Active Learning of Nondeterministic Systems from an ioco Perspective*

Michele Volpato and Jan Tretmans

No Institute Given

Abstract. This document contains the proofs to the theorems of the original paper. Some lemmas are also introduced.

Theorem 1. *Let $q_\delta \in \mathcal{LTS}(L_I, L_U \cup \{\delta\})$ be a valid suspension automaton. Then, there exists a labelled transition system $q \in \mathcal{LTS}(L_I, L_U)$ such that $Straces(q) = traces(q_\delta)$.*

Proof. Follows directly from Theorem 2 in [18].

□

Algorithm 1 Construct Hypothesis $\mathcal{H}$

Input: A closed and consistent observation table (S, E, T).
Output: A labelled transition system $\mathcal{H} = \langle Q, L_I, L_U \cup \{\delta\}, \rightarrow, q_0 \rangle$.

1: $Q = \{row(s) \mid s \in S\}$
2: $q_0 = row(\epsilon)$
3: **for each** $row(s) \in Q$ **do**
4: **for each** $\lambda \in L_I$ **do**
5: add $row(s) \xrightarrow{\lambda} row(s{\cdot}\lambda)$
6: **for each** $\lambda \in (L_U \cup \{\delta\})$ **do**
7: **if** $\lambda \in T(s, \epsilon)$ **then**
8: add $row(s) \xrightarrow{\lambda} row(s{\cdot}\lambda)$

Lemma 1. *Let (S, E, T) be a closed and consistent observation table such that S is prefix closed and let $\mathcal{H}$ be the hypothesis obtained by running Algorithm 1 on (S, E, T), then $\forall s \in (S \cup S{\cdot}L_\delta)$. ($\mathcal{H}$ **after** s) is equal to either $\{row(s)\}$, if $row(s)$ is defined, or $\emptyset$, if not.*

* This research is supported by the Dutch Technology Foundation STW, which is part of the Netherlands Organisation for Scientific Research (NWO), and which is partly funded by the Ministry of Economic Affairs.

Proof. We prove it by induction on the length of s. For length 0, trivially $(\mathcal{H}$ **after** $\epsilon) = \{row(\epsilon)\}$.

Let us assume it is true for any trace of length at most $k \geq 0$. Let $s \in (S \cup S{\cdot}L_\delta)$ be a trace of length $k+1$. We can decompose s into $s_1\lambda$ where s_1 is a trace of length k and $\lambda \in L_\delta$. The trace s_1 must be in S, because, either s is in $S{\cdot}L_\delta$ (thus $s_1 \in S$), or s is in S, which is prefix closed, and thus $s_1 \in S$.

$$\begin{aligned}
(\mathcal{H} \textbf{ after } s) &= (\mathcal{H} \textbf{ after } s_1\lambda) \\
&= ((\mathcal{H} \textbf{ after } s_1) \textbf{ after } \lambda) && \text{property of } \textbf{after} \\
&= (row(s_1) \textbf{ after } \lambda) && \text{induction hypothesis} \\
&= \{row(s_1\lambda)\} && \text{Algorithm 1, lines 5 and 8} \\
&= \{row(s)\}
\end{aligned}$$

If λ is enabled in $row(s_1)$ then $row(s_1\lambda)$ is defined and it is equal to the row of a valid state, because (S, E, T) is closed and consistent. If λ is not enabled in $row(s_1)$ then $row(s_1\lambda)$ is not defined and $(\mathcal{H}$ **after** $s_1\lambda) = \emptyset$ as expected.

□

Theorem 2. *If an observation table (S, E, T) is closed and consistent, S is prefix closed and E is suffix closed, then the hypothesis $\mathcal{H}$, obtained by running Algorithm 1 on it, is* compatible *with the function T, i. e., $\forall s \in (S \cup S{\cdot}L_\delta), \forall e \in E$. $out(\mathcal{H}$ **after** $s{\cdot}e) = T(s{\cdot}e)$.*

Proof. We prove it by induction on the length of e. Consider $e = \epsilon$ and s any element of $(S \cup S{\cdot}L_\delta)$.

$$\begin{aligned}
out(\mathcal{H} \textbf{ after } s{\cdot}e) &= out(\mathcal{H} \textbf{ after } s) \\
&= out(\{row(s)\}) && \text{Lemma 1} \\
&= T(s) && \text{Algorithm 1, line 8} \\
&= T(s{\cdot}e)
\end{aligned}$$

Let us assume it is true for any $e \in E$ of length at most $k \geq 0$. Let $e' \in E$ be a trace of length $k+1$. E is suffix closed, thus we can decompose e' into $a{\cdot}e$ for some $a \in L_\delta$ and some $e \in E$ of length k. The observation table is closed and $s \in (S \cup S{\cdot}L_\delta)$, thus $\exists s_1 \in S$. $row(s) = row(s_1)$. (Note that $s_1 a \in (S \cup S{\cdot}L_\delta)$).

$$\begin{aligned}
out(\mathcal{H} \textbf{ after } s{\cdot}e') &= out(\mathcal{H} \textbf{ after } sa{\cdot}e) \\
&= out((\mathcal{H} \textbf{ after } s) \textbf{ after } a{\cdot}e) && \text{property of } \textbf{after} \\
&= out(\{row(s)\} \textbf{ after } a{\cdot}e) && \text{Lemma 1} \\
&= out(\{row(s_1)\} \textbf{ after } a{\cdot}e) && row(s) = row(s_1) \\
&= out((\mathcal{H} \textbf{ after } s_1) \textbf{ after } a{\cdot}e) && \text{Lemma 1} \\
&= out(\mathcal{H} \textbf{ after } s_1 a{\cdot}e) && \text{property of } \textbf{after} \\
&= T(s_1 a{\cdot}e) && \text{induction hypothesis} \\
&= T(s_1{\cdot}e') \\
&= T(s{\cdot}e') && row(s) = row(s_1)
\end{aligned}$$

□

Theorem 3. *Let (S, E, T) be a closed and consistent observation table such that S is prefix closed and E is suffix closed, and let $\mathcal{H} = \langle Q, L_I, L_U \cup \{\delta\}, \rightarrow, q_0 \rangle$ be the input-ouput transition system induced by (S, E, T). For any deterministic input-ouput transition system $\mathcal{H}' = \langle Q', L_I, L_U \cup \{\delta\}, \rightarrow', q'_0 \rangle$ compatible with T, $|Q'| \geq |Q|$.*

Proof. Similarly to the function $row(\bullet)$ for observation tables, define, for all $q' \in Q'$, $row(q')$ as the finite function f, from E to $2^{(L_U \cup \{\delta\})}$ such that $f(e) = \{x \in (L_U \cup \{\delta\}) \mid \exists q'' \in Q' \,.\, q' \stackrel{ex}{\Rightarrow} q''\}$.

$\mathcal{H}'$ is consistent with T, thus $\forall s \in (S \cup S{\cdot}L_\delta), \forall e \in E \,.\, out(\mathcal{H}' \textbf{ after } s{\cdot}e) = T(s{\cdot}e)$. Let $(\mathcal{H}' \textbf{ after } s) = \{q'\}$, then $out(\mathcal{H}' \textbf{ after } s{\cdot}e) = out(\{q'\} \textbf{ after } e) = T(s{\cdot}e)$, and finally $row(q') = row(s)$. Since $row(s)$ ranges over all elements of S then $row(q')$ must ranges over all elements of Q and thus $\mathcal{H}'$ must have at least as many states as $\mathcal{H}$.

□

Algorithm 2 Close (S, E, T)

Input: An observation table (S, E, T).
Output: A closed observation table.

1: **while** $\exists s_1 \in (S{\cdot}L_\delta)$ such that $T(s_1{\cdot}\epsilon) \neq \emptyset$ and $\forall s \in S, row(s_1) \neq row(s)$ **do**
2: $\quad S \leftarrow S \cup \{s_1\}$
3: $\quad$ Complete (S, E, T) by asking membership queries
4: **return** the updated observation table (S, E, T)

Algorithm 3 Analyse counterexample c

Input: An observation table (S, E, T) and a counterexample c.
Output: A suffix-closed set E'.

1: Decompose c in $s{\cdot}v$ where
$\quad s$ is the longest prefix of c such that $s \in (S \cup S{\cdot}L_\delta)$
2: **return** $\{v' \in L_\delta^* \mid v' \text{ is a suffix of } v\}$

Theorem 4. *The observation table obtained by adding the result of Algorithm 3, executed on a closed and consistent observation table (S, E, T) and a counterexample c, to the set of suffixes E is not closed.*

Proof. Let $\mathcal{H}$ be the hypothesis induced by (S, E, T). c is a counterexample, thus $out(\mathcal{H} \textbf{ after } c) \neq out(\text{SUL} \textbf{ after } c)$. This means that c can be decompose

into a prefix c_p a label λ and a suffix c_s such that (SUL **after** $[c_p]_{\mathcal{H}}\lambda{\cdot}c_s$) $\neq$ (SUL **after** $[c_p\lambda]_{\mathcal{H}}{\cdot}c_s$). Note that $[c_p]_{\mathcal{H}}\lambda \notin S$ otherwise the previous inequality could not be true because of $[c_p]_{\mathcal{H}}\lambda = [c_p\lambda]_{\mathcal{H}}$. This implies $c_p\lambda \notin S$, thus $c_p\lambda$ cannot be shorter than the longest prefix s of c such that $s \in (S \cup S{\cdot}L_\delta)$ (Line 1 of Algorithm 3).

Line 2 adds to E the suffix closure of v, where v is the suffix of c related to the prefix s. This step adds c_s to E because c_s must is a suffix of v (being c_s shorter or equal to v and being both suffixes of c). After completing the table with membership queries, $T([c_p]_{\mathcal{H}}\lambda{\cdot}c_s) \neq T([c_p\lambda]_{\mathcal{H}}{\cdot}c_s$, but $[c_p\lambda]_{\mathcal{H}}$ was the (unique) element of S such that $row([c_p\lambda]_{\mathcal{H}}) = row([c_p]_{\mathcal{H}}\lambda$ before completing the table, thus, now, $\nexists s \in S \,.\, row(s) = row([c_p]_{\mathcal{H}}\lambda)$ and the table is not closed.

□

Theorem 5. *Given a closed observation table (S, E, T), the induced hypothesis $\mathcal{H}$ is non-blocking, anomaly-free and stable.*

Proof. We prove each property, one by one.

1. for each $s \in S$, $T(s{\cdot}\epsilon) \neq \emptyset$ (see Section **??**), thus in line 7 of Algorithm 1, the *if* clause is true for all states for at least an output $x \in L_U \cup \{\delta\}$. Thus there exists a state $row(s_1)$ such that $row(s_1) = row(sx)$ for at least an output x (or δ). The hypothesis $\mathcal{H}$ is *non-blocking*.
2. for each $s \in S$, $T(s\delta{\cdot}\epsilon) \subseteq \{\delta\}$ (see Section **??**), thus the only non-input transition leaving the state reached by executing $s\delta$ on $\mathcal{H}$, if any, is a δ transition. The hypothesis $\mathcal{H}$ is *anomaly-free*.
3. let $s \in S \,.\, row(s) = q$.
   If $s\delta \in S$ then $s\delta\delta \in (S \cup S{\cdot}L_\delta)$. Furthermore $\forall e \in E \,.\, T(s\delta\delta{\cdot}e) = out(\text{SUL } \textbf{after } s\delta\delta{\cdot}e) = out(\text{SUL } \textbf{after } s\delta{\cdot}e) = T(s\delta{\cdot}e)$ because δ appears only as self-loops in SUL. Thus $row(s\delta\delta) = row(s\delta)$. Thus $q' = q''$. Being represented by the same state in $\mathcal{H}$ (closedness of the observation table), q' and q'' accept the same language, which is even stronger than *stability*.
   If $s\delta \notin S$ then $s\delta \in (S{\cdot}L_\delta)$ and, given the closedness of the table, $\exists s_1 \in S \,.\, row(s\delta) = row(s_1)$ and thus $s_1\delta \in (S \cup S{\cdot}L_\delta)$.

$$
\begin{aligned}
& row(s_1) = row(s\delta) \\
& \Rightarrow T(s_1{\cdot}\epsilon) = \{\delta\} \\
& \Rightarrow out(\text{SUL } \textbf{after } s_1) = \{\delta\} \\
& \Rightarrow \forall \sigma \in L_\delta^* \,.\, out(\text{SUL } \textbf{after } s_1\sigma) \\
& \quad = out(\text{SUL } \textbf{after } s_1\delta\sigma) && \text{property of } \mathcal{LTS} \\
& \Rightarrow \forall e \in E \,.\, out(\text{SUL } \textbf{after } s_1 e) && \text{in particular} \\
& \quad = out(\text{SUL } \textbf{after } s_1\delta e) \\
& \Rightarrow \forall e \in E \,.\, T(s_1\delta{\cdot}e) = T(s_1{\cdot}e)
\end{aligned}
$$

   Thus, as in the previous case, $q' = row(s_1) = row(s_1\delta) = q''$. The hypothesis $\mathcal{H}$ is *stable*.

Algorithm 4 Check for *quiescence reducibility*

Input: An observation table (S, E, T).
Output: A set of suffixes E'.

1: **for each** $s_1 \in S \,.\, \{\delta\} \subsetneq T(s_1 \cdot \epsilon)$ **do**
2: $\quad s_2 \leftarrow s \in S \,.\, row(s) = row(s_1\delta)$
3: $\quad wait \leftarrow \{\langle s_1, s_2, \epsilon\rangle\}$ {*list of state pairs to be checked associated with the sequence of labels needed to reach them from the first pair*}
4: $\quad past \leftarrow \emptyset$ {*list of checked pairs*}
5: $\quad$ **while** $wait \neq \emptyset$ **do**
6: $\quad\quad$ Pick $\langle s_1, s_2, \sigma\rangle \in wait$
7: $\quad\quad$ **for each** $\lambda \in T(s_2 \cdot \epsilon) \cup L_I$ **do**
8: $\quad\quad\quad$ **if** $\lambda \notin T(s_1 \cdot \epsilon) \cup L_I$ **then**
9: $\quad\quad\quad\quad$ **return** $\{\sigma' \in L_\delta^* \mid \sigma'$ is a suffix of $\sigma\}$
10: $\quad\quad\quad$ **else**
11: $\quad\quad\quad\quad s_1' \leftarrow s \in S \,.\, row(s) = row(s_1\lambda)$
12: $\quad\quad\quad\quad s_2' \leftarrow s \in S \,.\, row(s) = row(s_2\lambda)$
13: $\quad\quad\quad\quad$ **if** $\langle s_1', s_2'\rangle \notin past \wedge s_1' \neq s_2'$ **then**
14: $\quad\quad\quad\quad\quad wait \leftarrow wait \cup \{\langle s_1', s_2', \sigma\lambda\rangle\}$
15: $\quad\quad wait \leftarrow wait \setminus \{\langle s_1, s_2, \sigma\rangle\}$
16: $\quad\quad past \leftarrow past \cup \{\langle s_1, s_2\rangle\}$
17: **return** $\emptyset$ {*no counterexample has been found*}

□

Lemma 2. *Let (S, E, T) be an observation table, then $\forall s \in S, T(s \cdot \epsilon) = \{\delta\} \Rightarrow row(s\delta) = row(s)$.*

Proof. First, note that $row(s\delta)$ is well defined, being $s\delta \in (S \cdot L_\delta)$. $T(s \cdot \epsilon) = \{\delta\} \Rightarrow \forall q \in (\text{SUL } \textbf{after } s), out(q) = \{\delta\}$. For each of these states, $q \stackrel{\lambda}{\Rightarrow} q'$ iff $q \stackrel{\delta\lambda}{\Longrightarrow} q'$, for each label λ enabled in q, because δ appears only in loops in SUL. Thus, $\forall e \in E, T(s\delta \cdot e) = out(\text{SUL } \textbf{after } s\delta e) = out(\text{SUL } \textbf{after } se) = T(s \cdot e)$, which implies $row(s\delta) = row(s)$.

□

Theorem 6. *Given a finite input-ouput transition system* SUL*, running Algorithm 5 on* SUL *will terminate with a valid suspension automaton $\mathcal{H}$ whose set of accepted traces is equivalent to the set of suspension traces of* SUL*, i. e.,* $\text{SUL} \in [\mathcal{H}]_\delta$.

Proof. The equivalence is guaranteed by the equivalence query. Furthermore, the algorithm terminates when it reaches Line 17, thus the table is closed (and consistent) and the hypothesis induced by the table is a valid suspension automaton, because of the *repeat* loop from Line 4 to Line 9.

To prove that the algorithm terminates, let be n the number of states of the minimal valid suspension automaton SUL_δ equivalent to SUL. Note that SUL_δ

Algorithm 5 Learn the input-ouput transition system SUL

Input: The set of input labels L_I
Output: A valid suspension automaton $\mathcal{H}$ s.t. SUL $\in [\mathcal{H}]_\delta$.

```
   // Initialize (S, E, T)
 1: S = E = {ε}
 2: Complete (S, E, T) by asking membership queries
   // Start learning
 3: loop
 4:   repeat
 5:     Close (S, E, T) using Algorithm 2
 6:     Check quiescence reducibility with Algorithm 4 on (S, E, T), obtaining a suffix-
        closed set E'
 7:     E ← E ∪ E'
 8:     Complete (S, E, T) by asking membership queries
 9:   until E' = ∅
10:   Construct the hypothesis H using Algorithm 1 on (S, E, T)
11:   Ask an equivalence query for H
12:   if a counterexample c is found then
13:     Analyse counterexample c using Algorithm 3 on (S, E, T) obtaining a suffix-
        closed set E'
14:     E ← E ∪ E'
15:     Complete (S, E, T) by asking membership queries
16:   else
17:     return H
```

is always consistent with the observation table, being T consistent with SUL. We need to show that that the cardinality of S increases strictly monotone up to n. The state S is changed only by Algorithm 2, and only if the observation table is not closed. If the table is not closed, then $|S| < n$, because, otherwise, closing it will end in adding at least one state, obtaining an hypothesis consistent with the table, but having more states than SUL_δ, which cannot be true, for Theorem 3. For every counterexample, found with either an equivalence query or Algorithm 4, the table must be closed again, for Theorem 4, thus there can only be a maximum of $n - 1$ of such counterexamples. Thus the algorithm will eventually reach Line 11 with a closed table such that $|S| = n$, obtaining no counterexample and thus returning the current induced hypothesis.

□