Bachelor's Thesis Computing Science

Radboud University Nijmegen

Sufficient Conditions for k-completeness using Observation Trees and Apartness

---

Author:
Ivo Melse
s1088677

First supervisor/assessor:
Professor Dr. Frits Vaandrager

Second assessor:
Dr. Jurriaan Rot

November 5, 2024

Abstract

We consider the classic problem of black-box conformance testing. Suppose that we have a known *specification* FSM/Mealy Machine S and a hidden *implementation* FSM/Mealy machine $\mathcal{M}$. We can use *k-complete test suites* to test if they are equivalent, under the assumption that $\mathcal{M}$ has at most k more states than S. We build on recent work in conformance testing and the $L^{\#}$ algorithm to prove a new set of sufficient conditions for k-completeness in terms of *Observation trees* and *apartness*.

Acknowledgements

First and foremost, I would like to thank my first supervisor Frits Vaandrager. None of this work would have been possible without his guidance. This thesis builds on his recent work on conformance testing and $L^{\#}$. Secondly, I would like to thank my second assessor Jurriaan Rot, in particular for involving me into the world of $L^{\#}$ and conformance testing in the first place. On a personal level, I would also like to extend my thanks to my family, friends and girlfriend for being supportive when writing a thesis was stressful. Furthermore, a special thanks to my sister for proofreading.

Contents

1 Introduction **5**

2 Preliminaries **7**

2.1 Mealy Machines / FSMs 7

2.2 Apartness and equivalence 9

2.3 Conformance testing 11

2.4 Observation trees 12

2.5 The basis and frontiers 13

2.6 The candidate set 15

3 Research **17**

3.1 Proof for Theorem 3.1 18

3.1.1 Transitions and completeness 18

3.1.2 Transition apartness and non-apartness 19

3.1.3 Shortest witnesses 19

3.1.4 The functional simulation 20

3.1.5 Identification and basis 20

3.1.6 Applied Lemmas 21

3.1.7 The proof itself 22

3.2 Proof for Theorem 3.2 26

3.3 Counterexample to Hypothesis 3.1 28

4 Related Work **31**

4.1 Conformance testing 31

4.1.1 The W-method 31

4.1.2 The HSI-method 32

4.1.3 The ADS-method 33

4.1.4 The $UIOv$-method 33
4.2 Applying Theorem 3.1 34
4.3 Comparison to Vaandrager's Theorem 35
4.3.1 The number of state pairs 36
4.3.2 We definitely need the assumptions 37
4.4 Active automata learning and $L^{\#}$ 38
4.4.1 Active automata learning 38
4.4.2 $L^{\#}$ 39

5 Conclusions & Further work **41**

A Deferred proofs **45**
A.1 Transitions and completeness 45
A.1.1 Same transition 45
A.1.2 Defined words 45
A.1.3 Frontier prefix 46
A.2 Transition apartness and non-apartness 47
A.2.1 Transition apartness 47
A.2.2 Transition non-apartness 48
A.3 Shortest witnesses 49
A.3.1 Shortest witness consistency 49
A.4 The functional simulation 50
A.4.1 Functional simulation consistency 50
A.4.2 Observation tree consistency 51
A.5 Identification and basis 51
A.5.1 Bijective f^B 52
A.5.2 Identification 53
A.5.3 Basis size 54
A.5.4 Identified basis 55
A.6 Applied Lemmas 55
A.6.1 Two functional simulations 55
A.6.2 Applied transition apartness 56
A.6.3 Applied shortest witness consistency 56
A.7 Application of Theorem 3.1 56
A.8 Comparison to Vaandrager's Theorem 57
A.8.1 Equivalent Assumptions 57

A.8.2 Amount of frontier state pairs 58

Chapter 1

Introduction

If we have a known system and a hidden system (black box), can we somehow infer if they display exactly the same behaviour? Can we design a set of tests to show if the systems are equivalent? This is the main question posed in the field of black-box conformance testing.

The problem consists of two Mealy machines (a.k.a. Finite State Machines, FSM): A visible *specification* S and a hidden *implementation* $\mathcal{M}$. We want to design a finite set of test words which we call a test suite T. A test *passes* if the outputs of S and $\mathcal{M}$ are equal for this particular test, otherwise it *fails*. This idea goes back to work by Moore [10].

If it were possible, we would design *complete* test suites. These would be test suites where S and $\mathcal{M}$ are equivalent if and only if every test passes. Intuitively speaking, this is impossible because $\mathcal{M}$ can always have states that 'pretend' to be states in S up to a certain point, and then start behaving differently. Then, even though S and $\mathcal{M}$ are not equivalent, it would take a word of infinite length to show this.

However, we can do something similar to complete test suites by making an additional assumption about S and $\mathcal{M}$. We assume that for some $k \in \mathbb{N}$, $\mathcal{M}$ has *at most* k more states than S. Then we can make a test suite where S and $\mathcal{M}$ are equivalent if and only if every test passes. We call this a k-complete test suite.

Over the years various methods to construct k-complete test suites have been proposed. The first and most well-known of these is without a doubt the W-method by Vasilevskii and Chow [16] [3]. An advantage of the W-method is that its definition is relatively simple. The main downside is that it generates a lot of *redundant* tests. Removing these tests from the test suite would still result in a k-complete test suite. This is why various other test suites build on the W-method in order to decrease the amount of redundant tests. A few prevalent examples are the HSI-method [7] [11], the $UIOv$-method [2] and the ADS-method [6]. We will discuss these in

chapter 4.

A very recent development is a 2024 paper by Vaandrager [14] in which quite a different approach to conformance testing is presented. Instead of considering individual tests, a tree structure called an *Observation tree* is used to store test results. Vaandrager proves that a test suite is k-complete, given a set of assumptions on states in the Observation tree. This approach was mostly inspired by the $L^{\#}$ algorithm [15], and the concept of apartness [5].

In this bachelor thesis, we continue this idea. We show that for a set of slightly different assumptions on states in the observation tree, a test suite is also k-complete. We use a different, more direct approach to proving k-completeness than in previous work. Interestingly enough, it appears that our assumptions do not imply the results of Vaandrager, and neither the reverse.

Our results might help future researchers in designing more efficient test suites. Furthermore, they contribute towards a deeper theoretical understanding about conformance testing. Our results may also help in finding *necessary* conditions for k-completeness. These are conditions where if they do not hold, it follows that a test suite is *not* k-complete.

Apart from our work on k-completeness, we take a closer look at the *identification* property of states in observation trees. The goal was to find clever ways to identify states with the least amount of tests possible.

We provide a brief outline of the future chapters.

2. Preliminaries: We present the necessary background theory needed for our results in sufficient detail. In particular, the concepts of *apartness* and *Observation trees*.

3. Research: We provide a Theorem about k-completeness. Additionally, we provide a counterexample to a hypothesis about identification and prove a related Theorem about identification.

4. Related work: We give an overview of most methods for generating test suites from the literature. We show that the k-completeness of each of these is a Corollary from our Theorem about k-completeness.

5. Conclusion & future work: We discuss the implications of our results and propose applications and future work.

Chapter 2

Preliminaries

This chapter is inspired by the 2022 $L^{\#}$ paper by Vaandrager *et al.* [15] and the 2024 conformance testing paper by Vaandrager [14].

2.1 Mealy Machines / FSMs

A *Mealy Machine*, also known as a *Finite State Machine*, is a state machine where each transition has both an *input symbol* and an *output symbol*. They were first created by George H. Mealy in 1955 [8]. Figure 2.1 shows an example of a simple Mealy Machine called $\mathcal{M}$.

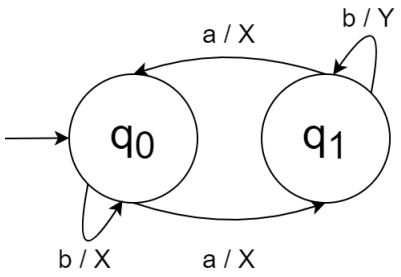

Figure 2.1: Simple example of a Mealy Machine called $\mathcal{M}$

The *input alphabet* of Mealy Machine $\mathcal{M}$ is $I = \{a, b\}$, and the *output alphabet* of Mealy Machine $\mathcal{M}$ is $O = \{X, Y\}$. The set of *states* of $\mathcal{M}$ is $Q^{\mathcal{M}} = \{q_0, q_1\}$.

Transitions are denoted as arrows. Take for example $q_0 \xrightarrow{a/X} q_1$. Here a/X means that for this transition, the input symbol is a and the output symbol is X. q_0 is left of the arrow and q_1 is at the right of the arrow, hence the transition is from state q_0 to q_1.

Formally, transitions are defined by *transition function* δ and *output function* λ, which describe for some state $q \in Q^{\mathcal{M}}$ and some letter $i \in I$, to what state the transition should go, and what should be the output symbol of this transition. For the transition $q_0 \xrightarrow{a/X} q_1$, $\delta(q_0, a) = q_1$, and that $\lambda(q_0, a) = X$.

We extend the domain of these functions to define them for words of length greater than one. Then we simply continue to the next transition, using the next input letter. In Figure 2.1, $\delta(q_0, aa) = \delta(q_1, a) = q_0$ and $\lambda(q_0, aa) = \lambda(q_0, a) \cdot \lambda(q_1, a) = X \cdot X = XX$. Note that we use the symbol $\cdot$ for concatenation, but we leave this implicit most of the time.

Definition 2.1 (Mealy machine)**.** *A (partial)* ***Mealy machine*** $\mathcal{M}$ *is a tuple* $\mathcal{M} = (Q, q_0, \delta, \lambda)$*, where:*

- $Q^{\mathcal{M}}$ *is the finite set of* ***states*** *of* $\mathcal{M}$*.*
- q_0 *is the* ***initial state****.*
- *the (partial)* ***transition function*** $\delta : Q \times I^* \rightharpoonup Q$*, which satisfies the equations:*

$$\begin{aligned} \delta(q, \epsilon) &= q \\ \delta(q, i\sigma) &= \delta(\delta(q, i), \sigma) \end{aligned}$$

- *the (partial)* ***output function*** $\lambda : Q \times I^* \rightharpoonup O$*, which satisfies the equations:*

$$\begin{aligned} \lambda(q, \epsilon) &= \epsilon \\ \lambda(q, i\sigma) &= \lambda(q, i) \cdot \lambda(\delta(q, i), \sigma) \end{aligned}$$

  Note that $\delta(q, i) \downarrow$ *if and only if* $\lambda(q, i) \downarrow$*.*

We might write $\delta(\mathcal{M}, \sigma)$ *or* $\lambda(\mathcal{M}, \sigma)$*. This is equivalent notation to* $\delta(q_0^{\mathcal{M}}, \sigma)$ *or* $\lambda(q_0^{\mathcal{M}}, \sigma)$*.*

As we see in Definition 2.1, for some state $q \in Q$ and some letter $i \in I$, it is not always the case that $\lambda(q, i) \downarrow$. If the state's transition and output functions are defined for all letters, the state is called *complete*. If not it is called *partial*.

We can also extend this notion to Mealy machines themselves, not just their states. We say that a Mealy machine is complete, if all of its states are complete.

Definition 2.2 (Complete and partial)**.** *Let $\mathcal{M}$ be a Mealy machine.*

*A state $q \in Q^{\mathcal{M}}$ is **complete** if $\lambda(q, i) \downarrow$ for all $i \in I$. If q is not complete, then it is **partial**.*

A subset $W \subseteq Q^{\mathcal{M}}$ is complete if $\forall q \in W : q$ is complete. Otherwise W is partial.

$\mathcal{M}$ is complete if $Q^{\mathcal{M}}$ is complete. Otherwise $\mathcal{M}$ is partial.

We discussed transitions before, but now we formally define them. In this bachelor thesis, we will use both transitions and the λ and δ functions. They are equivalent notations and we might implicitly switch between them.

Definition 2.3 (Transition)**.** *Let $\mathcal{M}$ be a Mealy machine with states $q, r, \in Q^{\mathcal{M}}$, $\sigma \in I^*$, and $\chi \in O^*$, and let $n \in \mathbb{N}$, then:*

$$q \xrightarrow[n]{\sigma/\chi} r \iff \delta(q, \sigma) = r \wedge \lambda(q, \sigma) = \chi \wedge |\sigma| = n \wedge |\chi| = n$$

$q \xrightarrow{\sigma/\chi} r$*, then* $n = 1$.
$q \xrightarrow[*]{\sigma/\chi} r$*, then* $n \geqslant 0$
$q \xrightarrow[+]{\sigma/\chi} r$*, then* $n > 0$
We might omit σ, r, or χ if they are irrelevant.

2.2 Apartness and equivalence

The idea of *apartness* in this bachelor thesis is adapted from Vaandrager *et al.* in 2022 [15], although its origins can be traced through Geuvers & Jacobs in 2021 [5] back to Brouwer.

Suppose we have states q and r. Then q and r can be either *apart* or *non-apart*. If they are apart, this means that there is some word $\sigma \in I^*$ where $\lambda(q, \sigma)$ and $\lambda(r, \sigma)$ are both defined, but $\lambda(q, \sigma) \neq \lambda(r, \sigma)$. We then say that σ *witnesses* their apartness. If no there is no such witness they are non-apart.

Figure 2.2 shows partial states q, r and s, along with all their successors. States q and r are apart (denoted $q \# s$). This is shown by *witness* aa. (denoted $aa \vdash q \# s$). The word aa is a witness because $\lambda(q, aa) \downarrow$ and $\lambda(s, aa) \downarrow$ and $\lambda(q, aa) = XX \neq XY = \lambda(s, aa)$.

To the contrary, for q and r there is no such witness that tells them apart. Then q and r are non-apart, denoted by $q \not\# r$.

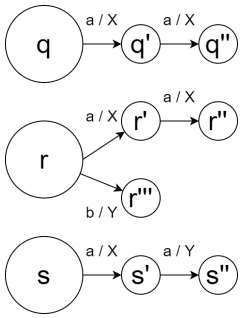

Figure 2.2: States q, r and s that demonstrate apartness.

Definition 2.4 (Apartness)**.** *Let $\mathcal{M}$ and $\mathcal{N}$ be (possibly equal, possibly partial) Mealy machines and $q \in Q^{\mathcal{M}}$ and $r \in Q^{\mathcal{N}}$:*

- *If there exists a word $\sigma \in I^*$, such that $\lambda(q, \sigma) \downarrow$ and $\lambda(r, \sigma) \downarrow$ and $\lambda(q, \sigma) \neq \lambda(r, \sigma)$, then q and r are **apart**, denoted $q \# r$. Moreover, we say that σ **witnesses** that q and r are apart, denoted $\sigma \vdash q \# r$. σ is called a **witness** or **separating sequence**.*
- *If q and r are not apart, then $\forall \sigma \in I^*, \lambda(q, \sigma) \downarrow \wedge \lambda(r, \sigma) \downarrow : \lambda(q, \sigma) = \lambda(r, \sigma)$. We say that states q and r are **non-apart**, denoted $q \not\# r$.*

Note that in most testing literature a *witness* is known as a *separating sequence*.

There is a useful property of apartness, called *Weak co-transivity*. For the proof, we refer to Vaandrager *et al.* in 2022 [15].

Lemma 2.1 (Weak co-transivity, Lemma 2.8 in [15])**.** *In every Mealy machine $\mathcal{M}$,*

$$\sigma \vdash r \# s \wedge \delta(q, \sigma) \downarrow \Longrightarrow r \# q \vee s \# q$$

We define a notion that is similar to apartness and non-apartness but it works on complete Mealy machines instead of states, namely *Mealy machine equivalence*.

Definition 2.5. *Let $\mathcal{M}$ and $\mathcal{N}$ be complete Mealy machines:*

- *$\mathcal{M}$ and $\mathcal{N}$ are **equivalent**, if for all $\sigma \in I^* : \lambda(\mathcal{M}, \sigma) = \lambda(\mathcal{N}, \sigma)$. We denote this as $\mathcal{M} \approx \mathcal{N}$.*
- *If $\mathcal{M}$ and $\mathcal{N}$ are not equivalent, then $\exists \sigma \in I^* : \lambda(\mathcal{M}, \sigma) \neq \lambda(\mathcal{N}, \sigma)$ We say that $\mathcal{N}$ and $\mathcal{M}$ are **inequivalent**. We denote this as $\mathcal{M} \not\approx \mathcal{N}$.*

2.3 Conformance testing

In order to explain conformance testing, we first need to define the *specification* and the *implementation*. The fixed *specification* S is some Mealy machine where we have access to all states and transitions. The fixed *implementation* $\mathcal{M}$ is a Mealy machine for which we do not have access to the inner workings. We can only give input sequences to $\mathcal{M}$ and observe the outputs. We assume that both S and $\mathcal{M}$ are complete, and that S is minimal.

The problem of conformance testing can be stated as follows: we would like to know if S is a correct specification of $\mathcal{M}$. That is, whether $S \approx \mathcal{M}$ or $S \not\approx \mathcal{M}$.

For this purpose, we create a *test suite*. This is a set of words that are fed into both S and $\mathcal{M}$ in order to see whether they give the same output. We would like a test suite to be *complete*, which means that $S \approx \mathcal{M}$, if and only if all tests in the test suite get the same output from both S and $\mathcal{M}$.

Definition 2.6 (Test suite)**.** *A* ***test suite*** *is a finite subset* $T \subset I^*$*.*

Test suites are always finite. As it turns out, this means that a test suite can never be complete. This is illustrated by Figure 2.3. It takes a word of length greater than $n \in \mathbb{N}$ to show that $S \approx \mathcal{M}$, but n can be arbitrarily large.

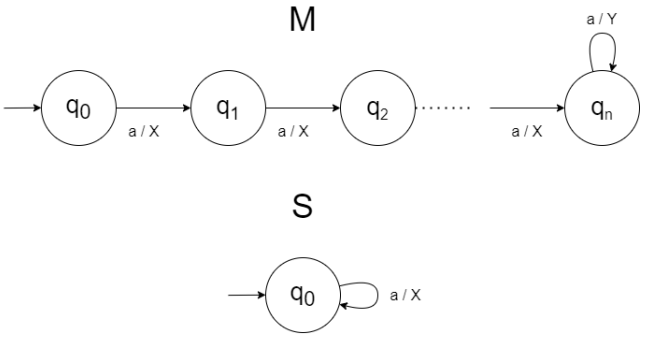

Figure 2.3: Example that demonstrates that no test suite is complete

To resolve this, we have to resort to making an additional assumption about the implementation. Namely that $\mathcal{M}$ has at most $k \in \mathbb{N}$ more states than S, i.e. $|Q^{\mathcal{M}}| \leqslant |Q^S| + k$. If we work under this assumption, we are doing *k-conformance testing*, with test suites that are *k-complete*.

Definition 2.7 (k-complete test suite)**.** *A* ***test suite*** T *is* ***k-complete****, if and only if* $|Q^{\mathcal{M}}| \leqslant |Q^S| + k$ *and* $\forall \sigma \in T, \lambda(S, \sigma) = \lambda(\mathcal{M}, \sigma)$ *implies that* $S \approx \mathcal{M}$*.*

As aforementioned, there exists various methods to generate test suites that are k-complete.

2.4 Observation trees

We might think of test suites not as a set of individual tests, but and *Observation tree*. This idea is by no means new, going back at least to Simão *et al.* in 2012 [12]. The basic idea is that the information obtained from each test is stored in a Mealy machine that takes the form of a tree.

In graph theory, a tree is a graph where there is exactly one path to each state. We apply a similar idea to Mealy machines: If for each state in Mealy machine $\mathcal{T}$, there is exactly one word that reaches this state, then $\mathcal{T}$ is a tree. Figure 2.4 (left, middle) shows two examples of Trees. One of these also happens to be an observation tree.

Definition 2.8 (Tree and access sequence)**.** *Let $\mathcal{T}$ be a Mealy Machine, and let state $q \in Q^{\mathcal{T}}$. An **access sequence** of q, denoted as access(q) is a word in I^* such that $\delta(\mathcal{T}, \sigma) = q$.*

*A **Tree** $\mathcal{T}$ is a partial Mealy machine for which it holds that for each state $q \in Q^{\mathcal{T}}$, there exists exactly one **access sequence**.*

Like aforementioned, we would like to use a tree structure to store the information that we have obtained from each test. Suppose that we have a (hidden) implementation $\mathcal{M}$ and a test suite T. For each test word $\sigma \in T$, we take the output in $\mathcal{M}$, $\lambda(q_0^{\mathcal{M}}, \sigma)$. Then we store this output information in an Observation tree.

Figure 2.4 shows examples of (Observation) trees. We see an example $\mathcal{T}_1$ (left) of an observation tree obtained by running $T = \{a, bab\}$ on $\mathcal{M}$ (right). Indeed, for all words σ in the test suite T, $\lambda(\mathcal{T}_1, \sigma)$ is defined and it matches the output from $\mathcal{M}$. We also see a non-example $\mathcal{T}_2$ (middle). There, we have one transition $t_0 \xrightarrow{b/Y} t_2$ that does not match $\mathcal{M}$ (right).

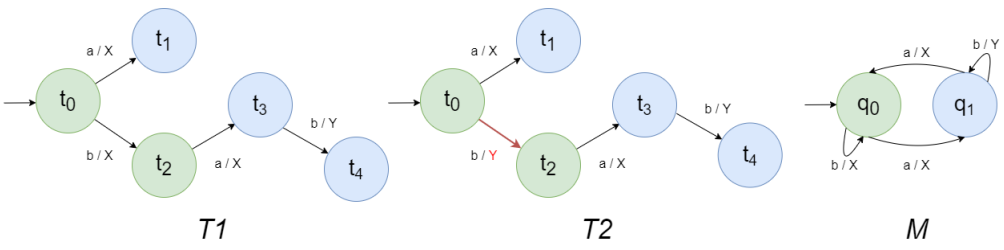

Figure 2.4: Example $\mathcal{T}_1$ (left) of an observation tree for $\mathcal{M}$ (right), with $T = \{a, bab\}$ on $\mathcal{M}$. $\mathcal{T}_2$ (middle) is a Tree, but not an observation tree for $\mathcal{M}$.

The states in $\mathcal{T}_1$ and $\mathcal{M}$ are colored. This coloring has a purpose: A word leads to a blue state in $\mathcal{T}_1$ if and only if it leads to a blue state in $\mathcal{M}$. The same holds for green states.

The formal definition of an observation tree is based on this idea. We have a function $f : Q^{\mathcal{T}_1} \to Q^{\mathcal{M}}$ where $f(t_0) = f(t_2) = q_0$ (green), and $f(t_1) = f(t_3) = f(t_4) = q_1$ (blue). We call this a *functional simulation.*

Definition 2.9 (Functional simulation)**.** *A* ***Functional simulation*** $f : \mathcal{T} \to \mathcal{M}$ *between Mealy Machines* $\mathcal{T}$ *and* $\mathcal{M}$ *is defined as a map* $f : Q^{\mathcal{T}} \to Q^{\mathcal{M}}$ *where* $f(q_0^{\mathcal{T}}) = q_0^{\mathcal{M}}$*, and* $q \xrightarrow{i/o} q'$ *implies* $f(q) \xrightarrow{i/o} f(q')$*.*

Then $\mathcal{T}$ simply is an observation tree for $\mathcal{M}$ if such a functional simulation exists.

Definition 2.10 (Observation tree)**.** $\mathcal{T}$ *is an* ***observation tree*** *for* $\mathcal{M}$ *if* $\mathcal{T}$ *is a tree, and there exists some functional simulation* $f : \mathcal{T} \to \mathcal{M}$*.*

There is a Lemma on functional simulations that we will use in chapter 3 later on.

Lemma 2.2 (Lemma 2.7 in [15])**.** *For a functional simulation* $f : Q^{\mathcal{T}} \to Q^{\mathcal{M}}$ *and* $q, r \in Q^{\mathcal{T}}$ *:*

$$q \# r \implies f(q) \# f(r)$$

We fix $\mathcal{T}$ as an observation tree for both S and $\mathcal{M}$.

2.5 The basis and frontiers

In this section, we will be using the notion of apartness (and non-apartness) to divide the states of $\mathcal{T}$ into a number of disjoint sets: the *basis* and the *frontiers.*

We start by explaining the intuition. Consider the colored states in Figure 2.4 again. One may think of the basis B as a subset of $Q^{\mathcal{T}}$ where every color appears at least once, and for every state $q \in B \subseteq Q^{\mathcal{T}}$, the predecessor of q is also in B.

Figure 2.5 shows an example of an observation tree with a basis and frontiers. Note that for this particular combination of $\mathcal{T}$ and S, there are multiple possibilities for the basis. In this example we have $B = \{t_0, t_1\}$, but $B = \{t_0, t_1, t_2\}$ is also an option.

The 0-*level frontier* F^0 is then the set of successors of B which are not in B themselves. F^1 is the set of successors of F^0, F^2 is the set of successors of F^1, etc.

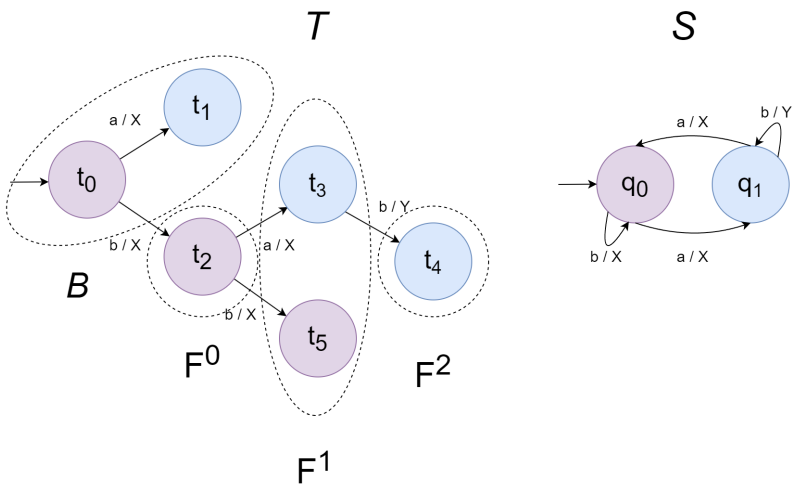

Figure 2.5: An observation tree $\mathcal{T}$ (left) for specification S (right). The basis set B of $\mathcal{T}$ is indicated, as well as the frontiers.

Now we formally define the basis using *state covers*. A state cover of a Mealy machine S is a set of words such that all states in S are reached. In Figure 2.5, $\{\epsilon, a\}$ is an example of a state cover for $\mathcal{M}$. A state cover must be finite, as well as *prefix-closed*. The latter means that for every word in the state cover, its prefix must also be present.

Definition 2.11 ((Exact) state cover)**.** *A set of words $W \subseteq I^*$ is **prefix-closed**, if for all $\sigma \in W$, all prefixes of σ are also in W.*

*A **state cover** $A^S \subseteq I^*$ of Mealy machine S is a finite, prefix-closed set such that: for every state $q \in Q^S : \exists \sigma \in A^S, \delta(S, \sigma) = q$.*

*A state cover is **exact** if there are no two words $v, w \in A^S, v \neq w$, such that $\delta(S, v) = \delta(S, w)$.*

Note that a state cover only exists if there are no unreachable states.

We fix A^S as a state cover of S.

Definition 2.12 (Basis)**.** *The **basis** $B \subseteq Q^{\mathcal{T}}$ of a tree $\mathcal{T}$, given a state cover A^S is defined as $B = \{q \in Q^{\mathcal{T}} | access(q) \in A^S\}$.*

Definition 2.13 (k-level frontier)**.** *For $k \in \mathbb{Z}$:*

$$F^k = \begin{cases} \{q' \in Q^{\mathcal{T}} \backslash B \mid \exists q \in B, i \in I : q' = \delta^{\mathcal{T}}(q,i)\} & \text{if } k = 0 \\ \{q' \in Q^{\mathcal{T}} \mid \exists q \in F^{k-1}, i \in I : q' = \delta^{\mathcal{T}}(q,i)\} & \text{if } k > 0 \\ \varnothing & \text{if } k < 0 \end{cases}$$

$$F^{<k} = \bigcup_{i<k} F^i$$

$$F^{\leqslant k} = \bigcup_{i\leqslant k} F^i$$

We fix B to be the basis of $\mathcal{T}$ with state cover A^S, and we fix the k-level frontiers accordingly.

2.6 The candidate set

For each state q in $Q^{\mathcal{T}}$ we define a *candidate set*. This is the set of states in B that are not apart from q.

Definition 2.14 (Candidate set)**.** *For each state in $q \in Q^{\mathcal{T}}$, q has a candidate set $C(q) = \{q' \in B \mid q' \not\# q\}$.*
*If $|C(q)| = 1$, then q is **identified**.*
A subset $W \subset Q^{\mathcal{T}}$ is identified if $\forall q \in W : q$ is identified.

Figure 2.6 is an example for these candidate sets. We have an observation tree $\mathcal{T}$ (left) for Mealy machine $\mathcal{M}$, with basis $B = \{t_0, t_1\}$.

- The only information we have about t_2 is that it responds X to input a. It is neither apart from t_0 nor t_1 because these both $\lambda(t_0, a) = X$ and $\lambda(t_1, a) = X$. Therefore both remain in the candidate set $C_{t_2} = \{t_0, t_1\}$. t_2 is not identified.
- For t_3 we know that $t_3 \# t_0$ because $\lambda(t_0, b) = X \neq \lambda(t_3, b) = Y$. Therefore only t_1 remains in the candidate set $C_{t_3} = \{t_1\}$. t_1 is identified.
- For t_4 we know that $t_4 \# t_1$ because $\lambda(t_1, b) = Y \neq \lambda(t_4, b) = X$. Therefore only t_0 remains in the candidate set $C_{t_4} = \{t_0\}$. t_4 is identified.

Because we have no observations for F^1 in this particular observation tree, it is the case here that for each $q \in F^1$, simply $C(q) = B = \{t_0, t_1\}$.

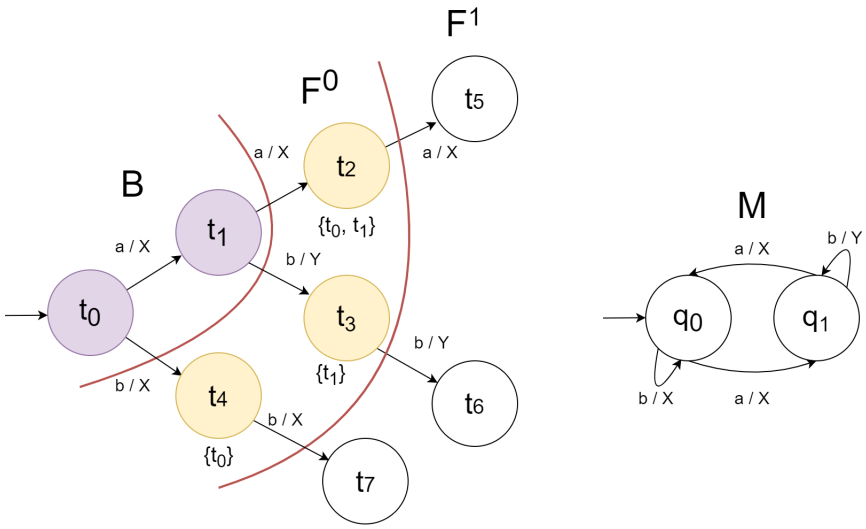

Figure 2.6: Illustration of an observation tree (left) for Mealy machine $\mathcal{M}$ (right). The candidate sets of each state in F^0 are added.

We present another useful Lemma here. If the basis B is identified, then every two non-equal states are apart. The proof has been deferred to Appendix A.5.4.

Lemma 2.3 (Identified basis, A.5.4)**.** *If B is identified, then:*

$$\forall q, r \in B : q = r \vee q \# r$$

Chapter 3

Research

Our most notable result is the k-completeness Theorem.

Theorem 3.1 (k-completeness)**.** *Let $\mathcal{T}$ be an observation tree for both S (with functional simulation f) and for $\mathcal{M}$. Let A^S be a state cover for S. B is a basis set of $\mathcal{T}$, based on A^S. The k-level frontiers $F^0 \dots F^k$ are based on B.*

If $B \cup F^{<k}$ is complete and $B \cup F^{\leqslant k}$ is identified, $|Q^{\mathcal{M}}| \leqslant |Q^S| + k$, and Assumption 3.1 holds:

***Assumption* 3.1.** *In $\mathcal{T}$:*

$$\forall q, r \in F^{\leqslant k} : q \underset{+}{\longrightarrow} r \implies f(q) = f(r) \vee q \# r$$

then:

$$S \approx \mathcal{M}$$

Two less notable results are a counterexample for Hypothesis 3.1 and a proof of Theorem 3.2 which is based on the Hypothesis.

Hypothesis 3.1 (Strong cascading identification)**.** *Let $\mathcal{T}$ be an observation tree for both S (with functional simulation f). Let A^S be a state cover for S. B is a basis set of $\mathcal{T}$, based on A^S. The k-level frontiers $F^0 \dots F^k$ are based on B.*

If $B \cup F^0$ is identified and $F^{\leqslant k}$ is complete, then:

$$F^k \text{ is identified } \implies F^{<k} \text{ is identified}$$

Theorem 3.2 (Cascading identification)**.** *Let $\mathcal{T}$ be an observation tree for both S (with functional simulation f). Let A^S be a state cover for S. B is a basis set of $\mathcal{T}$, based on A^S. The k-level frontiers $F^0 \dots F^k$ are based on B.*

If A^S is exact, $B \cup F^{<k}$ is complete, and Assumption 3.2 holds:

***Assumption* 3.2.**

$$\forall q \in F^{<k} : \forall s \in B : (f(q) = f(s) \vee q\#s) \vee (\forall i \in I : \delta(s, i) \in B)$$

then:

$$F^k \textit{ is identified} \Longrightarrow F^{\leqslant k} \textit{ is identified}$$

We prove Theorem 3.1 in section 3.1. Furthermore, we give a proof for Theorem 3.2 in section 3.2, and a counterexample to Hypothesis 3.1 in section 3.3.

3.1 Proof for Theorem 3.1

In sections 3.1.1-3.1.6, we provide a series of Lemmas which are used in the proof of Theorem 3.1 in section 3.1.7. For each Lemma, a reference of the section of the Appendix with the proof is included in the title.

We avoid restating this boilerplate information in each Lemma.

- If $\mathcal{T}$ is an observation tree for S, we assume that we have functional simulation $f : \mathcal{T} \to S$ and a state cover A^S of S. The basis B for $\mathcal{T}$ is based on A^S and the k-level frontiers $F^0 \dots F^k$ are based on this B.
- If $\mathcal{T}$ is an observation tree for $\mathcal{M}$, we assume that we have functional simulation $g : \mathcal{T} \to \mathcal{M}$.

3.1.1 Transitions and completeness

These Lemmas have in common that they work on transitions and completeness, but have nothing to do with apartness or identification.

Lemma 3.1 (Same transition, A.1.1)**.** *Let $\mathcal{T}$ be a Tree.*
For $q, r, q', r' \in Q^{\mathcal{T}}, v \in I^*$ with $q \xrightarrow[*]{v} q'$ and $r \xrightarrow[*]{v} r'$:

$$q = r \Longleftrightarrow q' = r'$$

Lemma 3.2 relates to the completeness of the frontiers: If $F^{<k}$ is complete, then it follows for any state in F^0 that all words of length less than or equal to k are defined.

Lemma 3.2 (Defined words, A.1.2)**.** *Let $\mathcal{T}$ be an observation tree for S.*
If $F^{<k}$ is complete, then:

$$\forall q \in F^0 : \forall \sigma \in I^* : |\sigma| \leqslant k \Longrightarrow \lambda(q, \sigma) \downarrow$$

Lemma 3.3 is somewhat similar to Lemma 3.2, but it is concerned with the completeness of the basis instead of the frontiers.

Lemma 3.3 (Frontier prefix, A.1.3)**.** *Let $\mathcal{T}$ be an observation tree for S. If B is complete, then $\forall q, \in B, \forall \sigma \in I^*, \lambda(q, \sigma)\uparrow$:*

$$\exists q' \in F^0 : \exists v \in \textit{proper-prefix}(\sigma) : q \xrightarrow[+]{v} q'$$

3.1.2 Transition apartness and non-apartness

We present two interesting properties of apartness, namely *transition apartness* and *transition non-apartness*. We could say that they are opposites.

Lemma 3.4 (Transition apartness, A.2.1)**.** *Let $\mathcal{N}$ be a (partial) Mealy machine.*
For all $q, r \in Q^{\mathcal{N}}$, $v, w \in I^*$:

$$vw \vdash q\#r \iff v \vdash q\#r \vee w \vdash \delta(q, v)\#\delta(r, v)$$

Lemma 3.5 (Transition non-apartness, A.2.2)**.** *Let $\mathcal{N}$ be a (partial) Mealy machine.*
For all $q, r \in Q^{\mathcal{N}}$:

$$(1)\ \forall \sigma \in I^*, \delta(q, \sigma)\downarrow \wedge \delta(r, \sigma)\downarrow : \exists v \in \textit{prefix}(\sigma) : \quad \lambda(q, v) = \lambda(r, v) \wedge \delta(q, v) \not\# \delta(r, v) \implies q \not\# r$$

and

$$(2)\ q \not\# r \implies$$
$$\forall v \in I^*, \delta(q, v)\downarrow \wedge \delta(r, v)\downarrow : \qquad \lambda(q, v) = \lambda(r, v) \wedge \delta(q, v) \not\# \delta(r, v)$$

3.1.3 Shortest witnesses

In our proof for Theorem 3.1, the notion of the *shortest witness* plays a central role.

Definition 3.1 (Shortest witness)**.** *Let $\mathcal{N}$ be a (partial) Mealy machine. A word $\sigma \in I^*$ is the **shortest witness** for states $q, r, \in Q^{\mathcal{N}}$, notated $\sigma \vdash^s q\#r$, if: $\sigma \vdash q\#r \wedge \neg(\exists \sigma' : \sigma' \vdash q\#r \wedge |\sigma'| < |\sigma|)$.*

An interesting property of a shortest witness is *shortest witness consistency*. It is similar to *transition apartness* (Lemma 3.4), which is also used in the proof.

Lemma 3.6 (Shortest witness consistency, A.3.1)**.** *Let $\mathcal{N}$ be a (partial) Mealy machine.*
For all $q, r \in Q^{\mathcal{N}}, \forall v \in I^*, \forall w \in I^+$:

$$vw \vdash^s q \# r \implies w \vdash^s \delta(q, v) \# \delta(r, v)$$

3.1.4 The functional simulation

The following Lemmas are all related to observation trees and their functional simulations.

Lemma 3.7 quite directly follows from the definition of the functional simulation.

Lemma 3.7 (Functional simulation consistency, A.4.1)**.** *Let $\mathcal{T}$ be an observation tree for $\mathcal{N}$ with functional simulation h.*
For all $q, r \in Q^{\mathcal{T}}, v \in I^*$ with $q \xrightarrow[*]{v} r$:

$$h(q) \xrightarrow[*]{v} h(r)$$

Lemma 3.8 is largely based on Lemma 3.7. It states that a state in $\mathcal{T}$ should be non-apart from its corresponding state in $\mathcal{N}$. This also stems from the definition of the functional simulation.

Lemma 3.8 (Observation tree consistency, A.4.2)**.** *Let $\mathcal{T}$ be an observation tree for $\mathcal{N}$ with functional simulation h.*

$$\forall q \in Q^{\mathcal{T}} : q \not\# h(q)$$

3.1.5 Identification and basis

These three Lemmas are all related to identification and the basis B.

Lemma 3.9 (Bijective f^B, A.5.1)**.** *Let $\mathcal{T}$ be an observation tree for S. Let f^B such that $\forall q \in B : f^B(q) = f(q)$.*

$$(1)\ A^S \text{ is exact} \implies f^B : B \to Q^S \text{ is injective}$$
$$(2)\ B \text{ is complete} \implies f^B : B \to Q^S \text{ is surjective}$$

Lemma 3.10 allows us to reason about identification in terms of the functional simulation. This simplifies our proofs since we can abstract away from reasoning about candidate sets.

Lemma 3.10 (Identification, A.5.2)**.** *Let $\mathcal{T}$ be an observation tree for S with B is complete and A^S is exact.*
For $q \in Q^{\mathcal{T}}$:

$$(1)\, q \text{ is identified} \iff \forall s \in B : f(q) = f(s) \vee q \# s$$
$$(2)\, C(q) = \{c_q\} \implies f(q) = f(c_q)$$

Lemma 3.11 (Basis size, A.5.3)**.** *Let $\mathcal{T}$ be an observation tree for S where B is complete.*

$$(1)\, B \text{ is identified} \implies A^S \text{ is exact}$$
$$(2)\, A^S \text{ is exact} \implies |B| = |Q^S|$$

3.1.6 Applied Lemmas

We can state our proof for Theorem 3.1 more concisely if we use weaker versions of some of the previous Lemmas. Lemma 3.12, 3.13 and 3.14 are based on Lemma 3.8, 3.4 and 3.6 respectively.

Lemma 3.12 (Applied observation tree consistency, A.6.1)**.** *Let $\mathcal{T}$ be an observation tree for both S and $\mathcal{M}$.*
For all $q \in Q^{\mathcal{T}}, \sigma \in I^*$:

$$(1)\, \sigma \vdash f(q) \# g(q) \implies \lambda(q_0^{\mathcal{T}}, \sigma) \uparrow$$

and equivalently:

$$(2)\, \lambda(q_0^{\mathcal{T}}, \sigma) \downarrow \implies \neg(\sigma \vdash f(q) \# g(q))$$

Lemma 3.13 (Applied transition apartness, A.15)**.** *Let $\mathcal{T}$ be an observation tree for both S and $\mathcal{M}$.*
For all $q, q' \in Q^{\mathcal{T}}$ and $\forall v, w \in I^*$ where $q \xrightarrow{v} q'$:

$$vw \vdash f(q) \# g(q) \iff v \vdash f(q) \# g(q) \vee w \vdash f(q') \# g(q')$$

Lemma 3.14 (Applied shortest witness consistency, A.16)**.** *Let $\mathcal{T}$ be an observation tree for both S and $\mathcal{M}$.*
For all $q, q' \in Q^{\mathcal{T}}$ and $\forall v \in I*, \forall w \in I^+$ where $q \xrightarrow{v} q'$

$$vw \vdash^s f(q) \# g(q) \implies w \vdash^s f(q') \# g(q')$$

3.1.7 The proof itself

Using the Lemmas presented in sections 3.1.1-3.1.6, we can now prove Theorem 3.1.

Theorem 3.1 (k-completeness)**.** *Let $\mathcal{T}$ be an observation tree for both S (with functional simulation f) and for $\mathcal{M}$. Let A^S be a state cover for S. B is a basis set of $\mathcal{T}$, based on A^S. The k-level frontiers $F^0 \dots F^k$ are based on B.*
If $B \cup F^{<k}$ is complete and $B \cup F^{\leqslant k}$ is identified, $|Q^{\mathcal{M}}| \leqslant |Q^S| + k$, and Assumption 3.1 holds:

> ***Assumption* 3.1.** *In $\mathcal{T}$:*
>
> $$\forall q, r \in F^{\leqslant k} : q \underset{+}{\longrightarrow} r \Longrightarrow f(q) = f(r) \vee q \# r$$

then:

$$S \approx \mathcal{M}$$

Proof. We provide a proof by contra-positive. We show that:

$$B \cup F^{<k} \text{ is complete} \wedge B \cup F^{\leqslant k} \text{ is identified} \wedge \text{ Assumption 3.1 holds} \wedge S \not\approx \mathcal{M} \Longrightarrow |Q^{\mathcal{M}}| > |Q^S| + k$$

We assume everything at the left side of the implication. We first claim that the following then holds:

1. $\exists p \in F^0, \exists w \in I^* : w \vdash^s f(p) \# g(p) \wedge$
   $\neg(\exists q' \in F^0 : \exists w' \in I^* : w' \vdash^s f(q') \# g(q') \wedge |w'| < |w|)$.

Now let $w \in I^*$ and $p \in F^0$ such that these satisfy the conditions of claim 1. (These will be fixed throughout this proof.) We define the following subset P_w of $Q^{\mathcal{T}}$ based on this word w. Intuitively, it includes all frontier states that are *on the path* of this word w.

$$P_w = \{q \in F^{\leqslant k} \mid \exists x \in \text{proper-prefix}(w) : p \underset{*}{\xrightarrow{x}} q\}$$

Now we make three claims about P_w. Each of these is substantiated in its own subsection.

2. $\forall q, r \in P_w, q \neq r : g(q) \neq g(r)$. (Here we use Assumption 3.1).
3. $\forall q \in P_w, \forall r \in B : g(q) \neq g(r)$.
4. $|P_w| = k + 1$.

Now we make yet another claim **A**, which we immediately prove:
$\forall q, r \in B \cup P_w, q \neq r : g(q) \neq g(r)$.

Let states $q, r \in B \cup P_w, q \neq r$: Now we have three cases:

- Case 1, $q, r \in B$: We have that B is identified. Then we have per Lemma 2.3 that $q = r \vee q\#r$. Since we have that $q \neq r$, we get that $q\#r$. Now apply Lemma 2.2 to get that $g(q)\#g(r)$, and therefore $g(q) \neq g(r)$.
- Case 2, $q, r \in P_w$: In this case we get from claim 2 that $g(q) \neq g(r)$.
- Case 3, $q \in P_w, r \in B$: (Note that we can assume this without loss of generality) In this case we get from claim 3 that $g(q) \neq g(r)$.

In each of these cases, we have shown that $g(q) \neq g(r)$. Therefore, **A** holds. Since g is a total function, it follows from **A** that $|Q^{\mathcal{M}}| \geqslant |B \cup P_w|$.

We get from Lemma 3.11 that $|B| = |Q^S|$, and from claim 4, that $|P_w| > k$. Then:

$$|Q^{\mathcal{M}}| \geqslant |B \cup P_w| = |B| + |P_w| = |Q^S| + |P_w| > |Q^S| + k$$

We have now shown that $|Q^{\mathcal{M}}| > |Q^S| + k$. We have proven Theorem 3.1 using a proof by contra-positive. □

Claim 1

$$\exists p \in F^0, \exists w \in I^* : w \vdash^s f(p)\#g(p) \wedge$$
$$\neg(\exists q' \in F^0 : \exists w' \in I^* : w' \vdash^s f(q')\#g(q') \wedge |w'| < |w|)$$

Proof. We have that $S \not\approx \mathcal{M}$. Then it follows by definition that $\exists \sigma \in I^*$, such that $\sigma \vdash q_0^S \# q_0^{\mathcal{M}}$. Let σ be such a witness.

By definition of the functional simulation, it follows that $f(q_0^{\mathcal{T}}) = q_0^S$ and $g(q_0^{\mathcal{T}}) = q_0^{\mathcal{M}}$, So $\sigma \vdash f(q_0^{\mathcal{T}})\#g(q_0^{\mathcal{T}})$. We get from Lemma 3.12 (1) and the fact that $\sigma \vdash f(q_0^{\mathcal{T}})\#g(q_0^{\mathcal{T}})$, that $\lambda(q_0^{\mathcal{T}}, \sigma) \uparrow$.

From the definition of the basis, it follows that $q_0^{\mathcal{T}} \in B$. Now we can apply Lemma 3.3 with $q = q_0^{\mathcal{T}}$, $\sigma = \sigma$. We get that $\exists q \in F^0$ where $\exists v \in$ proper-prefix(σ) such that $q_0^{\mathcal{T}} \xrightarrow{v} q$. Since $v \in$ proper-prefix(σ), let $w \in I^*$ such that $\sigma = vw$.

We have that $\sigma = vw \vdash f(q_0^{\mathcal{T}})\#f(q_0^{\mathcal{T}})$. Now we apply Lemma 3.13 with $q = q_0^{\mathcal{T}}$, $q' = q$ $v = v$, $w = w$. We get that $v \vdash f(q_0^{\mathcal{T}})\#g(q_0^{\mathcal{T}}) \vee w \vdash f(q)\#g(q)$.

We have that $q_0^{\mathcal{T}} \xrightarrow{v} q$ and therefore that $\lambda(q_0^{\mathcal{T}}, v) \downarrow$. Then it follows from Lemma 3.12 (2) that $\neg(v \vdash f(q)\#g(q))$. Then it follows that $w \vdash f(q)\#g(q)$.

Since $f(q)\#g(q)$, it follows that there exists a shortest witness u_q that separates $f(q)$ and $g(q)$, i.e. $\exists u_q \in I^* : u_q \vdash^s f(q)\#g(q)$.

Now we have shown that a state $q \in F^0$ exists for which there exists a shortest witness $u_q \in I^* : u_q \vdash^s f(q)\#g(q)$. This might be true for multiple states. Simply select on of these states p such that there is no other state $q' \in F^0$ where $|u_{q'}| < |u_p|$, and let $w = u_p$.

Then we have proven claim 1. □

Claim 2

$\forall q, r \in P_w, q \neq r : g(q) \neq g(r)$

Proof. By definition of P_w and the fact that $q, r \in P_w$, we have that $\exists x, u \in$ proper-prefix(w), such that $p \xrightarrow[*]{x} q$ and $p \xrightarrow[*]{u} r$. We also have that $q \neq r$. Then we can assume without loss of generality that $|x| < |u|$, and therefore that $\exists y, z \in I^+ : p \xrightarrow[*]{x} q \xrightarrow[+]{y} r$, where $u = xy$ and $w = xyz$. ($y \geqslant 1$ since $|x| < |u| = |xy|$ and $z \geqslant 1$ since $u = xy$ is a proper prefix of $w = xyz$.)

We consider two cases: Either $f(q) = f(r)$ or $f(q) \neq f(r)$. We show that in either of these, $g(q) \neq g(r)$.

Case $f(q) \neq f(r)$:

We have that $q \xrightarrow[+]{y} r$. By Assumption 3.1, then we get that $f(q) = f(r) \vee q\#r$. We already have that $f(q) \neq f(r)$. So it follows that $q\#r$. Then we apply Lemma 2.2 to get that $g(q)\#g(r)$, and therefore that $g(q) \neq g(r)$.

Case $f(q) = f(r)$:

This is a proof by contradiction. Suppose that $g(q) = g(r)$.

Since $f(q) = f(r)$, and $p \xrightarrow[*]{x} q \xrightarrow[+]{y} r$, it follows from Lemma 3.7, that $f(p) \xrightarrow[*]{x} f(q) \xrightarrow[+]{y} f(q) \xrightarrow[*]{z}$.

Likewise, since $g(q) = g(r)$, and $p \xrightarrow[*]{x} q \xrightarrow[+]{y} r$, it follows from Lemma 3.7, that $g(p) \xrightarrow[*]{x} g(q) \xrightarrow[+]{y} g(q) \xrightarrow[*]{z}$.

Now from Lemma 3.14, it follows that since $w = xyz \vdash^s f(p)\#g(p)$, that both $yz \vdash^s f(q)\#g(q)$ and $z \vdash^s f(q)\#g(q)$. However, since $|y| \geqslant 1$, it follows that $|z| < |yz|$.

According to the definition of shortest witnesses, this is a contradiction. Therefore our assumption that $g(q) = g(r)$ must be wrong. Then it follows that $g(q) \neq g(r)$.

Then we have shown that $\forall q, r \in P_w, q \neq r : g(q) \neq g(r)$. As such, we have proven claim 2. □

Claim 3

$\forall q \in P_w, r \in B, g(q) \neq g(r)$

Proof. Let $q \in P_w$ and $r \in B$.

By definition of P_w, and the fact that $q \in P_w$, we have that $\exists x \in$ proper-prefix$(w) : p \xrightarrow[*]{x} q$. Then $\exists y, z \in I^*$ such that $yz \in I^+$ and $w = xyz$.

We consider two cases: either $f(q) = f(r)$ or $f(q) \neq f(r)$. We show that in either of these, $g(q) \neq g(r)$.

Case $f(q) \neq f(r)$:

We have that $B \cup F^{\leqslant k}$ is identified. Since $q \in P_w \subseteq B \cup F^{\leqslant k}$, we also have that q is identified. By Lemma 3.10 (applies because A^S is exact, this in turn follows from Lemma 3.11 and the fact that B is identified and complete) with $q = q$ and $s = r$, it then follows that $f(q) = f(r) \vee q\#r$.

We have that $f(q) \neq f(r)$, therefore $q\#r$. Then we apply Lemma 2.2. We get that $g(q)\#g(r)$, and consequently, $g(q) \neq g(r)$.

Case $f(q) = f(r)$:

This is a proof by contradiction: Suppose that $g(q) = g(r)$.

Since $f(q) = f(r)$ and $p \xrightarrow[*]{x} q$, it follows from Lemma 3.7 that $f(p) \xrightarrow[*]{x} f(r)$.

Likewise, since $g(q) = g(r)$, it follows from Lemma 3.7 that $g(p) \xrightarrow[*]{x} g(r)$.

Show that $\exists r' \in F^0 : g(r) \xrightarrow[+]{y} g(r') \xrightarrow[+]{z}$ and $f(r) \xrightarrow[+]{y} f(r') \xrightarrow[+]{z}$:

From claim 1 and the fact that $w = xyz$, we have that $xyz \vdash^s f(p)\#g(p)$. Then by Lemma 3.14, it follows that $yz \vdash^s f(r)\#g(r)$.

It follows from Lemma 3.12 that $\lambda(r, yz) \uparrow$.

We have that $r \in B$ and that B is complete. Then we may apply Lemma 3.3 with $q = r$, $\sigma = y$, $v = x$, $w = yz$. We get that $\exists r' \in F^0, \exists y \in$ proper-prefix$(yz) : r \xrightarrow[+]{y} r'$. Then it follows that $|y| \geqslant 1$.

Then $\lambda(r, y) \downarrow$. It also follows that $|z| \geqslant 1$ since otherwise it would follow that $\lambda(r, yz) \downarrow$, which is a contradiction.

From Lemma 3.7, it follows that $g(r) \xrightarrow[+]{y} g(r') \xrightarrow[+]{z}$ and $f(r) \xrightarrow[+]{y} f(r') \xrightarrow[+]{z}$.

Since $yz \vdash^s f(r)\#g(r)$, and $z \in I^+$ (since $|z| \geqslant 1$), it follows from Lemma 3.14 that $z \vdash^s f(r')\#g(r')$.

Since $|y| \geqslant 1$, it follows that $|z| < |yz| \leqslant |xyz| = |w|$.

Now we have shown that: $\exists r' \in F^0 : \exists z \in I^* : \ z \vdash^s f(r')\#g(r') \wedge |z| < |w|$.

However, according to claim 1: $\neg(\exists r' \in F^0 : \exists z \in I^* : z \vdash^s f(r')\#g(r') \wedge |z| < |w|)$

This is a contradiction. Therefore the assumption that $g(q) = g(r)$ must

be wrong. It follows that $g(q) \neq g(r)$.

Then we have shown that $\forall q \in P_w, r \in B, g(q) \neq g(r)$. Hence we have proven claim 3. □

Claim 4

$|P_w| = k + 1$

Proof. We show that $|w| > k$ by contradiction:

Suppose that $|w| \leqslant k$. Then it follows from Lemma 3.2 (applies because $F^{<k}$ is complete) with $q = p$, $\sigma = w$, we get that $\lambda(p, w) \downarrow$. From claim 1, we have that $w \vdash f(p)\#g(p)$. Then it follows from Lemma 3.12 that $\lambda(p, w) \uparrow$. This is a contradiction, so it must be the case that $|w| > k$.

Now it follows from the definition of P_w that $|P_w| = k + 1$:

Since $|w| > k$, it follows that proper-prefix(w) contains $|w|$ words of length $0, \ldots k, \ldots |w| - 1$. Then if we take only the words with length $\leqslant k$, we are left with $k + 1$ words of length $0, \ldots k$.

Then by definition of P_w, it follows that for each of these words, there is a state $q \in P_w$. Hence $|P_w| = k + 1$.

□

3.2 Proof for Theorem 3.2

We require Assumption 3.2 in for Theorem 3.2 work. Intuitively speaking, we assume that for any state $s \in B$ where not all of the successors of s are in B, $q \in F^{<k}$ is already identified 'relative to' s. Then we prove that in the opposite case, q is also is identified.

Theorem 3.2 (Cascading identification)**.** *Let $\mathcal{T}$ be an observation tree for both S (with functional simulation f). Let A^S be a state cover for S. B is a basis set of $\mathcal{T}$, based on A^S. The k-level frontiers $F^0 \ldots F^k$ are based on B.*

If A^S is exact, $B \cup F^{<k}$ is complete, and Assumption 3.2 holds:

***Assumption* 3.2.**

$$\forall q \in F^{<k} : \forall s \in B : (f(q) = f(s) \vee q\#s) \vee (\forall i \in I : \delta(s, i) \in B)$$

then:

$$F^k \text{ is identified} \Longrightarrow F^{\leqslant k} \text{ is identified}$$

Proof. We present a proof by induction on the n-level frontier with $n \leqslant k$. We already have that F^k is identified (base case with $n = k$). Then we show the inductive case where $n < k$.

Let $q \in F^n$. We show that $f(q) = f(s) \vee q\#s$.

Suppose that $\neg(\forall s \in B : \neg(\forall i \in I : \delta(s,i) \in B))$, then we get from Assumption 3.2, that $\forall s \in B : f(q) = f(s) \vee q\#s$. Then it follows from Lemma 3.10 (applies since A^S is exact and B is complete) that q is identified.

Suppose that $\forall s \in B, \forall i \in I : \delta(s,i) \in B$:

Proof by contradiction. Suppose that q is not identified. Then either $C(q) = \varnothing$ or $\exists r, t \in C(q) : r \neq t$.

Case $C(q) = \varnothing$:

Suppose that $C(q) = \varnothing$. Then by definition of the candidate set, it follows that $\forall s \in B : q\#s$. Then it follows according to Lemma 2.2, that $f(q)\#f(s)$ and therefore $f(q) \neq f(s)$.

Then $\exists f(q) \in Q^S : \neg(\exists s \in B : f(q) = f(s))$. Then $f^B : B \to Q^S$ is not surjective. But according to Lemma 3.9 (2) (applies since B is complete), f^B is surjective. This is a contradiction.

Case $\exists r, t \in C(q) : r \neq t$:

Let $r, t \in C(q) \subseteq B, r \neq t$. Then since $\forall s \in B, \forall i \in I : \delta(s,i) \in B$, it follows that $\delta(r,i), \delta(t,i) \in B$.

By definition of the candidate set and the fact that $t \in C(q)$, it follows that $q\#t$. Then apply Lemma 3.5 (2) with $q = q$, $r = t$, and $v = i$. We get that $\delta(q,i)\#\delta(t,i)$.

Similarly, by definition of the candidate set and the fact that $r \in C(q)$, it follows that $q\#r$. Then apply Lemma 3.5 (2) with $q = q$, $r = r$, and $v = i$. We get that $\delta(q,i)\#\delta(r,i)$.

Let $r' = \delta(r,i)$ and $t = \delta(t',i)$. Then from Lemma 3.1 (negated, left to right) and the fact that $r \neq t$, $r \underset{*}{\to} r'$ and $t \underset{*}{\to} t'$, we get that $r' \neq t'$.

Since $q'\#r'$ and $q'\#t'$, it follows that $r', t' \in C(q')$. Now we have shown that $\exists q', r' \in C(q') : q' \neq r$. Now since $C(q')$ has at least two unequal elements, it follows that $C(q') \geqslant 2$. Then, according to the definition of identification on states, q' is not identified.

But $q' \in F^{n+1}$ and F^{n+1} is identified. Then q' is identified. This is a contradiction.

We have shown that assuming that q is not identified always leads to a contradiction. Therefore it must be the case that q is identified.

Since we have shown that an arbitrary element $q \in F^n$ is identified, it follows that F^n is identified. So we have shown the inductive case. Then we have proven by induction that $F^{<k}$ is identified. □

3.3 Counterexample to Hypothesis 3.1

Initially, we thought that Hypothesis 3.1 would hold. It turns out that this is not the case. Finding a counterexample proved to be difficult.

In order to find a counterexample for Hypothesis 3.1, we have to find a tree $\mathcal{T}$ for complete Mealy machine S, where there is some $q \in F^{<k}$ that is not identified, even though all of its successors in F^k are identified. Furthermore, we have just proven that we need a situation in which Assumption 3.2 does not hold.

With this in mind, we came up with the following counterexample presented in Figure 3.1. In order to keep the figure reasonably readable and concise, we had to resort to some unusual notation:

- Symbol x is an element of I. A transition using this x, $q \xrightarrow{x/o} t$ is in fact a shorthand notation for two transitions: $q \xrightarrow{a/o} t'$ and $q \xrightarrow{b/o} t''$. If there is a circle labeled t, it in fact represents states t' and t'' (with $t' \neq t''$ but $t \approx t'$). We then call the transition $q \xrightarrow{x/o} t$.
- A dotted arrow is a series of transitions where the states are left implicit. It is similar to having a $*$ under a transition arrow.
- The γ on the left simply means that we repeat the states and transitions within the square on the right that is also labeled γ.
- The dotted lines around circles are semantically equivalent to the non-dotted lines and highlight states of interest.

Now we show that this is, in fact, a counterexample to Hypothesis 3.1.

Hypothesis 3.1. *Let $\mathcal{T}$ be an observation tree for both S (with functional simulation f). Let A^S be a state cover for S. B is a basis set of $\mathcal{T}$, based on A^S. The k-level frontiers $F^0 \dots F^k$ are based on B.*
If A^S is exact, and $B \cup F^0$ is identified, and $B \cup F^{\leqslant k}$ is complete, then:

$$F^k \text{ is identified} \implies F^{<k} \text{ is identified}$$

In Figure 3.1, we have here an observation tree $\mathcal{T}$ (top right, bottom left, bottom right) for a specification S (top left), with $I = \{\text{a, b}\}$ and $O = \{R, G, Y, ?, B, P\}$. All states in $Q^{\mathcal{T}}$ up to F^2 are shown, other states are left implicit.

We have exact state cover $A^S = \{\epsilon, a, b, ba, bb, aa, aaa, aaaa, aaaaa, ab, aba, abaa, abaaa\}$. B is based on this A^S. $B \cup F^0$ is identified, and $B \cup F^{\leqslant k}$ is complete. Furthermore, F^2 is identified.

Now take $k = 2$. Then if Hypothesis 3.1 were true, it would follow that F^1 is identified.

However, F^1 is not identified:

> We have $p01, b01 \in F^1$ where $C(po1) = \{p0, b0\}$ and $C(bo1) = \{p0, b0\}$. Then by definition of identification, it follows that $p01$ and $b01$ are not identified. Then by definition of identification on sets, it follows that F^1 is not identified, since $p01, b01 \in F^1$.

Then this is a counterexample to Hypothesis 3.1.

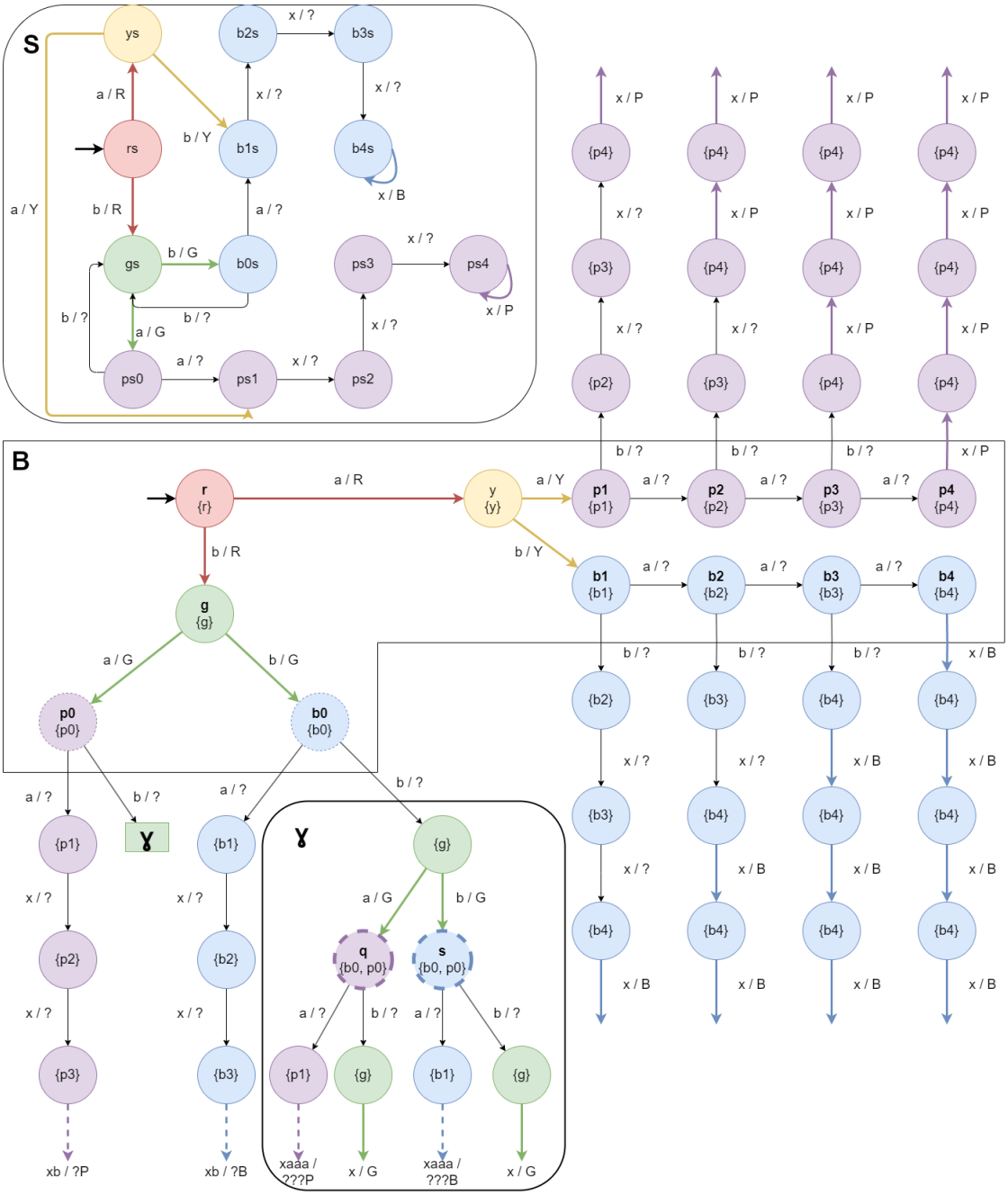

Figure 3.1: Counterexample to Hypothesis 3.1.

Chapter 4

Related Work

4.1 Conformance testing

This section is based on chapter 2 of the PhD-thesis of Joshua Moerman [9]. Note that the terms witness and *separating sequence* are synonyms.

As was mentioned in chapter 1, various methods for generating test suites have been proposed. We will discuss most of these methods here, and show that their correctness follows from Theorem 3.1.

4.1.1 The W-method

As aforementioned, the W-method is relatively simple. We need the following three components:

- State cover A^S of the specification S.
- The set $I^{\leqslant k+1}$, which contains all words with input alphabet I of length less than or equal to $k+1$.
- Characterization set W_S of S. (See Definition 4.1)

Definition 4.1 (Characterization set)**.** *For some Mealy machine $\mathcal{N}$, a* ***characterization set*** *$W_\mathcal{N}$ is a subset of I^*, such that for any two states $q, r \in Q^\mathcal{N}$ there exists a word $\sigma \in W_\mathcal{N}$ such that $\sigma \vdash q\#r$.*

Then we can define the W-method as:

$$T_w = A^S \cdot I^{\leqslant k+1} \cdot W_S$$

Based on the W-method was, Fujiwara *et al.* created the more efficient W_p-method in 1991 [4]. We will not discuss this method here. The next HSI-method is based on the W_p method.

4.1.2 The HSI-method

HSI stands for "*harmonized state identifiers*". The method was proposed by Luo *et al.* in 1995 [7] and Petrenko *et al.* in 1993 [11]. In addition to the aforementioned A^S and $I^{\leqslant k+1}$, we also need a separating family $\mathcal{H}$ (instead of a characterization set W_S).

In order to define a *separating family*, we first explain the concepts of *state identifiers* and *harmonization*. We also define concatination on families, which is different from regular concatonation.

Definition 4.2 (State identifiers)**.** *Let $\mathcal{N}$ be a Mealy machine. A **state identifier** of a state $s \in Q^{\mathcal{N}}$ is a set $W_s \subset I^*$ such that for every state $q \in Q^{\mathcal{N}}$ where $q \# s$, there exists a word $\sigma \in W_s$ such that $\sigma \vdash q \# s$.*

Note that state identifiers are different from characterization sets: state identifiers only contain witnesses between one state and all other states that are apart from this state. On the other hand, characterization sets contain a witnesses for *any* pair of apart states. Taking the union of the state identifiers of all states (except one) would result in a characterization set.

Definition 4.3 (Harmonization and separating family)**.** *A set of state identifiers $\mathcal{H}$ is **harmonized** if for any two state identifiers $W_s, W_t \in \mathcal{H}$, $\exists \sigma \in W_s \cap W_t$ such that $\sigma \vdash s \# t$. A harmonized set of separating sequences is called a **separating family**.*

We explain the intuition behind harmonization. It might seem that it follows by definition that any set of state identifiers should be harmonized. However, here is a counterexample:

Suppose that we have Mealy machine $\mathcal{M}$ in Figure 4.1. Then $W_s = \{a\}$ is a set of state identifiers for s (because $a \vdash s \# t$) and $W_t = \{b\}$ is a set of state identifiers for t (because $b \vdash t \# s$). However, $W_s \cap W_t = \varnothing$, so $\{W_s, W_t\}$ is not harmonized. If we take $W_t' = \{a\}$, however, then $\{W_s, W_t'\}$ is harmonized because we use the same witness for states s and t: $W_s \cap W_t' = \{a\}$ and $a \vdash s \# t$.

Intuitively speaking, having a harmonized set of separating families ensures that the *same seperating sequence* is being used for the *same pair of states* in every state identifier.

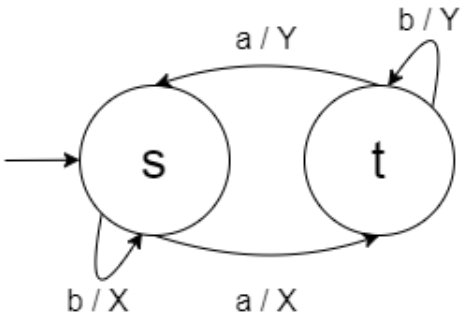

Figure 4.1: Example Mealy machine $\mathcal{M}$ to explain harmonization

Now for the HSI method, we leverage this idea of assuring that the same witness is being used for the same pair of states. We need to define concatenation on separating families with special operator $\odot$ to ensure that the right separating family is used for each state:

Definition 4.4 (Family concatenation)**.** *Let X be a set of words and $\mathcal{Y}$ be a family. Then $X \odot \mathcal{Y} = \{x \cdot y \mid x \in X, y_s \in \mathcal{Y}$ where $x = access(s)\}$*

Consider the following example: in our specification S, we have states $q, r \in Q^S$ with access sequences a_q and a_r, and state identifiers $W_q = \{v, u\}$ and $W_r = \{w, t\}$. Then: $\{a_q, a_r\} \odot \{W_q, W_r\} = \{a_q v,\ a_q u,\ a_r w,\ a_r t\}$. The right separating family is used for each state.

We define the HSI-method as follows.

$$T_{HSI} = A^S \cdot I^{\leqslant k+1} \odot \mathcal{H}$$

4.1.3 The ADS-method

The ADS-method can be simply considered as a particular case of the HSI-method, where only one word is required to identify a state, i.e. $|W_s| = 1$ for all $W_s \in \mathcal{H}$. We call this particular word an *adaptive distinguishing sequence*. For the example that we presented in Figure 4.1, this is actually the case. This particular technique does not work on every Mealy machine because not every Mealy machine admits an ADS. Moerman [9] describes ADS in more detail.

4.1.4 The $UIOv$-method

Like the HSI-method, the $UIOv$-method can be considered a generalization of the ADS-method. Instead of relying on harmonization, it uses UIO-sequences.

Definition 4.5 (UIO)**.** *Let $\mathcal{N}$ be a Mealy machine. Given a state $s \in Q^{\mathcal{N}}$, a **Unique Identifying Output (UIO)** for s is a word $\sigma \in I^*$ such that for all states $t \in Q\mathcal{N}$ with $s \# t$, $\sigma \vdash s \# t$.*

Now we let $\mathcal{U}$ be a set of UIO-sequences for all states in S. Then we may define the $UIOv$-method as follows:

$$T_{UIOv} = A^S \cdot I^{\leqslant k} \cdot \mathcal{U} \cup A^S \cdot I^{\leqslant k+1} \odot \mathcal{U}$$

Note that we use $\mathcal{U}$ twice here: in the first instance, we simply take a concatenation, meaning that we use all elements of $\mathcal{U}$. In the second instance we use family concatenation; we only use the specific UIO that corresponds to the correct state.

4.2 Applying Theorem 3.1

In this section, we demonstrate how to apply our Theorem 3.1 to show that any of the aforementioned test suites are k-complete, under the assumption that A^S is exact.

When we say that a test suite T *generates* a tree $\mathcal{T}$, we mean that $\mathcal{T}$ is a tree such that $\forall \sigma \in T : \delta(q_0^{\mathcal{T}}, \sigma) = \delta(q_0^S, \sigma)$.

The proofs of the following Lemmas can be found in Appendix A.7.

Lemma 4.1 (Complete tree)**.** *Let test suite T with $A^S \cdot I^{\leqslant k+1} \subseteq T$.*

Then T generates a tree $\mathcal{T}$ where $B \cup F^{<k}$ is complete.

Lemma 4.2. *Let test suite T with $A^S \cdot I^{\leqslant k+1} \cdot D \subseteq T$.*

Then T generates a tree $\mathcal{T}$ where for all $q \in B \cup F^{\leqslant k}$ and $\delta(q, D) \downarrow$

Lemma 4.3. *Let test suite T and $\mathcal{Y}$ either a separating family or family of UIOs, such that $A^S \cdot I^{\leqslant k+1} \odot \mathcal{Y} \subseteq T$.*

then T generates a tree $\mathcal{T}$ where for all $q \in B \cup F^{\leqslant k} : \delta(q, y_q) \downarrow$.

Corollary 4.1. *The W-method, HSI-method, ADS-method and $UIOv$-method are k-complete.*

Proof. Assume that $|Q^{\mathcal{M}}| \leqslant |Q^S| + k$. We show that Theorem 3.1 can by applied.

We let $\mathcal{T}$ be a tree generated by T on S. Let f be the functional simulation from $Q^{\mathcal{T}}$ to S. Let f be the functional simulation from $\mathcal{T} \to Q^S$.

It follows from Lemma 4.1 that $B \cup F^{<k}$ is complete.

Let $q \in B \cup F^{\leqslant k}$ and $r \in B \cup F^{<k}$, such that $f(q) \neq f(r)$. We show that for every test method it follows that $\exists \sigma \in I^*$: where σ is defined on both q and r, and $\sigma \vdash f(q) \# f(r)$.

- For the W-method: By definition of the characterization set W, we have that $\exists \sigma \in W : \sigma \vdash f(q) \# f(r)$. From applying Lemma 4.2 with $D = W$ we get that $\lambda(q, \sigma) \downarrow$ and $\lambda(r, \sigma) \downarrow$.
- For the HSI-method and ADS-method: We have that there exists a separating family $\mathcal{H}$ which is harmonized. Therefore there exist W_q and W_r which are separating families for q and r respectively, with $\exists \sigma \in W_q \cap W_r$ with $\sigma \vdash f(q) \# f(r)$. From applying Lemma 4.3, we get that $\lambda(q, \sigma) \downarrow$ and $\lambda(r, \sigma) \downarrow$.
- For the $UIOv$-method: We have a family of $UIOs$, $\mathcal{U}$. From $A^S \cdot I^{\leqslant k+1} \odot \mathcal{U}$ and Lemma 4.3, we get that $\lambda(q, UIO_{f(q)}) \downarrow$. From $A^S \cdot I^{\leqslant k} \cdot \mathcal{U}$ and Lemma 4.2 with $D = \mathcal{U}$, we get that $\lambda(r, UIO_{f(q)}) \downarrow$.

  By definition of $UIO_{f(q)}$, it follows that $UIO_{f(q)} \vdash f(q) \# f(r)$.

Now we get from Lemma 3.8 (with $h = f$) that $f(q) \not\# q$ and (with $h = g$, $q = r$) that $f(r) \not\# r$. Then by definition of non-apartness, $\lambda(q, \sigma) = \lambda(f(q), \sigma) \neq \lambda(f(r), \sigma) = \lambda(r, \sigma)$. Thus by definition of apartness, $q \# r$.

Assumption 3.1 immediately follows from this.

We have shown that $f(q) = f(r) \vee f \# r$. We apply 3.10 (possible since A^S is exact). We get that q is identified. Since q is an arbitrary element in $B \cup F^{\leqslant k}$, it follows that $B \cup F^{\leqslant k}$ is identified.

Then all the conditions that Theorem 3.1 requires are met. □

4.3 Comparison to Vaandrager's Theorem

Our work is most similar to a paper by Vaandrager in 2024 [14]. Vaandrager shows k-completeness given slightly different conditions from ours.

Theorem 4.1 (Theorem 3.5 in [14])**.** *Let $\mathcal{T}$ be an observation tree for both S and for $\mathcal{M}$.*

If $B \cup F^{<k}$ is complete and $B \cup F^{\leqslant k}$ is identified, and $|Q^{\mathcal{M}}| \leqslant |Q^S| + k$, and Assumption 4.1 holds:

***Assumption* 4.1.**

$$\forall r \in B, \forall t' \in F^k, \forall t'' \in F^{<k} : r \# t' \Longrightarrow r \# t'' \vee t' \# t''$$

then:

$$S \approx \mathcal{M}$$

The only thing that differs between the two Theorems is the assumption. Note that in Vaandrager's paper, a different definition of *complete* is used, but we have restated Theorem 4.1 using our definition.

At first glance, it might not be clear how Vaandrager's Assumption 4.1 and our Assumption 3.1 relate. It turns out, however, that we can restate Vaandrager's Assumption such that it uses the same terminology as our Assumption.

Lemma 4.4 (Equivalent assumptions, A.8.1)**.** *Let $\mathcal{T}$ be an observation tree. If $B \cup F^{\leqslant k}$ is identified, then:*

$$\textit{Assumption 4.1} \iff \forall q \in F^{<k}, \forall r \in F^k : f(q) = f(r) \vee q \# r$$

We can now see that the assumptions are actually quite similar. There is an important difference however about for *which* pairs of frontier states $q, r \in F^{\leqslant k}$ we require that $f(q) = f(r) \vee q \# r$. Vaandrager requires this for $q \in F^{<k}$ and $r \in F^k$. Whereas we require this for $q, r \in F^{\leqslant k}$ such that $q \underset{+}{\longrightarrow} r$.

Interestingly enough, neither assumption seems to imply the other. However, they do both imply a third condition. It would then make sense that this is a sufficient condition to prove k-completeness.

Conjecture 4.1. *Let $\mathcal{T}$ be an observation tree for both S and for $\mathcal{M}$. If $B \cup F^{<k}$ is complete and $B \cup F^{\leqslant k}$ is identified, and $|Q^{\mathcal{M}}| \leqslant |Q^S| + k$, and Assumption 4.2 holds:*

***Assumption* 4.2.**

$$\forall q \in F^{<k}, \forall r \in F^k, q \underset{+}{\longrightarrow} r : f(q) = f(r) \vee q \# r$$

then:

$$S \approx \mathcal{M}$$

We have attempted a proof for Conjecture 4.1 that is similar to our proof for Theorem 3.1. In our proof for Claim 2 (section 3.1.7), we show that $\forall q, r \in P_w : g(q) \neq g(r)$, where P_w is our set of *path states*. We have tried to show by a proof by contradiction that Assumption 4.2 implies that $\forall q, r \in P_w : g(q) \neq g(r)$, but we were unable to do so. If Conjecture 4.1 holds, then a different proof strategy may be required.

4.3.1 The number of state pairs

We can compare our Assumption 3.1 with Vaandrager's Assumption 4.1 in terms of performance.

Let w be the number of states $q, r \in F^{\leqslant k}$ where we require that $f(q) = f(r) \vee q \# r$. Let $n = |F^0|$ and $m = |I|$, and k such that $|Q^{\mathcal{M}}| \leqslant |Q^S| + k$.

For Vaandrager's Assumption: 4.1, $w = \frac{n^2 m^k (m^k - 1)}{m-1}$.

For our Assumption: 3.1, $w = \frac{nm(km^{k+1} - (k+1)m^k + 1)}{(m-1)^2}$.

Both of these statements are proven in Appendix section A.8.2. For illustration, if we take $n = 3$, $m = 5$ and $k = 2$, then $w = 1350$ for Vaandrager's assumption and $w = 165$ for our assumption.

In conclusion, our Theorem 3.1 requires less states $q, r \in F^{\leqslant k}$ where it is required that $f(q) = f(r) \vee q\#r$, although the number still grows exponentially.

It should be noted that from the proof of Corollary 4.1, it already follows that in all the testing methods that we discussed, $\forall q, r \in F^{\leqslant k} : f(q) = f(r) \vee q\#r$. It remains to be seen if there are scenarios in which Theorem 3.1 catches more redundant tests than Theorem 4.1, in particular if we take into account harmonized state identifiers.

4.3.2 We definitely need the assumptions

Vaandrager also gives an example that shows that we actually need an assumption: In Figure 4.2, we have a specification S and an implementation $\mathcal{M}$ where $S \not\approx \mathcal{M}$, since input word $rrrlll$ tells them apart. We take $k = 1$, then $|Q^M| \leqslant |Q^S| + k$. In Figure 4.3, we have an observation tree for both S and $\mathcal{M}$ where $B \cup F^{\leqslant k}$ is identified and $B \cup F^{<k}$ is complete.

Suppose that $B \cup F^{\leqslant k}$ is identified and $B \cup F^{<k}$ is complete were sufficient conditions for k-completeness. Then this would imply that $S \approx \mathcal{M}$. However, we have that $S \not\approx \mathcal{M}$.

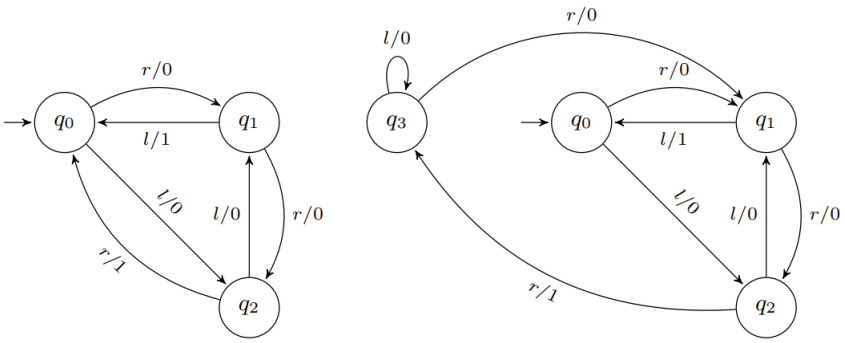

Figure 4.2: Mealy machines S and $\mathcal{M}$. Taken from Example 3.8 in [14]

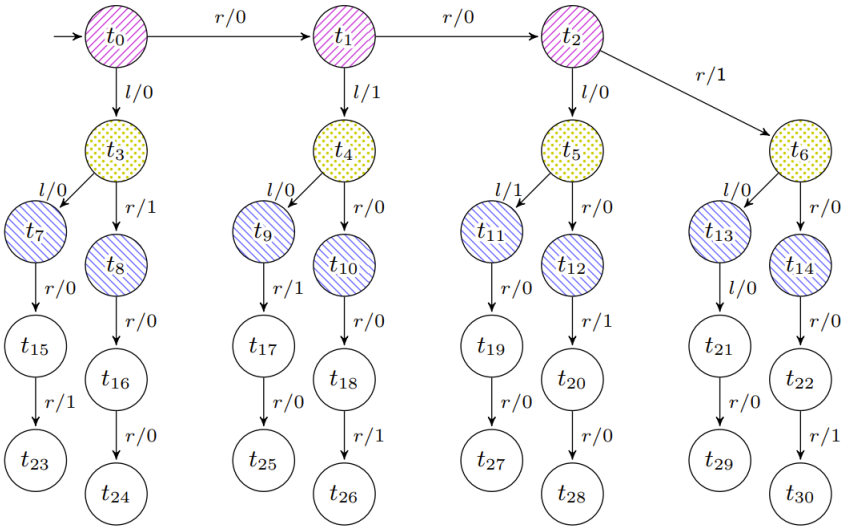

Figure 4.3: Observation tree $\mathcal{T}$. Taken from Example 3.8 in [14]

Neither Vaandrager's Assumption 4.1 nor our Assumption 3.1 holds, because $t_6 \in F^0$ and $t_{13} \in F^1$, $t_6 \xrightarrow[+]{} t_{13}$ and $f(t_6) \neq f(t_{13}) \wedge t_6 \not\# t_{13}$.

4.4 Active automata learning and $L^\#$

Since much of our work is inspired by the $L^\#$ algorithm, it makes sense to give a brief overview. $L^\#$ solves a problem called *active automata learning*.

4.4.1 Active automata learning

Active automata learning is a problem that was first introduced by Dana Angluin in 1987 [1] for various types of state machines. In the case of Mealy Machines, it works as follows. Suppose that you have some black-box Mealy Machine $\mathcal{M}$. We call this the *SUL (System Under Learing)*. Then you have a *Learner* and a *Teacher*. The Learner tries to infer $\mathcal{M}$ by asking two kinds of queries to the Teacher:

1. In *Output queries*, the Learner gives some input word $\sigma \in I^*$ to the Teacher. The Teacher then returns $\lambda(q_0^S, \sigma)$.

2. In *Equivalence queries*, the Learner gives some *hypothesis* Mealy Machine $\mathcal{H}$ to the teacher and asks whether $\mathcal{H} \approx \mathcal{M}$. If this is the case, the Teacher returns `yes` and the Learning is succesfull. If not it returns `no` along with a counterexample σ.

In figure 4.4, a graphical representation of the active automata learning problem is given.

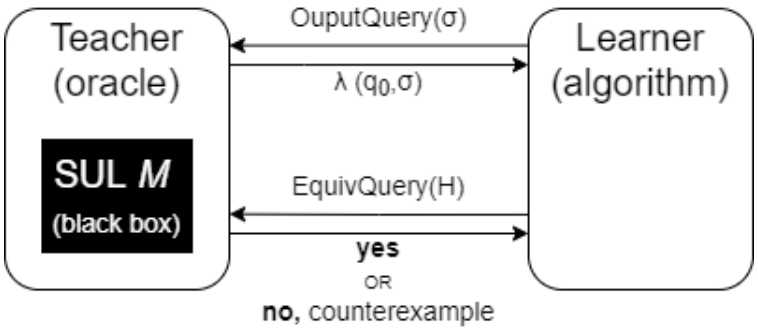

Figure 4.4: Graphical overview of the active learning problem. The *Learner* (right) tries to infer the SUL $\mathcal{M}$ by asking queries to the *Teacher* (left)

Active automata learning algorithms do not have to implement these two types of queries. They treat the teacher like an *oracle*. For real-world implementations, it is obviously necessary to implement these. Output queries are very easy to implement. It suffices to just feed the input word σ into the $\mathcal{M}$ and return the output.

Equivalence queries are harder. The link between conformance testing and automata learning lies here. Since the SUL $\mathcal{M}$ is a black box, these are impossible to *implement* because this would require a complete test suite. We discussed in chapter 2 why this is impossible. In practice, test suites are used that do not guarantee any k-completeness, but are optimized to find counterexamples quickly.

4.4.2 $L^{\#}$

For a long time, the field of active automata learning was almost exclusively concerned with optimizing L^*. This was until Vaandrager *et al.* [15] introduced a novel approach to active automata learning in 2022. Most of the concepts discussed in chapter 2 upon which this bachelor thesis builds (notably apartness and observation trees), are simply adopted from this $L^{\#}$ algorithm.

Briefly speaking, $L^{\#}$ uses an observation tree to store information about the implementation, that it gets from output queries and equivalence queries. This idea was in turn inspired by Soucha and Bogdanov in 2019 [13] who first considered Observation trees for learning algorithms. The notion of apartness in the $L^{\#}$ algorithm was inspired by work by Geuvers & Jacobs [5].

Particularly related to this bachelor thesis, is that $L^{\#}$ also works with an observation tree with a prefix-closed basis. In the case of $L^{\#}$, this contains all

states that have been found to be apart from each other so far. $L^{\#}$ expands the basis state by state, and at a certain point it creates a hypothesis $\mathcal{H}$, which is directly based on the states in the basis.

Chapter 5

Conclusions & Further work

Our most important result is without a doubt Theorem 3.1. We have shown different conditions for k-completeness than in previous work. This provides us with a greater understanding of the requirements of k-completeness.

In the related work section, we discuss that our Theorem 3.1 requires less pairs $q, r \in F^{\leqslant k} : f(q) = f(r) \vee q\#r$ than Vaandrager's Theorem. To the contrary, we were not able to show that the conditions of Vaandrager's Theorem imply our conditions. Our Theorem might filter out more redundant tests than Vaandrager's, although this requires additional research to verify.

We have provided a proof using a novel proof technique (shortest witnesses) to show k-completeness. This differs from the technique that Vaandrager [14] uses to show k-completeness (bisimulations). Our proof technique could be used by future researchers to prove truly weaker conditions for k-completeness using observation trees, in particular Conjecture 4.1. It must be noted that we are not entirely convinced that this conjecture actually holds.

Our less notable result is an enquiry into the idea of cascading identification. We have shown with a counterexample that F^k is identified *does not* imply that $F^{<k}$ is identified. However, using an additional assumption, we were able to prove Theorem 3.2. This is still a reasonably interesting result.

We provide some directions for future research.

- Look for different or weaker conditions for k-completeness.
- Vaandrager's Theorem 4.1 might be provable using a proof by shortest witnesses. Conversely, our Theorem 3.1 might be provable using bisimulations.
- It might also be possible to show that Vaandrager's Assumption 4.1 implies our Assumption 3.1, under the condition that $B \cup F^{\leqslant k}$ is

identified and $B \cup F^{<k}$ is complete, or the reverse.

- It might be possible to prove Conjecture 4.1. This would also solve the aforementioned.
- Theorem 3.1 might be used to either create more efficient test suites, or prune existing ones.
- Given a test suite T, it might be interesting to look into if certain tests are more likely to run into a counterexample than others. This could reduce the amount of tests that need to be executed to find a counterexample.
- Theorem 3.1 might be used to integrate learning and testing more efficiently in $L^{\#}$, if we are interested in k-completeness guarantees while learning.
- Related to our enquiry about identification, we present Conjecture 5.1. Suppose that we have a tree $\mathcal{T}$ where F^k is identified. Then it would follow from Conjecture 5.1 that in most cases, we can make a Tree $\mathcal{T}'$ which is a copy of $\mathcal{T}$ such that $r \in F^{<k}$ is identifed *without* adding any words to r, but instead adding words to basis states.

Conjecture 5.1 (Weak cascading identification)**.** *Let $\mathcal{T}$ be an observation tree for S with A^S is exact.*

$$\begin{aligned} F^k \textit{ is identified} \implies \\ \forall q \in F^{<k} : \forall s \in B, f(q) = f(s) : \\ \exists \sigma \in I^*, \lambda(q, \sigma) \downarrow : \sigma \vdash q \# f(s) \end{aligned}$$

Bibliography

[1] D. Angluin. Learning regular sets from queries and counterexamples. *Information and computation*, 75(2):87–106, 1987.

[2] W. Y. L. Chan, C. T. Vuong, and M. R. Otp. An improved protocol test generation procedure based on uios. *SIGCOMM Comput. Commun. Rev.*, 19(4):283–294, aug 1989.

[3] T. S. Chow. Testing software design modeled by finite-state machines. *IEEE transactions on software engineering*, (3):178–187, 1978.

[4] S. Fujiwara, G. v. Bochmann, F. Khendek, M. Amalou, and A. Ghedamsi. Test selection based on finite state models. *IEEE Transactions on Software Engineering*, 17(6):591–603, 1991.

[5] H. Geuvers and B. Jacobs. Relating apartness and bisimulation. *Logical Methods in Computer Science*, 17, 2021.

[6] D. Lee and M. Yannakakis. Testing finite-state machines: state identification and verification. *IEEE Transactions on Computers*, 43(3):306–320, 1994.

[7] G. Luo, A. Petrenko, and G. v Bochmann. Selecting test sequences for partially-specified nondeterministic finite state machines. In *Protocol Test Systems: 7th workshop 7th IFIP WG 6.1 international workshop on protocol text systems*, pages 95–110. Springer, 1995.

[8] G. H. Mealy. A method for synthesizing sequential circuits. *The Bell System Technical Journal*, 34(5):1045–1079, 1955.

[9] J. Moerman. *Nominal techniques and black box testing for automata learning*. PhD thesis, [Sl]:[Sn], 2019.

[10] E. F. Moore. Gedanken-experiments on sequential machines. *Automata studies*, 34:129–153, 1956.

[11] A. Petrenko, N. V. Yevtushenko, A. Lebedev, and A. Das. Nondeterministic state machines in protocol conformance testing. In *Protocol Test Systems*, 1993.

[12] A. Simão, A. Petrenko, and N. Yevtushenko. On reducing test length for fsms with extra states. *Software Testing, Verification & Reliability*, 22:435–454, 09 2012.

[13] M. Soucha and K. Bogdanov. Observation Tree Approach: Active Learning Relying on Testing. *The Computer Journal*, 63(9):1298–1310, 07 2019.

[14] F. Vaandrager. A new perspective on conformance testing based on apartness. *manuscript. Radboud University*, 2024.

[15] F. Vaandrager, B. Garhewal, J. Rot, and T. Wißmann. A new approach for active automata learning based on apartness. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 223–243. Springer, 2022.

[16] M. Vasilevskii. Failure diagnosis of automata. *Cybernetics*, 9(4):653–665, 1973.

Appendix A

Deferred proofs

A.1 Transitions and completeness

A.1.1 Same transition

Proof for Lemma 3.1

Lemma A.1 (Same transition)**.** *Let $\mathcal{T}$ be a Tree.*
For $q, r, q', r' \in Q^{\mathcal{T}}, v \in I^*$ with $q \xrightarrow[*]{v} q'$ and $r \xrightarrow[*]{v} r'$:

$$q = r \iff q' = r'$$

Proof. **From left to right**:

From $q \xrightarrow{v/o} q'$ and $r \xrightarrow{v/o'} r'$, we have that $q' = \delta(q, v)$ and $r' = \delta(r, v)$. We have that $q = r$. Then $q' = \delta(q, v) = \delta(r, v) = r'$.

From right to left:

We have that $q' = r'$

By the definition of a Tree, it follows that there is exactly one $\text{access}(q), \text{access}(r) \in I^*$ such that $\delta(\mathcal{T}, \text{access}(q)) = q$, $\delta(\mathcal{T}, \text{access}(r)) = r$.

Now suppose that $q \neq r$. Then it follows that $\text{access}(q) \neq \text{access}(r)$. Because $q \xrightarrow{v} q'$ and $r \xrightarrow{v} r'$, it follows that $\text{access}(q') = \text{access}(q)v \neq \text{access}(r)v = \text{access}(r')$.

But since $q' = r'$, by definition, $\text{access}(q') = \text{access}(r')$. This is a contradiction, therefore the assumption that $q \neq r$ must be false. Therefore $q = r$. □

A.1.2 Defined words

Proof for Lemma 3.2

Lemma A.2 (Defined words)**.** *Let $\mathcal{T}$ be an observation tree for S. If $F^{<k}$ is complete, then:*

$$\forall q \in F^0 : \forall \sigma \in I^* : |\sigma| \leqslant k \Longrightarrow \lambda(q, \sigma) \downarrow$$

Proof. Let $q \in F^0$ and $\sigma \in I^*$ such that $|\sigma| \leqslant k$.
By induction on σ, we show that $\exists r' \in F^0 : q \xrightarrow[*]{\sigma} r'$:

Base case: $\sigma = \epsilon$

Let $r' = q$. $\delta(r', \sigma) = \delta(q, \epsilon) = q$, $q \in F^0$. Then $\exists r' \in F^0 : q \xrightarrow[*]{\sigma} r'$

Inductive case: $\sigma = vi$ and $|\sigma| \leqslant k$

From the Inductive Hypothesis we get that $\exists r \in F^{|v|} : q \xrightarrow[*]{v} r$. Since $|\sigma| = |vi| \leqslant k$ it follows that $|v| < k$. Then $F^{|v|}$ is complete. Therefore, r is complete. Then $\exists r' \in F^{|v|+1} : r \xrightarrow[*]{i} r'$.

Then by definition of transitions and the fact that $|\sigma| = |vi| = |v| + 1$, we get that $\exists r' \in F^{|\sigma|} : q \xrightarrow[*]{\sigma} r'$

By induction, for all $\sigma \in I^*, |\sigma| \leqslant k : \exists r' \in Q^{\mathcal{T}} : q \xrightarrow[*]{\sigma} r'$, and equivalently $\lambda(q, \sigma) \downarrow$. □

A.1.3 Frontier prefix

Proof for Lemma 3.3

Lemma A.3 (Frontier prefix)**.** *Let $\mathcal{T}$ be an observation tree. If B is complete, then $\forall q, \in B, \forall \sigma \in I^*, \lambda(q, \sigma) \uparrow$:*

$$\exists q' \in F^0 : \exists v \in \textit{proper-prefix}(\sigma) : q \xrightarrow[+]{v} q'$$

Proof. Let $q, \in B$ and let $\sigma \in I^*, \lambda(q, \sigma) \uparrow$.
We do a proof by induction on the word w, which is a prefix of σ.

Base case: $w = i$

Because B is complete, it follows that q is complete. Then $\exists r' \in Q^{\mathcal{T}} : q \xrightarrow[*]{i} r'$.

$\lambda(q, i) \downarrow$ and $\lambda(q, \sigma) \uparrow$. Therefore $w = i \neq \sigma$. Since w is a prefix of σ, and $w \neq \sigma$, it follows that $w \in$ proper-prefix(σ).

Inductive case: $w = vi$.

From the Inductive Hypothesis we get that $\exists r \in B : q \xrightarrow[*]{v} r$ where $v \in$ proper-prefix(σ). Because B is complete, it follows that r is complete. Then $\exists r' \in Q^{\mathcal{T}} : r \xrightarrow{i} r'$. Then by definition of transitions and the fact that $w = vi$, it follows that $q \xrightarrow{w} r'$.

$\lambda(r, w) \downarrow$ and $\lambda(r, \sigma) \uparrow$. Therefore $w \neq \sigma$. Since w is a prefix of σ, and $w \neq \sigma$, it follows that $w \in \text{proper-prefix}(\sigma)$.

Now, according to the definition of the k-level frontiers, we have two cases:

Case 1: $r' \in B$

Then we have that $\exists r' \in B : q \xrightarrow[*]{w} r'$. We can apply the inductive case again. Since the basis B is finite, it follows that Case 1 cannot be applied forever (this would require an infinite basis set B)

Case 2: $r' \in F^0$

Now we have shown that $\exists r' \in F^0 : q \xrightarrow[*]{w} r'$. Since $|w| = |vi| \geqslant |i| = 1$, it also follows that $q \xrightarrow[+]{w} r'$.

Then we have proven by induction that $\exists q' \in F^0 : \exists w \in \text{proper-prefix}(\sigma) : q \xrightarrow[+]{v} q'$. □

A.2 Transition apartness and non-apartness

A.2.1 Transition apartness

Proof for Lemma 3.4

Lemma A.4 (Transition apartness)**.** *Let $\mathcal{N}$ be a (partial) Mealy machine. For all $q, r \in Q^{\mathcal{N}}$, $v, w \in I^*$:*

$$vw \vdash q\#r \iff v \vdash q\#r \vee w \vdash \delta(q, v)\#\delta(r, v)$$

Proof. **From left to right**:

We do a proof by contradiction. Suppose that $\neg(v \vdash q\#r \vee w \vdash \delta(q, v)\#\delta(r, v))$. Then it follows by De Morgan's laws that $\neg(v \vdash q\#r)$ and $\neg(w \vdash \delta(q, v)\#\delta(r, v))$.

Then by definition of apartness, $\lambda(q, v) = \lambda(r, v)$ and $\lambda(\delta(q, v), w) = \lambda(\delta(r, v), w)$. But then by definition of the λ function:

$$\lambda(q, vw) = \lambda(q, v) \cdot \lambda(\delta(q, v), w) = \lambda(r, v) \cdot \lambda(\delta(r, v), w) = \lambda(r, vw)$$

Then by definition of apartness, $\neg(vw \vdash q\#r)$.

Since we have shown a contradiction, our assumption must be false. Therefore $v \vdash q\#r \vee w \vdash \delta(q, v)\#\delta(r, v)$.

From right to left:

We consider two cases:

Case $v \vdash q\#r$:

By definition of apartness, $\lambda(q, v) \neq \lambda(r, v)$.

Case $w \vdash \delta(q,v)\#\delta(r,v)$:

By definition of apartness, $\lambda(\delta(q,v),w) \neq \lambda(\delta(r,v),w)$.

Then in either case, it follows by definition of the λ function that:

$$\lambda(q,vw) = \lambda(q,v) \cdot \lambda(\delta(q,v),w) \neq \lambda(r,v) \cdot \lambda(\delta(r,v),w) = \lambda(r,vw)$$

Then it follows by definition of apartness that $vw \vdash q\#r$.

□

A.2.2 Transition non-apartness

Proof for Lemma 3.5

Lemma A.5 (Transition non-apartness)**.** *Let* $\mathcal{N}$ *be a (partial) Mealy machine.*
For all $q, r \in Q^{\mathcal{N}}$*:*

$$(1)\ \forall \sigma \in I^*, \delta(q,\sigma)\downarrow \wedge \delta(r,\sigma)\downarrow\colon \exists v \in \mathit{prefix}(\sigma):\quad \lambda(q,v) = \lambda(r,v) \wedge \delta(q,v) \not\# \delta(r,v) \implies q \not\# r$$

and

$$(2)\ q \not\# r \implies \forall v \in I^*, \delta(q,v)\downarrow \wedge \delta(r,v)\downarrow\colon \qquad \lambda(q,v) = \lambda(r,v) \wedge \delta(q,v) \not\# \delta(r,v)$$

Proof. **For 1**:

Let $\sigma \in I^*, q, r \in Q^{\mathcal{N}}$ such that $\lambda(q,\sigma)\downarrow$ and $\lambda(r,\sigma)\downarrow$.

Now from our assumptions we have that there exists some $v, w \in I^*$ such that $\sigma = v \cdot w$, $\lambda(q,v) = \lambda(r,v)$, and $\delta(q,v) \not\# \delta(r,v)$.

Since we have that $\lambda(q,v) = \lambda(r,v)$, it follows by definition of apartness that $\neg(v \vdash q\#r)$. Since we have that $\delta(q,v) \not\# \delta(r,v)$, it follows by definition of non-apartness that $\neg(w \vdash \delta(q,v)\#\delta(r,v))$.

Then we have that $\neg(v \vdash q\#r) \wedge \neg(w \vdash \delta(q,v)\#\delta(r,v))$. By De Morgan's laws, this equivalent to $\neg(v \vdash q\#r \vee w \vdash \delta(q,v)\#\delta(r,v))$.

Now we can apply Lemma 3.4. Since it is a bi-implication, having the right side false implies that the left side is false. Then it follows that $\neg(vw \vdash q\#r)$. Since $\sigma = vw$, also $\neg(\sigma \vdash q\#r)$.

Since we have shown that for any arbitrary element $\sigma \in I^*$ that $\neg(\sigma \vdash q\#r)$, it follows by definition of non-apartness that $q \not\# r$.

For 2:

Let $v \in I^*$ such that $\lambda(q,v)\downarrow \wedge \lambda(r,v)\downarrow$. Now let $w \in I^*$ such that $\lambda(\delta(q,v),w)\downarrow \wedge \lambda(\delta(r,v),w)\downarrow$.

Since $q \not\# r$, it follows by definition of non-apartness that $\neg(vw \vdash q\#r)$. Then it follows from Lemma 3.4 that $\neg(v \vdash q\#r \vee w \vdash \delta(q,v)\not\#\delta(r,v)$. By De Morgan's laws, this is equivalent to $\neg(v \vdash q\#r)$ and $\neg(w \vdash \delta(q,v)\not\#\delta(r,v))$.

From $\neg(v \vdash q\#r)$, it follows by definition of apartness that $\lambda(q,v) = \lambda(r,v)$.

From $\neg(w \vdash \delta(q,v)\not\#\delta(r,v))$, it follows by definition of apartness that $\lambda(\delta(q,v),w) = \lambda(\delta(r,v),w)$.

Then we have shown that $\forall w \in I^*, \lambda(\delta(q,v),w) \downarrow \wedge \lambda(\delta(r,v),w) \downarrow$: $\lambda(\delta(q,v),w) = \lambda(\delta(r,v),w)$. Then by definition of non-apartness, $\delta(q,v)\not\#\delta(r,v)$.

□

A.3 Shortest witnesses

A.3.1 Shortest witness consistency

Proof for Lemma 3.6

Lemma A.6 (Shortest witness consistency)**.** *Let $\mathcal{N}$ be a (partial) Mealy machine.*
For all $q, r \in Q^{\mathcal{N}}, \forall v \in I^*, \forall w \in I^+$:

$$vw \vdash^s q\#r \implies w \vdash^s \delta(q,v)\#\delta(r,v)$$

Proof. Let $q, r \in Q^{\mathcal{N}}, v, w \in I^*$ such that $vw \vdash^s q\#r$. Let $q' = \delta(q,v)$ and $r' = \delta(r,v)$.

We do a proof by contradiction. Suppose $\neg(w \vdash^s q'\#r')$. Then, by definition of a shortest witness it follows that: either $\neg(w \vdash q'\#r')$ or $\exists \sigma' \in I^* : \sigma' \vdash q'\#r' \wedge |\sigma'| < |w|$.

Case $\neg(w \vdash q'\#r')$:

Since $w \in I^+$, we have that $|w| \geqslant 1$. Then $|v| < v + 1 \leqslant |vw|$.

From the definition of $vw \vdash^s q\#r$, we get that $\neg(\exists \sigma' \in I^* : \sigma' \vdash q'\#r' \wedge |\sigma'| < |vw|)$. Since $|v| < |vw|$, it then follows that $\neg(v \vdash q\#r)$.

So we have that $\neg(v \vdash q\#r)$, and $\neg(w \vdash q\#r)$. Then according to De Morgan's laws, $\neg(v \vdash q\#r \vee w \vdash q'\#r')$, which is also equivalent to $\neg(v \vdash q\#r \vee w \vdash \delta(q,v)\#\delta(r,v))$.

Now we can apply Lemma 3.4. Since it is a bi-implication, having the right side false implies that the left side is false. Then it follows that $\neg(vw \vdash q\#r)$. But we have that $vw \vdash^s q\#r$, this is a contradiction.

The assumption that $\neg(w \vdash q'\#r')$ leads to a contradiction.

Case $\exists \sigma' \in I^* : \sigma' \vdash q'\#r' \wedge |\sigma'| < |w|$:

Let $\sigma' \in I^*$ such that $\sigma' \vdash q'\#r'$, and $|\sigma'| < |w|$. Then it follows that $|v\sigma'| = |v| + |\sigma'| < |v| + |w| = |vw|$

Since $\sigma' \vdash q'\#r'$ (and equivalently, $\sigma' \vdash \delta(q,\sigma')\#\delta(r,\sigma')$), it follows by Lemma 3.4 that $v\sigma' \vdash q\#r$.

So we have shown that $\exists v\sigma' \in I* : v\sigma' \vdash q\#r \wedge |v\sigma'| < |vw|$.

But from the definition of $vw \vdash^s q\#r$, we get that $\neg(\exists v\sigma' \in I* : v\sigma' \vdash q\#r \wedge |v\sigma'| < |vw|)$. This is a contradiction.

The assumption that $\exists \sigma' \in I^* : \sigma' \vdash q'\#r' \wedge |\sigma'| < |vw|$ leads to a contradiction.

Now we have shown that assuming that $\neg(w \vdash^s q'\#r')$. leads to a contradiction in all cases. Then it follows that $w \vdash^s q'\#r'$, and equivalently, $w \vdash^s \delta(q,v)\#\delta(r,v)$. □

A.4 The functional simulation

A.4.1 Functional simulation consistency

Proof for Lemma 3.7

Lemma A.7 (Functional simulation consistency)**.** *Let $\mathcal{T}$ be an observation tree for $\mathcal{N}$ with functional simulation h.*
For all $q, r \in Q^{\mathcal{T}}, v \in I^*$ with $q \xrightarrow[*]{v} r$:

$$h(q) \xrightarrow[*]{v} h(r)$$

Proof. Let $q, r \in Q^{\mathcal{T}}, v \in I^*$ with $q \xrightarrow[*]{v} r$. We do a proof by induction on v.

Base case: $v = \epsilon$

Since $q \xrightarrow[*]{v} r$ we have that $r = \delta(q,v)$. Since $v = \epsilon$, it follows that $r = \delta(q,v) = \delta(q,\epsilon) = q$.

Likewise, we have that $\delta(h(q),v) = \delta(h(q),\epsilon) = h(q)$. Therefore $h(q) \xrightarrow[*]{v} h(q)$, and by extension $h(q) \xrightarrow[*]{v} h(r)$.

Inductive case: $v = wi$

We have that $q \xrightarrow[*]{v} r$. Since $v = wi$, let $r, r' \in Q^{\mathcal{T}}$ such that $q \xrightarrow[*]{w} r' \xrightarrow{i} r$.

From the inductive hypothesis, it follows that $h(q) \xrightarrow[*]{w} h(r')$.

By definition of the functional simulation, it follows from $r' \to r$ that $h(r') \to h(r)$. Then $h(q) \xrightarrow[*]{w} h(r') \to h(r)$. And then since $v = wi$, it also follows that $h(q) \xrightarrow[*]{v} h(r)$.

Then we have shown using induction that $h(q) \xrightarrow[*]{v} h(r)$. □

A.4.2 Observation tree consistency

Proof for Lemma 3.8

Lemma A.8 (Observation tree consistency)**.** *Let $\mathcal{T}$ be an observation tree for $\mathcal{N}$ with functional simulation h, then:*

$$\forall q \in Q^{\mathcal{T}} : q \not\# h(q)$$

Proof. Let $q \in Q^{\mathcal{T}}$, and let $v \in I* : \lambda(q, v) \downarrow$. We show that $\lambda(q, v) = \lambda(h(q), v)$ by a proof by induction.

Base case $v = \epsilon$:

By definition of the λ function, $\lambda(q, v) = \epsilon = \lambda(h(q), v)$.

Inductive case $v = wi$

We have that $\lambda(q, vi) \downarrow$. Then let $r, r' \in Q^{\mathcal{T}}$ and $u, \sigma \in O^*$ such that $q \xrightarrow[*]{w/u} r \xrightarrow{i/o} r'$. Then $u = \lambda(q, w)$ and $o = \lambda(r, i)$.

We get from Lemma 3.7 that $h(q) \xrightarrow[*]{w} h(r)$. We get from the Inductive Hypothesis that $u = \lambda(q, w) = \lambda(h(q), w)$.

From the definition of the functional simulation and the fact that $r \xrightarrow{i/o} r'$, it follows that $h(r) \xrightarrow{i/o} h(r')$, so $\lambda(r, i) = o = \lambda(h(r), i)$.

Then $\lambda(q, v) = \lambda(q, w) \cdot \lambda(r, i) = \lambda(h(q), w) \cdot \lambda(h(r), i) = \lambda(h(q), v)$.

Now we have shown that for all $v \in I* : \lambda(q, v) \downarrow$: $\lambda(q, v) = \lambda(h(q), v)$. Then by definition of non-apartness, it follows that $q \not\# h(q)$. □

A.5 Identification and basis

Lemma A.9 is never directly used, but it plays a role in the proof of Lemma 3.9 and Lemma 3.11.

Lemma A.9 (Complete B)**.** *Let $\mathcal{T}$ be an observation tree for B where B is complete.*

$$\forall \sigma \in A^S : \exists q \in B : q_0^{\mathcal{T}} \xrightarrow[*]{\sigma} q$$

Proof. Let $\sigma \in A^S$ We do a proof by induction on σ.

Base case $\sigma = \epsilon$:

By definition of transitions, $q_0^{\mathcal{T}} \xrightarrow[*]{\epsilon} q_0^{\mathcal{T}}$. Since $\epsilon \in A^S$ (because S always has a q_0^S), it follows by definition of B that $q_0^{\mathcal{T}} \in B$. Therefore with $q = q_0^{\mathcal{T}}$, it follows that $\exists q \in B : q_0^{\mathcal{T}} \xrightarrow[*]{\sigma} q$

Inductive case $\sigma = vi$

By the Inductive Hypothesis, we have that $\exists q' \in B : q_0^{\mathcal{T}} \xrightarrow[*]{v} q'$. Since B is complete, it follows that $\exists q \in Q^{\mathcal{T}} : q' \xrightarrow{i} q$.

Then $q_0^{\mathcal{T}} \xrightarrow[*]{v} q' \xrightarrow{i} q$, and since $\sigma = vi$ also $q_0^{\mathcal{T}} \xrightarrow[*]{\sigma} q$. Since $\sigma \in A^S$ (We may always assume this because A^S is prefix-closed), it follows by definition of B that $q \in B$.

Then we have shown that $\exists q \in B : q_0^{\mathcal{T}} \xrightarrow[*]{\sigma} q$.

□

A.5.1 Bijective f^B

Proof for Lemma 3.9

Lemma A.10 (Bijective f^B)**.** *Let $\mathcal{T}$ be an observation tree for S. Let f^B such that $\forall q \in B : f^B(q) = f(q)$.*

$$(1)\ A^S \text{ is exact} \Longrightarrow f^B : B \to Q^S \text{ is injective}$$
$$(2)\ B \text{ is complete} \Longrightarrow f^B : B \to Q^S \text{ is surjective}$$

Proof. **1. Injective**: Show that if A^S is exact, $\forall q, r \in B : f(q) = f(r) \Longrightarrow q = r$:

We do a proof by contradiction. Suppose that $\exists q, r \in B$ such that $f(q) = f(r)$ and $q \neq r$.

Since $q, r \in B \subseteq Q^{\mathcal{T}}$, it follows by definition of Trees that $\exists v, w \in I^* : q_0^{\mathcal{T}} \xrightarrow{v} q$ and $q_0^{\mathcal{T}} \xrightarrow{w} r$ with $v \neq w$. And it follows by definition of B that $v, w \in A^S$.

From Lemma 3.7, and the fact that $q_0^S = f(q_0^{\mathcal{T}})$, it follows that $q_0^S \xrightarrow{v} f(q)$ and $q_0^S \xrightarrow{w} f(r)$. Then since $f(q) = f(r)$, it follows that $\delta(q_0^S, v) = f(q) = f(r) = \delta(q_0^S, w)$

But then we have that $\exists v, w \in I^* : v \neq w$ where $\delta(q_0^S, v) = \delta(q_0^S, w)$. Then it follows by definition of the state cover that A^S is not exact. But we have that A^S is exact. This is a contradiction, therefore $q = r$.

2. Surjective: Show that if B is complete, $\forall r \in Q^S : \exists q \in B : f(q) = r$

Let $r \in Q^S$. It follows by definition of the state cover A^S, that $\exists \sigma \in A^S$, such that $q_0^S \xrightarrow[*]{\sigma} r$.

By Lemma A.9, we have that $\exists q \in B : q_0^{\mathcal{T}} \xrightarrow[*]{v} q$.

Now we apply Lemma 3.7 with $h = f$ and $q = q_0^{\mathcal{T}}$, $v = \sigma$ and $r = q$. We get that $f(q_0^{\mathcal{T}}) \xrightarrow[*]{\sigma} f(q)$. Then since $q_0^S = f(q_0^{\mathcal{T}})$, it follows that $q_0^S \xrightarrow[*]{\sigma} f(q)$.

Since $q_0^S \xrightarrow[*]{\sigma} r$ and $q_0^S \xrightarrow[*]{\sigma} f(q)$ it follows that $r = f(q)$. Then we have shown that $\exists q \in B : f(q) = r$.

□

A.5.2 Identification

Proof for Lemma 3.10

Lemma A.11 (Identification)**.** *Let $\mathcal{T}$ be an observation tree for S with B is complete and A^S is exact.*
For all $q \in Q^{\mathcal{T}}$:

$$(1)\ q \text{ is identified} \Longrightarrow \forall s \in B : f(q) = f(s) \vee q \# s$$
$$(2)\ C(q) = \{c_q\} \Longrightarrow f(q) = f(c_q)$$

Proof. We get from Lemma 3.9 and the fact that B is complete and A^S is exact, that $f^B : B \to Q^S$ is bijective.

1. From left to right:

Let $q \in Q^{\mathcal{T}}$ such that q is identified. Then by definition of the candidate set, it follows that $C(q) = 1$. Let $c_q \in B$ where $\underline{C(q) = \{c_q\}}$.

Show that $\forall s \in B, s \neq c_q : q \# s \wedge f(q) \neq f(s)$:

Let state $s \in B, s \neq c_q$. Then $s \notin C(q)$, hence by definition of the candidate set, it follows that $q \# s$. Then by Lemma 2.2, it follows that $f(q) \# f(s)$ and therefore $f(q) \neq f(s)$.

Show that $\underline{f(q) = f(c_q)}$:

Proof by contradiction. Suppose $f(q) \neq f(c_q)$. Then it follows by (1) that $\forall s \in B : f(q) \neq f(s)$.

Then $\exists f(q) \in B : \neg(\exists s \in B : f(s) = f(q))$. This violates the surjective property of f^B. Therefore the assumption must be false, it follows that $f(q) = f(c_q)$.

1. From right to left:

Let $q \in Q^{\mathcal{T}}$. We have that $\forall s \in B : f(q) = f(s) \vee q \# s$.

Show that $|C(q)| \leqslant 1$:

Let $s, t \in C(q)$. We show that $s = t$.

Since $s, t \in C(q)$, it follows that $q \not\# s$ and that $q \not\# t$. We have that $f(q) = f(s) \vee q \# s$, and $f(q) = f(t) \vee q \# t$. Then it follows that $f(q) = f(s) = f(t)$.

We have $f(s) = f(t)$, and $s, t \in B$. Then it follows from the injective property that $f(s) = f(t)$ implies $s = t$. So $s = t$.

Then $C(q) \leqslant 1$.

Show that $|C(q)| \geqslant 1$:

Proof by contradiction. Suppose that $C(q) = \varnothing$. Then by definition of the candidate set, it follows that $\forall s \in B : q \# s$. Then it follows according to Lemma 2.2, that $f(q) \# f(s)$ and therefore $f(q) \neq f(s)$.

Then $\exists f(q) \in Q^S : \neg(\exists s \in B : f(q) = f(s))$. Then $f^B : B \to Q^S$ is not surjective. This is a contradiction. Then the assumption must be false. It follows that $|C(q)| \geqslant 1$.

Then $|C(q)| = 1$ and by definition of the candidate set, q is identified.

For 2

This follows from our proof for 1. From left to right: we let $c_q \in B$ such that $C(q) = \{c_q\}$, and we show that $f(q) = f(c_q)$. (Both are underlined.)

□

A.5.3 Basis size

Proof for Lemma 3.11

Lemma A.12 (Basis size)**.** *Let $\mathcal{T}$ be an observation tree for S where B is complete.*

$$(1)\ B \text{ is identified} \implies A^S \text{ is exact}$$
$$(2)\ A^S \text{ is exact} \implies |B| = |Q^S|$$

Proof. **For 1**:

Proof by contradiction. Suppose that $\exists v, w \in A^S, v \neq w : \lambda(q_0^S, v) = \lambda(q_0^S, w)$.

By Lemma A.9 (and the fact that $v, w \in A^S$ and B is complete), it follows that $\exists q, r \in B$ such that $q_0^{\mathcal{T}} \xrightarrow[*]{v} q$ and $q_0^{\mathcal{T}} \xrightarrow[*]{w} r$. Since $v \neq w$, it follows by the definition of Trees that $q \neq r$.

Then from Lemma 3.7 and $f(q_0^{\mathcal{T}}) = q_0^S$, it follows that $q_0^S \xrightarrow[*]{v} f(q)$ and $q_0^S \xrightarrow[*]{w} f(r)$. Then it follows from $\lambda(q_0^S, v) = \lambda(q_0^S, w)$ that $f(q) = f(r)$.

From Lemma 2.3 and the fact that B is identified, it follows that $q = r \vee q \# r$. Then $q \# r$. But then from Lemma 2.2 it follows that $f(q) \# f(r)$ and therefore $f(q) \neq f(r)$.

This is a contradiction. It follows that $\neg(\exists v, w \in I^*, v \neq w : \lambda(q_0^S, v) = \lambda(q_0^S, w))$. Then by definition of state covers, A^S is exact.

For 2:

We have that B is complete and that A^S is exact, then it follows from Lemma 3.9 that $f^B : B \to Q^S$ is bijective. Then it follows that $|B| = |Q^S|$.

□

A.5.4 Identified basis

Proof for Lemma 2.3

Lemma A.13 (Identified basis)**.** *If B is identified, then:*

$$\forall q, r \in B : q = r \vee q \# r$$

Proof. Let $q, r \in B, q \neq r$:

Proof by contradiction, suppose that $\neg(q = r \vee q \# r)$. Then it follows by De Morgan's laws that $q \neq r \wedge q \not\# r$. It is always the case that $q \not\# q$. Then by definition of the candidate set, $q, r \in C(q)$ with $q \neq r$. Then $|C(r)| > 1$.

Since B is identified and $q \in B$, it follows that q is identified. Then $|C(q)| = 1$. This is a contradiction with $|C(r)| > 1$. Then our assumption must be false.

Now we have shown that for all $q, r \in B : q = r \vee q \# r$. □

A.6 Applied Lemmas

The proofs for these are rather trivial if we apply the other Lemmas.

A.6.1 Two functional simulations

Proof for Lemma 3.12

Lemma A.14 (Applied observation tree consistency)**.** *Let $\mathcal{T}$ be an observation tree for both S and $\mathcal{M}$.*

For all $q \in Q^{\mathcal{T}}, \sigma \in I^*$:

$$(1)\ \sigma \vdash f(q) \# g(q) \Longrightarrow \lambda(q_0^{\mathcal{T}}, \sigma) \uparrow$$

and equivalently:

$$(2)\ \lambda(q_0^{\mathcal{T}}, \sigma) \downarrow \Longrightarrow \neg(\sigma \vdash f(q) \# g(q))$$

Proof. Let $q \in Q^T$ and $\sigma \in I^*$ such that $\sigma \vdash f(q) \# g(q)$. We prove **1**. Then it is trivial that **2** is equivalent.

Proof by contradiction: Suppose that $\sigma \vdash f(q) \# g(q)$ and $\lambda(q_0^{\mathcal{T}}, \sigma) \downarrow$. By Lemma 3.8, it follows that $q \not\# f(q)$, and $q \not\# g(q)$. Then since $\lambda(q_0^{\mathcal{T}}, \sigma) \downarrow$, it follows that $\lambda(q, \sigma) = \lambda(f(q), \sigma)$ and $\lambda(q, \sigma) = \lambda(g(q), \sigma)$. Since $\sigma \vdash f(q) \# g(q)$, it follows that $\lambda(f(q), \sigma) \neq \lambda(g(q), \sigma)$.

But then we have that:

$$\lambda(q, \sigma) = \lambda(f(q), \sigma) \neq \lambda(g(q), \sigma) = \lambda(q, \sigma)$$

It is obviously impossible that $\lambda(q, \sigma) \neq \lambda(q, \sigma)$. This is a contradiction, our assumption must be wrong. Therefore, $\lambda(q_0^{\mathcal{T}}, \sigma) \uparrow$. □

A.6.2 Applied transition apartness

Proof for Lemma 3.13

Lemma A.15 (Applied transition apartness)**.** *Let $\mathcal{T}$ be an observation tree for both S and $\mathcal{M}$.*
For all $q, q' \in Q^{\mathcal{T}}$ and $\forall v, w \in I^*$ where $q \xrightarrow{v} q'$:

$$vw \vdash f(q) \# g(q) \iff v \vdash f(q) \# g(q) \vee w \vdash f(q') \# g(q')$$

Proof. Let $q, q' \in Q^{\mathcal{T}}$ and $\forall v, w \in I^*$ where $q \xrightarrow{v} q'$.
We have that $q \xrightarrow{v} q'$. Then from Lemma 3.7, we get that also $f(q) \xrightarrow{v} f(q')$ and $g(q) \xrightarrow{v} g(q')$. Then $\delta(f(q), v) = f(q')$ and $\delta(g(q), v) = g(q')$.
Now our property follows from applying Lemma 3.4 with $v = v$, $w = w$, $q = f(q)$, $r = g(q)$, $\delta(q, v) = f(q')$ and $\delta(r, v) = g(q')$. □

A.6.3 Applied shortest witness consistency

Proof for Lemma 3.14

Lemma A.16 (Applied shortest witness consistency)**.** *Let $\mathcal{T}$ be an observation tree for both S and $\mathcal{M}$.*
For all $q, q' \in Q^{\mathcal{T}}$ and $\forall v \in I*, \forall w \in I^+$ where $q \xrightarrow{v} q'$

$$vw \vdash^s f(q) \# g(q) \Longrightarrow w \vdash^s f(q) \# g(q)$$

Proof. Let $q, q' \in Q^{\mathcal{T}}$ and $\forall v \in I^*$ and $w \in I^+$ where $q \xrightarrow{v} q'$.
We have that $q \xrightarrow{v} q'$. Then from Lemma 3.7, we get that also $f(q) \xrightarrow{v} f(q')$ and $g(q) \xrightarrow{v} g(q')$. Then $\delta(f(q), v) = f(q')$ and $\delta(g(q), v) = g(q')$.
Now our property follows from applying Lemma 3.6 with $v = v$, $w = w$, $q = f(q)$, $r = g(q)$, $\delta(q, v) = f(q')$ and $\delta(r, v) = g(q')$. □

A.7 Application of Theorem 3.1

Proof for Lemma 4.1

Lemma A.17 (Complete tree)**.** *For any test suite T, if $A^S \cdot I^{\leqslant k+1} \subseteq T$, then T generates a tree $\mathcal{T}$ where $B \cup F^{<k}$ is complete.*

Proof.

- Let $s \in B$. Then by definition of B, it follows that $\text{access}(s) \in A^S$. Since I contains only words of length less than $k+1$, $I \subseteq I^{\leqslant k+1}$. Now, since $A^S \cdot I \subseteq T$, it follows that $\delta(q_0^{\mathcal{T}}, \text{access}(s) \cdot i) \downarrow$ for all $i \in I$. Then it follows that $\delta(s, i) \downarrow$ Then by definition, s is complete.

- Let $n < k$ and let $q \in F^n$. By definition of the k-level frontiers, it follows that $\exists s \in B : \exists v \in I^*$ where $s \xrightarrow[+]{v} q$ where $|v| = n + 1$.

  Let I^{n+2} be the set of words in I^* with length $n + 2$. Since $n < k$, we have that $I^{n+2} \subseteq I^{\leqslant k+1}$. Now, since $A^S \cdot I^{n+2} \subseteq T$, it follows that $\delta(q_0^{\mathcal{T}}, \text{access}(s) \cdot v') \downarrow$ for all $v' \in I^{n+2}$. Now let $i \in I$ Then it follows that $\delta(s, vi) \downarrow$ since $vi \in I^{n+2}$. Then it follows that $\delta(q, i) \downarrow$. Then by definition, q is complete.

Then by definition of completeness on sets, it follows that $B \cup F^{\leqslant k}$ is complete. □

Proof for Lemma 4.2

Lemma A.18. *For any test suite T, if $A^S \cdot I^{\leqslant k+1} \cdot D \subseteq T$ where $D \subseteq I^*$, then T generates a tree $\mathcal{T}$ where for all $q \in B \cup F^{\leqslant k}$ and for all $\sigma \in D$, $\delta(q, \sigma) \downarrow$*

Proof. Let $q \in B \cup F^{\leqslant k}$. Then it follows that $\text{access}(q) \in A^S \cdot I^{\leqslant k+1}$. Let $\sigma \in D$. Then $\delta(q_0^{\mathcal{T}}, \text{access}(q) \cdot \sigma) \downarrow$. Then it also follows that $\delta(q, \sigma) \downarrow$. □

Proof for Lemma 4.3

Lemma A.19. *For any test suite T, if $A^S \cdot I^{\leqslant k+1} \odot \mathcal{Y} \subseteq T$ where $\mathcal{Y}$ is a separating family or family of UIOs, then T generates a tree $\mathcal{T}$ where for all $q \in B \cup F^{\leqslant k}$ and for all $\sigma \in W_q$, $\delta(q, \sigma) \downarrow$.*

Proof. Let $q \in B \cup F^{\leqslant k}$. Then it follows that $\text{access}(q) \in A^S \cdot I^{k \leqslant k+1}$.

If $\mathcal{Y}$ is a separating family, let $\sigma \in W_q$. Then by definition of $\odot$, it follows that $\delta(q_0^{\mathcal{T}}, \text{access}(q) \cdot \sigma) \downarrow$. Then it also follows that $\delta(q, \sigma) \downarrow$.

If $\mathcal{Y}$ is a family of UIOs, by definition of $\odot$, it follows that $\delta(q_0^{\mathcal{T}}, \text{access}(q) \cdot UIO(q)) \downarrow$. Then it also follows that $\delta(q, UIO(q)) \downarrow$. □

A.8 Comparison to Vaandrager's Theorem

Proof for Lemma 4.4

A.8.1 Equivalent Assumptions

Lemma A.20 (Equivalent Assumptions)**.** *Let $\mathcal{T}$ be an observation tree. If $B \cup F^{\leqslant k}$ is identified, then:*

$$\forall r \in B, \forall t' \in F^k, \forall t'' \in F^{<k} : r \# t' \implies r \# t'' \vee t' \# t''$$

$$\iff$$

$$\forall q \in F^{<k}, \forall r \in F^k : f(q) = f(r) \vee q \# r$$

Proof. **Left to right**

Let $t' \in F^k, t'' \in F^{<k}$.

We have that t' and t'' are identified since $t', t'' \in B \cup F^{\leqslant k}$. Then let $s, s'' \in B$ such that $C(t') = \{s'\}$ and $C(t'') = \{s''\}$. Then it follows from Lemma 3.10 (2) that $f(t') = f(s')$ and $f(t'') = f(s'')$. Either $s' = s''$ or $s' \neq s''$

Case $s = s''$.

Then $f(t) = f(s) = f(s'') = f(t'')$. Therefore, $f(t') = f(t'')$.

Case $s \neq s''$

Then since $C(t') = \{s'\}$ and $C(t'') = \{s''\}$, it follows by definition of the candidate set that $t' \# s''$ and $t'' \# s'$.

Since $s'' \# t'$ and $t' \in F^k$, $t'' \in F^{<k}$, it follows from the left-hand side of the implication, that $s'' \# t'' \vee t' \# t''$. We already have that $t'' \not\# s''$. Then it must be the case that $t' \# t''$.

Then we have shown that $\forall t \in F^{<k}, \forall t'' \in F^k : f(t) = f(t'') \vee t \# t''$.

Right to left

Let $t \in F^{<k}$ and $t'' \in F^k$. Let $r \in B$ such that $r \# t'$. From the right-hand side of the implication, we get that $f(t') = f(t'') \vee t' \# t''$.

Case $f(t') = f(t'')$.

Proof by contradiction. Suppose that $r \not\# t''$.

Then by definition of the candidate set, $C(t'') = \{r\}$. Then it follows from Lemma 3.10 (2) that $f(t') = f(t'') = f(r)$.

From Lemma 2.2 and the fact that $r \# t'$, we get that $f(r) \# f(t')$, so $f(r) \neq f(t')$.

But then $f(r) = f(t')$ and $f(r) \neq f(t')$. This is a contradiction, the assumption must be wrong. Therefore, $r \# t''$.

Case $t' \# t''$ Trivial.

Then we have shown that $\forall r \in B, \forall t' \in F^k, \forall t'' \in F^{<k} : r \# t' \implies r \# t'' \vee t' \# t''$.

□

A.8.2 Amount of frontier state pairs

Let $m = |I|$ and $n = |F^0|$. And let w be the amount of pairs of states $q, r \in F^{\leqslant k}$ for which we require that $f(q) = f(r) \vee q \# r$.

Show that if $F^{<i}$ complete, $|F^i| = n \cdot m^i$:

Base case $i = 0$: Then $|F^i| = |F^0| = n = n \cdot m^0 = n \cdot m^i$.

Inductive case $i > 0$:

If F^{i-1} is complete, then every state in F^{i-1} has $m = |I|$ direct successors. By definition, $|F^i|$ is exactly the amount of successors of F^{i-1}.

Hence $|F^i| = |F^{i-1}| \cdot m =^{IH} n \cdot m^{i-1} \cdot m = n \cdot m^i$.

For Assumption 4.1:

We require that $\forall q \in F^{<k}, r \in F^k : f(q) = f(r) \vee q\#r$. (Equivalence to Assumption 4.1 is shown in Lemma 4.4)

Then we require for $|F^k| \cdot |F^{<k}|$ states that $f(q) = f(r) \vee q\#r$. Then:

$$\begin{aligned} w &= |F^k| \cdot |F^{<k}| \\ &= |F^k| \cdot \sum_{i=0}^{k-1} |F^i| \\ &= nm^k \cdot \sum_{i=0}^{k-1} |F^i| \\ &= nm^k \cdot \sum_{i=0}^{k-1} nm^i \\ &= n^2 m^k \cdot \sum_{i=0}^{k-1} m^i \\ &= n^2 m^k \cdot \frac{m^k - 1}{m - 1} \\ &= \frac{n^2 m^k (m^k - 1)}{m - 1} \end{aligned}$$

For Assumption 3.1:

We require that $\forall q, r \in F^{\leqslant k}, q \underset{+}{\longrightarrow} r : f(q) = f(r) \vee q\#r$. Let $r \in F^i$.

By definition of the i-level frontier, it follows that r has i (indirect) predecessors in $F^{\leqslant k}$. Then it follows that F^i has $|F^i| \cdot i$ (indirect) predecessors in $F^{\leqslant k}$. Hence we take the sum of $|F^i| \cdot i$ for all frontier layers up to k.

Then:

$$\begin{aligned} w &= \sum_{i=0}^{k} |F^i| i \\ &= \sum_{i=0}^{k} n m^i i \\ &= n \sum_{i=0}^{k} m^i i \\ &= n \cdot \frac{m(km^{k+1} - (k+1)m^k + 1)}{(m-1)^2} \\ &= \frac{nm(km^{k+1} - (k+1)m^k + 1)}{(m-1)^2} \end{aligned}$$