Rough Finite State Automata and Rough Languages

R Arulprakasam, R Perumal, M Radhakrishnan and V R Dare

1, 2, 3 Department of Mathematics, Faculty of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu.
4 Department of Mathematics, Madras Christian College, Tambaram, Tamil Nadu.

E-mail: r.arulprakasam@yahoo.com, rajkumardare@yahoo.com

Abstract. Sumita Basu [1, 2] recently introduced the concept of a rough finite state (semi)automaton, rough grammar and rough languages. Motivated by the work of [1, 2], in this paper, we investigate some closure properties of rough regular languages and establish the equivalence between the classes of rough languages generated by rough grammar and the classes of rough regular languages accepted by rough finite automaton.

1. Introduction

The first extension of formal languages was stochastic language. Based on the concept of fuzzy set introduced by Zadeh [6], Wee [5], defined the mathematical formulation of fuzzy automata and initiated the studies of fuzzy languages accepted by fuzzy automata. Pawlak’s [4] introduced the concept of rough sets in 1982, like fuzzy set theory, is another mathematical approach to deal with imprecise, uncertain or incomplete information and knowledge. It has rapidly drawn attention of both mathematicians and computer scientists due to its ability to model many aspects of artificial intelligence and cognitive sciences, particularly in the areas of knowledge acquisition, decision analysis and expert systems. Formal languages, finite automata and grammars, which generate them, cannot be successfully applied to model a real language, as they do not include the incorrectness and ambiguity of natural languages. Imprecision of real language can be incorporated in the formal language theory with the help of rough sets. Rough set approximation of language was done by Paun et al.’s [3] in 1997. They investigated the possibility of approximating a language when only a part of the string (called the sub string) is known. By varying the length of the sub string a sequence of approximations may be obtained which converge to the language. Polkowski [3], has introduced the equivalence relations among strings, which leads to the lower and upper approximations of languages and proved that, the approximations of context-free languages to be regular languages. Following the advent of rough set theory, Sumita Basu [1, 2] recently introduced the concept of a rough finite state (semi)automaton, rough grammar and rough languages. Motivated by the work of [1, 2], in this paper, we investigate some closure properties of rough regular languages and establish the equivalence between the classes of rough languages generated by rough grammar and the classes of rough regular languages accepted by rough finite automaton.

2. Preliminaries and Notations

In this section, we recall the concept of a rough finite-state automaton and rough grammar in details [1, 2].
Definition 2.1 A pair \((X, R)\) is called an approximation space, where \(X\) is a nonempty set and \(R\) is an equivalence relation on \(X\). For \(x \in X\), \([x]_R = \{y \in X \mid x Ry\}\) is called an equivalence class or a block of \(x\) under \(R\). Let \(X/R = \{[x]_R \mid x \in X\}\).

Definition 2.2 Let \((X, R)\) be an approximation space and \([x]_R\) be the equivalence class of \(x\) under \(R\). Then lower approximation and upper approximation of \(A \subseteq X\) are, respectively, defined to be the sets

\[
A = \{x \in X : [x]_R \subseteq A\} \quad \text{and} \quad \overline{A} = \{x \in X : [x]_R \cap A \neq \emptyset\}.
\]

For an approximation space \((X, R)\), \(A \subseteq X\) is called a definable set if it is an union of equivalence classes under \(R\) and a pair \((L, U)\) of definable sets is called a rough set in \((X, R)\), if \(L \subseteq U\), and if any equivalence class of \(X\) is a singleton set \([x]\) such that \(x \in U\), then \(x \in L\).

Definition 2.3 A rough finite-state automaton (or RFSA) is a 4-tuple \(M = (Q, R, \Sigma, \delta)\) where \(Q\) is a finite nonempty set of states, \(R\) is an equivalence relation on \(Q\), \(\Sigma\) is a finite set of inputs and \(\delta : Q \times \Sigma \to A\), where \(A = \{\{A, \overline{A}\} : A \subseteq Q\}\) is a map (called the rough transition map) such that for each \((q, a) \in Q \times \Sigma\), \(\delta(q, a) = (\{A, \overline{A}\})\) being a rough set in \((Q, R)\) for some \(A \subseteq Q\).

We shall denote \(A\) and \(\overline{A}\) as \(\delta(q, a)\) and \(\overline{\delta(q, a)}\) respectively and we will write the set of all rough sets \(\{\{A, \overline{A}\} : A \subseteq Q\}\) in the approximation space \((Q, R)\) as \(A\).

Example 2.4 Consider the RFSA \(M = (Q, R, \Sigma, \delta)\) where \(Q = \{q_1, q_2, q_3, q_4, q_5\}\), \(R\) is an equivalence relation on \(Q\) with \(Q/R = \{\{q_1, q_3\}, \{q_2\}, \{q_4, q_5\}\}\), \(\Sigma = \{a, b\}\) and the rough transition map \(\delta\) is given by the following table:

| \(Q\)   | \(\delta(q, a)\)                                      | \(\delta(q, a)\)                                      |
|---------|------------------------------------------------------|------------------------------------------------------|
| \(q_1\) | \(\emptyset, \{q_1, q_3\}\)                        | \(\{q_2\}, \{q_2\} \cup \{q_1, q_3\}\)             |
| \(q_2\) | \(\{q_1, q_3\}, \{q_2\} \cup \{q_1, q_3\}\)      | \(\{q_4, q_5\}, \{q_4, q_5\} \cup \{q_1, q_3\}\)  |
| \(q_3\) | \(\emptyset, \{q_1, q_3\} \cup \{q_2\}\)          | \(\{q_2\}, \{q_2\} \cup \{q_1, q_3\}\)             |
| \(q_4\) | \(\{q_4, q_5\}, \{q_4, q_5\} \cup \{q_2\}\)      | \(\{q_2\}, \{q_2\} \cup \{q_1, q_3\}\)             |
| \(q_5\) | \(\{q_2\}, \{q_2\} \cup \{q_1, q_3\}\)           | \(\{q_1, q_3\}, \{q_2\} \cup \{q_1, q_3\} \cup \{q_4, q_5\}\) |

Definition 2.5 Let \(M = (Q, R, \Sigma, \delta)\) be a RFSA and \(D\) be the set of all definable sets generated by \(R\) over \(Q\). Then the block transition map \(\delta^D : D \times \Sigma \to A\) is defined as follows: \(\forall D' \in D\) and \(\forall a \in \Sigma\),

\[
\delta^D(D', a) = \left(\delta^D(D', a), \overline{\delta^D(D', a)}\right), \quad \text{where}
\]

\[
\overline{\delta^D(D', a)} = \bigcup \{\delta(q, a) : q \in B \subseteq D', B \in Q/R\}
\]

and

\[
\delta^D(D', a) = \bigcup \{\delta(q, a) : q \in B \subseteq D', B \in Q/R\}.
\]
Example 2.6 Consider the RFSA given in Example 2.4. Then the block transitions can be evaluated as:

\[
\begin{align*}
\delta^D(\{q_1, q_3\} \cup \{q_2\}, a) &= \left(\delta^D(\{q_1, q_3\} \cup \{q_2\}, a), \delta^D(\{q_1, q_3\} \cup \{q_2\}, a)\right) \\
&= (\{q_1, q_3\}, \{q_1, q_3\} \cup \{q_2\}), \text{ since} \\
\delta^D(\{q_1, q_3\} \cup \{q_2\}, a) &= \bigcup \left\{ \delta(q, a) : q \in B \subseteq \{q_1, q_3\} \cup \{q_2\} \right\} \\
&= \delta(q_1, a) \cup \delta(q_3, a) \cup \delta(q_2, a) \\
&= \{q_1, q_3\} \text{ and} \\
\delta^D(\{q_1, q_3\} \cup \{q_2\}, a) &= \bigcup \left\{ \delta(q, a) : q \in B \subseteq \{q_1, q_3\} \cup \{q_2\} \right\} \\
&= \delta(q_1, a) \cup \delta(q_3, a) \cup \delta(q_2, a) \\
&= \{q_1, q_3\} \cup \{q_2\}.
\end{align*}
\]

The rest of the block transitions can be computed similarly.

Definition 2.7 Let \( M = (Q, R, \Sigma, \delta) \) be a RFSA. Define \( \delta^* : Q \times \Sigma^* \to A \) as follows: \( \forall q \in Q, \forall x \in \Sigma^*, \text{ and } \forall a \in \Sigma, \)

\[
\begin{align*}
(i) \ \delta^*(q, e) &= ([q], [q]), \forall q \in Q \text{ and} \\
(ii) \ \delta^*(q, xa) &= \left(\delta^*(q, xa), \delta^*(q, x)\right), \text{ where} \\
\delta^*(q, xa) &= \delta^D(\delta^*(q, x), a) \text{ and} \\
\delta^*(q, xa) &= \delta^D(\delta^*(q, x), a).
\end{align*}
\]

Definition 2.8 Let \( M = (Q, R, \Sigma, \delta) \) be a RFSA. Then the block transition map \( \delta^D : D \times \Sigma \to A \) can be extended to a map \( \delta^{*D} : D \times \Sigma^* \to A \) as follows: \( \forall D' \in D \) and \( \forall x \in \Sigma^*, \)

\[
\begin{align*}
\delta^{*D}(D', x) &= \left(\delta^{*D}(D', x), \delta^{*D}(D', x)\right) \text{ where} \\
\delta^{*D}(D', x) &= \bigcup \left\{ \delta^*(q, x) : q \in B \subseteq D', B \in \Sigma^* \right\} \text{ and} \\
\delta^{*D}(D', x) &= \bigcup \left\{ \delta^*(q, x) : q \in B \subseteq D', B \in \Sigma^* \right\}.
\end{align*}
\]

Definition 2.9 A rough finite state machine (or rough finite automaton) is 6-tuple \( M = (Q, R, \Sigma, \delta, I, F) \) where \( (Q, R, \Sigma, \delta) \) is a RFSA, \( I \) is a definable set in \( (Q, R) \), called an initial configuration, and \( F \subseteq Q \) is the set of final states of \( M \).

Definition 2.10 Let \( M = (Q, R, \Sigma, \delta, I, F) \) be a RFA. Then the set of strings definitely accepted (resp. possibly accepted) by \( M \) is denoted by \( \beta_M \) (resp. \( \beta_M^{\prime} \)), and defined by

\[
\beta_M = \left\{ x \in \Sigma^* : \delta(I, x) \cap F \neq \emptyset \right\}
\]

\[
\beta_M^{\prime} = \left\{ x \in \Sigma^* : \delta(I, x) \cap F \neq \emptyset \right\}.
\]

The rough regular language of \( M \), is defined to be the rough set \( L(M) = (\beta_M, \beta_M^{\prime}) \).
Definition 2.11 [2] Let \( V \) be a finite and nonempty set of variables and \( R_V \) an equivalence relation on \( V \). The pair \((V, R_V)\) will be called the approximation space of variables. Let \( T \) be a finite and nonempty set of terminals and \( R_T \) an equivalence relation on \( T \). The pair \((T, R_T)\) will be called the approximation space of terminals. Pair \((P, R_P)\) is the approximation space of productions. Let \((S_N, \overline{S}_N)\) be a rough set in \((V, R_V)\) such that \( S_N \subseteq V_N \) and \( \overline{S}_N \subseteq V_N \).

Note: \( V \) and \( T \) are so chosen that \( V \cap T = \emptyset \).

Definition 2.12 [2] Let \((V_N, \overline{V}_N)\) be a rough set in \((V, R_V)\) and \((T_N, \overline{T}_N)\) be a rough set in \((T, R_T)\). Let for \( p_i, p'_j \in P, i = 1, 2, \cdots k, j = 1, 2, \cdots n \)

\[
P_1 = \bigcup_{i=1}^{k} R_p[p_i] \text{ and } P_2 = \bigcup_{j=1}^{n} R_p[p'_j].
\]

Thus \( P_i (i = 1, 2) \) are union of some of the equivalence classes of \( R_P \).

Let, \( P_1 \) be the restriction of \( P_1 \) to \((V_N \cup T_N)^* \times (V_N \cup T_N)^* \) and \( P_2 \) be the restriction of \( P_2 \) to \((V_N \cup T_N)^* \times (V_N \cup T_N)^* \).

Definition 2.13 [2] A rough grammar (denoted by \( G \)) as \( G = (G, \overline{G}) \) where \( G = (V_N, T_N, P_N, S_N) \) and \( \overline{G} = (\overline{V}_N, \overline{T}_N, \overline{P}_N, \overline{S}_N) \).

Definition 2.14 [2] Let \( L \) be the language generated by \( G \). Then

\[
L = L(G) = \{ w : w \in T_N^* \text{ and for } s \in S_N, s \Rightarrow_G w \}.
\]

Let \( \overline{L} \) be the language generated by \( \overline{G} \). Then

\[
\overline{L} = \overline{L}(
\overline{G}) = \{ w : w \in \overline{T}_N^* \text{ and for } s \in \overline{S}_N, s \Rightarrow_{\overline{G}} w \}.
\]

The language generated by \( G = (G, \overline{G}) \) is denoted by \( L(G) \) and is defined as \( L(G) = (L, \overline{L}) \).

Definition 2.15 A rough grammar \( G = (G, \overline{G}) \) is of type \((i, j)\) if \( G \) is a type \(i\) grammar and \( \overline{G} \) is a type \(j\) grammar. Then \(i, j\) may take the values 0, 1, 2, 3. Accordingly there may be sixteen types of rough grammar.

3. Main Results

In this section, we investigate some closure properties of rough regular languages and establish the equivalence between the classes of rough languages generated by rough grammar and the classes of rough regular languages accepted by rough finite automaton.

Theorem 3.1 If \( L_1 = (L_1, \overline{L}_1) \) and \( L_2 = (L_2, \overline{L}_2) \) be rough regular languages over \( \Sigma \), then \( L_1 \cap L_2 \) is a rough regular language over \( \Sigma \).

Proof. Let \( L_1 \) and \( L_2 \) are rough regular languages recognized by rough finite automata \( M_1 = (Q_1, R_1, \Sigma, \delta_1, I_1, F_1) \) and \( M_2 = (Q_2, R_2, \Sigma, \delta_2, I_2, F_2) \) respectively. Construct a rough finite automaton \( M = (Q, R, \Sigma, \delta, I, F) \) where

\[
Q = Q_1 \times Q_2
\]

\[
I = I_1 \times I_2
\]

\[
F = F_1 \times F_2
\]
For all \((q_1, q_2) \in Q_1 \times Q_2\) and \(x \in \Sigma\), \(\delta : Q \times \Sigma \to A\) is a transition function such that
\[
\delta ((q_1, q_2), x) = \left( \left( \delta_1 ((q_1, x), \delta_2 ((q_2, x)) \right), \left( \delta_1 ((q_1, x), \delta_2 ((q_2, x)) \right) \right).
\]
The relation \(R\) on \(Q = Q_1 \times Q_2\) is defined as \(((p_1, p_2), (q_1, q_2)) \in R\) if \((p_1, q_1) \in R_1\) and \((p_2, q_2) \in R_2\). Then \(L_1(M_1) \cap L_2(M_2) = L(M)\). Therefore \(L_1 \cap L_2\) is a rough regular language.

**Theorem 3.2** If \(L_1 = (L_1, T_1)\) and \(L_2 = (L_2, T_2)\) be rough regular languages over \(\Sigma\), then \(L_1 \cup L_2\) is a rough regular language.

**Proof.** Let \(L_1\) and \(L_2\) are rough regular languages recognized by rough finite automata \(M_1 = (Q_1, R, \Sigma, \delta_1, I_1, F_1)\) and \(M_2 = (Q_2, R, \Sigma, \delta_2, I_2, F_2)\) respectively. Consider the rough finite auton\(M = (Q, R, \Sigma, \delta, I, F)\) where
\[
Q = Q_1 \cup Q_2 \text{ and } Q_1 \cap Q_2 = \emptyset,
I = I_1 \cup I_2,
F = F_1 \cup F_2\text{ and}
\]
For all \((q_1, q_2) \in Q_1 \times Q_2\) and \(x \in \Sigma\), \(\delta : Q \times \Sigma \to A\) is a transition function such that
\[
\delta ((q_1, q_2), x) = \left( \left( \delta_1 ((q_1, x), \delta_2 ((q_2, x)) \right), \left( \delta_1 ((q_1, x), \delta_2 ((q_2, x)) \right) \right).
\]
Then \(L_1(M_1) \cup L_2(M_2) = L(M)\). Therefore \(L_1 \cup L_2\) is a rough regular language.

**Theorem 3.3** If \(L = (L, T)\) is a rough regular language recognized by rough finite automaton \(M\), then there exists a \((3, 3)\)-rough grammar \(G = (G, \bar{G})\) such that \(L = L(G)\).

**Proof.** Let \(M = (Q, R, \Sigma, \delta, I, F)\) be a rough finite automaton, that recognizes a rough regular language \(L\). Construct the \((3, 3)\)-rough grammar \(G = (G, \bar{G})\) where \(G = (V_N, T_N, P_N, S_N)\) and \(\bar{G} = (\overline{V_N}, \overline{T_N}, \overline{P_N}, \overline{S_N})\). Let
\[
V_N = \overline{V_N} = V_N = Q,
T_N = \overline{T_N} = T_N = \Sigma\text{ and}
S_N = \overline{S_N} = \{I\}.
\]
For each rough transition \(\delta(q, a) = (A, \overline{A})\) being a rough set in \((Q, R)\) for some \(A \subseteq Q\). Then the production
\[
P_1 = \{ q \to aA \} \cup \{ q \to a : A \cap F \neq \emptyset \} \text{ and}
\]
\[
P_2 = \{ q \to a\overline{A} \} \cup \{ q \to a : \overline{A} \cap F \neq \emptyset \}.
\]
We can easily see that a rough regular language \(L\) recognized by rough finite automaton \(M\) is equal \(L(G)\).

**Theorem 3.4** Let \(G = (G, \bar{G})\) be a \((3, 3)\)-rough grammar, then there is a rough finite automaton \(M\) such that \(L(M) = L(G)\).

**Proof.** Let \(L(G)\) be a rough language generated by \((3, 3)\)-rough grammar \(G = (G, \bar{G})\). We shall define a rough finite automaton \(M\) to accept \(L(G)\). Consider a rough finite automaton \(M = (Q, R, \Sigma, \delta, I, F)\) where
\[
Q = V_N \cup \{ q \}, q \notin V_N,
I = S_N \text{ and}
F = \{ q \} \cup \{ A : A \in V_N \text{ and } A \rightarrow \lambda \text{ is in } P_2 \}.
\]
The rough transition \(\delta\) is defined as follows:
if \(q \to aA\) is a rule in \(P_1\) at \(G\) and \(q \rightarrow \overline{aA}\) is a rule in \(P_2\) at \(G\), then \(\delta(q, a) = (A, \overline{A})\). From the construction of rough finite automaton we can prove that corresponding to derivation of a word \(u\) in \(G\), there is a sequence of moves of \(M\) which accepts \(u\). Conversely, corresponding to a sequence of moves of \(M\) which accepts \(u\), there is a derivation of \(u\) in \(G\). Hence \(L(G) = L(M)\).
References

[1] S. Basu, Rough grammar and rough language, *Foundations of Computing and Decision Sciences*, 28 (2003), 129-140.
[2] S. Basu, Rough finite state automata, *Cybernetics and Systems*, 36 (2005), 107-124.
[3] Gheorghe Paun, Lech Polkowski and Andrzej Skowron, Rough set approximations of languages, *Fundamenta Informaticae*, 33(1998), 1-165.
[4] Z. Pawlak, Rough sets, *International Journal of Computer and Information Sciences*, 11 (1982), 341-356.
[5] W.G. Wee, On generalizations of adaptive algorithm and application of the fuzzy sets concept to pattern classification, Ph.D. Thesis, Purdue University, 1967.
[6] L.A. Zadeh, Fuzzy sets, *Information and Control*, 8 (1965), 338-353.