CS 4510: Automata and Complexity **Fall 2019**

Lecture 11: Learning a DFA

October 2, 2019

Lecturer: Santosh Vempala *Scribe: Yanan Wang*

11.1 Review

11.1.1 Definition

Definition 11.1 *A* ***Membership query*** *consists of a string* $w \in \Sigma^*$*. The answer is either yes or no depending on whether* w *is in this language.*

Definition 11.2 ***Equivalence queries*** *take a hypothesis DFA. If the hypothesis DFA is identical to* M*, the output is yes. Otherwise the output is no and a counterexample (A counterexample is a string that is accepted by the hypothesis DFA but not by* M *or vise versa).*

11.1.2 Observation Table

The Angluin's Algorithm maintains a **observation table** which holds the set of candidate states of a DFA S and a set of query strings E. The observation table has two properties as follows.

- Rows are labeled by strings, which are candidate states. And they are prefix closed.
- Columns are query strings, which are suffix closed.

There are two attributes of the observation table.

- **Closed**: $row(s) \in S \implies \forall a \in \Sigma, row(s \cdot a) \in S$.
- **Consistent**: $\forall s_1, s_2, row(s_1) = row(s_2) \implies \forall a \in \Sigma, row(s_1 \cdot a) = row(s_2 \cdot a)$.

The rule for the entries of observation table is defined as:

$$T(s,e) = \begin{cases} 1 & \text{for} \quad s \cdot e \in L \\ 0 & \text{for} \quad s \cdot e \notin L \end{cases}$$

The interpretation of $T(s,e)$ is that $T(s,e)$ is 1 if and only if s concatenated by e is a string in L.

11.2 Angluin's Algorithm

1 Start with $S = E = \{\varepsilon\}$ and $T(\varepsilon), T(0), T(1)$ are given.
2 **if** *the observation table is not consistent, i.e.,* $\exists a \in \Sigma$, *s.t.* $row(s_1) = row(s_2)$, *but* $row(s_1 \cdot a) \neq row(s_2 \cdot a)$, *i.e.,* $\exists e \in E$, *s.t.,* $T(s_1 \cdot a \cdot e) \neq T(s_2 \cdot a \cdot e)$ **then**
3   add $a \cdot e$ and all its suffixes to E, complete the observation table
4 **if** *the observation table is not closed, i.e.,* $\exists t \in S \cdot E$, *s.t.* $row(t) \neq row(s), \forall s \in S$ **then**
5   add t and all its prefixes to S, complete the observation table
6 Repeat $2-5$ until the observation table is consistent and closed.
7 **if** *the observation table is consistent and closed* **then**
8   proposed a DFA based on it **if** *there is an counterexample* x **then**
9   add x and all its prefixes to S, complete table.
10   Repeat $2-7$
11   **else**
12   The proposed DFA is what we want.

Given a closed and consistent observation table, we define a corresponding DFA over alphabet Σ with set of states Q, initial state q_0, set of final states F and transition function δ as followed:

- $Q = \{row(s) : s \in S\}$,
- $q_0 = row(\epsilon)$,
- $F = \{row(s) : s \in S \text{ and } T(s) = 1\}$
- $\delta(rows(s), a) = row(s \cdot a)$

For new DFA M we just constructed, each state represents a unique row in observation matrix. This DFA is a feasible DFA. Furthermore, if other DFAs that is acceptance consistent with M but is not identical to M must have more states than M.

Lemma 11.3 *DFA* $M(T)$ *is consistent with table* T *and* M *is the smallest such DFA.*

Proof: To prove the lemma, we have to prove two things:

1. $\delta(q_0, s) = row(s)$
2. $\delta(q_0, s \cdot e) \in F, \iff T(s \cdot e) = 1$

First, we prove the first part. We conduct induction on $|s|$.
Base: $\delta(q_0, \varepsilon) = row(\varepsilon)$ holds.
We assume 1. holds for $|s| = k$.
For $|s| = k+1$, $\delta(q_0, s) = \delta(q_0, s_1 \cdot a) = \delta(\delta(q_0, s_1), a) = \delta(row(s_1), a) = row(s_1 \cdot a) = row(s), \exists a \in \Sigma, s = s_1 \cdot a$.
Hence, the first part is correct.
We can do the similar induction for the second part and we leave this part for you to complete.

■

From 11.3, we know that any DFA consistent with table T must have distinct states corrosponding to distinct rows of S.

Lemma 11.4 *At most* $n-1$ *conjectures including not closed or not consistent.*

Let's consider the time complexity of Angluin's Algorithm. Given n states, alphabet size k and the longest counterexample m, the time complexity is linear with $m \cdot n \cdot k$, i.e., $poly(n, m, k)$.