = \{\overline{u_1}u_{i1}\overline{u_2}...\overline{u_m}u_{im}\overline{u}_{m+1}|u_{i1} \in V^* \&...\&u_{im} \in V^*\}, \tag{68}$$

and

$$A_{t+1} = W_t \cap I_t = \{w|w \in W_t \&(\exists u_{i1} \in V^*)...(\exists u_{im} \in V^*)\overline{u_1}u_{i1}\overline{u_2}...\overline{u_m}u_{im}\overline{u}_{m+1} = w\}, \tag{69}$$

so Eq. (69) is the definition of M-semantics of the term's query language to SDB; as seen, $w \in A_{t+1}$, if $w \in W_t$, and word equation $s = w$ has at least one solution.

Example 7. Consider database W_t, containing three facts:
SENSOR 1 IS AT GREEN VALLEY,
SENSOR 2 IS AT BLUE LAKE,
AREA LOWER FOREST IS SMOKED.
If query $s =$ *SENSOR a*, which purpose, as seen, is to select all facts with information about sensor installation, then

$$A_{t+1} = \left\{ \begin{matrix} SENSOR & 1 & IS & AT & GREEN\ VALLEY, \\ SENSOR & 2 & IS & AT & BLUE\ LAKE \end{matrix} \right\},$$

and solution of word equations
SENSOR a = SENSOR 1 IS AT GREEN VALLEY
and
SENSOR a = SENSOR 2 IS AT BLUE LAKE
are, respectively,

$$\{a \rightarrow 1 \quad IS \quad AT \quad GREEN\ VALLEY\}$$

and

$$\{a \rightarrow 2 \quad IS \quad AT \quad BLUE\ LAKE\}. \blacksquare$$

However, the application of the term's query language to databases with incomplete information, containing sentential forms of CF grammar with scheme D_t, being DS metadabase, is not so simple and needs more sophisticated mathematical background.

Let G be CF grammar, corresponding metadabase D (lower index t for simplicity is omitted). We shall call **word equation on context-free language** $L(G)$ couple

$$<s = s', \delta>, \tag{70}$$

where the first component $s = s'$ is the word equation in the sense (58), called here *kernel*, while

$$\delta = \{\gamma_1 \to \beta_1, ..., \gamma_l \to \beta_l\}, \tag{71}$$

is the so-called suffix, which defines domains (sets of values) of variables γ_1, ..., γ_l, entering terms s and s', by means of strings β_1, ..., β_l, containing terminal and nonterminal symbols of grammar G. Kernel and suffix must satisfy the so-called sentential condition

$$\{s[\delta], s'[\delta]\} \subseteq SF(G), \tag{72}$$

i.e., strings, being the result of application of substitution δ to terms s and s', must be sentential forms of grammar G. As seen, δ is the generalization of substitution (60), so it will be called lower *SF-substitution*.

WECFL (70) may be read "$s = s'$, where δ."

Domain of variable γ_i is the set of strings in terminal alphabet V, which are generated from string β_i by application of rules of grammar G. This domain is denoted as

$$\mathrm{V}(\gamma_i, \delta) = \left\{u | \gamma_i \to \beta_i \in \delta \& \beta_i \overset{*}{\Rightarrow} u \& u \in V^*\right\} \tag{73}$$

(from here we shall use $\overset{*}{\Rightarrow}$ in the sense $\underset{G}{\overset{*}{\Rightarrow}}$).

Suffix δ defines *set of terminal substitutions* to terms s and s', denoted

$$\sum_{\delta} = \underset{\substack{u_1 \in V(\gamma_1, \delta) \\ \cdots \\ u_l \in V(\gamma_l, \delta)}}{\cup} \{\{\gamma_1 \to u_1, ..., \gamma_l \to u_l\}\}. \tag{74}$$

As it is easy to see, direct consequence of the sentential condition (72) and definition (74) is

$$\{s[d], s'[d]\} \subseteq L(G), \tag{75}$$

for every $d \in \sum_{\delta}$.

If terminal substitution d is such that

$$s[d] \equiv s\left[d'\right], \tag{76}$$

it is called *solution of WECFL* (70). Set of solutions of WECFL (70), which is infinite in general case, is denoted $D[s = s', \delta]$.

Function V may be applied to every term, so

$$\mathrm{V}(s, \delta) = \left\{s[d] | d \in \sum_{\delta}\right\} \subseteq L(G), \tag{77}$$

$$\mathrm{V}(s', \delta) = \left\{s'[d] | d \in \sum_{\delta}\right\} \subseteq L(G). \tag{78}$$

Example 8. Let metadatabase be the same as in Example 2, and WECFL is

$$< AREA\ GREEN\ VALLEY\ IS a = b\ AT\ 15.00,$$
$$\{a \to <state> AT <time>,\ b \to AREA <name\ of\ area> IS <state>\}>.$$

As seen, this equation satisfies sentential condition, because

$$s[\delta] = AREA\ \ GREEN\ \ VALLEY\ IS < state > AT < time > \in SF(G_t),$$
$$s'[\delta] = AREA < name\ of\ \ area > IS < state > AT\ \ 15.00 \in SF(G_t).$$

According to Eq. (73),

$V(a,\delta) = \{IN\ NORMAL\ STATE\ AT\} \cdot T \cup \{SMOKED\ AT\} \cdot TV(b,\delta) = \{AREA\} \cdot S \cdot \{IS\ IN\ NORMAL\ STATE\ AT\} \cup \{AREA\} \cdot S \cdot \{IS\ SMOKED\}$, where T is the set of strings, explicating time (00.00, 00.01, ..., 23.58, 23.59), while S is the set of names of the monitored areas.

Terminal substitution

$$s = a \to SMOKED\ AT\ 15.00, \quad b \to AREA\ \ GREEN\ \ VALLEY\ IS\ SMOKED$$

is the solution of the presented WECFL. ■

As seen, in general case the set of solutions of WECFL may be infinite, and the problem is to find finite representation of this set.

Let us consider two sentential forms x and x' of unambiguous and acyclic CF grammar G. Each of them defines generated (derived) from it set of strings, being words of language $L(G)$:

$$W_x = \left\{w | x \overset{*}{\Rightarrow} w \& w \in V^*\right\}, \tag{79}$$

$$W_y = \left\{w | y \overset{*}{\Rightarrow} w \& w \in V^*\right\}. \tag{80}$$

And therefore SF x and x' are finite representations of sets W_x and W_y, both being subsets of language $L(G)$. This obstacle serves as background for the following statement, representing necessary solution.

Statement 1 [51]. If

$$W = W_x \cap W_{x'} \neq \{\varnothing\}, \tag{81}$$

then there exists SF y such that

$$W = \left\{w | x \overset{*}{\Rightarrow} y \& x' \overset{*}{\Rightarrow} y \& y \overset{*}{\Rightarrow} w \& w \in V^*\right\}. \blacksquare \tag{82}$$

Verbally, non-empty intersection of sets W_x and W_y is the subset of language $L(G)$, in which words are generated from *SF y*, which itself is generated from *SF x* and x' simultaneously.

Example 9. Consider SF $s[d]$ and $s'[d]$ from Example 8. As seen, *SF*

$$y = AREA\ \ GREEN\ \ VALLEY\ IS < state > AT\ \ 15.00$$

is the finite representation of intersection $W_{s[d]} \cap W_{s'[d]}$. ■

Thus SF y from Eq. (82) is nothing else than required finite representation of the non-empty intersection (81).

This finding is a basis for constructing the set of solutions $D[s = s', \delta]$. Let us begin from the case where all variables, having place in WECFL (or, just the same, in term ss'), enter it once, i.e., there is no more than one occurrence of any variable in ss'.

Obviously, if

$$W = V(s,\delta) \cap V(s',\delta) = \{\varnothing\}, \tag{83}$$

then WECFL (70) does not have a solution, i.e.,

$$D[s = s', \delta] = \{\varnothing\}, \tag{84}$$

and if $W \neq \{\varnothing\}$, then, since $s[\delta]$ and $s'[\delta]$ are sentential forms of CF grammar G, there exists finite representation of set W, being SF y generated (derived) from $s[\delta]$ and $s'[\delta]$ simultaneously.

From this place it is clear, that finite representation of the set $D[s = s', \delta]$ is set

$$\overline{\delta} = \{\gamma_1 \to \overline{\beta}_1, ..., \gamma_l \to \overline{\beta}_l\}, \tag{85}$$

such that

$$W = \{w | s[\overline{\delta}] = s'[\overline{\delta}] = y \& y \overset{*}{\Rightarrow} w \& w \in V^*\}. \tag{86}$$

It is easy to verify that $\overline{\beta}_1$, ..., $\overline{\beta}_l$ are strings, containing terminal and nonterminal symbols, and being generated from strings β_1, ..., β_l, respectively, by $s[\delta] \Rightarrow y^*$ and $s'[y] \Rightarrow y^*$.

Set $\overline{\delta}$ will be named *unifier of WECFL* (70). In accordance with Eqs. (54) and (55), we shall consider lower so-called maximal unifiers, corresponding to

$$y = \inf\{s[\delta], s'[\delta]\}, \tag{87}$$

where y is the maximal lower bound of the considered two-element set.

Example 10. Let metadatabase be the same as in Example 2, and WECFL is

$$<AREA\ a\ IS\ SMOKED\ AT\ t\ =\ b\ AT\ 15.00,$$
$$\{a \to <name\ of\ area>,\quad t \to <time>,\quad b \to AREA <name\ of\ area> IS <state>\}>.$$

As seen,

$$s[\delta] = AREA <name\ of\ area> IS\ SMOKED\ AT <time>,$$

$$s'[\delta] = AREA <name\ of\ area> IS <state> AT\ 15.00,$$

$$y = \inf\{s[\delta], s'[\delta]\} = AREA <name\ of\ area> IS\ SMOKED\ AT\ 15.00,$$

and thus

$$\overline{\delta} = \{a \to <name\ of\ area>,\qquad t \to 15.00,$$

$$b \to AREA <name\ of\ area> IS\ SMOKED\}.\blacksquare$$

Now we may return to DBI and introduce the so-called term data manipulation language (*TDML*), being the set of the so-called augmented terms $<s, d>$, where s is the term and d is the SF-substitution. M-semantics of this language is similar to Eqs. (49), (54), and (55) and is obtained by replacement of *SF* y by couple $<s, d>$:

$$A_{t+1}^{s,d} = \left\{x | x \in X_t \& s[d] \overset{*}{\Rightarrow} x\right\}, \tag{88}$$

$$\overline{A}_{t+1}^{s,d} = \{x | x \in X_t \& \exists \inf\{s[d], x\}\}, \tag{89}$$

$$\overline{\overline{A}}_{t+1}^{s,d} = \{\inf\{s[d], x\} | x \in X_t \& \exists \inf\{s[d], x\}\}. \tag{90}$$

Moreover, from now we may use augmented terms or even their sets as N-facts. Corresponding equations, which describe M-semantics of TDML, much more useful from the practical point of view, are as follows:

$$A_{t+1}^{s,d} = \left\{ <\bar{s},\bar{d}> | <\bar{s},\bar{d}> \in X_t \& s[d] \overset{*}{\Rightarrow} \bar{s}\left[\bar{d}\right] \right\}, \tag{91}$$

$$\overline{A}_{t+1}^{s,d} = \left\{ <\bar{s},\bar{d}> | <\bar{s},\bar{d}> \in X_t \& \exists \inf \left\{ s[d], \bar{s}\left[\bar{d}\right] \right\} \right\}, \tag{92}$$

$$\overline{\overline{A}}_{t+1}^{s,d} = \left\{ D\left[s = \bar{s}, d \cup \bar{d} \right] | <\bar{s},\bar{d}> \in X_t \right\}. \tag{93}$$

As may be seen, the last definition provides the most informative reply, containing maximal unifiers of WECFL, each corresponding N-fact, entering BDI.

The concerned reader may find the detailed consideration of WECFL, DBI algorithmics (including N-facts fusion), and key theoretical issues of SDB/DBI internal organization, providing associative access to the stored data as well as their compression, in [38–40].

All the said about TDML is sufficient for consideration of already mentioned knowledge representation, called augmented Post systems, being core of the deductive capabilities of "Set of Strings" Framework. APS are described in the separate chapter of this book.

Author details

Igor Sheremet
Financial University under the Government of Russian Federation, Moscow, Russia

*Address all correspondence to: sheremet@rfbr.ru

IntechOpen

© 2019 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. CC BY

References

[1] Smolan R, Erwitt J. The Human Face of Big Data. Abingdon, UK: Marston Book Services; 2012. p. 224

[2] Roberts FS. What is Big Data and how has it changed. In: Invited Talk at International Conference on Data Intensive Systems Analysis for Geohazard Studies; July 18–21, 2016; Sochi, Russia; 2016

[3] Chen M, Mao S, Zhang Y, Leung VCM. Big Data: Related Technologies, Challenges, and Future Prospects. NY: Springer; 2014. p. 100. DOI: 10.1007/978-3-319-06245-7

[4] Mayer-Schonberger V, Cukier R. Big Data: A Revolution that Will Transform how we Live, Work, and Think. Canada: Eamon Dolan/Houghton Mifflin Harcourt; 2013. p. 242

[5] Labrinidis A, Jagadish HV. Challenges and opportunities with big data. Proceedings of the VLDB Endowment. 2012;**5**(12):2032-2033

[6] Maheshwari A. Data Analytics Made Accessible. Seattle, WA: Amazon Digital Services; 2015. p. 156

[7] Marz N, Warren J. Big Data: Principles and Best Practices of Scalable Real-time Data Systems. NY: Manning Publications; 2015. p. 328

[8] Codd EF. The Relational Model for Database Management: Version 2. Boston, MA: Addison Wesley; 1990. p. 567

[9] Date CJ. An Introduction to Database Systems. Pearson; 2003. p. 1034

[10] Sumathi S, Esakkirajan S. Fundamentals of Relational Database Management Systems. NY: Springer; 2007. p. 776

[11] Mensah K. Oracle Database Programming Using Java and Web Services. Amsterdam: Elsevier; 2006. p. 1120

[12] Vaswani V. MySQL Database Usage & Administration. NY: McGraw Hill; 2009. p. 368

[13] Pendyala V. Veracity of Big Data: Machine Learning and Other Approaches to Verifying Truthfulness. NY: Apress; 2018. p. 177

[14] McNeill C. Veracity: The Most Important "V" of Big Data. GutCheck. January 23, 2018. Available from: www.gutcheckit.com/blog/veracity-big-data

[15] Normandeau K. Beyond Volume, Variety and Velocity is the Issue of Big Data Veracity. Inside Big Data. September 12, 2003. Available from: www.insidebigdata.com/2013/09/12/beyond-volume-variety-velocity-issue-big-data-veracity

[16] Walker M. Data Veracity. Data Science Central. November 28, 2012. Available from: www.datasciencecentral.com/profiles/blogs/data-veracity

[17] Siewert SB. Big Data in the Cloud. Data Velocity, Volume, Variety, Veracity. Available from: www.ibm.com/developerworks/library/bd-bigdatacloud

[18] Tan P-N, Steinbach M, Kumar V. Introduction to Data Mining. London, UK: Pearson/Addison Wesley; 2005. p. 400

[19] Witten IH, Frank E, Hall MA. Data Mining: Practical Machine Learning Tools and Techniques. 3rd ed. Burlington, MA: Morgan Kaufmann; 2011. p. 664

[20] Data Mining. Project Report Online. December 27, 2017. Available from: www.projectreportonline.com/data-mining/

[21] Adriaans P, Zantinge D. Data Mining. London, UK: Pearson; 2005. p. 458

[22] Chomsky N. Syntactic Structures. 2nd ed. Berlin, New York: Mouton de Gruyter; 2002. p. 117

[23] Post EL. Formal reductions of the general combinatorial decision problem. American Journal of Mathematics. 1943; **65**:197-215

[24] Teeling M. What is Data Integrity? Learn How to Ensure Database Data Integrity via Checks, Tests, & Best Practices. May 14, 2012. Available from: www.veracode.com/blog/2012/05/what-is-data-integrity

[25] Haloin T, Morgan T. Information Modelling and Relational Databases. Burlington, MA: Morgan Kaufmann; 2010. p. 976

[26] Date C. Database Design and Relation Theory: Normal Forms and all that Jazz. Sebastopol, CA: O'Reilly Media, Inc.; 2012. p. 260

[27] Silberschatz A, Korth H, Sudarshan S. Database Systems Concepts. NY: McGraw Hill; 2012

[28] Garcia-Molina H, Ullman JD, Widom J. Database Systems: The Complete Book. 2nd ed. London, UK: Pearson Prentice Hall; 2009

[29] Fagin R, Vardi MY. The theory of data dependencies—A survey. In: Anshel M, Gewirtz W, editors. Mathematics of Information Processing. Proceedings of Symposia in Applied Mathematics. Vol. 34. 1986. pp. 19-71

[30] Demetrovics J, Libkin L, Muchnik IB. Functional dependencies in relational databases: A lattice point of view. Discrete Applied Mathematics. 1992;**40**(2):155-185. DOI: 10.1016/0166-218X(92)90028-9

[31] Megid YA, El-Tazi N, Fahmy A. Using functional dependencies in conversion of relational databases to graph databases. In: DEXA 2018: Database and Expert Systems Applications. Lecture Notes in Computer Science. Vol. 11030; 2018. pp. 350-357

[32] Wiil UK. Experience with hyperbase: A hypertext database supporting collaborative work. SIGMOD Record. 1993;**22**(4):19-25

[33] De Bra P, Pechenitzkiy M. Dynamic and adaptive hypertext: Generic frameworks, approaches and techniques. In: Proceedings of the 20th ACM Conference on Hypertext and Hypermedia. 2009. pp. 387-388. DOI: 10.1145/1557914.1558003

[34] Bagchi A, Lahoti G. Relating web pages to enable information-gathering tasks. In: Proceedings of the 20th ACM Conference on Hypertext and Hypermedia. 2009. pp. 109-118. DOI: 10.1145/1557914.1557935

[35] Lescovec J. Human navigation in networks. In: Proceedings of the 23th ACM Conference on Hypertext and Hypermedia. 2012. pp. 143-144. DOI: 10.1145/2309996.2310020

[36] Hall W. From hypertext to linked data: The ever evolving web. In: Proceedings of the 22th ACM Conference on Hypertext and Hypermedia. 2011. pp. 3-4. DOI: 10.1145/1995996.1995969

[37] Hara Y, Botafogo R. Hypermedia databases: a specification and formal language. In: Proceedings of Database and Expert Systems Applications DEXA'94; Athens, Greece; September 7–9, 1994. pp. 521-530. DOI: 10.1007/3-540-58435-8-218

[38] Sheremet IA. Intelligent Software Environments for Information Processing Systems. Moscow: Nauka; 1994. p. 544. (In Russian)

[39] Sheremet IA. Grammatical Codings. Hannover: EANS; 2012. p. 54

[40] Sheremet IA. Augmented Post Systems: The Mathematical Framework for Data and Knowledge Engineering in Network-Centric Environment. Berlin: EANS; 2013. p. 395

[41] Sheremet I. Data and knowledge bases with incomplete information in a "Set of Strings" framework. International Journal of Engineering and Applied Sciences. 2016;**3**(3):90-103

[42] McCreary D, Kelly A. Making Sense of NoSQL. A Guide for Managers and the Rest of us. NY: Manning Publications; 2013. p. 312

[43] Tiwari S. Professional NoSQL. Birmingham, UK: Packt Publishing; 2011. p. 384

[44] Moniruzzaman AB, Hossain SA. NoSQL Database: New Era of Databases for Big Data Analytics—Classifications, Characteristics and Comparison; 2013. arXiv: 1307.0191

[45] Kessin Z. Building Web Applications with Erlang. Sebastopol, CA: O'Reilly Media, Inc.; 2012. p. 156

[46] Oracle Berkeley DB. Available from: www.oracle.com/technetwork/database/database-technologies/berkeleydb/overview/index.html

[47] Redis. Available from: www.redis.io

[48] Cunningham L. World's Largest Database Runs on Postgres? Available from: https://it.toolbox.com/blogs/lewiscunningham/worlds-largest-database-runs-on-postgres-052808

[49] Meduna A. Formal Languages and Computation: Models and their Applications. London, UK: CRC Press; 2014. p. 315

[50] Kolmogorov AN. Three approaches to the definition of notion "quantity of information". In: The Selectas. Vol. 3 "Information Theory and Theory of Algorithms". Moscow: Nauka; 2005. (In Russian)

[51] Sheremet IA. Word Equations on Context-Free Languages. Hannover: EANS; 2011. p. 44

[52] Schultz KU, editor. Word equations and related topics. Lecture Notes in Computer Science. Vol. 572. NY: Springer Verlag; 1992. p. 256

[53] Lothaire M. Algebraic Combinatorics on Words. Cambridge, UK: Cambridge University Press; 2002. p. 504

[54] Makanin GS. The problem of solvability of equations in a free semigroup. Mathematics of the USSR-Sbornik. 1977;**32**(2):129-198

[55] Makanin GS. Equations in a free group. Mathematics of the USSR-Izvestiya. 1983;**21**(3):483-546

[56] Makanin GS. Solvability of universal and positive theories in a free group. Mathematics of the USSR-Izvestiya. 1985;**25**(1):75-83