A Proofs of Theorem 1 and Theorem 2

A.1 Proof of Theorem 1

Proof. For a given finite sample S, according to the definition of uCFAs, the state-transition diagram of the learnt uCFA is the same with that of the learnt CFA, and the learnt uCFA and CFA have the same way for string recognition [21]. The learnt CFA is deterministic [21], do does for the learnt uCFA.

The differences between the learnt uCFA and the learnt CFA focus on the functions of counting, however, this does not matter with recognizing a string. Let $\mathcal{A}_u$ and $\mathcal{A}$ denote the learnt uCFA and the learnt CFA, respectively. According to the Proposition 2 in [21], for any string $s \in S$, $s \in \mathcal{L}(\mathcal{A})$ ($\mathcal{L}(\mathcal{A}) \supseteq S$), then $s \in \mathcal{A}_u$. Thus, $\mathcal{L}(\mathcal{A}_u) \supseteq S$

A.2 proof of Theorem 2

Proof. For any finite sample S, an uCFA is first learnt, then is transformed to an uSORE. Let $\mathcal{A}_u$ and $r_{\%}$ denote the learnt uCFA and the learnt uSORE, respectively. According to Theorem 1, $\mathcal{L}(\mathcal{A}_u) \supseteq S$. The learnt uCFA $\mathcal{A}_u$ is transformed to an uSORE $r_{\%}$ by using algorithm $Soa2Sore$, which respects $\mathcal{A}_u$ as an SOA. For an expression r derived from an SOA A, there is $\mathcal{L}(r) \supseteq \mathcal{L}(A)$ (Theorem 27 in [13]). Then, $\mathcal{L}(r_{\%}) \supseteq \mathcal{L}(\mathcal{A}_u)$. Therefore, $\mathcal{L}(r_{\%}) \supseteq S$.