7 Appendix: Proofs

7.1 Proof of Theorem 1

Proof. For a SOREFC r, if there exists an SFA($\&, \#$) $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(r)$, then, for the ith subexpression of the form $r_i = r_{i_1} \& r_{i_2} \& \cdots \& r_{i_k}$ $(i, k \in \mathbb{N}, k \geq 2)$ in r, there is a shuffle marker $\&_i$ in the SFA($\&, \#$) $\mathcal{A}$ for starting to recognize the strings derived by r_i. For each subexpression r_{i_j} $(1 \leq j \leq k)$ in r_i, there is a concurrent marker $||_{ij}$ in the SFA($\&, \#$) $\mathcal{A}$ for starting to recognize the symbols or strings derived by r_{i_j}.

For the SOREFC r, the sum of the number of binary operators is at most $|\Sigma| - 1$. Suppose that the number of the operators $\&$ is n_2, the number of the other binary operators is n_2'. Then, there is $n_2 + n_2' = |\Sigma| - 1$. In worst case, for the syntax tree T of r, and each node v in T labelled by $\&$, the parent node of v is labelled by other binary operator. Then, there is $n_2' = n_2$, the maximum number of the operator $\&$ is $\lceil \frac{|\Sigma|-1}{2} \rceil$. For each node v in T labelled by $\&$, since the parent node of v is labelled by other binary operator, SFA($\&, \#$) $\mathcal{A}$ can have a corresponding shuffle mark $\&_i$ for the ith operator $\&$ in r. Thus, SFA($\&, \#$) $\mathcal{A}$ has at most $\lceil \frac{|\Sigma|-1}{2} \rceil$ shuffle markers.

For each subexpression r_{i_j} in r_i, there is a concurrent marker $||_{ij}$ in the SFA($\&, \#$) $\mathcal{A}$ for starting to recognize the symbols or strings derived by r_{i_j}. If the subexpression $r_{i_j} = a \in \Sigma$, then the number of the concurrent markers is not over $|\Sigma|$. Thus, the maximum number of the concurrent markers in SFA($\&, \#$) $\mathcal{A}$ is $|\Sigma|$. SFA($\&, \#$) $\mathcal{A}$ has at most $|\Sigma|$ concurrent markers.

For syntax tree T of r, the number of leaf node v_0 is $|\Sigma|$. Since the sum of the number of binary operators is at most $|\Sigma| - 1$, the number of the node v_1 with two childs in T is at most $|\Sigma| - 1$. For each node of v_0 and v_1, suppose that the parent node is labelled by a counting operator, then the number of the nodes labelled by counting operators is at most $2|\Sigma| - 1$. Since a SOREFC forbids immediately nested counters, for each node v_c in T labelled by counting operator, the parent node of v_c is not labelled by another counting operator if v_c has not siblings. Then, there are at most $2|\Sigma| - 1$ counting operators in r. Since a loop marker can be used to mark a counter (see the corresponding definition in Section 3.1), SFA($\&, \#$) $\mathcal{A}$ has at most $2|\Sigma| - 1$ loop markers.

7.2 Proof of Theorem 2

Proof. An SFA($\&, \#$) recognizes a string by treating symbols in a string individually. A symbol y in a string s is recognized if and only if the current state p is reached such that $y \in p$. The end symbol $\dashv$ is added into the end of the string s. Let p_y denote the state (a set of nodes) p including symbol y. The next symbol of y in the string $s \dashv$ is read if and only if y has been recognized at the state p_y. H is the node transition graph of an SFA($\&, \#$) $\mathcal{A}$. The number of nodes in H is at most $3.5|\Sigma|$ (including q_0 and q_f). Assume that the current symbol y is read and the current state is q:

1. q is a set: $|q| \geq 1$ and $\exists v \in \{||_{ij}\}_{i \in \mathbb{D}_\Sigma, j \in \mathbb{P}_\Sigma} \cup \Sigma \cup \{+_k\}_{k \in \mathbb{B}_\Sigma} : v \in q \wedge y \in H. \succ (v)$.

   The state q ($q \notin \{q_0, q_f\}$) includes at most $3.5|\Sigma|$ nodes. For an SFA$(\&, \#)$, it takes $\mathcal{O}(|\Sigma|)$ time to search the node v in q, and it takes $\mathcal{O}(H.V) = \mathcal{O}(|\Sigma|)$ time to decide whether $y \in H. \succ (v)$. If $y \in H. \succ (v)$, then SFA$(\&, \#)$ $\mathcal{A}$ will reach the state $p_y = q \backslash \{v\} \cup \{y\}$, y will be recognized. Hence, for the current state q, SFA$(\&, \#)$ $\mathcal{A}$ takes $\mathcal{O}(|\Sigma|^2)$ time to decide whether y can be recognized.

2. q is a set: $|q| \geq 1$ and $\exists v \in \{\&_i, +_k\}_{i \in \mathbb{D}_\Sigma, k \in \mathbb{B}_\Sigma} \wedge v \in q : y \in H.R(v)$.
   For an SFA$(\&, \#)$, it takes $\mathcal{O}(|\Sigma|)$ time to search the node v in q, and it takes $\mathcal{O}(|\Sigma|)$ time to decide whether $y \in H.R(v)$. If $y \in H.R(v)$, then the state q will transit to state $q' = q \backslash \{\&_i\} \cup H. \succ (v)$.

   If there exists $i \in \mathbb{D}_\Sigma : v = \&_i$, then the node $||_{ij}$ ($||_{ij} \in H. \succ (\&_i), j \in \mathbb{P}_\Sigma$) will be used to check whether $y \in H. \succ (||_{ij})$. Case (1) will be considered. Thus, for the current state q, SFA$(\&, \#)$ $\mathcal{A}$ takes $\mathcal{O}(|\Sigma|^2)$ time to decide whether y can be recognized.

   If there exists $k \in \mathbb{B}_\Sigma : v = +_k$, SFA$(\&, \#)$ $\mathcal{A}$ takes $\mathcal{O}(|\Sigma|)$ time to check whether $y \in H. \succ (v)$. Case (2) is still satisfied. However, there are at most $2|\Sigma| - 1$ loop markers. Then, SFA$(\&, \#)$ $\mathcal{A}$ will take $\mathcal{O}(|\Sigma|)$ time to check all loop markers until Case (1) is satisfied. Hence, for the current state q, SFA$(\&, \#)$ $\mathcal{A}$ takes $\mathcal{O}(|\Sigma|^3)$ time to decide whether y can be recognized.

3. q is a set: $|q| \geq 1$, $\exists W \subseteq q$ and $\exists i \in \mathbb{D}_\Sigma \forall x \in W : x \in H. \prec (\&_i) \wedge |W| = |H. \succ (\&_i)|$: $y \in H.R(\&_i)$.
   SFA$(\&, \#)$ $\mathcal{A}$ takes $\mathcal{O}(|\Sigma|^2)$ time to search the set W from q. If $y \in H.R(\&_i)$, then Case (2) is considered. Otherwise, the current state q will transit to state $q' = q \backslash W \cup \{z | z \in H. \succ (w), w \in W\}$. SFA$(\&, \#)$ $\mathcal{A}$ takes $\mathcal{O}(|\Sigma|)$ time to check whether $y \in q'$ or $q' = q_f$. Hence, for the current state q, SFA$(\&, \#)$ $\mathcal{A}$ takes $\mathcal{O}(|\Sigma|^3)$ to decide whether y can be recognized.

4. $q = q_0$.
   If $y \in H. \succ (q_0)$, then SFA$(\&, \#)$ $\mathcal{A}$ takes $\mathcal{O}(|\Sigma|)$ time to transit to the state $\{y\}$. Otherwise, in the node transition graph of SFA$(\&, \#)$ $\mathcal{A}$, the node $v \in \{\&_i, +_k\}_{i \in \mathbb{D}_\Sigma, k \in \mathbb{B}_\Sigma}$ is searched and checked whether $y \in H.R(v)$. Then, for the current state q, SFA$(\&, \#)$ $\mathcal{A}$ takes $\mathcal{O}(|\Sigma|^2)$ time to transit to state $\{\&_i\}$ or state $\{+_k\}$. Case (2) is satisfied. Hence, for the current state q, SFA$(\&, \#)$ $\mathcal{A}$ takes $\mathcal{O}(|\Sigma|^3)$ time to decide whether y can be recognized.

For the current symbol $y \in \Sigma_s$ and the current state q, SFA$(\&, \#)$ $\mathcal{A}$ takes $\mathcal{O}(|\Sigma|^3)$ time to decide whether y can be recognized. When the last symbol of s is recognized, the end symbol $\dashv$ needs to be recognized. SFA$(\&, \#)$ $\mathcal{A}$ takes $\mathcal{O}(|H.V|) = \mathcal{O}(|\Sigma|)$ time to decide whether the state q can transit to the final state q_f. For the current state q and the current symbol $y \in \Sigma_s$, in SFA$(\&, \#)$, there are at most $|\Sigma|$ states which are used to check whether those states include the node y. Thus, for SFA$(\&, \#)$ $\mathcal{A}$, it takes $\mathcal{O}(|s||\Sigma|^3)$ time to decide whether s can be recognized.

7.3 Proof of Theorem 3

Proof. For a given SOREFC r, SFA($\&, \#$) $\mathcal{A}$ is constructed by using algorithm $ConsEquSFA^{\#}_{\&}$. First, by traversing the syntax tree of r, we find all subexpressions with shuffle operator, and replace them by the corresponding subexpressions with shuffle markers and concurrent markers. We also find all subexpressions with counting, and replace them by the corresponding subexpressions with loop markers. Let r' denote the final obtained expression, where the number of all markers is at most $3.5|\Sigma|$. r' can be obtained in $\mathcal{O}(3.5|\Sigma||r|)$ ($|r| > |\Sigma|$) time (in algorithm 2, in line 12). Then, it takes $\mathcal{O}(|r'| + |G.E|)$ ($|r'| < 3.5|\Sigma| + |r|$) time to construct an equivalent Glushkov automaton $G(V, E)$ for r' [10], and it takes $\mathcal{O}(|G.V|)$ ($|r'| > |G.V|$) time to transform G into the node transition graph of SFA($\&, \#$). Finally, SFA($\&, \#$) $\mathcal{A}$ is obtained. Since G includes $3.5|\Sigma| + |\Sigma|$ nodes at most, there is $|G.E| < (3.5|\Sigma| + |\Sigma|)^2 = \frac{81}{4}|\Sigma|^2$. For any given SOREFC r, it takes $\mathcal{O}(3.5|\Sigma||r| + |r'| + |G.E| + |G.V|) = \mathcal{O}(\frac{81}{4}|\Sigma||r|)$ time to construct an SFA($\&, \#$), i.e., the time complexity of algorithm $ConsEquSFA^{\#}_{\&}$ is $\mathcal{O}(\frac{81}{4}|\Sigma||r|)$.

$\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(r)$. In algorithm $ConsEquSFA^{\#}_{\&}$, $\mathcal{A}$ is transformed from the Glushkov automaton G, the node transition graph is obtained by adding node q_f and its corresponding edges in G (see line 13 in algorithm 2). Then, for a string $s \in \mathcal{L}(\mathcal{A})$, there exists a string $s' \in \mathcal{L}(G)$ that the string s can be obtained by removing all shuffle markers, loop markers and concurrent markers from s'. Since G is the Glushkov automaton constructed for r', there is $\mathcal{L}(G) = \mathcal{L}(r')$ [10]. Then, there is $s' \in \mathcal{L}(r')$. r' is obtained by following steps: replacing all subexpressions with shuffle operators by the corresponding subexpressions with shuffle markers and concurrent markers; replacing all subexpressions with counting operators by the corresponding subexpressions with loop markers (in lines 3, 5, 9 and 11 in algorithm 2). Since $s \in \mathcal{L}(\mathcal{A})$, when the string s or a substring in s matches each counter $q_c \in V_c(\mathcal{A}.H)$ in SFA($\&, \#$) $\mathcal{A}$, there exists the range of counting result $[\mathcal{A}.H.T.l(q_c), \mathcal{A}.H.T.u(q_c)]$ lying in $\mathcal{A}.range(q_c)$. Each range in $\mathcal{A}.range$ is obtained from the corresponding counting operator in r, so there is $s \in \mathcal{L}(r)$. Otherwise, $s' \in \mathcal{L}(r')$ and $s \in \mathcal{L}(\mathcal{A})$ will not be hold. Then, there is $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(r)$.

$\mathcal{L}(r) \subseteq \mathcal{L}(\mathcal{A})$. For any string $s \in \mathcal{L}(r)$, there is a string $s' \in \mathcal{L}(r')$. s can be obtained by removing all shuffle markers, loop markers and concurrent markers from s'. Since the Glushkov automaton G is equivalent to r', there is $s' \in \mathcal{L}(G)$. In algorithm $ConsEquSFA^{\#}_{\&}$, since the node transition graph can be obtained by adding node q_f and its corresponding edges in G, and each range in $\mathcal{A}.range$ is obtained from the corresponding counting operator in r, when the string s or a substring in s matches each counter $q_c \in V_c(\mathcal{A}.H)$ in SFA($\&, \#$) $\mathcal{A}$, there exists the range of counting result $[\mathcal{A}.H.T.l(q_c), \mathcal{A}.H.T.u(q_c)]$ lying in $\mathcal{A}.range(q_c)$. So there is $s \in \mathcal{L}(\mathcal{A})$. Otherwise, $s' \in \mathcal{L}(G)$ and $s \in \mathcal{L}(r)$ will not be hold. Then, there is $\mathcal{L}(r) \subseteq \mathcal{L}(\mathcal{A})$.

Hence, $\mathcal{L}(\mathcal{A}) = \mathcal{L}(r)$. For a SOREFC r, an equivalent SFA($\&, \#$) $\mathcal{A}$ can be constructed in $\mathcal{O}(\frac{81}{4}|\Sigma||r|)$ time.