International Journal of Machine Learning and Cybernetics (2022) 13:4013–4032
https://doi.org/10.1007/s13042-022-01637-0

ORIGINAL ARTICLE

Generalized rough and fuzzy rough automata for semantic computing

Swati Yadav[1] · S. P. Tiwari[1] · Mausam Kumari[2] · Vijay K. Yadav[2]

Received: 5 September 2021 / Accepted: 21 August 2022 / Published online: 21 September 2022

Abstract
The classical automata, fuzzy finite automata, and rough finite state automata are some formal models of computing used to perform the task of computation and are considered to be the input device. These computational models are valid only for fixed input alphabets for which they are defined and, therefore, are less user-friendly and have limited applications. The semantic computing techniques provide a way to redefine them to improve their scope and applicability. In this paper, the concept of semantically equivalent concepts and semantically related concepts in information about real-world applications datasets are used to introduce and study two new formal models of computations with semantic computing (SC), namely, a rough finite-state automaton for SC and a fuzzy finite rough automaton for SC as extensions of rough finite-state automaton and fuzzy finite-state automaton, respectively, in two different ways. The traditional rough finite-state automata can not deal with situations when external alphabet or semantically equivalent concepts are given as inputs. The proposed rough finite-state automaton for SC can handle such situations and accept such inputs and is shown to have successful real-world applications. Similarly, a fuzzy finite rough automaton corresponding to a fuzzy automaton is also failed to process input alphabet different from their input alphabet, the proposed fuzzy finite rough automaton for SC corresponding to a given fuzzy finite automaton is capable of processing semantically related input, and external input alphabet information from the dataset obtained by real-world applications and provide better user experience and applicability as compared to classical fuzzy finite rough automaton.

Keywords Semantic computing · Semantic relations · Fuzzy finite automata · Rough finite state automata · Fuzzy finite rough automata

1 Introduction

Semantic computing is a technology that maps the semantics of the user with that of content to design and operate computer content to satisfy the needs and intentions of users better and create a more meaningful user experience. SC addresses the derivation and matching of the semantics of computational content and that of naturally expressed user intentions and brings together those disciplines concerned with connecting the intentions of humans with computational content by retrieving, using and manipulating existing content according to user's goals, and by creating, rearranging and managing content that matches the author's intentions (cf., [42]). SC technologies have been classified into five classes, namely, Semantic Analysis, Semantic Integration, Semantic Services, Service Integration, Semantic Interface (cf., [17, 42] for details) and shown useful in, Education [25], Business Intelligence [26], Healthcare [45], Internet of things [59], several branches of Computer Science and Biomedical System [17], Speech and Language Processing [10], spoken document summarization [15], Theoretical Computer Science [20, 22, 61], and in many more disciplines for details see references of these cited work.

✉ S. P. Tiwari
sptiwarimaths@gmail.com

Swati Yadav
yswatimaths@gmail.com

Mausam Kumari
mausam.maths@gmail.com

Vijay K. Yadav
vkymaths@gmail.com

1 Department of Mathematics and Computing, Indian Institute of Technology (Indian School of Mines), Dhanbad 826004, India

2 Department of Mathematics, Central University of Rajasthan, NH-8, Bandarsindari, Ajmer, Rajasthan 305817, India

The concept of semantic similarity and semantic relatedness are two different research topics and play a significant role in many applications of SC. In fact, the semantic relatedness [2, 7, 14, 16, 46, 66] quantifies the likeness between two concepts like "person" and "student." The semantic relatedness tries to compute the semantic proximity between two concepts which can be related but not similar, like "car and wheel" and "coffee and cup". On the other hand, semantic similarity identifies the concepts having common characteristics. The computational methods of semantic similarity developed so far use different information, knowledge resources and approaches, e.g., historical google search patterns [13], feature-based approaches using Wikipedia [23], Wikipedia-based information [24], multiple information sources [27], contextual correlation [31, 37] and many of them have proven to be useful in some specific applications of computational intelligence. The task of SC is to extract information from databases in the sense of clarification of words and to explain the meaning of various constituents of sentences (words or phrases) or sentences themselves in a natural language via semantic relatedness or semantic similarity among constituents. In literature, commonly used word similarity datasets, are R*G* [37], M*C* [31], W*ordSim*-353 [12], M*EN* [7] and S*imLex*-999 [18]. The similarity data sets typically contain a broad range of semantic relations such as synonymy, antonymy, hypernymy, co-hypernymy, meronymy and topical relatedness (cf., [6]). Among the above-mentioned similarity data set S*imLex*-999, a standard gold resource for evaluating distributional semantic models in contrast to gold standards such as W*ordSim*-353 and M*EN* explicitly quantifies similarity rather than association or relatedness so that pairs of entities that are associated but not actually similar (Freud, psychology) have a low rating (cf., [18]). The measurements of semantic similarity based on taxonomy, features or information content of concepts, are significant in various applications like natural language processing, synonym detection [28], key sentence extraction [66] and misspelling detection and correction. In most of the research work associated with SC, the word "computing" in the phrase "semantic computing' means computational implementations of semantics reasoning (e.g., ontology reasoning, rule reasoning, semantic query, and semantic search) and is irrelevant to the formal theory of computing [22]. The methods of semantic similarities used in the literature have been classified into five categories (cf., [61]), but these methods of SC lack formal computation theory because these methods are based on experiments rather than computation. Jiang [22], first introduce formal models of computation for SC based on classical automata. The models of computing based on semantic similarity proposed by [20, 22, 61] requires more support from the theories of computation, whereas, before these work, most of the research about semantic similarity was done using experiments. It is worth mentioning here that our motive is not to get into details of SC or semantic relatedness methods of SC, but the main purpose of this paper is to use concepts of SC, equivalent concepts (which are special kind of semantic relations), and semantically related concepts of SC to propose new formal computational models which are efficient to capture incomplete or insufficient and vague information of dataset obtained from real-world applications rather define or propose new methods for SC. These proposed models are different from the formal computational models of SC proposed in [20, 22, 61] in the sense that our models handle incomplete or insufficient and vague information of datasets obtained from real-world applications.

The importance of classical automata in the theory of computation is well known. Since the notion of fuzzy sets was introduced by Zadeh [65] in 1965, for representing uncertainty, it has been extensively used in automata theory. The notion of fuzzy automata was first presented by Santos [38] in 1968, and the mathematical formulation of fuzzy automata was proposed by Wee [60]. After that, multidirectional research in the area of fuzzy automata and languages is reported in the literature. The algebraic aspects of fuzzy automata and languages have been studied in [32, 51]. The minimal realization problem of fuzzy languages has been studied algebraically in [21], by category-theoretic approach in [50, 53, 55], and in bicategory theoretic setting in [52, 54, 63] to brings closer the gap between classical automata theory and natural languages. The fuzzy automata and languages have been shown helpful in many applications like supervisory control [43], learning systems [60], heart problem deduction [8]. Both finite automata [19] and fuzzy finite automata [9, 21] are the formal model of computation with values and therefore have limited applications. To overcome this issue, Ying in [64], proposed new kind of fuzzy automata whose input may be a string of fuzzy subsets of the input alphabet, instead of a string of symbols from the input alphabet, under the name a formal model of computing with words. Recently, L. Wei et al. [61] proposed the concept of fuzzy automata under semantic similarity for computing with words, where "words" means probability distributions over the alphabet, and the proposed fuzzy automata can compute with vague and imprecise data as inputs. The author's in [61] model a fuzzy automaton under semantic similarity having successful application in weather forecasting. Yuncheng Jiang [22], proposed a different understanding of "semantic computing" from computation theory perspective, where classical finite automata and fuzzy finite automata are redefined to accept equivalent concepts and semantically related concepts.

In general, the information about real-world application datasets is erroneous, inexact, or uncertain. We can understand erroneous, inexact, or uncertain information about real-world applications datasets as follows. Given a data set, a data error is an atomic value (or a cell) that is different from its given ground truth, it may be classified as either quantitative or qualitative ones, e. g., for the animal data set, one of the denial constraints states that "if there are two captures of the same animal indicated by the same tag number, then the first capture must be marked as original", in this case, any cell that participates in at least one violation is considered as erroneous [1]. The information that Mr. *X* will reach at place *Y* about 6 O'clock is inexact. To understand dataset uncertainty, we consider the case of observational gridded climate datasets from [68], where the general term "dataset uncertainty" is used to refer to both representational and non-representational uncertainty. Representational uncertainty of a dataset is uncertainty regarding how accurately the dataset represents the phenomenon it aims to measure. Non-representational uncertainties arise because abstract properties of datasets, for example, the resolution and the baseline period in global temperature datasets, are more or less adequate for a specific purpose. These non-representational uncertainties do not arise because of the dataset's relationship with reality but because of the intended purpose to which the dataset is put [68]. In general, the uncertainty in information about real-world datasets may arise due to (i) sources that are difficult to trust, (ii) sources of data from where it comes or how it was calculated, and (iii) noise such as inaccurate posts in social media or information posted by bots, (iv) abnormalities, e.g., when two trusted resources report different values for the same thing, (v) inherent uncertainty, e.g., probabilistic information, (vi) ambiguity, i.e., unclear data, e.g., a natural language filled with a vague statement.

An adequate tool to handle insufficient and incomplete information in the analysis of the various type of datasets obtained by real-world applications is the rough set theory (RST) introduced by Pawlak [35]. Like fuzzy set theory (which models the vagueness of information in the dataset), RST is another extension of classical set theory used to enlist and model the knowledge from information contained in datasets [47]. The insufficient information in the dataset is encountered if there is an object in the dataset which have the same values for all features, but the associated outcome has a different value, while vagueness of information in the dataset is a result of an evaluation of subjective concepts like young, warm, beautiful, intelligent etc. The original definition of a rough set proposed by Pawlak was based on equivalence relation. But this confining demand for equivalence limits the application scope of RST. For application purposes, many extensions of rough sets like fuzzy rough sets, rough fuzzy sets, IT2 fuzzy rough sets, IT2 rough fuzzy sets, tolerance rough fuzzy sets, rough sets based on Galois connections, soft rough fuzzy sets, and soft fuzzy rough sets have been studied (cf., [11, 29, 30, 39, 41, 62, 67]). RST has been proved an essential method in cognitive sciences, decision making, data mining, and artificial intelligence. The main advantage of using RST is that no preliminary information about data is required. RST describes dependencies between attributes, evaluates the significance of attributes, and deals with inconsistent data. "RST grows on the assumption that, with every object of the universe of discourse, one can associate some information (data, knowledge). Objects characterized by the same information are indiscernible in view of the available information about them. Any set of all indiscernible objects is called an elementary set and forms a basic granule of knowledge about the universe. Any set of objects, being a union of some elementary sets, is referred to as crisp (precise); otherwise, a set is rough (imprecise, vague). Consequently, each rough set has boundary-line cases. Therefore, a rough set can be replaced by a pair of crisp sets, called the lower and the upper approximation. The upper approximation contains objects which possibly belong to the set, and the lower approximation consists of all objects which undoubtedly belong to the set" [44]. The difference between the upper and lower approximation is called the boundary of the set. A set is said to be definable if its boundary is empty. Equivalently, a definable set is a union of elementary sets.

A dataset is said to be inconsistent if there is an object in the dataset which have the same values for all attributes, but the associated outcome has a different value. The outcome of such a dataset can not be decided precisely due to insufficient information about dataset attributes; the objects in the dataset characterized by the same information are indiscernible because of the available information about them. In such a case, indiscernible relation defines a rough set on object class of dataset. The systems associated with such a dataset can not be modeled by any computational model, classical automata, or fuzzy finite-state automata. To overcome on such issues, Basu [5], proposed the concept of rough finite state automata (RFSA). In RFSA presented by Basu, the transition function was defined in such a way that in a given state, when an input is provided, the output next state is a rough set of the states in a certain way, and the idea was further extended to design a 'recognizer' that accept imprecise statements. After that, Basu [4], introduced and studied the concept of rough grammar and rough languages generated by them, whereas the approximation of languages in the rough set was studied in [36]. The concept of rough finite-state machine introduced by Basu [5] is further studied by Sharan, Srivastava, and Tiwari [40] to characterize rough finite state automata, by Tiwari, Sharan and Singh [49] to discuss coverings of product of rough transformation semigroup, and by Tiwari and Sharan [48] to introduce the concept of product of rough finite state machine. R. Arulprakasam et al. [3] established a correspondence between rough languages generated by rough grammar and rough languages accepted by rough finite-state automata. Tyagi and Tripathi [56, 58], introduced rough automata, rough grammar, and rough languages into a fuzzy environment. Recently, Pal et al. [34] described fuzzy rough automata corresponding to a fuzzy finite automata based on complete residuated lattices, where transition map, after given an input, returns L-fuzzy rough set of states as next state, and Swati et al. [62] introduced interval type-2 fuzzy finite rough automata having application in COVID-19 patient deduction.

In the theory of computing, computations are mainly represented by effective models like classical automata, fuzzy finite automata, and rough finite-state automata. These models are considered input devices. However, these mathematical models are defined for fixed input alphabet or state set. In the case of some applications (see an example of weather forecasting in [61]), where these computational models are applied, needs to change the states or the user wants to use any one of these automata. Still, with their input symbols/words that are different from the input symbols/words of automata considered, these automata fail to adjust according to such changes. We can't change the behavior of these automata, but from the theories of semantic similarity, there always exists a semantically similar state or input symbol, and we can use that semantic relation (cf., [57]) to extend these automata with similar state or input symbol. Thus the concept of semantic similarity facilitates the application of these automata in such applications. In 2019, Jiang [22], executed the idea of SC in formal computation theory and proposed formal models of SC, where classical automata, fuzzy finite automata, and pushdown automata have been extended for SC. Recently, in [61], a new generalization of fuzzy automata has been proposed for real-world applications (e.g., in weather forecasting, see [61]), where automata need to deal with new states that are not in the state set of automata.

The fact that mathematical models like classical automata, fuzzy finite automata, and rough finite state automata are input devices defined for fixed input alphabet or state set and observation that SC-based models of classical automata and fuzzy finite automata proposed in [20, 22, 61] can use semantically equivalent and semantically related input alphabet or state set from dataset to model real-world problems, motivate us to define new formal computing models of SC based on rough finite state automata and fuzzy finite rough automata. The advantages of these two new standard models of computations with SC can be seen as follows:

- the introduced generalized rough finite-state automata for SC can accept semantically equivalent incomplete and insufficient input information (see Example 3.1), and the external inputs from the dataset obtained by real-world applications. In contrast, the computing model rough finite state automata in [5] accept only incomplete and insufficient input information from the dataset obtained from real-world applications, and the computing model finite automata based on SC in [22] accept only semantically equivalent input information and external input from the dataset obtained by real-world applications.
- the introduced generalized fuzzy finite rough automata for SC can accept semantically related input (see Examples 4.1 and 4.2), and external input alphabet information from the dataset obtained from real-world applications in vague and incomplete environment, whereas the computing model fuzzy finite state automata accept crisp input information from the dataset obtained by real-world applications and computing model finite automata based on SC in [22] accept crisp input and semantically related input, and external input alphabet information from the dataset obtained by real-world applications.

The structure of the paper is as follows: Sect. 2 consists of those primary notions and concepts which are required in subsequent sections. In Sect. 3, for a given rough finite-state automaton, we introduce new formal models of computation called rough finite-state automata for SC through two different approaches. In Sect. 4, corresponding to a fuzzy finite automaton, we introduce fuzzy finite rough automata for SC, one under semantically related concepts and another with respect to external alphabets. These two formal models of computation can deal with the concepts of ambiguity as well as impreciseness that arise in natural languages.

2 Preliminaries

This section briefly recalls basic definitions of rough sets, fuzzy sets, and rough sets and discusses some concepts related to rough finite state automata and SC.

2.1 Semantic Relationship

Semantic relationships are basically the interrelations that exist between the meanings of words (semantic relationships at the word level), between meanings of phrases, and between meanings of sentences. This paper is mainly focused on the relationship between the meaning of words. At the word level, the semantic relationships of words like antonymy, synonymy, class inclusion, part-whole, and case relationships are studied. Antonymy and synonymy are the most general semantic relationships between words. Hyponymy or class inclusion is the semantic relationship that exists between two (or more) words in such a way that the meaning of one word includes (or contains) the meaning of other words(s). The latter word is a general term referred to as hypernyms (or super-type, super-concepts, super-categories). The former word is specific, referred to as hyponyms (or subtype, subconcepts, subcategories). The semantic relatedness [2, 7, 14, 16, 46, 66], quantifies the likeness between two concepts and tries to compute the semantic proximity between two concepts which can be related but not similar. On the other hand, semantic similarity identifies the concepts having common characteristics. Now, we recall the following definition of SC from [61]

Definition 2.1 [61] Let U be a set of concepts or symbols. The **semantic similarity** of two concepts $a, b \in U$ is defined as a function $sim : U \times U \rightarrow [0, 1]$ such that

1. sim(a, a) = 1
2. sim(a, b) = sim(b, a).

2.2 Rough set, fuzzy set and fuzzy rough set

In this subsection, we recall those concepts from theory of rough sets, fuzzy sets and fuzzy rough sets from [11, 33, 35, 65] which we need in subsequent sections.

Definition 2.2 [35] Let U be a non-empty universe and $R \subseteq U \times U$ be an equivalence relation on U. The pair (U, R) is called an **approximation space**. For any $u \in U$, $[u]_R = \{v \in U : (u, v) \in R\}$ is called an **equivalence class** or a block of u with respect to R. The family of all equivalence classes $\{[u]_R : u \in U\}$ is called the **quotient set** denoted by U/R and it defines a partition of U over R. If $u, v \in U$ belongs to the same equivalence class then they are said to be indiscernible.

Definition 2.3 [35] Let (U, R) be an approximation space and $[u]_R$ be the equivalence class of u under R. Given an arbitrary set $E \subseteq U$, E may not be described specifically in (U, R) and may be identified by a pair of **lower and upper approximations** defined as follows:

$$\underline{R}(E) = \cup\{u \in U : [u]_R \subseteq E\} = \cup\{[u]_R : [u]_R \subseteq E\}$$

$$\overline{R}(E) = \cup\{u \in U : [u]_R \cap E \neq \phi\} = \cup\{[u]_R : [u]_R \cap E \neq \phi\}.$$

The pair $(\underline{R}(E), \overline{R}(E))$ is called a **rough set in** (U, R) if $\underline{R}(E) \subseteq \overline{R}(E)$.

Remark 2.1 [40] (i) For $E \subseteq U$, lower and upper approximations of E in (U, R) can be denoted as $\underline{E}$ and $\overline{E}$, in this case, a rough set in (U, R) can be viewed as a pair $(\underline{E}, \overline{E})$. (ii) For all $u \in U$, if we denote its equivalence class $[u]_R$ in (U, R) simply by $[u]$, then the pair $([u, [u])$ is a rough set in (U, R) and that $\underline{\{u\}} = [u] = \overline{\{u\}}$.

(iii) Given an approximation space (U, R) and $E \subseteq U$, $\underline{E}$ and $\overline{E}$ are interpreted as the collection of those elements of U that are **definitely** and **possibly** belongs to E, respectively. Further, E is called definable (or exact) in (U, R) iff $\underline{E} = \overline{E}$. For any $E \subseteq U$, $\underline{E}$ and $\overline{E}$ are definable sets in (U, R).

Remark 2.2 [35] The **boundary**BnX **of** X is given by $\overline{R}(X) - \underline{R}(X)$, and thus consists of elements possibly, but not definitely, in X. A set $X \subseteq U$ is said to be **definable** in (U, R) if and only if $BnX = \phi$. In particular, a definable set is union of equivalence classes or blocks in (U, R).

Definition 2.4 [65] Let U be a universe of discourse. Then **a fuzzy subset of** U is a mapping from U into unit interval $[0, 1]$. If A is a fuzzy subset of U and $u \in U$, then $A(u)$ or $\mu_A(u)$ is the **membership degree of** u **in**A. We use $\mathcal{F}(U)$ to denote the **set of all fuzzy subsets** of U.

Suppose that $\lambda \in [0, 1]$ and A, B are fuzzy sets in U. Then we define the following operations of fuzzy sets, for each $u \in U$:

1. **scale product**: $(\lambda \cdot A)(u) = min\{\lambda, A(u)\}$.
2. **union**: $(A \cup B)(u) = max\{A(u), B(u)\}$.
3. **intersection**: $(A \cap B)(u) = min\{A(u), B(u)\}$.
4. **multiplication**: $(\lambda \times A)(u) = \lambda \times A(u)$.

Definition 2.5 [33] Let U be a non-empty universe. Then a **fuzzy relation on** U is a map $R : U \times U \rightarrow [0, 1]$. The fuzzy relation R is called

(i) **reflexive** if $R(u, u) = 1, \forall u \in U$;
(ii) **symmetric** if $R(u_1, u_2) = R(u_2, u_1), \forall u_1, u_2 \in U$; and
(iii) **transitive** if $R(u_1, u_2) \wedge R(u_2, u_3) \leq R(u_1, u_3)$, $\forall u_1, u_2, u_3 \in U$.

A fuzzy relation R on U is called **fuzzy equivalence relation on** U, if it is reflexive, symmetric and transitive.

Generalizations of rough sets to the fuzzy environment was first initiated by Dubois and Prade [11]. They considered the approximation of rough sets in fuzzy approximation spaces and approximation of fuzzy sets in crisp approximation spaces. The former one gives us fuzzy rough set.

Definition 2.6 [11] Let U be a non-empty universe and R be a fuzzy binary relation on U such that (U, R) is the fuzzy approximation space. Then for any $X \in \mathcal{F}(U)$, the **fuzzy rough set of** X is the pair $(\underline{R}(X), \overline{R}(X))$, where $\underline{R}(X)$ *and* $\overline{R}(X)$ are fuzzy sets of U with membership functions defined, respectively, as

$$\underline{R}(X)(u) = \bigwedge_{v \in U} \{(1 - R(u,v)) \vee \mu_X(v)\}$$

$$\text{and} \quad \overline{R}(X)(u) = \bigvee_{v \in U} \{R(u,v) \wedge \mu_X(v)\},$$

where $\vee$ and $\wedge$ denotes the join and meet operations, respectively.

2.3 Fuzzy finite automata

Herein, we recall the concepts of a fuzzy finite automaton and fuzzy languages from [32].

Definition 2.7 [32] A **fuzzy-finite automaton** (FFA) is a five-tuple $M = (Q, X, \delta, I, H)$, where

1. Q is a finite non-empty set of states.
2. X is a finite input alphabet.
3. $\delta : Q \times X \to \mathcal{F}(Q)$ is a mapping such that for any $q \in Q$ *and* $x \in X$, $\delta(q, x)$ is a fuzzy subset of Q.
4. $I : Q \to [0, 1]$ is the fuzzy set of initial states, and
5. $H : Q \to [0, 1]$ is the fuzzy set of final states.

Remark 2.3 [32] The transition function δ is extended to $\delta^* : Q \times X^* \to \mathcal{F}(Q)$ such that for all $q \in Q$, $\epsilon \in X^*$(empty string),

$$\delta^*(q, \epsilon) = \{1/q\}$$

$$\delta^*(q, wx)(p) = \vee\{\delta^*(q, w)(r) \wedge \delta(r, x)(p) : r \in Q\},$$

for all $w \in X^*$ *and* $x \in X$.

Definition 2.8 [32] A **fuzzy language accepted by** M, denoted by f_M is a fuzzy subset of X^*, i.e., $f_M : X^* \to [0, 1]$ such that for any $w \in X^*$,

$$f_M(w) = \vee\{I(q) \wedge \delta^*(q, w)(p) \wedge H(p) : q, p \in Q\}.$$

2.4 Rough finite state automata

Herein, we recall the concepts of rough finite state automata (rough automata) and rough languages from Basu [4, 5].

Definition 2.9 [5] A **rough finite state automaton** (RFSA) is a system $M = (Q, R, X, \delta, I, H)$, where

(i) Q is the finite non-empty set of states;
(ii) R is an equivalence relation on Q, i.e., (Q, R) is an approximation space;
(iii) X is the finite set of input symbols;
(iv) I is a definable set in (Q, R), called the initial configuration;
(v) $H \subseteq Q$ is the set of final states or accepting states;
(vi) $\delta : Q \times X \to \mathbf{A}$, where $\mathbf{A} = \{(\underline{A}, \overline{A}) : A \subseteq Q\}$, is a map called **rough transition map** of M such that for each $(q, x) \in Q \times X$,

$$\delta(q, x) = (\underline{A}, \overline{A})$$

be a rough set of Q in (Q, R) with lower approximation $\underline{\delta(q, x)}$ and upper approximation $\overline{\delta(q, x)}$.

Remark 2.4 (i) Throughout, the set of all rough set $\{(\underline{A}, \overline{A}) : A \subseteq Q\}$ in the approximation space (Q, R) is just denoted by $\mathbf{A}$.

(ii) Every finite state automaton can be viewed as an RFSA (cf., [40]).

Remark 2.5 In the above definition of RFSA, an input in a state result in a rough set of states called the lower and the upper approximation. It differs from both the concept of automaton and fuzzy automaton in the sense that input in a state in these cases results in a single state/subset of Q, or a fuzzy (sub)set of Q.

Definition 2.10 [3, 5, 40] Let $M = (Q, R, X, \delta, I, H)$ be a RFSA and $\mathbf{D}$ be the class of all definable sets in (Q, R). Then, for any $D' \in \mathbf{D}$, the **block transition map** of M is a map $\delta^D : \mathbf{D} \times X \to \mathbf{A}$ defined as follows:

$$\delta^D(D', x) = \left(\underline{\delta^D(D', x)}, \overline{\delta^D(D', x)}\right), \quad \textit{where}$$

$$\underline{\delta^D(D', x)} = \cup\{\underline{\delta(q, x)} : q \in B \subseteq D', B \in Q/R\}$$

$$\text{and} \quad \overline{\delta^D(D', x)} = \cup\{\overline{\delta(q, x)} : q \in B \subseteq D', B \in Q/R\}.$$

Throughout, or a nonempty set X, X^* denote set of all words on X, i.e., finite strings of elements of X under concatenation of strings with empty string $\epsilon \in X^*$. Now, we have the following definition.

Definition 2.11 [5, 40] The transition map δ of M can be extended to $\delta^* : Q \times X^* \to \mathbf{A}$ in the following way:

(i) $\delta^*(q, \epsilon) = \left([q], [q]\right)$, for all $q \in Q$ and empty string $\epsilon \in X^*$.
(ii) For all $q \in Q, w \in X^*$, *and* $x \in X$, $\delta^*(q, wx) = \left(\underline{\delta^*(q, wx)}, \overline{\delta^*(q, wx)}\right)$, where $\underline{\delta^*(q, wx)} = \underline{\delta^D(\underline{\delta^*(q, w)}, x)}$ and $\overline{\delta^*(q, wx)} = \delta^D(\overline{\delta^*(q, w)}, x)$

Definition 2.12 [3, 5, 40] Let $M = (Q, R, X, \delta, I, H)$ be a RFSA and $\mathbf{D}$ be the class of all definable sets in (Q, R). Then the block transition map δ^D of M can be extended to $\delta^{*^D} : \mathbf{D} \times X^* \to \mathbf{A}$ such that for all $D' \in \mathbf{D}$ *and* $w \in X^*$:

$$\delta^{*^D}(D',w) = \left(\underline{\delta^{*^D}(D',w)}, \overline{\delta^{*^D}(D',w)}\right), \ \textit{where}$$
$$\underline{\delta^{*^D}(D',w)} = \cup\{\underline{\delta^*(q,w)} : q \in B \subseteq D', B \in Q/R\} \ \text{and} \ \overline{\delta^{*^D}(D',w)} = \cup\{\overline{\delta^*(q,w)} : q \in B \subseteq D', B \in Q/R\}.$$

Definition 2.13 [40] Let $M = (Q, R, X, \delta, I, H)$ be a RFSA, then for all $D' \in \mathbf{D}$, $w \in X^*$ and $a \in X$,

$$\delta^{*^D}(D',wa) = \left(\underline{\delta^{*^D}(D',wa)}, \overline{\delta^{*^D}(D',wa)}\right), \ \textit{where}$$
$$\underline{\delta^{*^D}(D',wa)} = \underline{\delta^D(\underline{\delta^*(q,w)},a)}$$
$$\text{and} \quad \overline{\delta^{*^D}(D',wa)} = \overline{\delta^D(\overline{\delta^*(q,w)},a)}.$$

Definition 2.14 [4, 5] Let $M = (Q, R, X, \delta, I, H)$ be a RFSA. Then, **the behaviour of** M is a rough subset of X^* denoted by $\beta_M = (\underline{\beta_M}, \overline{\beta_M})$, where $\underline{\beta_M} = \{w \in X^* : \underline{\delta^*(I,w)} \cap H \neq \phi\}$ is the set of strings **definitely accepted by** M, and $\overline{\beta_M} = \{w \in X^* : \overline{\delta^*(I,w)} \cap H \neq \phi\}$ is the set of strings **possibly accepted by** M.

Like classical automata and FFA, RFSA is also a mathematical model used for computation purposes in computational theory where data is given in the form of decision table having incomplete information. The automaton was presented as a recognizer of rough sets that accepts rough regular languages [3]. Formally, after an input set is provided the rough finite state automaton permits a state to transit to a rough set of states. Obviously, once RFSA is defined its input set is also fixed just like the case of finite automata.

Jiang [22] suggested interpretation of "semantic computing" in the view of formal computational theory, when classical automata is made to read some unknown symbol having a semantic relation with the known symbol in alphabets of automata. For example, suppose a finite automata M is given with input alphabet Σ, and a transition $\delta(q_1, W) = q_2$, where $W \in \Sigma$. Also, suppose that W and U are synonyms (i.e., $W \equiv U$) or U is a subconcept of W (i.e., $U \leq W$), but $U \notin \Sigma$, so that M cannot deal with $\delta(q_1, U)$ since $U \notin \Sigma$, however it is not true due to existence of some semantic relations between W and U such as $W \equiv U$, thus we have that $\delta(q_1, U) = \delta(q_1, W) = q_2$ if $W \equiv U$. Thus, "semantic computing" can be considered as a mechanism of computing where inputs can be known or unknown symbols (cf., [22]). A new automata for SC having internal and external alphabets can be derived by extending classical automata using semantic similarity of words of automata and external words based on different ontologies. In this paper, we introduce a model of computation called rough automata for SC which deals with ambiguity and an other model of computation for SC which capture both ambiguity and impreciseness involved in natural languages.

3 Rough finite state automata as a formal model for semantic computing

In this section, we extend RFSA to RFSA for SC. First, we shall use the equivalent concepts of words or symbols to extend RFSA to RFSA for SC. Later we shall use concepts of the external alphabet introduced in [22] to define rough finite-state automata for SC with respect to the external alphabet.

3.1 Rough finite state automata for SC using equivalent concept

Herein, corresponding to a given RFSA, a RFSA for SC under equivalent concepts is introduced.

Definition 3.1 Let $M = (Q, R, X, \delta, I, H)$ be a RFSA defined in Definition 2.9. A **rough finite state automaton for SC under equivalent concepts** $(RFSA)_{SCEC}$ corresponding to M is the system $M' = (Q, R, X, Y, \delta, \gamma, I, H)$, where

(i) Q, R, I, H are same as defined for rough automata. (see Definition 2.9)
(ii) X is the internal alphabet of M.
(iii) $Y = \{y | y \in \Omega, y \notin X \ \textit{and} \ \exists \, x \in X \ \textit{such that} \ y \equiv x\}$ is the external alphabet for M, where Ω is the set of all symbols.
(iv) $X \cup Y$ is the alphabet set for M'.
(v) $\gamma : Q \times X \rightarrow \mathbf{A}$ is the internal rough transition function, i.e., transition map of M.
(vi) $\gamma : Q \times Y \rightarrow \mathbf{A}$ is the external transition function of M defined as:

$$\gamma(q,y) = (\underline{\gamma(q,y)}, \overline{\gamma(q,y)}), \ \textit{where}$$
$$\underline{\gamma(q,y)} = \underline{\delta(q,x)} \ \textit{and} \ \overline{\gamma(q,y)} = \overline{\delta(q,x)} \ \text{for some } x \in X \text{ such that } x \equiv y.$$

(vii) $\delta \cup \gamma : Q \times (X \cup Y) \rightarrow \mathbf{A}$ is the transition map of M'.

Definition 3.2 Let $M' = (Q, R, X, Y, \delta, \gamma, I, H)$ be a $RFSA_{SCEC}$ and $\mathbf{D}$ be the class of all definable sets in (Q, R). Then, **the block transition map**

$$(\delta \cup \gamma)^D : \mathbf{D} \times (X \cup Y) \rightarrow \mathbf{A}$$

of M' is defined $\forall D' \in \mathbf{D}$, $x \in X \cup Y$ as: $(\delta \cup \gamma)^D(D',x) = \left(\underline{(\delta \cup \gamma)^D(D',x)}, \overline{(\delta \cup \gamma)^D(D',x)}\right)$, where

$$\underline{(\delta \cup \gamma)^D(D',x)} = \cup\{\underline{(\delta \cup \gamma)(q,x)} : q \in B \subseteq D', B \in Q/R\}$$
$$\text{and} \quad \overline{(\delta \cup \gamma)^D(D',x)} = \cup\{\overline{(\delta \cup \gamma)(q,x)} : q \in B \subseteq D', B \in Q/R\}.$$

Definition 3.3 The rough transition map $\delta \cup \gamma$ of $(RFSA)_{SCEC}$ M' can be extended to $(\delta \cup \gamma)^* : Q \times (X \cup Y)^* \to \mathbf{A}$ as follows:

(i) $(\delta \cup \gamma)^*(q, \epsilon) = \Big([q], [q]\Big)$ for all $q \in Q$.

(ii) $(\delta \cup \gamma)^*(q, wx) = \Big(\underline{(\delta \cup \gamma)^*(q, wx)}, \overline{(\delta \cup \gamma)^*(q, wx)}\Big)$, such that $\underline{(\delta \cup \gamma)^*(q, wx)} = \underline{(\delta \cup \gamma)^D\Big(\underline{(\delta \cup \gamma)^*(q, w)}, x\Big)}$ and $\overline{(\delta \cup \gamma)^*(q, wx)} = \overline{(\delta \cup \gamma)^D\Big(\overline{(\delta \cup \gamma)^*(q, w)}, x\Big)}$, for all $q \in Q, w \in (X \cup Y)^*$ *and* $x \in X \cup Y$.

Definition 3.4 Let $M' = (Q, R, X, Y, \delta, \gamma, I, H)$ be a $RFSA_{SCEC}$. The **rough language** $L(M')$ **accepted by** M' is a pair $(\underline{L(M')}, \overline{L(M')})$, where

$$\underline{L(M')} = \{w \in (X \cup Y)^* \mid \underline{(\delta \cup \gamma)^*(I, w)} \cap H \neq \phi\}$$
$$\text{and} \quad \overline{L(M')} = \{w \in (X \cup Y)^* \mid \overline{(\delta \cup \gamma)^*(I, w)} \cap H \neq \phi\},$$

where $\underline{L(M')}$ is the **set of strings definitely accepted by** M' and $\overline{L(M')}$ is the **set of strings possibly accepted by** M'. Language accepted by M is a rough subset of $(X \cup Y)^*$, called **rough regular language**.

Now, we provide a real-life example to demonstrate how proposed RFSA for SC under equivalent concepts corresponding to a given RFSA can be used to decide the result of an interview. Suppose a RFSA M is defined to determine the results of an interview for conditional attributes and decision attribute of the dataset given in Table 1. This RFSA M can not process the conditional attribute values Y if filled by a candidate, but M_1' the proposed RFSA for SC, can do it by using a semantically equivalent relationship present between elements of X and Y.

Example 3.1 (Example 5 of Ref. [5] continued) In the data Table 1 given below from an interview, we have qualifications of candidates as conditional attributes and selection/rejection of candidate as decision attribute.

Table 1 Decision table for Example 3.1

	Diploma	Experience	Knowledge of English	Decision
x_1	MBA	Medium	No	Reject
x_2	MBA	Low	No	Reject
x_3	MCE	Low	Yes	Accept
x_4	M.Sc.	High	Yes	Accept
x_5	M.Sc.	Medium	No	Reject
x_6	M.Sc.	High	Yes	Accept
x_7	MBA	High	No	Accept
x_8	MCE	Low	Yes	Reject

Then the rough finite state automaton that accept the result of the interview is $M = (Q, R, X, \delta, I, H)$, where $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\}$ and $R = \{[q_0], [q_1], [q_2], [q_3, q_8], [q_4, q_6], [q_5], [q_7]\}$, $I = \{[q_0]\}, H = \{q_3, q_4, q_6, q_7\}$, $X = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6\}$, where σ_1 = (MBA, Medium, No), σ_2 = (MBA, Low, No), σ_3 = (MCE, Low, Yes), σ_4 = (M.Sc., High, Yes), σ_5 = (M.Sc., Medium, No), σ_6 = (MBA, High, No), and transition map δ is defined in Table 2.

Then we can conclude that

$$\underline{L(M)} = \{\sigma_4, \sigma_6\} \textit{ and } \overline{L(M)} = \{\sigma_3, \sigma_4, \sigma_6\}.$$

Now, let a user want to use the above rough finite-state automaton M with the following conditional attribute values filled by a candidate of the interview, he/she fails to do so because the new input symbols are not in the alphabet of given rough finite-state automaton M.

ρ_1 = (Master of Business Administration, Average, No),
ρ_2 = (Master of Business Administration, Poor, No),
ρ_3 = (Master of Civil Engineering, poor, Yes),
ρ_4 = (Master of Science, Great, Yes),
ρ_5 = (Master of Science, Average, No),
ρ_6 = (Master of Business Administration, Great, No).

But concept of semantic similarity suggested that

$$\sigma_1 \equiv \rho_1, \sigma_2 \equiv \rho_2, \sigma_3 \equiv \rho_3, \sigma_4 \equiv \rho_4, \sigma_5 \equiv \rho_5, \sigma_6 \equiv \rho_6.$$

Therefore, he/she may obtain $M' = (Q, R, X, Y, \delta, \gamma, I, H)$ as semantic extension of M in following manner, the notions Q, R, X, δ, I, H are same as defined for M. Let external alphabet Y of M be given by $Y = \{\rho_1, \rho_2, \rho_3, \rho_4, \rho_5, \rho_6\}$ and for the class of all definable set $\mathbf{D}$ of Q, the rough transition map $\gamma : Q \times Y \to \mathbf{A}$ of M' be defined as

Table 2 Transition table for Example 3.1

Q/X	σ_1	σ_2	σ_3	σ_4	σ_5	σ_6
q_0	$([q_1], [q_1])$	$([q_2], [q_2])$	$([q_1], [q_3, q_8] \cup [q_1])$	$([q_4, q_6], [q_4, q_6])$	$([q_5], [q_5])$	$([q_7], [q_7])$

$\gamma(I,\rho_1) = (\underline{\gamma(I,\rho_1)}, \overline{\gamma(I,\rho_1)});$
$\underline{\gamma(I,\rho_1)} = \underline{\delta(I,\sigma_1)} = \underline{\delta(q_0,\sigma_1)} = [q_1];$
$\overline{\gamma(I,\rho_1)} = \overline{\delta(I,\sigma_1)} = \overline{\delta(q_0,\sigma_1)} = [q_1].$

Similarly,

$\gamma(I,\rho_2) = \delta(I,\sigma_2) = ([q_2],[q_2]);$
$\gamma(I,\rho_3) = \delta(I,\sigma_3) = ([q_1],[q_3,q_8] \cup [q_1]);$
$\gamma(I,\rho_4) = \delta(I,\sigma_4) = ([q_4,q_6],[q_4,q_6]);$
$\gamma(I,\rho_5) = \delta(I,\sigma_5) = ([q_5],[q_5]);$
$\gamma(I,\rho_6) = \delta(I,\sigma_6) = ([q_7],[q_7]).$

Then, user may conclude that $\underline{L(M')} = \{\rho_4,\rho_6\}$ and $\overline{L(M')} = \{\rho_3,\rho_4,\rho_6\}$. Thus, rough finite state automata equipped with semantic similarity concept, process semantically equivalent concepts of inputs with same outcomes and provide a better user experience, while traditional rough finite automata M can not access the inputs from Y because members of Y are not in its original input set X.

Remark 3.1 For practical applications, the alphabet Y can be obtained through various methods by collecting information from available knowledge sources like Wikipedia and Google or by exploiting linked data and using ontologies provided by users. An efficient and less time-consuming algorithm to define Y using description logic and ontologies has been described in [22].

Now, let a user wants to use the above rough finite-state automaton M with the following conditional attribute values filled by a candidate of the interview, he/she fails to do so because the new input symbols are not in the alphabet of given rough finite-state automaton M.

Theorem 3.1 *Let $M = (Q,R,X,\delta,I,H)$ be a RFSA and $M' = (Q,R,X,Y,\delta,\gamma,I,H)$ be $RFSA_{SCEC}$ obtained by extension of M under equivalent concepts. Then M' satisfies the following properties:*

1. *For any $w \in X^*$, if $w \in \underline{L(M)}$ (resp. $\overline{L(M)}$), then $w \in \underline{L(M')}$ (resp. $\overline{L(M')}$).*
2. *For $w' \in (X \cup Y)^*$, if $w' \in \underline{L(M')}$ (resp. $\overline{L(M')}$) then $\exists$ $w \in X^*$ such that $w \in \underline{L(M)}$ (resp. $\overline{L(M)}$)*

Proof

1. Let $w \in X^* \subseteq (X \cup Y)^*$ be such that $w \in \underline{L(M)}$. Then by Definition 2.14, $\underline{\delta^*(I,w)} \cap H \neq \phi$. Now, since $w \in X^*$ we get that

   $\underline{(\delta \cup \gamma)^*}(I,w) = \underline{\delta^*(I,w)}$ *and* $\overline{(\delta \cup \gamma)^*}(I,w) = \overline{\delta^*(I,w)}$.

   . Thus $\underline{(\delta \cup \gamma)^*(I,w)} \cap H \neq \phi$ as $\underline{\delta^*(I,w)} \cap H \neq \phi$. Thus $w \in \underline{L(M')}$. Similarly, if $w \in \overline{L(M)} \implies \overline{\delta^*(I,w)} \cap H \neq \phi \implies \overline{(\delta \cup \gamma)^*}(I,w) \cap H \neq \phi \implies w \in \overline{L(M')}$.
2. Let $w' \in (X \cup Y)^*$ such that $w' \in \underline{L(M')}$ (*resp.* $\overline{L(M)}$) Then $\underline{(\delta \cup \gamma)^*(I,w')} \cap H \neq \phi$ (*resp.* $\overline{(\delta \cup \gamma)^*(I,w')} \cap H \neq \phi$). We have following cases:

   (a) If $w' \in X^*$, then

   $\underline{(\delta \cup \gamma)^*(I,w')} = \underline{\delta^*(I,w')}$ *and*
   $\overline{(\delta \cup \gamma)^*(I,w')} = \overline{\delta^*(I,w')}$.

   Then for $w' \in \underline{L(M')}$,

   $\underline{\delta^*(I,w')} \cap H \neq \phi \implies w' \in \underline{L(M)}$,

   and for $w' \in \overline{L(M')}$,

   $\overline{\delta^*(I,w')} \cap H \neq \phi \implies w' \in \overline{L(M)}$.

   (b) If $w' \notin X^*$. Let $w' = a_1a_2 \dots a_n$, where $a_i \in X \cup Y$ for $1 \le i \le n$. Consider,

$$
\begin{aligned}
&\underline{(\delta \cup \gamma)^*(I,w')} = \underline{(\delta \cup \gamma)^*(I,a_1a_2\dots a_n)} \\
&= \underline{(\delta \cup \gamma)^D\Big(\underline{(\delta \cup \gamma)^*(I,a_1a_2\dots a_{n-1})}, a_n\Big)} \\
&= \underline{(\delta \cup \gamma)^D\Big(\underline{(\delta \cup \gamma)^D\Big(\underline{(\delta \cup \gamma)^*(I,a_1a_2\dots a_{n-2})}, a_{n-1}\Big)}, a_n\Big)} \\
&\vdots \\
&= \underline{(\delta \cup \gamma)^D\Big(\underline{(\delta \cup \gamma)^D\Big(\dots \underline{(\delta \cup \gamma)^D\Big(\underline{(\delta \cup \gamma)(I,a_1)}, a_2\Big)}, \dots, a_{n-1}\Big)}, a_n\Big)}.
\end{aligned}
$$

Now, if $a_1 \in X$ then $\underline{(\delta \cup \gamma)}(I, a_1) = \underline{\delta(I, a_1)}$. If not, then $\underline{(\delta \cup \gamma)}(I, a_1) = \underline{\gamma(I, a_1)} = \underline{\delta(I, b_1)}$ for some $b_1 \in X$ such that $b_1 \equiv a_1$. Thus we will have

$$\underline{(\delta \cup \gamma)^D\left(\underline{(\delta \cup \gamma)^D\left(\dots \underline{(\delta \cup \gamma)^D\left(\underline{(\delta \cup \gamma)(I, a_1)}, a_2\right)}, \dots, a_{n-1}\right)}, a_n\right)}$$

$$= \underline{(\delta \cup \gamma)^D\left(\underline{(\delta \cup \gamma)^D\left(\dots \underline{(\delta \cup \gamma)^D\left(\underline{\delta(I, b_1)}, a_2\right)}, \dots, a_{n-1}\right)}, a_n\right)}.$$

Continuing the same process for all remaining symbols we get;

$$\underline{(\delta \cup \gamma)^D\left(\underline{(\delta \cup \gamma)^D\left(\dots \underline{(\delta \cup \gamma)^D\left(\underline{(\delta \cup \gamma)(I, a_1)}, a_2\right)}, \dots, a_{n-1}\right)}, a_n\right)}$$

$$= \underline{\delta^D\left(\underline{\delta^D\left(\dots \underline{\delta^D\left(\underline{\delta(I, b_1)}, b_2\right)}, \dots, b_{n-1}\right)}, b_n\right)}$$

$$= \underline{\delta^*(I, b_1 b_2 \dots b_n)}.$$

Let $w = b_1 b_2 \dots b_n$ and since $b_i \in X, 1 \leq i \leq n$, we have $w \in X^*$. Thus, finally we get that for $w' = a_1 a_2 \dots a_n \in (X \cup Y)^*$;

$$\underline{(\delta \cup \gamma)^*(I, w')} = \underline{\delta^*(I, w)}$$

where, $w' \equiv w$. Since $w' \in \underline{L(M')} \implies \underline{(\delta \cup \gamma)^*(I, w')} \cap H \neq \phi \implies \underline{\delta^*(I, w)} \cap H \neq \phi \implies w \in \underline{L(M)}$.

Steps for the upper approximation follow in the similar manner.

3.2 Rough finite state automata for SC with respect to external alphabet

In Definition 3.1, we have taken $X \cup Y$ as the alphabet set for RFSA for SC M' under equivalent concept and the transition map for M' is $\delta \cup \gamma$. But in some problems, users may wish to use their own alphabets Y (external alphabet, cf., [22]) only. In that case, the RFSA for SC can be formulated as in Definition 3.5.

Definition 3.5 Let $M = (Q, R, X, \delta, I, H)$ be a RFSA defined in Definition 2.9. A **rough finite state automaton with respect to external alphabet** $Y\,(RFSA)_{SCEA}$ is an eight-tuple $M' = (Q, R, X, Y, \delta, \gamma, I, H)$, where

(i) Y and γ are same as that of Definition 3.1.
(ii) Y is the alphabet for M' and γ is the transition map for M'.

M' is said to be semantic expansion of M w.r.t. external words or symbols.

Definition 3.6 The **block transition map of** M' is $\gamma^D : \mathbf{D} \times Y \to \mathbf{A}$ defined for all blocks $D' \in \mathbf{D}$, and $y \in Y$ as:

$$\gamma^D(D', y) = \left(\underline{\gamma^D(D', y)}, \overline{\gamma^D(D', y)}\right), \quad where$$

$$\underline{\gamma^D(D', y)} = \cup \{\underline{\gamma(q, y)} : q \in B \subseteq D', B \in Q/R\} \quad \text{and} \quad \overline{\gamma^D(D', y)}$$
$$= \cup \{\overline{\gamma(q, y)} : q \in B \subseteq D', B \in Q/R\}.$$

Definition 3.7 To define the notion of rough language accepted by M', the rough transition map γ of M' can be extended to $\gamma^* : Q \times Y^* \to \mathbf{A}$:

(i) $\gamma^*(q, \epsilon) = \left([q], [q]\right)$, for all $q \in Q$.
(ii) $\gamma^*(q, wy) = \left(\underline{\gamma^*(q, wy)}, \overline{\gamma^*(q, wy)}\right)$, such that

$$\underline{\gamma^*(q, wy)} = \underline{\gamma^D\left(\underline{\gamma^*(q, w)}, y\right)} \quad \text{and} \quad \overline{\gamma^*(q, wy)} = \overline{\gamma^D\left(\overline{\gamma^*(q, w)}, y\right)},$$

for all $q \in Q, w \in Y^*$ *and* $y \in Y$.

Definition 3.8 Let $M' = (Q, R, X, Y, \delta, \gamma, I, H)$ be a $RFSA_{SCEA}$. The **language accepted by** M' denoted as $L(M') = (\underline{L(M')}, \overline{L(M')})$ is a rough subset of Y^*, where

$$\underline{L(M')} = \{w \in Y^* \mid \underline{\gamma^*(I, w)} \cap H \neq \phi\}$$
$$\text{and} \quad \overline{L(M')} = \{w \in Y^* \mid \overline{\gamma^*(I, w)} \cap H \neq \phi\}.$$

The $\underline{L(M')}$ is the **set of strings definitely accepted by** M' and $\overline{L(M')}$ is the **set of strings possibly accepted by** M'.

Suppose $M = (Q, R, X, \delta, I, H)$ is a RFSA with input alphabet $X = \{x_1, x_2, \dots, x_n\}$ such that $x_1 \not\equiv x_2 \not\equiv \dots \not\equiv x_n$. Let O be the ontology provided by some user willing to apply on M using his/her own symbols or words. Then an external alphabet $Y = \{y_1, y_2, \dots, y_m\}$ generated using algorithm given in [22] is subsumed by X in the sense of semantics denoted as $Y \subseteq_s X$, i.e., for any $y_i \in Y, 1 \leq i \leq m$, $\exists\, x_j \in X, 1 \leq j \leq n$ such that $x_j \equiv y_i$ in O. Thus $|Y| \leq |X|$.

For semantic extension $M' = (Q, R, X, Y, \delta, \gamma, I, H)$ of RFSA $M = (Q, R, X, \delta, I, H)$ with respect to external alphabet, the string w **definitely (or possibly) accepted by** $RFSA_{SCEA}$ M' is said to be **definitely (or possibly) accepted by RFSA** M at semantic level and denoted as $w \in_s \underline{L(M)}$ (*or* $\overline{L(M)}$).

Theorem 3.2 *Let* $M = (Q, R, X, \delta, I, H)$ *be a RFSA and* $M' = (Q, R, X, Y, \delta, \gamma, I, H)$ *be a* $RFSA_{SCEA}$, *where* $X = \{x_1, x_2, \dots, x_n\}$ *and* $Y = \{y_1, y_2, \dots, y_m\}$. *Then*

1. *If* $Y \subseteq_s X$, *then for any* $w' \in Y^*$ *such that* $w' \in \underline{L(M')}$ (*or* $w' \in \overline{L(M')}$), *there exists* $w \in X^*$ *such that* $w \in \underline{L(M)}$ (*or* $w \in \overline{L(M)}$).
2. *If* $Y =_s X$.

(a) *for any* $w \in X^*$ *such that* $w \in \underline{L(M)}$ (*resp.* $\overline{L(M)}$), ∃ $w' \in Y^*$ *such that* $w' \in \underline{L(M')}$ (*resp.* $\overline{L(M')}$),

(b) *for any* $w' \in Y^*$ *such that* $w' \in \underline{L(M')}$ (*resp.* $\overline{L(M')}$), ∃ $w \in X^*$ *such that* $w \in \underline{L(M)}$ (*resp.* $\overline{L(M)}$).

Proof

1. Let $Y \subseteq_s X$ and $w' = y_1y_2\dots y_k$, where $y_i \in Y, 1 \le i \le k$. Then $w' \in \underline{L(M')}$ implies that $\underline{\gamma^*(I,w')} \cap H \neq \phi$. Now, by Definition 2.11

$$\begin{aligned}\underline{\gamma^*(I,w')} &= \underline{\gamma^*(I,y_1y_2\dots y_k)}\\ &= \underline{\gamma^D\left(\underline{\gamma^*(I,y_1y_2\dots y_{k-1})}, y_k\right)}\\ &= \underline{\gamma^D\left(\underline{\gamma^D\left(\underline{\gamma^*(I,y_1y_2\dots y_{k-2})}, y_{k-1}\right)}, y_k\right)}\\ &\ \ \vdots\\ &= \underline{\gamma^D\left(\underline{\gamma^D\left(\dots\underline{\gamma^D\left(\underline{\gamma(I,y_1)}, y_2\right)}, \dots, y_{k-1}\right)}, y_k\right)}\end{aligned}$$

and $w' \in \overline{L(M')}$ implies that $\overline{\gamma^*(I,w')} \cap H \neq \phi$.

$$\begin{aligned}\overline{\gamma^*(I,w')} &= \overline{\gamma^*(I,y_1y_2\dots y_k)}\\ &= \overline{\gamma^D\left(\overline{\gamma^*(I,y_1y_2\dots y_{k-1})}, y_k\right)}\\ &= \overline{\gamma^D\left(\overline{\gamma^D\left(\overline{\gamma^*(I,y_1y_2\dots y_{k-2})}, y_{k-1}\right)}, y_k\right)}\\ &\ \ \vdots\\ &= \overline{\gamma^D\left(\overline{\gamma^D\left(\dots\overline{\gamma^D\left(\overline{\gamma(I,y_1)}, y_2\right)}, \dots, y_{k-1}\right)}, y_k\right)}.\end{aligned}$$

Since $Y \subseteq_s X$, corresponding to each $y_i, 1 \le i \le k$, ∃ $x_j \in X, 1 \le j \le k$ such that $x_j \equiv y_i$. Then $\underline{\gamma(I,y_1)} = \underline{\delta(I,x_1)}$ (see Definition 3.1). Thus continuing the process for the rest symbols, we finally get a string $w = x_1x_2\dots x_k \in X^*$ such that

$$\begin{aligned}\underline{\gamma^*(I,w')} &= \underline{\gamma^D\left(\underline{\gamma^D\left(\dots\underline{\gamma^D\left(\underline{\gamma(I,y_1)}, y_2\right)}, \dots, y_{k-1}\right)}, y_k\right)}\\ &= \underline{\delta^D\left(\underline{\delta^D\left(\dots\underline{\delta^D\left(\underline{\delta(I,x_1)}, x_2\right)}, \dots, x_{k-1}\right)}, x_k\right)}\\ &= \underline{\delta^*(I,w)},\end{aligned}$$

and

$$\begin{aligned}\overline{\gamma^*(I,w')} &= \overline{\gamma^D\left(\overline{\gamma^D\left(\dots\overline{\gamma^D\left(\overline{\gamma(I,y_1)}, y_2\right)}, \dots, y_{k-1}\right)}, y_k\right)}\\ &= \overline{\delta^D\left(\overline{\delta^D\left(\dots\overline{\delta^D\left(\overline{\delta(I,x_1)}, x_2\right)}, \dots, x_{k-1}\right)}, x_k\right)}\\ &= \overline{\delta^*(I,w)}.\end{aligned}$$

Then $w' \in \underline{L(M')} \implies w \in \underline{L(M)}$ and $w' \in \overline{L(M')} \implies w' \in \overline{L(M)}$.

2. If $Y =_s X$.

(a) Let $w = x_1x_2\dots x_l; x_i \in X, 1 \le i \le l$ be any string in X^* such that $w \in \overline{L(M)}$. Now,

$$\begin{aligned}\overline{\delta^*(I,w)} &= \overline{\delta^*(I,x_1x_2\dots x_l)}\\ &= \overline{\delta^D\left(\overline{\delta^*(I,x_1x_2\dots x_{l-1})}, x_l\right)}\\ &= \overline{\delta^D\left(\overline{\delta^D\left(\overline{\delta^*(I,x_1x_2\dots x_{l-2})}, x_{l-1}\right)}, x_l\right)}\\ &\ \ \vdots\\ &= \overline{\delta^D\left(\overline{\delta^D\left(\dots\overline{\delta^D\left(\overline{\delta(I,x_1)}, x_2\right)}, \dots, x_{l-1}\right)}, x_l\right)}.\end{aligned}$$

Since $Y =_s X$, corresponding to each $x_i, 1 \le i \le l$, ∃ $y_i \in Y$, $1 \le i \le l$ such that $x_i \equiv y_i$. Then $\overline{\delta(I,x_1)} = \overline{\gamma(I,y_1)}$ (see definition of γ in Definition 3.1). Thus continuing in this manner, we shall get a string $w' = y_1y_2\dots y_l \in Y^*$ such that

$$\begin{aligned}\overline{\delta^*(I,w)} &= \overline{\delta^D\left(\overline{\delta^D\left(\dots\overline{\delta^D\left(\overline{\delta(I,x_1)}, x_2\right)}, \dots, x_{l-1}\right)}, x_l\right)}\\ &= \overline{\gamma^D\left(\overline{\gamma^D\left(\dots\overline{\gamma^D\left(\overline{\gamma(I,y_1)}, y_2\right)}, \dots, y_{l-1}\right)}, y_l\right)}\\ &= \overline{\gamma^*(I,w')}.\end{aligned}$$

Similarly, $\underline{\delta^*(I,w)} = \underline{\gamma^*(I,w')}$. Now,

$$w \in \overline{L(M)} \implies \overline{\delta^*(I,w)} \cap H \neq \phi \implies \overline{\gamma^*(I,w')} \cap H \neq \phi.$$

Thus, $w' \in \overline{L(M')}$, and in same manner, $w \in \underline{L(M)} \implies w' \in \underline{L(M')}$.

(b) Since $Y =_s X$, this implies $Y \subseteq_s X$ and we have proved in part 1 of this theorem that for any $w' \in Y^*$ such that $w' \in \underline{L(M')}$ (*resp.* $\overline{L(M')}$) there exists $w \in X^*$ such that $w \in \underline{L(M)}$ (*resp.* $\overline{L(M)}$).

4 Fuzzy finite rough automata as a model for semantic computing

In previous section, we have extended the computational model RFSA to RFSA for SC under equivalent concepts which are special kind of semantic relations. In this section, we first define fuzzy finite rough automaton corresponding to a fuzzy finite automaton taking the membership values in unit interval [0, 1], which we further extend for models of SC through two different approaches.

4.1 Fuzzy finite rough automaton corresponding to a fuzzy finite automaton

Definition 4.1 Let $M = (Q, X, \delta, I, H)$ be a FFA and let R be a fuzzy relation on Q such that (Q, R) being a fuzzy approximation space. Then the **fuzzy finite rough automaton (FFRA) corresponding to** M is a six-tuple $M_1 = (Q, R, X, \delta_1, I_1, H_1)$, where

(i) Q and X are same as in the case of FFA M of Definition 2.7.

(ii) $I_1 \in [0,1]^Q \times [0,1]^Q$ is a fuzzy rough set of initial states, i.e., $I_1 = (\underline{I_1}, \overline{I_1})$, where

$$\underline{I_1}(q) = \underline{R}(I)(q), \quad \overline{I_1}(q) = \overline{R}(I)(q), \quad \forall q \in Q.$$

(iii) $H_1 \in [0,1]^Q \times [0,1]^Q$ is a fuzzy rough set of final states, i.e., $H_1 = (\underline{H_1}, \overline{H_1})$, where

$$\underline{H_1}(q) = \underline{R}(H)(q), \quad \overline{H_1}(q) = \overline{R}(H)(q), \quad \forall q \in Q.$$

(iv) $\delta_1 : Q \times X \to [0,1]^Q \times [0,1]^Q$ is the fuzzy transition map such that for each $q \in Q$ and $x \in X$,

$$\delta_1(q, x) = (\underline{\delta_1(q, x)}, \overline{\delta_1(q, x)})$$

is a fuzzy rough set of states, where

$$\underline{\delta_1(q, x)}(p) = \underline{R}(\delta(q, x))(p) \quad \text{and}$$
$$\overline{\delta_1(q, x)}(p) = \overline{R}(\delta(q, x))(p)$$

Remark 4.1 The fuzzy rough transition δ_1 can naturally be extended to

$$\delta_1{}^* : Q \times X^* \to [0,1]^Q \times [0,1]^Q$$

and defined as

$$\delta_1^*(q, \epsilon) = \left(1/q, 1/q\right) \qquad \forall q \in Q \tag{1}$$

and $\delta_1^*(q, wa) = \left(\underline{\delta_1^*(q, wa)}, \overline{\delta_1^*(q, wa)}\right)$, where

$$\underline{\delta_1^*(q, wa)}(p) = \vee\{\underline{\delta_1^*(q, w)}(r) \wedge \underline{\delta_1(r, a)}(p) : r \in Q\}, \text{ and} \tag{2}$$

$$\overline{\delta_1^*(q, wa)}(p) = \vee\{\overline{\delta_1^*(q, w)}(r) \wedge \overline{\delta_1(r, a)}(p) : r \in Q\}, \tag{3}$$

for all $q \in Q, p \in Q, w \in X^*$ *and* $a \in X$.

Definition 4.2 Let $M = (Q, X, \delta, I, H)$ be a FFA and $M_1 = (Q, R, X, \delta_1, I_1, H_1)$ be a FFRA corresponding to M and $w \in X^*$, then **the degree to which** w **is accepted definitely by** M_1 is:

$$d_{\underline{M_1}}(w) = \vee\{\underline{I_1}(q) \wedge \underline{\delta_1^*(q, w)}(p) \wedge \underline{H_1}(p) : q, p \in Q\}$$

and **the degree to which** w **is accepted possibly by** M_1 is given by

$$d_{\overline{M_1}}(w) = \vee\{\overline{I_1}(q) \wedge \overline{\delta_1^*(q, w)}(p) \wedge \overline{H_1}(p) : q, p \in Q\}.$$

Definition 4.3 The **fuzzy rough language accepted by** M_1 is $L(M_1) = (\underline{L(M_1)}, \overline{L(M_1)})$, where $\underline{L(M_1)}$ and $\overline{L(M_1)}$ are fuzzy sets of X^* and defined as

$$\underline{L(M_1)} = \{(w, d_{\underline{M_1}}(w)) : w \in X^*\}$$
$$\text{and} \quad \overline{L(M_1)} = \{(w, d_{\overline{M_1}}(w)) : w \in X^*\}.$$

4.2 Fuzzy finite rough automata for SC under semantically related concepts

Fuzzy finite automata for SC under semantically related concepts is defined in Jiang [22]. Herein, we introduce and study fuzzy finite rough automata for SC under semantically related concepts. Now, we begin with following definition.

Definition 4.4 Let $M_1 = (Q, R, X, \delta_1, I_1, H_1)$ be a (FFRA) corresponding to FFA $M = (Q, X, \delta, I, H)$. A **fuzzy finite rough automaton for SC under semantically related concepts** $(FFRA)_{SCRC}$ is a system $M_1' = (Q, R, X, Y, \delta_1, \gamma_1, I_1, H_1)$, where

(i) Q, R, I_1 and H_1 are same as defined for FFRA in Definition 4.1

(ii) X is the internal alphabet of M_1'.

(iii) $Y \subseteq \Omega$ is the external alphabet of M_1', where Ω is the set of all words or symbols.

(iv) $X \cup Y$ is the alphabet for M_1'.

(v) δ_1 is the internal transition map for M_1'.

(vi) $\gamma_1 : Q \times Y \to [0,1]^Q \times [0,1]^Q$ is the external transition map for M' defined as:

$$\gamma_1(q,y) = (\underline{\gamma_1(q,y)}, \overline{\gamma_1(q,y)})$$
$$\underline{\gamma_1(q,y)} = \bigcup_{x\in X}\{sim(y,x)\times \underline{\delta_1(q,x)}\}$$
$$\overline{\gamma_1(q,y)} = \bigcup_{x\in X}\{sim(y,x)\times \overline{\delta_1(q,x)}\},$$

where the terms

$$(sim(y,x)\times \underline{\delta_1(q,x)})(p) = sim(y,x)\times \underline{\delta_1(q,x)}(p),$$
$$(sim(y,x)\times \overline{\delta_1(q,x)})(p) = sim(y,x)\times \overline{\delta_1(q,x)}(p), \forall p \in Q,$$

denote the multiplication of parameter $sim(y,x)$ with fuzzy sets $\underline{\delta_1(q,x)}$ and $\overline{\delta_1(q,x)}$, respectively.

(vii) $\delta_1 \cup \gamma_1$ is the transition map for M'_1.

Remark 4.2 The use of semantic similarity to define external transition map γ_1 in Definition 4.4 may have different approaches as suggested in Remark 1 of [22].

Remark 4.3 The transition map $\delta_1 \cup \gamma_1$ of $(FFRA)_{SCRC}$ M'_1 can further be extended to $(\delta_1\cup\gamma_1)^* : Q\times(X\cup Y)^* \to [0,1]^Q \times [0,1]^Q$ as follows:

$$(\delta_1\cup\gamma_1)^*(q,\epsilon) = \left(1/q, 1/q\right) \quad \forall q \in Q$$
$$(\delta_1\cup\gamma_1)^*(q,wa) = \left(\underline{(\delta_1\cup\gamma_1)^*(q,wa)}, \overline{(\delta_1\cup\gamma_1)^*(q,wa)}\right), \quad \text{where}$$

$$\underline{(\delta_1\cup\gamma_1)^*(q,wa)}(p) = \vee\{\underline{(\delta_1\cup\gamma_1)^*(q,w)}(r) \wedge \underline{(\delta_1\cup\gamma_1)(r,a)}(p) : r\in Q\} \tag{4}$$

$$\overline{(\delta_1\cup\gamma_1)^*(q,wa)}(p) = \vee\{\overline{(\delta_1\cup\gamma_1)^*(q,w)}(r) \wedge \overline{(\delta_1\cup\gamma_1)(r,a)}(p) : r\in Q\} \tag{5}$$

for all $q\in Q, p\in Q, w\in (X\cup Y)^*$ *and* $a\in(X\cup Y)$.

Definition 4.5 Let $M'_1 = (Q,R,X,Y,\delta_1,\gamma_1,I_1,H_1)$ be a $(FFRA)_{SCRC}$ and $w\in(X\cup Y)^*$. Then **the degree to which** w **is accepted definitely by** M'_1 is defined as

$$d_{\underline{M'_1}}(w) = \vee\{\underline{I_1}(q)\wedge \underline{(\delta_1\cup\gamma_1)^*(q,w)}(p)\wedge \underline{H_1}(p) : q,p\in Q\}$$

and **the degree to which** w **is accepted possibly by** M'_1 is defined as

$$d_{\overline{M'_1}}(w) = \vee\{\overline{I_1}(q)\wedge \overline{(\delta_1\cup\gamma_1)^*(q,w)}(p)\wedge \overline{H_1}(p) : q,p\in Q\}.$$

Definition 4.6 The **language accepted by** $(FFRA)_{SCRC}$ M'_1 is a fuzzy rough subset of $(X\cup Y)^*$ denoted by $L(M'_1) = (\underline{L(M'_1)}, \overline{L(M'_1)})$, where $\underline{L(M'_1)}$ and $\overline{L(M'_1)}$ are fuzzy sets of $(X\cup Y)^*$ and defined as

$$\underline{L(M'_1)} = \{(w, d_{\underline{M'_1}}(w)) : w\in(X\cup Y)^*\}$$
$$\overline{L(M'_1)} = \{(w, d_{\overline{M'_1}}(w)) : w\in(X\cup Y)^*\}.$$

Following the notion of semantically related concepts introduced in Jiang [22], we now demonstrate how fuzzy finite rough automata for SC under semantically related concepts M'_1, corresponding to a fuzzy finite automaton M defined in Definition 4.4 can be designed to process an external alphabet Y which is not the part of original alphabet X of M, but members of Y are semantically related with elements of original input alphabet X of M.

Example 4.1 Consider a FFA $M=(Q,X,\delta,I,H)$, where $Q=\{q_0,q_1,q_2\}$, $X=\{a_1,a_2\}$, where a_1=Artificial Intelligence a_2=Autoimmune Disease and

$$I=\frac{0.5}{q_1}, \quad H=\frac{0.8}{q_2}$$

be the fuzzy sets of initial states and final states, respectively. The transition map for M is defined in the following diagram (Fig. 1):

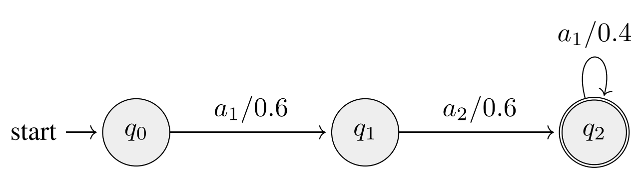

Fig. 1 Transition diagram for M for Example 4.1

The user may formulate FFRA M_1 corresponding to M as follows.

Let R be a fuzzy relation defined on Q given by the Table 3. Then the FFRA corresponding to M is $M_1=(Q,R,X,\delta_1,I_1,H_1)$, where Q and X are same as defined

Table 3 Fuzzy relation R for FFRA M_1 Example 4.1

	q_0	q_1	q_2
q_0	1	0.2	0.4
q_1	0.3	1	0
q_2	0	0.7	1

Table 4 Transition table for M_1 Example 4.1

Q	$\delta_1(q,a_1)$	$\delta_1(q,a_2)$
q_0	$\left(\frac{0.6}{q_1}, \frac{0.2}{q_0}+\frac{0.6}{q_1}+\frac{0.6}{q_2}\right)$	(ϕ,ϕ)
q_1	(ϕ,ϕ)	$\left(\frac{0.3}{q_2}, \frac{0.4}{q_0}+\frac{0.6}{q_2}\right)$
q_2	$\left(\frac{0.3}{q_2}, \frac{0.4}{q_0}+\frac{0.4}{q_2}\right)$	(ϕ,ϕ)

for M and I_1 being the fuzzy rough set of initial states such that:

$$\underline{I_1}(q_0) = \underline{R}(I)(q_0) = \underset{p\in Q}{\wedge}\{(1 - R(q_0,p)) \vee \mu_I(p)\} = 0$$

$$\underline{I_1}(q_1) = \underline{R}(I)(q_1) = \underset{p\in Q}{\wedge}\{(1 - R(q_1,p)) \vee \mu_I(p)\} = 0.5$$

$$\underline{I_1}(q_2) = \underline{R}(I)(q_2) = \underset{p\in Q}{\wedge}\{(1 - R(q_2,p)) \vee \mu_I(p)\} = 0$$

$$\overline{I_1}(q_0) = \overline{R}(I)(q_0) = \underset{p\in Q}{\vee}\{R(q_0,p) \wedge \mu_I(p)\} = 0.2$$

$$\overline{I_1}(q_1) = \overline{R}(I)(q_1) = \underset{p\in Q}{\vee}\{R(q_1,p) \wedge \mu_I(p)\} = 0.5$$

$$\overline{I_1}(q_2) = \overline{R}(I)(q_2) = \underset{p\in Q}{\vee}\{R(q_2,p) \wedge \mu_I(p)\} = 0.5.$$

and $H_1 \in [0,1]^Q \times [0,1]^Q$ be the fuzzy rough set of final states of M_1 such that

$$\underline{H_1}(q_0) = \underline{R}(H)(q_0) = \underset{p\in Q}{\wedge}\{(1 - R(q_0,p)) \vee \mu_H(p)\} = 0$$

$$\underline{H_1}(q_1) = \underline{R}(H)(q_1) = \underset{p\in Q}{\wedge}\{(1 - R(q_1,p)) \vee \mu_H(p)\} = 0$$

$$\underline{H_1}(q_2) = \underline{R}(H)(q_2) = \underset{p\in Q}{\wedge}\{(1 - R(q_2,p)) \vee \mu_H(p)\} = 0.3$$

$$\overline{H_1}(q_0) = \overline{R}(H)(q_0) = \underset{p\in Q}{\vee}\{R(q_0,p) \wedge \mu_H(p)\} = 0.4$$

$$\overline{H_1}(q_1) = \overline{R}(H)(q_1) = \underset{p\in Q}{\vee}\{R(q_1,p) \wedge \mu_H(p)\} = 0$$

$$\overline{H_1}(q_2) = \overline{R}(H)(q_2) = \underset{p\in Q}{\vee}\{R(q_2,p) \wedge \mu_H(p)\} = 0.8.$$

The fuzzy rough transition δ_1 for M_1 obtained using Definition 4.1 is given in the Table 4 below:

The above designed FFRA M_1 is able to process input from X and capture not only vagueness of next transition state but also the uncertainty involved in transition of M. Now, suppose a user want to use above fuzzy finite automata M_1 with input alphabet $Y = \{b_1, b\}$, where (b_1 = Fuzzy Logic and b_2 = Optic Neuritis), then user fails to use it with input alphabet Y because Y is not the part of input set of M_1. But, from Wikipedia category structure user can identify that b_1 is a subcategory of a_1 ($b_1 < a_1$) and b_2 is a subcategory of a_2 ($b_2 < a_2$) and their semantic relatedness is defined as $sim(a_i, b_i) = 0.8$ for $a_i < b_i$ or $b_i < a_i$ otherwise, $sim(a_i, b_j) = 0.2$;$i \neq j$. Then he/she may use Definition 4.4, to redefine M_1 to produce FFRA for SC under semantically related concepts M_1', with transition function given as

$$\underline{\gamma_1(q_0,b_1)(q_1)} = \left(sim(b_1,a_1) \times \underline{\delta_1(q_0,a_1)(q_1)}\right) \vee \left(sim(b_1,a_2) \times \underline{\delta_1(q_0,a_2)(q_1)}\right) = 0.48$$

$$\underline{\gamma_1(q_0,b_2)(q_1)} = \left(sim(b_2,a_1) \times \underline{\delta_1(q_0,a_1)(q_1)}\right) \vee \left(sim(b_2,a_2) \times \underline{\delta_1(q_0,a_2)(q_1)}\right) = 0.12$$

$$\overline{\gamma_1(q_0,b_1)} = \left(sim(b_1,a_1) \times \overline{\delta_1(q_0,a_1)}\right) \vee \left(sim(b_1,a_2) \times \overline{\delta_1(q_0,a_2)}\right) = \frac{0.16}{q_0} + \frac{0.48}{q_1} + \frac{0.48}{q_2}$$

$$\overline{\gamma_1(q_0,b_2)} = \left(sim(b_2,a_1) \times \overline{\delta_1(q_0,a_1)}\right) \vee \left(sim(b_2,a_2) \times \overline{\delta_1(q_0,a_2)}\right) = \frac{0.04}{q_0} + \frac{0.12}{q_1} + \frac{0.12}{q_2}.$$

Table 5 External transition of M_1' Example 4.1

Q	$\gamma_1(q,b_1)$	$\gamma_1(q,b_2)$
q_0	$\left(\frac{0.48}{q_1}, \frac{0.16}{q_0} + \frac{0.48}{q_1} + \frac{0.48}{q_2}\right)$	$\left(\frac{0.12}{q_1}, \frac{0.04}{q_0} + \frac{0.12}{q_1} + \frac{0.12}{q_2}\right)$
q_1	$\left(\frac{0.06}{q_2}, \frac{0.08}{q_0} + \frac{0.12}{q_2}\right)$	$\left(\frac{0.24}{q_2}, \frac{0.32}{q_0} + \frac{0.48}{q_2}\right)$
q_2	$\left(\frac{0.24}{q_2}, \frac{0.32}{q_0} + \frac{0.32}{q_2}\right)$	$\left(\frac{0.06}{q_2}, \frac{0.08}{q_0} + \frac{0.08}{q_2}\right)$

Thus, semantic extension of FFRA M_1 is $(FFRA)_{SCRC}$ $M_1' = (Q, R, X, Y, \delta_1, \gamma_1, I_1, H_1)$, where external transition γ_1 of M_1' is given in Table 5.

Theorem 4.1 *Let $M_1 = (Q, R, X, \delta_1, I_1, H_1)$ be a FFRA corresponding to FFA $M = (Q, X, \delta, I, H)$ and $(FFRA)_{SCRC}$ $M_1' = (Q, R, X, Y, \delta_1, \gamma_1, I_1, H_1)$ be the semantic extension of M_1 w.r.t. related concepts. Then the following properties hold*:

(1) *For any $w \in X^*$; $d_{\underline{M_1}}(w) = d_{\underline{M_1'}}(w)$ and $d_{\overline{M_1}}(w) = d_{\overline{M_1'}}(w)$.*

(2) *For $w' \in (X \cup Y)^*$, if w' is definitely (or possibly) accepted by M_1' with membership degree $d_{\underline{M_1'}}(w')$ (resp. $d_{\overline{M_1'}}(w')$) then there exists $w \in X^*$ such that $d_{\underline{M_1'}}(w') \leq d_{\underline{M_1}}(w)$ (resp. $d_{\overline{M_1'}}(w') \leq d_{\overline{M_1}}(w)$).*

Proof

(1) Let $w \in X^*$. Then $(\delta_1 \cup \gamma_1)^*(q,w) = \delta_1^*(q,w)$,

$$d_{\overline{M_1'}}(w) = \vee\{\overline{I_1}(q) \wedge \overline{(\delta_1 \cup \gamma_1)^*(q,w)}(p) \wedge \overline{H_1}(p) : q,p \in Q\}$$
$$= \vee\{\overline{I_1}(q) \wedge \overline{\delta_1^*(q,w)}(p) \wedge \overline{H_1}(p) : q,p \in Q\}$$
$$= d_{\overline{M_1}}(w), \quad \text{and}$$
$$d_{\underline{M_1'}}(w) = \vee\{\underline{I_1}(q) \wedge \underline{(\delta_1 \cup \gamma_1)^*(q,w)}(p) \wedge \underline{H_1}(p) : q,p \in Q\}$$
$$= \vee\{\underline{I_1}(q) \wedge \underline{\delta_1^*(q,w)}(p) \wedge \underline{H_1}(p) : q,p \in Q\}$$
$$= d_{\underline{M_1}}(w).$$

(2) Let $w' \in (X \cup Y)^*$ and $w' = a_1 a_2 \dots a_n$ such that $a_i \in (X \cup Y), 1 \leq i \leq n$.

(i) If $w' \in X^*$. Then using part (1), we conclude that

$$d_{\underline{M_1'}}(w') = d_{\underline{M_1}}(w') \ \ and \ \ d_{\overline{M_1'}}(w') = d_{\overline{M_1}}(w'), i.e.,$$

$$d_{\underline{M_1'}}(w') \leq d_{\underline{M_1}}(w') \quad \text{and} \quad d_{\overline{M_1'}}(w') \leq d_{\overline{M_1}}(w').$$

(ii) If $w' \notin X^*$. Let w' be definitely accepted by M_1' then we have that

$$d_{\underline{M_1'}}(w') = \vee\{\underline{I_1}(q) \wedge \underline{(\delta_1 \cup \gamma_1)^*(q,w)}(p) \wedge \underline{H_1}(p) : q,p \in Q\} > 0.$$

Also, by Equations 4 and 5, we have

$$\begin{aligned}
&\underline{(\delta_1 \cup \gamma_1)^*(q,w')}(p) = \underline{(\delta_1 \cup \gamma_1)^*(q,a_1a_2\dots a_n)}(p)\\
&= \vee\Big\{\underline{(\delta_1 \cup \gamma_1)^*(q,a_1a_2\dots a_{n-1})}(q_{n-1}) \wedge \underline{(\delta_1 \cup \gamma_1)(q_{n-1},a_n)}(p) : q_{n-1} \in Q\Big\}\\
&= \vee\Big\{\Big(\vee\{\underline{(\delta_1 \cup \gamma_1)^*(q,a_1a_2\dots a_{n-2})}(q_{n-2}) \wedge \underline{(\delta_1 \cup \gamma_1)(q_{n-2},a_{n-1})}(q_{n-1}) : q_{n-2} \in Q\}\Big)\wedge\\
&\qquad \underline{(\delta_1 \cup \gamma_1)(q_{n-1},a_n)}(p) : q_{n-1} \in Q\Big\}\\
&\quad \vdots\\
&= \vee\Big\{\Big(\vee\Big(\vee\dots\Big(\vee\Big\{\underline{(\delta_1 \cup \gamma_1)(q,a_1)}(q_1) \wedge \underline{(\delta_1 \cup \gamma_1)(q_1,a_2)}(q_2) : q_1 \in Q\Big\}\Big)\wedge\dots\Big)\wedge\\
&\qquad \underline{(\delta_1 \cup \gamma_1)(q_{n-2},a_{n-1})}(q_{n-1}) : q_{n-2} \in Q\Big)\wedge \underline{(\delta_1 \cup \gamma_1)(q_{n-1},a_n)}(p) : q_{n-1} \in Q\Big\}\\
&= \vee\Big\{\underline{(\delta_1 \cup \gamma_1)(q,a_1)}(q_1) \wedge \underline{(\delta_1 \cup \gamma_1)(q_1,a_2)}(q_2) \wedge \dots \wedge \underline{(\delta_1 \cup \gamma_1)(q_{n-1},a_n)}(p) : q_1,\dots,q_{n-1} \in Q\Big\}.
\end{aligned}$$

Now, if $a_1 \in X$ then $\underline{(\delta_1 \cup \gamma_1)(q,a_1)}(q_1) = \underline{\delta_1(q,a_1)}(q_1)$ otherwise we have

$$\underline{(\delta_1 \cup \gamma_1)(q,a_1)}(q_1) = \underline{\gamma_1(q,a_1)}(q_1) = \bigvee_{b_1 \in X}\{sim(a_1,b_1) \times \underline{\delta_1(q,b_1)}(q_1)\}.$$

Then

$$\begin{aligned}
&\vee\Big\{\underline{(\delta_1 \cup \gamma_1)(q,a_1)}(q_1) \wedge \underline{(\delta_1 \cup \gamma_1)(q_1,a_2)}(q_2) \wedge \dots \wedge \underline{(\delta_1 \cup \gamma_1)(q_{n-1},a_n)}(p) : q_1,\dots,q_{n-1} \in Q\Big\}\\
&= \vee\Big\{\Big(\bigvee_{b_1 \in X}\{sim(a_1,b_1) \times \underline{\delta_1(q,b_1)}(q_1)\}\Big) \wedge \underline{(\delta_1 \cup \gamma_1)(q_1,a_2)}(q_2) \wedge \dots\wedge\\
&\qquad \underline{(\delta_1 \cup \gamma_1)(q_{n-1},a_n)}(p) : q_1,\dots,q_{n-1} \in Q\Big\}.
\end{aligned}$$

Applying same condition to other symbols, we get

$$\begin{aligned}
&\underline{(\delta_1 \cup \gamma_1)^*(q,w')}(p) = \underline{(\delta_1 \cup \gamma_1)^*(q,a_1a_2\dots a_n)}(p)\\
&= \vee\Big\{\underline{(\delta_1 \cup \gamma_1)(q,a_1)}(q_1) \wedge \underline{(\delta_1 \cup \gamma_1)(q_1,a_2)}(q_2) \wedge \dots \wedge \underline{(\delta_1 \cup \gamma_1)(q_{n-1},a_n)}(p) : q_1,\dots,q_{n-1} \in Q\Big\}\\
&= \vee\Big\{\Big(\bigvee_{b_1 \in X}\{sim(a_1,b_1) \times \underline{\delta_1(q,b_1)}(q_1)\}\Big) \wedge \Big(\bigvee_{b_2 \in X}\{sim(a_2,b_2) \times \underline{\delta_1(q_1,b_2)}(q_2)\}\Big) \wedge \dots\\
&\qquad \wedge \Big(\bigvee_{b_n \in X}\{sim(a_n,b_n) \times \underline{\delta_1(q_{n-1},b_n)}(p)\}\Big) : q_1,\dots,q_{n-1} \in Q\Big\}.
\end{aligned}$$

Since $d_{\underline{M_1'}}(w') = \vee\{\underline{I_1}(q) \wedge \underline{(\delta_1 \cup \gamma_1)^*(q,w')}(p) \wedge \underline{H_1}(p) : q,p \in Q\} > 0$, we have that $sim(a_i,b_i) \in (0,1]$ and for $b_i = a_i \in X$, $sim(a_i,b_i) = 1$. Let us denote $w = b_1b_2\dots b_n$. Then

$$\begin{aligned}
\underline{\delta_1^*(q,w)}(p) &= \underline{\delta_1^*(q,b_1b_2\dots b_n)}(p)\\
&= \vee\Big\{\underline{\delta_1(q,b_1)}(q_1) \wedge \underline{\delta_1(q_1,b_2)}(q_2)\wedge\\
&\qquad \dots \wedge \underline{\delta_1(q_{n-1},b_n)}(p) : q_1,\dots,q_{n-1} \in Q\Big\}.
\end{aligned}$$

Clearly, since $sim(a_i,b_i) \in (0,1]$, we have

$$\underline{(\delta_1 \cup \gamma_1)^*(q,w')}(p) \leq \underline{\delta_1^*(q,w)}(p), \ \text{i.e.,}$$

$$\begin{aligned}
d_{\underline{M_1'}}(w') &= \vee\{\underline{I_1}(q) \wedge \underline{(\delta_1 \cup \gamma_1)^*(q,w)}(p) \wedge \underline{H_1}(p) : q,p \in Q\}\\
&\leq \vee\{\underline{I_1}(q) \wedge \underline{\delta_1^*(q,w)}(p) \wedge \underline{H_1}(p) : q,p \in Q\}\\
&= d_{\underline{M_1}}(w).
\end{aligned}$$

Thus, $d_{\underline{M'_1}}(w') \le d_{\underline{M_1}}(w)$. Similarly, if w' is possibly accepted by M'_1 then there exists $w \in X^*$ such that $d_{\overline{M'_1}}(w') \le d_{\overline{M_1}}(w)$.

In Example 4.2, given below, we show that string $b_1a_2b_2 \in (X \cup Y)^*$, the input alphabet of $(FFRA)_{SCRC}$ M'_1 is definitely accepted by M'_1 with degree 0.24 and it follows results in Theorem 4.1.

Example 4.2 Consider fuzzy finite rough automaton for SC defined in Example 4.1. Let $w = b_1a_2b_2 \in (X \cup Y)^*$. Then by Definition 4.5,

$$
\begin{aligned}
d_{\underline{M'_1}}(w) &= d_{\underline{M'_1}}(b_1a_2b_2)\\
&= \vee\{\underline{I_1}(q) \wedge \underline{(\delta_1 \cup \gamma_1)^*(q, b_1a_2b_2)}(p) \wedge \underline{H_1}(p) : q, p \in Q\}\\
&= \underline{I_1}(q_0) \wedge \underline{(\delta_1 \cup \gamma_1)^*(q_0, b_1a_2b_2)}(q_2) \wedge \underline{H_1}(q_2).
\end{aligned}
$$

By Equation (4), we have

$$
\underline{(\delta_1 \cup \gamma_1)^*(q_0, b_1a_2b_2)}(q_2) = \vee \left\{\underline{(\delta_1 \cup \gamma_1)(q_0, b_1)}(p) \wedge \underline{(\delta_1 \cup \gamma_1)(p, a_2)}(r) \wedge \underline{(\delta_1 \cup \gamma_1)(r, b_2)}(q_2) : p, r \in Q\right\}.
$$

Since $b_1, b_2 \in Y$ and $a_2 \in X$;

$$\underline{(\delta_1 \cup \gamma_1)(q_0, b_1)}(p) = \underline{\gamma_1(q_0, b_1)}(p),$$

$$\underline{(\delta_1 \cup \gamma_1)(r, b_2)}(q_2) = \underline{\gamma_1(r, b_2)}(q_2),$$

$$\underline{(\delta_1 \cup \gamma_1)(p, a_2)}(r) = \underline{\delta_1(p, a_2)}(r).$$

Thus,

$$
\begin{aligned}
\underline{(\delta_1 \cup \gamma_1)^*(q_0, b_1a_2b_2)}(q_2) &= \vee\{\underline{\gamma_1(q_0, b_1)}(p) \ \wedge\ \underline{\delta_1(p, a_2)}(r) \ \wedge\ \underline{\gamma_1(r, b_2)}(q_2) : p, r \in Q\}\\
&= \underline{\gamma_1(q_0, b_1)}(q_1) \ \wedge\ \underline{\delta_1(q_1, a_2)}(q_2) \ \wedge\ \underline{\gamma_1(q_2, b_2)}(q_2)\\
&= 0.48 \wedge 0.3 \wedge 0.24\\
&= 0.24.
\end{aligned}
$$

Then,

$$
\begin{aligned}
d_{\underline{M'_1}}(b_1a_2b_2) &= \underline{I_1}(q_0) \wedge \underline{(\delta_1 \cup \gamma_1)^*(q_0, b_1a_2b_2)}(q_2) \wedge \underline{H_1}(q_2)\\
&= 0.5 \wedge 0.24 \wedge 0.3\\
&= 0.24.
\end{aligned}
$$

Also, we have

$$
\begin{aligned}
&\vee\{\underline{\gamma_1(q_0, b_1)}(p) \wedge \underline{\delta_1(p, a_2)}(r) \wedge \underline{\gamma_1(r, b_2)}(q_2) : p, r \in Q\}\\
&= \vee\Big\{ \vee\{sim(b_1, a_1) \times \underline{\delta_1(q_0, a_1)}(p) : a_1 \in X\} \wedge \underline{\delta_1(p, a_2)}(r) \wedge\\
&\quad \{\vee\{sim(b_2, a_1) \times \underline{\delta_1(r, a_1)}(q_2) : a_1 \in X\}\} : p, r \in Q\Big\}\\
&\le \vee\{\underline{\delta_1(q_0, a_1)}(p) \wedge \underline{\delta_1(p, a_2)}(r) \wedge \underline{\delta_1(r, a_1)}(q_2) : p, r \in Q\}\\
&= \underline{\delta^*(q_0, a_1a_2a_1)}(q_2),
\end{aligned}
$$

i. e., corresponding to $w = b_1a_2b_2 \in (X \cup Y)^*$; we have $w' = a_1a_2a_1 \in X^*$ with $|w| = |w'|$, and

$$
\begin{aligned}
d_{\underline{M_1}}(a_1a_2a_1) &= \underline{I_1}(q_0) \wedge \underline{\delta_1^*(q, a_1a_2a_1)}(q_2) \wedge \underline{H_1}(q_2)\\
&= 0.5 \wedge 0.3 \wedge 0.3\\
&= 0.3.
\end{aligned}
$$

We conclude that $d_{\underline{M'_1}}(b_1a_2b_2) \le d_{\underline{M_1}}(a_1a_2a_1)$. Similarly, one can verify that $d_{\overline{M'_1}}(b_1a_2b_2) \le d_{\overline{M_1}}(a_1a_2a_1)$.

For an FFRA for SC under semantically related concepts defined in Definition 4.4 user can use both alphabets X and Y just like in the case of RFSA for SC under equivalent concepts (Definition 3.1). Suppose, for some practical applications; any user wants to apply FFRA for SC with their alphabet (cf., [22] for details) only. Then similar to Definition 3.5, $(FFRA)_{SCRC}$ is formalized in a more general way in accordance with the present situation.

4.3 Fuzzy finite rough automata for SC with respect to external alphabet

Herein, we define FFRA for SC with respect to external alphabets,

Definition 4.7 A **fuzzy finite rough automaton for SC with respect to external alphabet** $(FFRA)_{SCEA}$ is an eight tuples

$M'_1 = (Q, R, X, Y, \delta_1, \gamma_1, I_1, H_1)$, where

(i) $M_1 = (Q, R, X, \delta_1, I_1, H_1)$ is a FFRA.
(ii) Y and γ_1 are same as defined in Definition 4.4.
(iii) Y is the alphabet of M'_1.
(vi) $\gamma_1 : Q \times Y \to [0, 1]^Q \times [0, 1]^Q$ is the transition map for M'_1.

The fuzzy rough transition map γ_1 of $(FFRA)_{SCEA}$ M'_1 for SC w.r.t. external alphabet Y can further be extended to $\gamma_1^* : Q \times Y^* \to [0, 1]^Q \times [0, 1]^Q$ and defined as

$$\gamma_1^*(q, \epsilon) = \left(1/q, 1/q\right) \ \forall q \in Q$$

$$\gamma_1^*(q, wa) = \left(\underline{\gamma_1^*(q, wa)}, \overline{\gamma_1^*(q, wa)}\right), \quad \text{where}$$

$$\underline{\gamma_1^*(q,wa)}(p) = \vee\{\underline{\gamma_1^*(q,w)}(r) \wedge \underline{\gamma_1(r,a)}(p) \,:\, r \in Q\} \tag{6}$$

$$\overline{\gamma_1^*(q,wa)}(p) = \vee\{\overline{\gamma_1^*(q,w)}(r) \wedge \overline{\gamma_1(r,a)}(p) \,:\, r \in Q\}, \tag{7}$$

for all $q \in Q, p \in Q, w \in Y^*$ *and* $a \in Y$.

Definition 4.8 Let $M_1' = (Q,R,X,Y,\delta_1,\gamma_1,I_1,H_1)$ be a $(FRA)_{SCEA}$ and $w \in Y^*$, then **the degree to which** w **is accepted definitely by** M_1' is

$$d_{\underline{M_1'}}(w) = \vee\{\underline{I_1}(q) \wedge \underline{\gamma_1^*(q,w)}(p) \wedge \underline{H_1}(p) \,:\, q,p \in Q\}$$

and **the degree to which** w **is accepted possibly by** M_1' is

$$d_{\overline{M_1'}}(w) = \vee\{\overline{I_1}(q) \wedge \overline{\gamma_1^*(q,w)}(p) \wedge \overline{H_1}(p) \,:\, q,p \in Q\}.$$

Definition 4.9 The **language accepted by** M_1' is a fuzzy rough subset of Y^* denoted by $L(M_1') = (\underline{L(M_1')}, \overline{L(M_1')})$, where $\underline{L(M_1')}$ and $\overline{L(M_1')}$ are fuzzy sets of Y^*:

$$\underline{L(M_1')} = \{(w, d_{\underline{M_1'}}(w)) \,:\, w \in Y^*\}$$

$$\overline{L(M_1')} = \{(w, d_{\overline{M_1'}}(w)) \,:\, w \in Y^*\}.$$

Theorem 4.2 *Let* $(FFRA)_{SCEA}\, M_1' = (Q,R,X,Y,\delta_1,\gamma_1,I_1,H_1)$ *be a semantic extension of FFRA* $M_1 = (Q,R,X,\delta_1,I_1,H_1)$ *with respect to external alphabet. Then for any* $w' \in Y^*$ *such that* $d_{\underline{M_1'}}(w') > 0$ *or* $d_{\overline{M_1'}}(w') > 0$, *there exists a string* $w \in X^*$ *such that* $d_{\underline{M_1'}}(w') \le d_{\underline{M_1}}(w)$ *or* $d_{\overline{M_1'}}(w') \le d_{\overline{M_1}}(w)$.

Proof Let $w' = y_1y_2\dots y_m$, where $y_i \in Y, 1 \le i \le m$ be definitely accepted by $(FFRA)_{SCEA}\, M_1'$. i.e.,

$$d_{\underline{M_1'}}(w') = \vee\{\underline{I_1}(q) \wedge \underline{\gamma_1^*(q,w')}(p) \wedge \underline{H_1}(p) \,:\, q,p \in Q\} > 0.$$

Now,

$$\begin{aligned}
\underline{\gamma_1^*(q,w')}(p) &= \underline{\gamma_1^*(q,y_1y_2\dots y_m)}(p)\\
&= \vee\left\{\underline{\gamma_1^*(q,y_1y_2\dots y_{m-1})}(q_{m-1}) \wedge \underline{\gamma_1(q_{m-1},y_m)}(p) \,:\, q_{m-1} \in Q\right\}\\
&= \vee\left\{\left(\vee\{\underline{\gamma_1^*(q,y_1y_2\dots y_{m-2})}(q_{m-2}) \wedge \underline{\gamma_1(q_{m-2},y_{m-1})}(q_{m-1}) \,:\, q_{m-2} \in Q\}\right)\wedge\right.\\
&\qquad \left.\underline{\gamma_1(q_{m-1},a_m)}(p) \,:\, q_{m-1} \in Q\right\}\\
&\ \ \vdots\\
&= \vee\left\{\left(\vee\left(\vee\dots\left(\vee\left\{\underline{\gamma_1(q,y_1)}(q_1) \wedge \underline{\gamma_1(q_1,y_2)}(q_2) \,:\, q_1 \in Q\right\}\right)\wedge\dots\right)\wedge\right.\right.\\
&\qquad \left.\left.\underline{\gamma_1(q_{m-2},y_{m-1})}(q_{m-1}) \,:\, q_{m-2} \in Q\right) \wedge \underline{\gamma_1(q_{m-1},y_m)}(p) \,:\, q_{m-1} \in Q\right\}\\
&= \vee\left\{\underline{\gamma_1(q,y_1)}(q_1) \wedge \underline{\gamma_1(q_1,y_2)}(q_2) \wedge \dots \wedge \underline{\gamma_1(q_{m-1},y_m)}(p) \,:\, q_1,\dots,q_{m-1} \in Q\right\}.
\end{aligned}$$

Now, by definition of γ_1 (Definition 4.4), we have

$$\underline{\gamma_1(q,y_1)}(q_1) = \bigvee_{x_1 \in X}\{sim(y_1,x_1) \times \underline{\delta_1(q,x_1)}(q_1)\}$$

$$\underline{\gamma_1(q_1,y_2)}(q_2) = \bigvee_{x_2 \in X}\{sim(y_2,x_2) \times \underline{\delta_1(q_1,x_2)}(q_2)\}$$

$$\vdots$$

$$\underline{\gamma_1(q_{m-1},y_m)}(p) = \bigvee_{x_m \in X}\{sim(y_m,x_m) \times \underline{\delta_1(q_{m-1},x_m)}(p)\}.$$

Also, since $d_{\underline{M_1'}}(w') > 0$, we have $sim(y_i,x_i) \in (0,1]$ for $1 \le i \le m$. Thus,

$$\begin{aligned}
&\vee\left\{\underline{\gamma_1(q,y_1)}(q_1) \wedge \underline{\gamma_1(q_1,y_2)}(q_2) \wedge \dots \wedge \underline{\gamma_1(q_{m-1},y_m)}(p) \,:\, q_1,\dots,q_{m-1} \in Q\right\}\\
&= \vee\left\{\left(\bigvee_{x_1 \in X}\left\{sim(y_1,x_1) \times \underline{\delta_1(q,x_1)}(q_1)\right\}\right) \wedge \left(\bigvee_{x_2 \in X}\left\{sim(y_2,x_2) \times \underline{\delta_1(q_1,x_2)}(q_2)\right\}\right) \wedge \dots \wedge\right.\\
&\qquad \left.\left(\bigvee_{x_m \in X}\left\{sim(y_m,x_m) \times \underline{\delta_1(q_{m-1},x_m)}(p)\right\}\right) \,:\, q_1,\dots,q_{m-1} \in Q\right\}\\
&\quad = \vee\left\{\vee\left\{\left(sim(y_1,x_1) \times \underline{\delta_1(q,x_1)}(q_1)\right) \wedge \left(sim(y_2,x_2) \times \underline{\delta_1(q_1,x_2)}(q_2)\right) \wedge \dots \wedge\right.\right.\\
&\qquad \left.\left.\left(sim(y_m,x_m) \times \underline{\delta_1(q_{m-1},x_m)}(p)\right) \,:\, x_1,x_2,\dots,x_n \in X\right\} \,:\, q_1,\dots,q_{m-1} \in Q\right\}\\
&\quad \le \vee\left\{\underline{\delta_1(q,x_1)}(q_1) \wedge \underline{\delta_1(q_1,x_2)}(q_2) \wedge \dots \wedge \underline{\delta_1(q_{m-1},x_m)}(p) \,:\, q_1,\dots,q_{m-1} \in Q\right\}\\
&\quad (\text{ since } sim(y_i,x_i) \in (0,1],\ \ 1 \le i \le m)\\
&\quad = \underline{\delta_1^*(q,x_1x_2\dots x_m)}(p).
\end{aligned}$$

Let us represent $x_1x_2 \dots x_m = w$ and clearly $w \in X^*$. Thus, corresponding to $w' = y_1y_2 \dots y_m \in Y^*$, we got a string $w \in X^*$ with $|w| = |w'|$ such that

$$\underline{\gamma_1^*(q, w')}(p) \le \underline{\delta_1^*(q, w)}(p)$$

and

$$\overline{\gamma_1^*(q, w')}(p) \le \overline{\delta_1^*(q, w)}(p)$$

(proceeding the same steps performed for transition of lower approximation). Then

$$\underline{I_1}(q) \wedge \underline{\gamma_1^*(q, w')}(p) \wedge \underline{H_1}(p) \le \underline{I_1}(q) \wedge \underline{\delta_1^*(q, w)}(p) \wedge \underline{H_1}(p)$$

and

$$\overline{I_1}(q) \wedge \overline{\gamma_1^*(q, w')}(p) \wedge \overline{H_1}(p) \le \overline{I_1}(q) \wedge \overline{\delta_1^*(q, w)}(p) \wedge \overline{H_1}(p)$$

$$\implies \vee\{\underline{I_1}(q) \wedge \underline{\gamma_1^*(q, w')}(p) \wedge \underline{H_1}(p) : q, p \in Q\} \le \vee\{\underline{I_1}(q) \wedge \underline{\delta_1^*(q, w)}(p) \wedge \underline{H_1}(p) : q, p \in Q\}$$

$$and \quad \vee\{\overline{I_1}(q) \wedge \overline{\gamma_1^*(q, w')}(p) \wedge \overline{H_1}(p) : q, p \in Q\} \le \vee\{\overline{I_1}(q) \wedge \overline{\delta_1^*(q, w)}(p) \wedge \overline{H_1}(p) : q, p \in Q\}$$

$$\implies d_{\underline{M_1'}}(w') \le d_{\underline{M_1}}(w) \quad \text{and} \quad d_{\overline{M_1'}}(w') \le d_{\overline{M_1}}(w).$$

In Example 4.3, given below we show that string $b_1b_2 \in Y^*$ from the input alphabet Y of $(FFRA)_{SCEA}$ M_1' is definitely accepted by M_1' with degree 0.24, where as string $w' = a_1a_2 \in X^*$ from the input alphabet X of FFRA M_1 is definitely accepted by M_1 and it follows results in Theorem 4.2.

Example 4.3 Consider FFRA for SC defined in Example 4.1. Let $w = b_1b_2 \in Y^*$. Then by Definition 4.8,

$$\begin{aligned} d_{\underline{M_1'}}(w) = d_{\underline{M_1'}}(b_1b_2) &= \vee\{\underline{I_1}(q) \wedge \underline{\gamma_1^*(q, b_1b_2)}(p) \wedge \underline{H_1}(p) : q, p \in Q\} \\ &= \underline{I_1}(q_0) \wedge \underline{\gamma_1^*(q_0, b_1b_2)}(q_2) \wedge \underline{H_1}(q_2). \end{aligned}$$

By Equation (6), we have

$$\underline{\gamma_1^*(q_0, b_1b_2)}(q_2) = \vee\Big\{\underline{\gamma_1(q_0, b_1)}(p) \wedge \underline{\gamma_1(p, b_2)}(q_2) : p \in Q\Big\}.$$

Then,

$$\begin{aligned} d_{\underline{M_1'}}(b_1b_2) &= \underline{I_1}(q_0) \wedge \underline{\gamma_1(q_0, b_1)}(q_1) \wedge \underline{\gamma_1(q_1, b_2)}(q_2) \wedge \underline{H_1}(q_2) \\ &= 0.5 \wedge 0.48 \wedge 0.24 \wedge 0.3 \\ &= 0.24. \end{aligned}$$

Also, we have

$$\begin{aligned} &\vee\{\underline{\gamma_1(q_0, b_1)}(p) \wedge \underline{\gamma_1(p, b_2)}(q_2) : p \in Q\} \\ &= \vee\Big\{ \vee\{sim(b_1, a_1) \times \underline{\delta_1(q_0, a_1)}(p) : a_1 \in X\} \wedge \\ &\{\vee\{sim(b_2, a_2) \times \underline{\delta_1(p, a_2)}(q_2) : a_2 \in X\}\} : p \in Q\Big\} \\ &\le \vee\{\underline{\delta_1(q_0, a_1)}(p) \wedge \underline{\delta_1(p, a_2)}(q_2)\} = \underline{\delta^*(q_0, a_1a_2)}(q_2), \end{aligned}$$

(since $b_1, b_2 \in Y$ with semantic relation $b_1 < a_1$, $b_2 < a_2$ and we have defined $sim(b_i, a_i) = 0.8$, $i \in 1, 2$ and $sim(b_i, a_j) = 0.2$ for $i \ne j$), i.e., corresponding to $w = b_1b_2 \in Y^*$; we have $w' = a_1a_2 \in X^*$ with $|w| = |w'|$, and

$$\begin{aligned} d_{\underline{M_1}}(a_1a_2) &= \vee\{\underline{I_1}(q) \wedge \underline{\delta_1^*(q, a_1a_2)}(p) \wedge \underline{H_1}(p) : q, p \in Q\} \\ &= 0.5 \wedge 0.3 \wedge 0.3 \\ &= 0.3. \end{aligned}$$

Here, we conclude that $d_{\underline{M_1'}}(b_1b_2) \le d_{\underline{M_1}}(a_1a_2)$. In a similar way, one can verify that $d_{\overline{M_1'}}(b_1b_2) \le d_{\overline{M_1}}(a_1a_2)$.

5 Conclusion

Motivated by the usefulness of semantic computing techniques to handle information from the dataset obtained by real-world applications and their applications in the theory of computation proposed and studied in [20, 22, 61], we have introduced two new models of computation for SC, one is RFSA for SC, and another is FFRA for SC corresponding to a given FFSA. The proposed RFSA for SC is a mathematical model of natural language, which not only captures the incomplete and insufficient information in the dataset obtained from real-world applications but also accepts semantically equivalent incomplete and insufficient input information (see Example 3.1), and also external input from the dataset obtained by real-world applications. However, traditional RFSA in [5] accepts only incomplete and insufficient input information from the dataset obtained from real-world applications, and computing model finite automata for SC in [22] accepts only semantically equivalent input information and external input from the dataset obtained by real-world applications. Our second proposed model FFRA for SC corresponding to a given FFSA, is another mathematical model of computation which can accept semantically related input (see Examples 4.1 and 4.2), and external input alphabet information from the dataset obtained from real-world applications, in the vague and incomplete environment, whereas the computing model FFSA accept crisp input information from the dataset obtained by real-world applications and computing model finite automata based on SC in [22] accept crisp input and semantically related input, and external input alphabet information from the dataset obtained by real-world applications. For an FFRA for SC under semantically related concepts defined in Definition 4.4 user can use both alphabets X and Y just like in the case of RFSA for SC under equivalent concepts (Definition 3.1). Suppose, for some practical applications; any user wants to apply FFRA for SC with their alphabet (cf., [22] for details) only. Then similar to Definition 3.5, $(FFRA)_{SCRC}$ is formalized in a more general way in accordance with the present situation. Moreover, We have shown that the proposed models of computations RFSA for SC and FFRA for SC provide better user experience and applications as compared to existing models of computations or existing models of computations for SC. We feel that concepts discussed in this paper help researchers to define more general hybrid models of computations for SC.

Acknowledgements The work of Vijay K. Yadav is supported by grant number F.30-414/2018(BSR), UGC, New Delhi, India. Mausam Kumari is thankful to CSIR, New Delhi, India, for providing financial assistance through JRF F.N. 09/1131(0033)/2019-EMR-1. The authors are greatly indebted to the Area Editor and the Referee(s) for their valuable observations and suggestions for improving the paper.

References

1. Abedjan Z, Chu X, Deng D, Fernandez RC, Ilyas IF, Ouzzani M, Papotti P, Stonebraker M, Tang N (2016) Detecting data errors: where are we and what needs to be done? Proc VLDB Endow 9(12):993–1004
2. Aouicha MB, Taieb MAH, Hamadou AB (2018) SISR: system for integrating semantic relatedness and similarity measures. Soft Comput 22(6):1855–1879
3. Arulprakasam R, Perumal R, Radhakrishnan M, Dare VR (2018) Rough finite state automata and rough languages. J Phys Conf Ser IOP Publish 1000:012155
4. Basu S (2003) Rough grammar and rough language. Found Comput Decis Sci 28(3):129–141
5. Basu S (2005) Rough finite-state automata. Cybern Syst 36(2):107–124
6. Batchkarov M, Kober T, Reffin J, Weeds J, Weir D (2016) A critique of word similarity as a method for evaluating distributional semantic models. In: Proceedings of the 1st Workshop on Evaluating Vector Space Representations for NLP, Berlin, Germany, August 12, pp 7-12
7. Bruni E, Taran NK, Baroni M (2014) Multimodal distributional semantics. J Artif Intell Res 49:1–47
8. Camacho A, Merayo M. G, Núñez M (2017) Using fuzzy automata to diagnose and predict heart problems. In: IEEE Congress on Evolutionary Computation (CEC), pp 846-853. https://doi.org/10.1109/CEC.2017.7969397
9. de Mendívil JRG, Garitagoitia JR (2014) Determinization of fuzzy automata via factorization of fuzzy states. Inform Sci 283:165–179
10. Deng L, Wang K, Guido RC (2010) A semantic and detection-based approach to speech and language processing. Sheu et al. (Ed) Semantic Computing, pp 49-68
11. Dubois D, Prade H (1990) Rough fuzzy sets and fuzzy rough sets. Int J Gen Syst 17:191–209
12. Finkelstein L, Gabrilovich E, Matias Y, Rivlin E, Solan Z, Wolfman G, Ruppin E (2001) Placing search in context:The concept revisited. In: Proceeding soft he 10th international conference on World Wide Web, 406–414
13. Gil JM, Montes JA (2013) Semantic similarity measurement using historical google search patterns. Inform Syst Front 15(3):399–410
14. Gracia J, Mena E (2008) Web-based measure of semantic relatedness. In: In International Conference on Web Information Systems Engineering, Springer, Berlin, Heidelberg, pp 136–150
15. Guido RC et al (2007) Spoken document summarization based on dynamic time warping and wavelets. Int J Seman Comput 1:347–357
16. Halawi G, Dror G, Gabrilovich E, Koren Y (2012) Large-scale learning of word relatedness with constraints. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, China, 1406–1414
17. Hao K, Gong Z, Huo C, Sheu PCY (2011) Semantic computing and computer science. Int J Seman Comput 5(1):95–120
18. Hill F, Reichart R, Korhonen A (2015) SimLex-999: evaluating semantic models with (genuine) similarity estimation. Comput Ling 41(4):665–695
19. Hopcroft JE, Motwani R, Ullman JD (2001) Introduction to automata theory, languages, and computation. Acm Sigact News 32(1):60–65
20. Huang G, Wasti SH, Wei L, Jiang Y (2019) A formal model for semantic computing based on generalized probabilistic automata. Entropy. https://doi.org/10.3390/e21090903
21. Ignjatović J, Ćirić M, Bogdanović S, Petković T (2010) Myhill-Nerode type theory for fuzzy languages and automata. Fuzzy Sets Syst 161: 1288-1324
22. Jiang Y (2019) A formal model of semantic computing. Soft Comput 23(14):5411–5429
23. Jiang Y, Zhang X, Tang Y, Nie R (2015) Feature-based approaches to semantic similarity assessment of concepts using Wikipedia. Inform Process Manag 51(3):215–234
24. Jiang Y, Bai W, Zhang X, Hu J (2017) Wikipedia-based information content and semantic similarity computation. Inform Process Manag 53(1):248–265
25. Jin J, Kim M, Rivett P (2015) Semantic computing for education. Int J Seman Comput 9(3):395–413
26. Kim J, Ostrowski DA, Yamaguch H, Sheu PCY (2013) Semantic computing and business intelligence. Int J Seman Comput 7(1):87–117
27. Li Y, Bandar ZA, Mclean D (2003) An approach for measuring semantic similarity between words using multiple information sources. IEEE Trans Knowled Data Eng 15(4):871–882
28. Lin D (1998) Automatic retrieval and clustering of similar words. In: 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, vol 2 pp 768–774
29. Madrid N, Moreno JM, Poussa ER (2020) Rough sets based on Galois connections. Int J Appl Math Comput Sci 30(2):299–313
30. Meng D, Zhang X, Qin K (2011) Soft rough fuzzy sets and soft fuzzy rough sets. Comput Math Appl 62(12):4635–4645
31. Miller G, Charles W (1991) Contextual correlates of semantic similarity. Lang Cogn Process 6(1):1–28

32. Mordeson JN, Malik DS (2002) Fuzzy automata and languages: theory and applications. CRC Press
33. Murali V (1989) Fuzzy equivalence relations. Fuzzy Sets Syst 30(2):155–163
34. Pal P, Tiwari S. P, Singh S (2021) *L*-fuzzy rough automaton: a mathematical model for natural languages. Int J Mach Learn Cybern: 1–17
35. Pawlak Z (1982) Rough sets. Int J Comput Inform Sci 11(5):341–356
36. Polkowski L, Skowron A (1997) Rough set approximations of languages. Fund Inform 32(2):149–162
37. Rubenstein H, Goodenough J (1965) Contextual correlates of synonymy. Commun ACM 8(10):627–633
38. Santos ES (1968) Maximin automata. Inform Control 13(4):363–377
39. Sharan S, Tiwari SP, Yadav VK (2015) Interval type-2 fuzzy rough sets and interval type-2 fuzzy closure spaces. Iran J Fuzzy Syst 12:113–125
40. Sharan S, Srivastava AAK, Tiwari SP (2017) Characterizations of rough finite-state automata. Int J Mach Learn Cybern 8:721–730
41. Sharma BK, Sharan S, Tiwari SP (2017) On minimization of deterministic automaton with rough output. In: Arumugam S, Bagga J, Beineke L, Panda B (eds) Theoretical computer science and discrete mathematics. ICTCSDM 2016. Lecture Notes in Computer Science, Springer, pp 10398
42. Sheu PCY, Yu H, Ramamoorthy CV, Joshi AK, Lotfi AZ (2010) Semantic computing. IEEE Press, Wiley, New York
43. Singh S, Tiwari SP (2021) Fuzzy partitioned discrete-event system and its supervisory control. Int J Gen Syst 50(3):339–365
44. Slowinski R, Vanderpooten D (2000) A generalized definition of rough approximations based on similarity. IEEE Trans Knowl Data Eng 12(2):331–336
45. Sohn M, Jeong S, Kim J, Lee HJ (2017) Crowdsourced healthcare knowledge creation using patient's health experience-ontologies. Soft Comput 21:5207–5221
46. Szumlanski S, Gomez F, Sims V. K (2013) A new set of norms for semantic relatedness measures. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, August 4-9 2013, 890–895
47. Tao-Rong Q, Qing L, Huang HK (2009) A granular computing approach to knowledge discovery in relational databases. Acta Autom Sinica 35(8):1071–1079
48. Tiwari SP, Sharan S (2015) Products of rough finite-state machines. Int J Multip Valued Log Soft Comput 25:339–356
49. Tiwari SP, Sharan S, Singh AK (2013) On coverings of products of rough transformation semigroups. Int J Found Comput Sci 24:375–391
50. Tiwari SP, Yadav VK, Singh AK (2014) Construction of a minimal realization and monoid for a fuzzy language: a categorical approach. J Appl Math Comput 47:401–416
51. Tiwari SP, Yadav VK, Singh AK (2015) On algebraic study of fuzzy automata. Int J Mach Learn Cybern 6(3):479–485
52. Tiwari SP, Yadav VK, Dubey MK (2016) Minimal realization for fuzzy behaviour: a bicategory-theoretic approach. Int J Fuzzy Syst 30(2):1057–1065
53. Tiwari SP, Yadav VK, Gautam V (2017) On minimal fuzzy realization for a fuzzy language: a categorical approach. J Multip Valued Log Soft Comput 28:361–374
54. Tiwari SP, Yadav VK, Pal P, Sharma BK (2018) Minimal fuzzy realization for fuzzy behaviour: a bicategory-theoretic approach. J Multip Valued Log Soft Comput 31:105–121
55. Tiwari SP, Yadav VK, Davvaz B, Renu (2018) Categorical approach to minimal realization for a fuzzy language. Fuzzy Sets Syst 351:122–137
56. Tripathi A, Tyagi K (2019) Rough fuzzy (final) automata. In: AIP Conference Proceedings. AIP Publishing LLC, vol 2061(1), pp 020019
57. Turney PD (2006) Similarity of semantic relations. Comput Linguist 32(3):379–416
58. Tyagi K, Tripathi A (2017) Rough fuzzy automata and rough fuzzy grammar. Int J Fuzzy Syst Appl (IJFSA) 6(1):36–55
59. Wang F, Hu L, Zhou J, Hu J, Zhao K (2017) A semantics-based approach to multi-source heterogeneous information fusion in the internet of things. Soft Comput 21:2005–2013
60. Wee WG, Fu KS (1969) A formulation of fuzzy automata and its application as a model of learning systems. IEEE Trans Syst Sci Cybern 5(3):215–223
61. Wei L, Huang G, Wasti SH, Hussain MJ, Jiang Y (2021) Generalized fuzzy automata with semantic computing. Soft Comput 25(7):5775–5789
62. Yadav S, Tiwari SP, Kumari M, Yadav VK (2022) An interval type-2 fuzzy model of computing with words via interval type-2 fuzzy finite rough automata with application in COVID-19 deduction. New Math Natl Computat 18(1):61–101
63. Yadav S, Tiwari SP, Kumari M, Yadav VK (2021) Bicategory-theoretic approach to minimal fuzzy realization for fuzzy behavior. New Math Natl Comput 17(3):775–799
64. Ying M (2002) A formal model of computing with words. IEEE Trans Fuzzy Syst 10:640–652
65. Zadeh LA (1996) Fuzzy sets. In Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh: 394-432
66. Zesch T (2010) Study of semantic relatedness of words using collaboratively constructed semantic resources. Technische Universitat, Ph.D. diss
67. Zhai J, Zhang S, Zhang Y (2016) An extension of rough fuzzy set. J Intell Fuzzy Syst 30(6):3311–3320
68. Zumwald M, Knüsel B, Baumberger C, Hadorn G. H, Bresch D. N, Knutti R (2020) Understanding and assessing uncertainty of observational climate datasets for model evaluation using ensembles. WIREs Clim Change. https://doi.org/10.1002/wcc.654

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.