Working Paper Series
ISSN 1177-777X

SYNTHESIS OBSERVATION EQUIVALENCE AND WEAK SYNTHESIS OBSERVATION EQUIVALENCE

Sahar Mohajerani, Robi Malik, Martin Fabian

Working Paper: 03/2012
July 30, 2012

Department of Computer Science
The University of Waikato
Private Bag 3105
Hamilton, 3240
New Zealand

SYNTHESIS OBSERVATION EQUIVALENCE AND WEAK SYNTHESIS OBSERVATION EQUIVALENCE

Sahar Mohajerani
Department of Signals and Systems
Chalmers University of Technology
Göteborg, Sweden
mohajera@chalmers.se

Robi Malik
Department of Computer Science
The University of Waikato
Hamilton, New Zealand
robi@waikato.ac.nz

Martin Fabian
Department of Signals and Systems
Chalmers University of Technology
Göteborg, Sweden
fabian@chalmers.se

July 30, 2012

Abstract

This working paper proposes an algorithm to simplify automata in such a way that compositional synthesis results are preserved in every possible context. It relaxes some requirements of synthesis observation equivalence from previous work, so that better abstractions can be obtained. The paper describes the algorithm, adapted from known bisimulation equivalence algorithms, for the improved abstraction method. The algorithm has been implemented in the DES software tool Supremica and has been used to compute modular supervisors for several large benchmark examples. It successfully computes modular supervisors for systems with more than 10^{12} reachable states.

1 Introduction

Compositional methods are of great interest in *supervisory control theory* [15], firstly in order to find more comprehensible supervisor representations, and secondly to overcome the problem of *state-space explosion* for systems with a large number of components.

Compositional synthesis [7, 10, 12] computes a supervisor for a large discrete event system by repeated *abstraction*. Individual system components are replaced by simpler versions obtained from abstraction, and synchronous composition is computed step-by-step on abstracted components. At each step, partial supervisors are computed, which in the end give a modular supervisor for the original system. In this way, state-space explosion is mitigated, making synthesis possible for very large systems.

Several methods of compositional synthesis exist that differ in how abstractions are computed. *Natural projection* is easy to compute, but it is restrictive and additional conditions must be imposed to ensure synthesis of least restrictive nonblocking supervisors [5, 16]. *Conflict-preserving* abstractions and *observation equivalence* are adequate for the synthesis of nonblocking supervisors, but least restrictiveness is only guaranteed if all observable events are retained in the abstraction [9, 17].

More recently, a stronger version of observation equivalence known as *synthesis observation equivalence* has been proposed [14]. Synthesis observation equivalence is adequate for compositional synthesis of least restrictive supervisors. It has been combined with other abstraction methods and used to compute supervisors for practical applications [12].

This working paper proposes a relaxation of synthesis observation equivalence, called *weak synthesis observation equivalence*, which achieves better abstraction. A polynomial complexity algorithm to compute the abstraction is presented.

This working paper is an extended version of [13]. After the preliminaries in section 2, weak synthesis observation equivalence is defined in section 3. The algorithm to compute it is given in section 4, followed by experimental results in section 5, and concluding remarks in section 6. Proofs of the technical results can be found in the appendix.

2 Preliminaries and Notation

2.1 Events and Languages

Discrete event systems are modelled using events and languages [15]. Events are taken from a finite alphabet Σ, which is partitioned into two disjoint subsets, the

set Σ_c of *controllable* events and the set Σ_u of *uncontrollable* events. The special event $\omega \in \Sigma_c$ denotes *termination*.

The set of all finite *traces* of elements of Σ, including the *empty trace* ε, is denoted by Σ^*. A subset $L \subseteq \Sigma^*$ is called a *language*. The concatenation of two traces $s, t \in \Sigma^*$ is written as st. A trace $s \in \Sigma^*$ is called a *prefix* of $t \in \Sigma^*$, written $s \sqsubseteq t$, if $t = su$ for some $u \in \Sigma^*$. For $\Omega \subseteq \Sigma$, the *natural projection* $P_\Omega \colon \Sigma^* \to \Omega^*$ is the operation that removes from traces $s \in \Sigma^*$ all events not in Ω.

2.2 Nondeterministic Automata

System behaviours are typically modelled by deterministic automata, but nondeterministic automata may arise as intermediate results during abstraction.

Definition 1 A (nondeterministic) finite-state automaton is a tuple $G = \langle \Sigma, Q, \to, Q^\circ \rangle$, where Σ is a finite set of events, Q is a finite set of *states*, $\to \subseteq Q \times \Sigma \times Q$ is the *state transition relation*, and $Q^\circ \subseteq Q$ is the set of *initial states*. G is *deterministic*, if $|Q^\circ| \leq 1$ and $x \xrightarrow{\sigma} y_1$ and $x \xrightarrow{\sigma} y_2$ always implies $y_1 = y_2$.

The transition relation is written in infix notation $x \xrightarrow{\sigma} y$, and is extended to traces in Σ^* by letting $x \xrightarrow{\varepsilon} x$ for all $x \in Q$, and $x \xrightarrow{s\sigma} z$ if $x \xrightarrow{s} y$ and $y \xrightarrow{\sigma} z$ for some $y \in Q$. Furthermore, $x \xrightarrow{s}$ means $x \xrightarrow{s} y$ for some $y \in Q$, and $x \to y$ means $x \xrightarrow{s} y$ for some $s \in \Sigma^*$. These notations also apply to state sets and to automata: $X \xrightarrow{s} Y$ for $X, Y \subseteq Q$ means $x \xrightarrow{s} y$ for some $x \in X$ and $y \in Y$, and $G \xrightarrow{s}$ means $Q^\circ \xrightarrow{s}$, etc. The *accepted language* of automaton G is $\mathcal{L}(G) = \{\, s \in \Sigma^* \mid G \xrightarrow{s} \,\}$.

The termination event ω marks the completion of tasks. It is required to be in the alphabet of every automaton, and states reached by ω cannot have any outgoing transitions. That is, if $x \xrightarrow{\omega} y$ then $y \xrightarrow{\sigma}$ does not hold for any $\sigma \in \Sigma$. Thus, ω only occurs as the final event of traces accepted by an automaton. The traditional set of marked states is $Q^\omega = \{\, x \in Q \mid x \xrightarrow{\omega} \,\}$ in this notation. For graphical simplicity, states in Q^ω are shaded in the figures of this working paper instead of explicitly showing ω-transitions.

When automata are brought together to interact, synchronisation occurs on shared events occurring synchronously or not at all. This is modelled by *synchronous composition* [8].

Definition 2 Let $G_1 = \langle \Sigma_1, Q_1, \to_1, Q_1^\circ \rangle$ and $G_2 = \langle \Sigma_2, Q_2, \to_2, Q_2^\circ \rangle$ be two automata. The *synchronous composition* of G_1 and G_2 is defined as

$$G_1 \parallel G_2 = \langle \Sigma_1 \cup \Sigma_2, Q_1 \times Q_2, \to, Q_1^\circ \times Q_2^\circ \rangle \tag{1}$$

where

$$(x_1, x_2) \xrightarrow{\sigma} (y_1, y_2) \text{ if } \sigma \in \Sigma_1 \cap \Sigma_2,\ x_1 \xrightarrow{\sigma}_1 y_1,\ x_2 \xrightarrow{\sigma}_2 y_2\,; \tag{2}$$

$$(x_1, x_2) \xrightarrow{\sigma} (y_1, x_2) \text{ if } \sigma \in \Sigma_1 \setminus \Sigma_2,\ x_1 \xrightarrow{\sigma}_1 y_2\,; \tag{3}$$

$$(x_1, x_2) \xrightarrow{\sigma} (x_1, y_2) \text{ if } \sigma \in \Sigma_2 \setminus \Sigma_1,\ x_2 \xrightarrow{\sigma}_2 y_2\,. \tag{4}$$

Another common automaton operation is the *quotient* modulo an equivalence relation on the state set.

Definition 3 Let $G = \langle \Sigma, Q, \rightarrow, Q^\circ \rangle$ be an automaton and let $\sim \subseteq Q \times Q$ be an equivalence relation. The *quotient automaton* of G modulo $\sim$ is

$$G/\!\sim = \langle \Sigma, Q/\!\sim, \rightarrow/\!\sim, \tilde{Q}^\circ \rangle\,, \tag{5}$$

where $\rightarrow/\!\sim \;=\; \{\, [x] \xrightarrow{\sigma} [y] \mid x \xrightarrow{\sigma} y \,\}$ and $\tilde{Q}^\circ = \{\, [x^\circ] \mid x^\circ \in Q^\circ \,\}$. Here, $[x] = \{\, x' \in Q \mid x \sim x' \,\}$ denotes the *equivalence class* of $x \in Q$, and $Q/\!\sim \;= \{\, [x] \mid x \in Q \,\}$ is the set of all equivalence classes modulo $\sim$.

2.3 Supervisory Control Theory

Given a *plant* automaton G and a *specification* automaton K, *supervisory control theory* [15] provides a method to synthesise a supervisor that restricts the behaviour of the plant such that the specification is always fulfilled. Two common requirements for the supervisor are *controllability* and *nonblocking*.

Definition 4 Let G and K be two automata using the same alphabet Σ. K is *controllable* with respect to G if, for every trace $s \in \Sigma^*$, every state x of K, and every uncontrollable event $\upsilon \in \Sigma_\mathrm{u}$ such that $K \xrightarrow{s} x$ and $G \xrightarrow{s\upsilon}$, it holds that $x \xrightarrow{\upsilon}$ in K.

Definition 5 An automaton $G = \langle \Sigma, Q, \rightarrow, Q^\circ \rangle$ is *nonblocking*, if for every state $x \in Q$ and every trace $s \in (\Sigma \setminus \{\omega\})^*$ such that $G \xrightarrow{s} x$ there exists $t \in \Sigma^*$ such that $x \xrightarrow{t\omega}$.

For a deterministic plant G, it is well-known [15] that there exists a supremal controllable and nonblocking sublanguage of $\mathcal{L}(G)$, which represents the *least restrictive* feasible supervisor. Algorithmically, it is more convenient to perform synthesis on the automaton G instead of this language, or more precisely on the lattice of *subautomata* of G [4]. This approach also works for nondeterministic automata.

Definition 6 [7] $G_1 = \langle \Sigma, Q_1, \rightarrow_1, Q_1^\circ \rangle$ is a *subautomaton* of $G_2 = \langle \Sigma, Q_2, \rightarrow_2, Q_2^\circ \rangle$, written $G_1 \subseteq G_2$, if $Q_1 \subseteq Q_2$, $\rightarrow_1 \subseteq \rightarrow_2$, and $Q_1^\circ \subseteq Q_2^\circ$.

Theorem 1 [7] Every deterministic automaton G has a supremal controllable and nonblocking subautomaton,

$$\sup\mathcal{CN}(G) = \sup\{\, K \subseteq G \mid K \text{ is controllable with respect to } G \text{ and nonblocking} \,\}\,. \tag{6}$$

Here, the supremal element is defined based on the subautomaton relationship (definition 6). The result is equivalent to that of traditional supervisory control theory [15]. That is, $\sup\mathcal{CN}(G)$ represents the behaviour of the least restrictive supervisor that disables only controllable events in G such that nonblocking is ensured.

The synthesis result $\sup\mathcal{CN}(G)$ can be computed by removing blocking and uncontrollable states from the plant, until a fixpoint is reached, and restricting the original automaton G to these states.

Definition 7 [10] The *restriction* of $G = \langle \Sigma, Q, \rightarrow, Q^\circ \rangle$ to $X \subseteq Q$ is

$$G_{|X} = \langle \Sigma, Q, \rightarrow_{|X}, Q^\circ \cap X \rangle\,, \tag{7}$$

where $\rightarrow_{|X} = \{\, (x,\sigma,y) \in \rightarrow \mid x, y \in X \,\} \cup \{\, (x,\omega,y) \in \rightarrow \mid x \in X \,\}$.

Note that restriction only removes transitions, not states. Moreover, transitions with the termination event ω are retained even if their successor state is not contained in X. Typically, some states become unreachable after restriction, and these states can be removed, but this is not considered further in this working paper.

Definition 8 [10] The *synthesis step operator* $\Theta_G \colon 2^Q \rightarrow 2^Q$ for $G = \langle \Sigma, Q, \rightarrow, Q^\circ \rangle$ is defined as $\Theta_G(X) = \Theta_G^{\mathrm{cont}}(X) \cap \Theta_G^{\mathrm{nonb}}(X)$, where

$$\Theta_G^{\mathrm{cont}}(X) = \{\, x \in X \mid \text{for all } \sigma \in \Sigma_{\mathrm{u}},\ x \xrightarrow{\sigma} y \text{ implies } y \in X \,\}\,;$$
$$\Theta_G^{\mathrm{nonb}}(X) = \{\, x \in X \mid x \xrightarrow{t\omega}_{|X} \text{ for some } t \in \Sigma^* \,\}\,.$$

Θ_G^{cont} captures controllability, and Θ_G^{nonb} captures nonblocking. The synthesis result for G is obtained by restricting G to the greatest fixpoint of Θ_G.

Theorem 2 [10] Let $G = \langle \Sigma, Q, \rightarrow, Q^\circ \rangle$ be a deterministic automaton. The synthesis step operator Θ_G has a greatest fixpoint $\mathrm{gfp}\Theta_G = \hat{\Theta}_G \subseteq Q$, such that $G_{|\hat{\Theta}_G}$ is the greatest subautomaton of G that is both controllable with respect to G and nonblocking, i.e.,

$$\sup\mathcal{CN}(G) = G_{|\hat{\Theta}_G}\,. \tag{8}$$

If the state set Q is finite, the sequence $X^0 = Q$, $X^{i+1} = \Theta_G(X^i)$ reaches this fixpoint in a finite number of steps, i.e., $\hat{\Theta}_G = X^n$ for some $n \geq 0$.

2.4 Compositional Synthesis

Most discrete event systems are *modular* and consist of several interacting components. Then the synthesis problem is to find a least restrictive, controllable and nonblocking supervisor for the synchronous composition of a set of plants

$$\mathcal{G} = \{G_1, G_2, \ldots, G_n\} \,. \tag{9}$$

Compositional methods seek to build the synchronous composition incrementally, replacing individual components G_i by simpler *abstractions* G_i'. Such simplification typically exploits a set $\Upsilon \subseteq \Sigma$ of *local* events. These events are used only in the automaton being abstracted and contribute substantially to its simplification.

The abstraction relation must ensure that the results obtained from the abstracted model are the same as for the original model. An appropriate condition that works for compositional synthesis is *synthesis abstraction*.

Definition 9 [14] Let G and H be deterministic automata with alphabet Σ. Then H is a *synthesis abstraction* of G with respect to $\Upsilon \subseteq \Sigma$, written $G \lesssim_{\mathrm{synth},\Upsilon} H$, if for every deterministic automaton $T = \langle \Sigma_T, Q_T, \rightarrow_T, Q_T^\circ \rangle$ such that $\Sigma_T \cap \Upsilon = \emptyset$ the following holds,

$$\mathcal{L}(G \parallel \mathrm{sup}\mathcal{CN}(H \parallel T)) = \mathcal{L}(G \parallel \mathrm{sup}\mathcal{CN}(G \parallel T)) \,. \tag{10}$$

Synthesis abstraction requires that the supervisor synthesised from the abstracted automaton H, in combination with every possible rest of the system T, yields the same language when controlling the system, as would the supervisor synthesised from the original automaton G together with T.

3 Synthesis Observation Equivalence

Synthesis abstraction describes, in a general way, the kind of abstraction feasible for compositional synthesis. This section presents a concrete method to simplify a given automaton such that synthesis abstraction is satisfied, and the following section presents an algorithm to implement this method.

The proposed method is based on *bisimulation* and *observation equivalence*, which are standard examples of branching equivalences [11]. For two states to be equivalent, they must have the same nondeterministic future. This requirement is described using an equivalence relation that is *stable* with respect to certain transition relations.

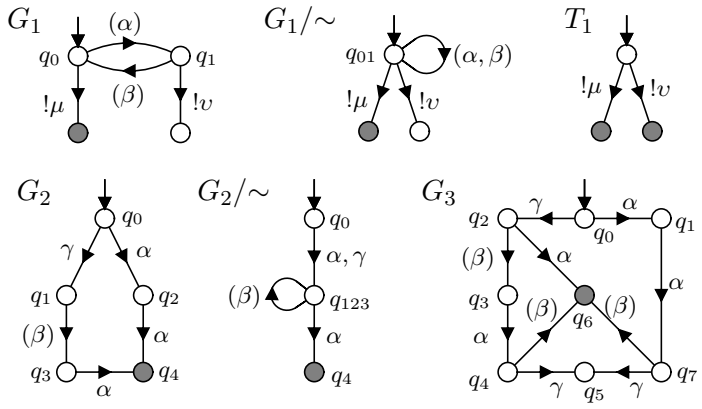

Figure 1: Example Automata. Uncontrollable events are prefixed with !, and local events have parentheses around them.

Definition 10 Let $\rightarrow \subseteq X \times X$ be a relation on a set X. An equivalence relation $\sim \subseteq X \times X$ is *stable* with respect to $\rightarrow$, if for all $x_1, x_2, y_1 \in X$ such that $x_1 \sim x_2$ and $x_1 \rightarrow y_1$ there exists $y_2 \in X$ such that $x_2 \rightarrow y_2$ and $y_1 \sim y_2$.

Definition 11 Let $G = \langle \Sigma, Q, \rightarrow, Q^\circ \rangle$ be an automaton. An equivalence relation $\sim \subseteq Q \times Q$ is called a *bisimulation* on G, if $\sim$ is stable with respect to $\stackrel{\sigma}{\rightarrow}$ for all $\sigma \in \Sigma$.

Definition 12 Let $G = \langle \Sigma, Q, \rightarrow, Q^\circ \rangle$ be an automaton with $\Sigma = \Omega \,\dot\cup\, \Upsilon$. An equivalence relation $\sim \subseteq Q \times Q$ is called an *observation equivalence* on G with respect to Υ, if $\sim$ is stable with respect to $\stackrel{\sigma}{\Rightarrow}$ for all $\sigma \in \Sigma$, where $x \stackrel{\sigma}{\Rightarrow} y$ if and only if $x \xrightarrow{t_1 P_\Omega(\sigma) t_2} y$ for some $t_1, t_2 \in \Upsilon^*$.

Unlike bisimulation, observation equivalence takes local events into account. Projection P_Ω is used in the definition of $\stackrel{\sigma}{\Rightarrow}$ to ensure that it covers both shared events $\sigma \in \Omega$ and local events $\sigma \in \Upsilon$.

Bisimulation and observation equivalence preserve all temporal logic properties [3]. Once an equivalence $\sim$ on G is found, the quotient automaton $G/\sim$ can be considered as an abstraction. For bisimulation this results in a synthesis abstraction, but it does not for observation equivalence [14].

Example 1 [14] Consider automata G_1 and T_1 in figure 1, where $\Upsilon = \{\alpha, \beta\}$ and $\Sigma_u = \{!\mu, !\upsilon\}$. States q_0 and q_1 are observation equivalent and merging them results in $G_1/\sim$. However, $G_1/\sim \parallel T_1$ does not have the same least restrictive

supervisor as $G_1 \parallel T_1$. A supervisor for $G_1 \parallel T_1$ can disable α to prevent blocking via $!\upsilon$, but after merging q_0 and q_1, disabling α is not enough to prevent the dangerous uncontrollable event $!\upsilon$.

While observation equivalence does not lead to synthesis abstraction in general, it can be strengthened [14] such that it does.

Definition 13 Let $G = \langle \Sigma, Q, \rightarrow, Q^\circ \rangle$ be an automaton with $\Sigma = \Omega \,\dot\cup\, \Upsilon$. An equivalence relation $\sim \subseteq Q \times Q$ is a *synthesis observation equivalence* on G with respect to Υ, if $\sim$ is stable with respect to $\overset{\Upsilon}{\Rightarrow}_{\mathrm{soe}}$, to $\overset{\sigma}{\Rightarrow}_{\mathrm{soe}}$ for each $\sigma \in \Sigma_{\mathrm{c}} \cap \Omega$, and to $\overset{\upsilon}{\Rightarrow}_{\mathrm{u}}$ for each $\upsilon \in \Sigma_{\mathrm{u}}$, defined as follows.

- $x \overset{\Upsilon}{\Rightarrow}_{\mathrm{soe}} y$ if there exists a path $x = z_0 \overset{\tau_1}{\rightarrow} \cdots \overset{\tau_k}{\rightarrow} z_k = y$ such that $\tau_1, \ldots, \tau_k \in \Upsilon$, and $\tau_j \in \Sigma_{\mathrm{c}}$ implies $x \sim z_j$ or $j = k$.
- $x \overset{\sigma}{\Rightarrow}_{\mathrm{soe}} y$ if there exists a path $x = z_0 \overset{\tau_1}{\rightarrow} \cdots \overset{\tau_k}{\rightarrow} z_k \overset{\sigma}{\rightarrow} y$ such that $\tau_1, \ldots, \tau_k \in \Upsilon$, and $\tau_j \in \Sigma_{\mathrm{c}}$ implies $x \sim z_j$.
- $x \overset{\upsilon}{\Rightarrow}_{\mathrm{u}} y$ if $x \xrightarrow{t_1 P_\Omega(\upsilon) t_2} y$ for some $t_1, t_2 \in (\Sigma_{\mathrm{u}} \cap \Upsilon)^*$.

Definition 13 modifies observation equivalence based on event types. Uncontrollable events are treated by $\Rightarrow_{\mathrm{u}}$ in the same way as in observation equivalence, except that the local events on the path must all be uncontrollable. Controllable events can be preceded by local events according to $\Rightarrow_{\mathrm{soe}}$, provided that states reached by controllable local events are equivalent to the start state of the path.

Example 2 Consider automaton G_2 in figure 1, where all events are controllable and $\Upsilon = \{\beta\}$. The equivalence relation $\sim$ with $q_1 \sim q_2 \sim q_3$ is a synthesis observation equivalence. For example, the transition $q_2 \overset{\alpha}{\rightarrow} q_4$ is matched by $q_1 \overset{\beta}{\rightarrow} q_3 \overset{\alpha}{\rightarrow} q_4$ where state q_3, reached by the local controllable event β, is equivalent to q_2. Merging the equivalent states results in the synthesis observation equivalent abstraction $G_2/\!\sim$ shown in figure 1.

The definition of $\overset{\sigma}{\Rightarrow}_{\mathrm{soe}}$ does not allow any local events *after* the controllable event σ. This is not necessary, and the condition can be relaxed as follows.

Definition 14 Let $G = \langle \Sigma, Q, \rightarrow, Q^\circ \rangle$ be an automaton with $\Sigma = \Omega \,\dot\cup\, \Upsilon$. An equivalence relation $\sim \subseteq Q \times Q$ is a *weak synthesis observation equivalence* on G with respect to Υ, if $\sim$ is stable with respect to $\overset{\Upsilon}{\Rightarrow}_{\mathrm{wsoe}}$, to $\overset{\sigma}{\Rightarrow}_{\mathrm{wsoe}}$ for each $\sigma \in \Sigma_{\mathrm{c}} \cap \Omega$, and to $\overset{\upsilon}{\Rightarrow}_{\mathrm{u}}$ for each $\upsilon \in \Sigma_{\mathrm{u}}$.

- $x \overset{\Upsilon}{\Rightarrow}_{\mathrm{wsoe}} y$ if $x \overset{\Upsilon}{\Rightarrow}_{\mathrm{soe}} z \overset{\Upsilon}{\Rightarrow}_{\mathrm{c}} y$ for some $z \in Q$.

- $x \overset{\sigma}{\Rightarrow}_{\mathrm{wsoe}} y$ if $x \overset{\sigma}{\Rightarrow}_{\mathrm{soe}} z \overset{\Upsilon}{\Rightarrow}_{\mathrm{c}} y$ for some $z \in Q$.
- $x \overset{\Upsilon}{\Rightarrow}_{\mathrm{c}} y$ if there exists a path $x = z_0 \overset{\tau_1}{\rightarrow} \cdots \overset{\tau_k}{\rightarrow} z_k = y$ such that $\tau_1, \ldots, \tau_k \in \Upsilon$, and $z_j \overset{u}{\rightarrow} z'$ for $u \in (\Sigma_{\mathrm{u}} \cap \Upsilon)^*$ implies $z' \sim z_i$ for some $0 \leq i \leq k$, and $z_j \overset{v}{\Rightarrow}_{\mathrm{u}} z'$ for $v \in \Sigma_{\mathrm{u}} \cap \Omega$ implies $y \overset{v}{\Rightarrow}_{\mathrm{u}} z''$ for some $z'' \sim z'$.

The modified relation $\Rightarrow_{\mathrm{wsoe}}$ allows for a path of local events after a controllable event, if local uncontrollable transitions outgoing from the path lead to a state equivalent to a state on the path, and shared uncontrollable transitions are also possible in the end state of the path.

Example 3 Consider automaton G_3 in figure 1, with all events controllable and $\Upsilon = \{\beta\}$. An equivalence relation with $q_1 \sim q_2 \sim q_3$ and $q_4 \sim q_7$ is a weak synthesis observation equivalence. For example, transition $q_2 \overset{\alpha}{\rightarrow} q_6$ is matched by $q_1 \overset{\alpha}{\rightarrow} q_7 \overset{\beta}{\rightarrow} q_6$, and state q_7 has no uncontrollable transitions outgoing. Note that states q_1 and q_2 are not synthesis observation equivalent, because the path $q_1 \overset{\alpha}{\rightarrow} q_7 \overset{\beta}{\rightarrow} q_6$ does not satisfy the conditions for $\overset{\alpha}{\Rightarrow}_{\mathrm{soe}}$.

As shown in appendix B, every synthesis observation equivalence also is a weak synthesis observation equivalence. Therefore, the following result confirms that both methods are feasible for compositional synthesis.

Theorem 3 Let $G = \langle \Sigma, Q, \rightarrow, Q^\circ \rangle$ be a deterministic automaton with $\Upsilon \subseteq \Sigma$, and let $\sim$ be a weak synthesis observation equivalence on G with respect to Υ such that $G/\!\sim$ is deterministic. Then $G \lesssim_{\mathrm{synth},\Upsilon} G/\!\sim$.

The proof follows from proposition 4 and proposition 6 in appendix A.

4 Algorithm

Given an automaton $G = \langle \Sigma, Q, \rightarrow, Q^\circ \rangle$ and a set Υ of local events, a coarsest weak synthesis observation equivalence relation can be computed by a partition refinement algorithm similar to [6]. This algorithm represents an equivalence relation as a *partition*, i.e., a set of *equivalence classes* each representing a set of equivalent states. The algorithm starts with an *initial partition* consisting of a single equivalence class, which is iteratively refined until a stable partition is reached. At each step, a *split* is performed on each known equivalence class C for each relation $\Rightarrow$ for which stability is required, separating states x with $x \Rightarrow C$ from other states. This principle is shown in algorithm 1.

Algorithm 1 Weak Synthesis Observation Equivalence

```
1: input G = ⟨Σ, Q, →, Q°⟩
2: partition ← {Q}
3: repeat
4:    for all C ∈ partition do
5:       for all σ ∈ Σ do
6:          SplitOn(partition, C, σ)
7:       end for
8:    end for
9: until there has been no further split
10: return partition
```

The bisimulation algorithm [6] performs clever bookkeeping when classes are split, which reduces the need to check whether further splits are necessary and ensures an overall time complexity of $O(|\rightarrow| \log |Q|)$. For observation equivalence, the *transitive closure* of the local event transitions needs to be computed, and this transitive closure computation dominates complexity. A partition based on observation equivalence can be computed in $O(|Q|^3)$ time complexity [2].

The partition refinement algorithm uses several data structures to facilitate the splitting of classes [6]. Each equivalence class is an object containing a list of the states in the class, and each state has a reference back to the class containing it. In addition, each equivalence class has a *split list* containing states to be split off from it.

The *SplitOn* algorithm (algorithm 2) performs the splitting for paths leading to a target class C, called a *splitter*. States with a path to the *splitter* based on each relation $\Rightarrow_{\mathrm{wsoe}}$ and $\Rightarrow_{\mathrm{u}}$ in definition 14 are separated from states without such a path. This is done by visiting each state *end* in the *splitter* and searching backwards for all states *src* with appropriate paths to *end*. These states are put in the split list of their class. After exploring the predecessors of all *end* states, the split lists are checked in lines 12–16. Classes with an empty split list or a split list containing all states in the class are left unchanged, other classes are split and replaced by two new classes.

For uncontrollable events, the source states for $\Rightarrow_{\mathrm{u}}$ are found by a standard backwards search (lines 2–6), whereas for controllable events a special procedure *BS* is used to follow the paths generated by $\Rightarrow_{\mathrm{wsoe}}$ (lines 8–10).

The procedure *BS* (algorithm 3) performs a backward search for a given controllable event σ and *end* state to find paths $x \Rightarrow_{\mathrm{soe}} z \Rightarrow_{\mathrm{c}} end$. It uses a *queue* of search records $\langle current, part, startclass \rangle$, each containing a *current* state, whether the search is in the first ($\Rightarrow_{\mathrm{soe}}$) or second ($\Rightarrow_{\mathrm{c}}$) *part* of the path, and the *startclass*

Algorithm 2 $SplitOn(partition \subseteq 2^Q, splitter \subseteq Q, \sigma \in \Sigma)$

```
1: if σ ∈ Σ_u then
2:    for all end ∈ splitter do
3:       for all src ⇒^σ_u end do
4:          move src to split list in [src]
5:       end for
6:    end for
7: else
8:    for all end ∈ splitter do
9:       BS(σ, end)
10:   end for
11: end if
12: for all class ∈ partition do
13:   if class has a non-trivial split list then
14:      split class and update partition
15:   end if
16: end for
```

(class of the yet unknown start state x) of the path. The search starts with the *end* state, in the second part of the path, and with an unassigned *startclass*, so the queue is initialised with the search record $\langle end, 2, \mathsf{none}\rangle$ in line 1.

When exploring a *current* state in the first part of the path, it is first checked whether this state can be the start of a path generated by $\Rightarrow_{\mathrm{soe}}$. This is possible if it belongs to the *startclass*, or if the *startclass* is unassigned, and in this case the *current* state is marked as a candidate to be split off from its class (lines 5–7).

Afterwards the loop in lines 8–14 scans all local transitions leading to the *current* state. If the event is uncontrollable, a new search record with the previous *startclass* is created in line 10. If the event is controllable, then based on definition 13 the *current* state must be equivalent to the yet unknown start state x of the path. If the *startclass* is unassigned or the same as the class of *current*, then *current* can potentially be x, so its class is used to form a new search record in line 12.

If the algorithm is in the second part of the path, it checks for possible predecessors according to $\Rightarrow_{\mathrm{c}}$. This is only needed for weak synthesis observation equivalence; synthesis observation equivalence is checked by the same algorithm if lines 16–32 are deleted from *BS*. These lines check, for each local transition leading to the *current* state, whether the source state *src* is *controllable*. This is done by exploring all states reachable by traces of local uncontrollable events. If one of these states is not equivalent to the *src*, *current*, or *end* state, or has a shared uncontrollable outgoing transition to a state with no matching state reachable from the

Algorithm 3 Backward Search $BS(\sigma \in \Sigma_{\mathrm{c}}, end \in Q)$

1: $queue \leftarrow \{\langle end, 2, \mathsf{none}\rangle\}$
2: **while** $queue \neq \emptyset$ **do**
3: remove $\langle current, part, startclass\rangle$ from *queue*
4: **if** $part = 1$ **then**
5: **if** $startclass \in \{[current], \mathsf{none}\}$ **then**
6: move *current* to split list in $[current]$
7: **end if**
8: **for all** transitions $src \xrightarrow{\upsilon} current$ with $\upsilon \in \Upsilon$ **do**
9: **if** $\upsilon \in \Sigma_{\mathrm{u}}$ **then**
10: add $\langle src, 1, startclass\rangle$ to *queue*
11: **else if** $startclass \in \{[current], \mathsf{none}\}$ **then**
12: add $\langle src, 1, [current]\rangle$ to *queue*
13: **end if**
14: **end for**
15: **else**
16: **for all** transitions $src \xrightarrow{\upsilon} current$ with $\upsilon \in \Upsilon$ **do**
17: $controllable \leftarrow \mathsf{true}$
18: **for all** $src \xrightarrow{u} succ$ with $u \in (\Sigma_{\mathrm{u}} \cap \Upsilon)^*$ **do**
19: **if** $succ \notin [src] \cup [current] \cup [end]$ **then**
20: $controllable \leftarrow \mathsf{false}$
21: **else**
22: **for all** $succ \xrightarrow{\gamma} succ'$ with $\gamma \in \Sigma_{\mathrm{u}} \cap \Omega$ **do**
23: **if not** $[end] \overset{\gamma}{\Rightarrow}_{\mathrm{u}} [succ']$ **then**
24: $controllable \leftarrow \mathsf{false}$
25: **end if**
26: **end for**
27: **end if**
28: **end for**
29: **if** *controllable* **then**
30: add $\langle src, 2, \mathsf{none}\rangle$ to *queue*
31: **end if**
32: **end for**
33: **if** $\sigma \in \Upsilon$ **then**
34: add $\langle current, 1, \mathsf{none}\rangle$ to *queue*
35: **else**
36: **for all** transitions $src \xrightarrow{\sigma} current$ **do**
37: add $\langle src, 1, \mathsf{none}\rangle$ to *queue*
38: **end for**
39: **end if**
40: **end if**
41: **end while**

end class, then the *src* state is not *controllable*. Otherwise, a new search record is created in line 30. The condition checked here is stronger than $\Rightarrow_{\mathrm{c}}$ in definition 14, which allows the target states of uncontrollable local transitions to be anywhere along the second part of the path. The algorithm still results in a weak synthesis observation equivalence relation, but not necessarily a coarsest one, as shown in appendix C. An exact implementation of $\Rightarrow_{\mathrm{c}}$ requires search records to store complete paths, making the algorithm exponential.

Next it is checked whether it is possible to move from the second part of the path to the first. This is possible if the event σ under consideration is local (line 34), or if there is a σ-transition to the *current* state (lines 36–38).

The algorithm terminates when the *queue* of search records is empty. To prevent duplicates, the *queue* is linked to a hash set to ensure that search records that have been enqueued once are never added to the *queue* again. The hash set is reset for each split operation, i.e., before line 8 in Algorithm 2.

Complexity. In the worst case, the main loop in line 3 of algorithm 1 is executed once for each state, giving up to $|Q|$ iterations. Inside the loop, a split on each class is performed. This causes each state to be processed once for each event, using either the loop in lines 2–6 or 8–10 of algorithm 2. The bodies of these loops are executed $|\Sigma||Q|$ times in total during each iteration of the main loop of algorithm 1. The splitting of classes after line 12 can be executed in lower complexity using the data structures outlined above.

The loop in lines 2–6 of algorithm 2 can be executed in $O(|Q|^2)$ time, assuming the transition relation $\Rightarrow_{\mathrm{u}}$ has been computed in advance. This is dominated by the loop in lines 8–10 which calls algorithm *BS*.

In the worst case, algorithm *BS* visits two search records for each combination of a state and class, i.e., up to $2|Q|^2$ search records. Each time, it executes either the loop in lines 8–14 or 16–32. The loop in lines 8–14 visits all local incoming transitions to a state, up to $|Q|$ operations if the local transitions are appropriately stored in advance. The loop in lines 16–32 also processes up to $|Q|$ local predecessor states, however each time the loop in lines 18–28 must be executed, potentially increasing complexity. Fortunately, this can be avoided by caching. The $\Rightarrow_{\mathrm{u}}$-successors of the *end* class can be computed in advance, and it can be checked for each state *src* whether it has exactly one successor class reachable by local uncontrollable events that is different from the class of *src* and from the *end* class, and that also passes the test in lines 22–26. By caching this successor class, it is possible to execute the loop in lines 18–28 only once for each state during the execution of the algorithm 3. With this caching, the complexity of algorithm *BS* is $O(|Q|^3)$.

Therefore, the execution of algorithm 1 involves $O(|Q|)$ iterations of the main loop, each performing $O(|\Sigma||Q|)$ search operations with of $O(|Q|^3)$ complexity.

Table 1: Experimental Results

			SOE		WSOE	
Model	**Aut**	**States**	**Time**	**States**	**Time**	**States**
agv	16	$2.6{\cdot}10^7$	17.8 s	107747	18.2 s	106169
agvb	17	$2.3{\cdot}10^7$	11.7 s	83577	11.5 s	82353
aip0alps	35	$3.0{\cdot}10^8$	0.9 s	867	0.9 s	867
fencaiwon09b	31	$8.9{\cdot}10^7$	0.1 s	73	0.1 s	73
fms_2003	31	$1.4{\cdot}10^7$	83.6 s	673868	69.7 s	444922
koordwsp_b	24	$1.1{\cdot}10^7$	0.5 s	756	0.4 s	743
tbed_noderailb	84	$3.1{\cdot}10^{12}$	5.7 s	18134	4.4 s	18134
tbed_uncont	84	$3.6{\cdot}10^{12}$	5.0 s	9148	4.4 s	9148

The worst-case time complexity to calculate a coarsest synthesis observation equivalence or a weak synthesis observation equivalence relation using this algorithm is $O(|\Sigma||Q|^5)$.

5 Experimental results

The synthesis observation equivalence and weak synthesis observation equivalence algorithms have been implemented in the DES software tool *Supremica* [1] and used within a compositional supervisor synthesis algorithm that computes modular supervisors [12].

This program has been used to compute synthesis abstractions for a set of benchmark examples that include complex industrial models and case studies taken from various application areas such as manufacturing systems and automotive body electronics. The automata in each example are iteratively composed and simplified, until a final abstraction is obtained and passed on to standard synthesis. All tests were run on a standard desktop PC using a single core 2.66 GHz microprocessor.

Table 1 shows for each test case the number of automata (Aut) in the model and the size of the reachable state space (States). It also shows the total runtime of compositional synthesis (Time) and the number of states in the final abstraction passed on to standard synthesis (States), when using synthesis observation equivalence (SOE) or weak synthesis observation equivalence (WSOE).

Supervisors can be calculated for all models in less than two minutes, with memory usage of no more than 600 MB. The size of the models is substantially reduced compared to the size of the original systems. Weak synthesis observation equivalence gives slightly less states than synthesis observation equivalence with about the same computational cost.

All examples are too large for supervisors to be computed by standard synthesis

alone, and abstraction using only bisimulation results in a final abstraction with at least $2 \cdot 10^6$ states for all test cases.

6 Conclusions

Weak synthesis observation equivalence has been introduced as a means of abstraction for compositional synthesis algorithms. Weak synthesis observation equivalence allows for better abstraction than previously possible with synthesis observation equivalence. A polynomial complexity algorithm for synthesis observation equivalence and weak synthesis observation equivalence has been proposed and implemented in the DES software tool Supremica. The experimental results show that the algorithm can compute abstractions of automata with several thousand states, making it possible to construct modular supervisors for systems with more than 10^{12} reachable states.

References

[1] Knut Åkesson, Martin Fabian, Hugo Flordal, and Robi Malik. Supremica—an integrated environment for verification, synthesis and simulation of discrete event systems. In *Proceedings of the 8th International Workshop on Discrete Event Systems, WODES'06*, pages 384–385, Ann Arbor, MI, USA, July 2006.

[2] Tommaso Bolognesi and Scott A. Smolka. Fundamental results for the verification of observational equivalence: a survey. In Harry Rudin and Colin H. West, editors, *Protocol Specification, Testing and Verification VII: Proceedings of IFIP WG6.1 7th International Conference on Protocol Specification, Testing and Verification*, pages 165–179, Amsterdam, The Netherlands, 1987. North Holland.

[3] Stephen D. Brookes and William C. Rounds. Behavioural equivalence relations induced by programming logics. In *Proceedings of 16th International Colloquium on Automata, Languages, and Programming, ICALP '83*, volume 154 of *LNCS*, pages 97–108. Springer-Verlag, 1983.

[4] Martin Fabian. *On Object Oriented Nondeterministic Supervisory Control*. PhD thesis, Chalmers University of Technology, Göteborg, Sweden, 1995.

[5] Lei Feng and W. M. Wonham. Computationally efficient supervisor design: Abstraction and modularity. In *Proceedings of the 8th International Work-*

shop on Discrete Event Systems, WODES'06, pages 3–8, Ann Arbor, MI, USA, July 2006.

[6] Jean-Claude Fernandez. An implementation of an efficient algorithm for bisimulation equivalence. *Science of Computer Programming*, 13:219–236, 1990.

[7] Hugo Flordal, Robi Malik, Martin Fabian, and Knut Åkesson. Compositional synthesis of maximally permissive supervisors using supervision equivalence. *Discrete Event Dynamic Systems: Theory and Applications*, 17(4):475–504, 2007.

[8] C. A. R. Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985.

[9] Petra Malik, Robi Malik, David Streader, and Steve Reeves. Modular synthesis of discrete controllers. In *Proceedings of 12th IEEE International Conference on Engineering of Complex Computer Systems, ICECCS '07*, pages 25–34, Auckland, New Zealand, 2007.

[10] Robi Malik and Hugo Flordal. Yet another approach to compositional synthesis of discrete event systems. In *Proceedings of the 9th International Workshop on Discrete Event Systems, WODES'08*, pages 16–21, Göteborg, Sweden, May 2008.

[11] Robin Milner. *Communication and concurrency*. Series in Computer Science. Prentice-Hall, 1989.

[12] Sahar Mohajerani, Robi Malik, and Martin Fabian. Nondeterminism avoidance in compositional synthesis of discrete event systems. In *Proceedings of the 7th International Conference on Automation Science and Engineering, CASE 2011*, pages 19–24, Trieste, Italy, 2011.

[13] Sahar Mohajerani, Robi Malik, and Martin Fabian. An algorithm for weak synthesis observation equivalence for compositional supervisor synthesis. In *Proceedings of the 11th International Workshop on Discrete Event Systems, WODES'12*, Guadalajara, Mexico, October 2012. to appear.

[14] Sahar Mohajerani, Robi Malik, Simon Ware, and Martin Fabian. On the use of observation equivalence in synthesis abstraction. In *Proceedings of the 3rd IFAC Workshop on Dependable Control of Discrete Systems, DCDS 2011*, pages 84–89, Saarbrücken, Germany, 2011.

[15] Peter J. G. Ramadge and W. Murray Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1):81–98, January 1989.

[16] Klaus Schmidt and Christian Breindl. On maximal permissiveness of hierarchical and modular supervisory control approaches for discrete event systems. In *Proceedings of the 9th International Workshop on Discrete Event Systems, WODES'08*, pages 462–467, Göteborg, Sweden, May 2008.

[17] Rong Su, Jan H. van Schuppen, and Jacobus E. Rooda. Model abstraction of nondeterministic finite-state automata in supervisor synthesis. *IEEE Transactions on Automatic Control*, 55(11):2527–2541, November 2010.

A Weak Synthesis Observation Equivalence

This appendix contains a proof of theorem 3, which states that weak synthesis observation equivalence implies synthesis abstraction. Following the line of [14], this is done by proving that weak synthesis observation equivalence implies *state-wise synthesis equivalence*.

Definition 15 [14] Let $G = \langle \Sigma, Q, \rightarrow, Q^\circ \rangle$ be an automaton. An equivalence relation $\sim \subseteq Q \times Q$ is a *state-wise synthesis equivalence* on G with respect to $\Upsilon \subseteq \Sigma$, if for all $x \in Q$, all deterministic automata $T = \langle \Sigma_T, Q_T, \rightarrow_T, Q_T^\circ \rangle$ such that $\Sigma_T \subseteq \Upsilon$, and for all states $x_T \in Q_T$ the following relations hold:

(i) if $(x, x_T) \in \hat{\Theta}_{G\|T}$, then $([x], x_T) \in \hat{\Theta}_{G/\sim\|T}$;

(ii) if $([x], x_T) \in \hat{\Theta}_{G/\sim\|T}$, then $(x, x_T) \in \hat{\Theta}_{G\|T}$.

State-wise synthesis equivalence means that for every equivalence class $\tilde{x}$, synthesis must remove either all or none of the states in $\tilde{x}$, in every possible context T. It is a known result [14] that this is a sufficient condition for synthesis abstraction.

Proposition 4 [14] Let $G = \langle \Sigma, Q, \rightarrow, Q^\circ \rangle$ be deterministic, and let $\sim$ be a state-wise synthesis equivalence on G with respect to $\Upsilon \subseteq \Sigma$ such that $G/\sim$ is deterministic. Then $G \lesssim_{\mathrm{synth},\Upsilon} G/\sim$.

Proof. It must be shown that for any deterministic automaton $T = \langle \Sigma_T, Q_T, \rightarrow_T, Q_T^\circ \rangle$ such that $\Sigma_T \cap \Upsilon = \emptyset$, equation (10) holds.

First, let $s \in \mathcal{L}(G \parallel \mathrm{sup}\mathcal{CN}(G \parallel T))$. This means $G \parallel \mathrm{sup}\mathcal{CN}(G \parallel T) \xrightarrow{s} (x_G, y_G, x_T)$, and since G is deterministic $x_G = y_G$. Let $s = \sigma_1 \cdots \sigma_n$, then $(x_0^G, x_0^T) \xrightarrow{\sigma_1}_{|\hat{\Theta}_{G\|T}} (x_1^G, x_1^T) \xrightarrow{\sigma_2}_{|\hat{\Theta}_{G\|T}} \cdots \xrightarrow{\sigma_n}_{|\hat{\Theta}_{G\|T}} (x_n^G, x_n^T) = (x_G, x_T)$ such that $(x_k^G, x_k^T) \in \hat{\Theta}_{G\|T}$ or $\sigma_k = \omega$ for $k = 0, \ldots, n$. By definition 15 (i), $([x_k^G], x_k^T) \in \hat{\Theta}_{G/\sim\|T}$ or $\sigma_k = \omega$ for $k = 0, \ldots, n$, and thus $([x_0^G], x_0^T) \xrightarrow{\sigma_1}_{|\hat{\Theta}_{G/\sim\|T}}$

$([x_1^G], x_1^T) \xrightarrow{\sigma_2}_{|\hat{\Theta}_{G/\sim\|T}} \cdots \xrightarrow{\sigma_n}_{|\hat{\Theta}_{G/\sim\|T}} ([x_n^G], x_n^T) = ([x_G], x_T)$. Therefore, $G \parallel \sup\mathcal{CN}(G/\sim \| T) \xrightarrow{s} (x_G, [x_G], x_T)$, which means $s \in \mathcal{L}(G \| \sup\mathcal{CN}(G/\sim \| T))$.

Conversely, let $s \in \mathcal{L}(G \| \sup\mathcal{CN}(G/\sim \| T))$. Since G and $G/\sim$ are deterministic, this means $G \| \sup\mathcal{CN}(G/\sim \| T) \xrightarrow{\sigma_1} (x_1^G, [x_1^G], x_1^T) \xrightarrow{\sigma_2} \cdots \xrightarrow{\sigma_n} (x_n^G, [x_n^G], x_n^T)$, where $s = \sigma_1 \cdots \sigma_n$. Since $([x_k^G], x_k^T) \in \hat{\Theta}_{G/\sim\|T}$ for $k = 0, \ldots, n$ by definition 15 (ii), $(x_k^G, x_k^T) \in \hat{\Theta}_{G\|T}$ or $\sigma_k = \omega$ for $k = 0, \ldots, n$. Therefore, $G \parallel \sup\mathcal{CN}(G \parallel T) \xrightarrow{\sigma_1} (x_1^G, x_1^G, x_1^T) \xrightarrow{\sigma_2} \cdots \xrightarrow{\sigma_n} (x_n^G, x_n^G, x_n^T)$, and thus it can be concluded that $s \in \mathcal{L}(G \parallel \sup\mathcal{CN}(G \parallel T))$. □

Proposition 6 below establishes the crucial result that every weak synthesis observation equivalence is a state-wise synthesis equivalence. Before that, lemma 5 establishes an auxiliary result about the paths in a quotient automaton resulting from weak synthesis observation equivalence.

Lemma 5 Let $G = \langle \Sigma, Q, \rightarrow, Q^\circ \rangle$ and $T = \langle \Sigma_T, Q_T, \rightarrow_T, Q_T^\circ \rangle$ be two automata with $\Sigma \cup \Sigma_T = \Omega \dot\cup \Upsilon$ and $\Upsilon \cap \Sigma_T = \emptyset$, and let $\sim$ be a weak synthesis observation equivalence on G with respect to Υ. Let $X \subseteq Q \times Q_T$ such that $([x], x_T) \in \hat{\Theta}_{G/\sim\|T}$ always implies $(x, x_T) \in X$. Furthermore, let $(x_1, x_1^T) \xrightarrow{\sigma} (x_2, x_2^T)$ such that $([x_1], x_1^T) \xrightarrow{\sigma}_{|\hat{\Theta}_{G/\sim\|T}} ([x_2], x_2^T)$. Then for all states $y_1 \in Q$ such that $x_1 \sim y_1$, there exist $t_1, t_2 \in \Upsilon^*$ and $y_2 \in Q$ such that $(y_1, x_1^T) \xrightarrow{t_1 P_\Omega(\sigma) t_2}_{|X} (y_2, x_2^T)$ and $x_2 \sim y_2$.

Proof. Let $x_1, x_2, y_1 \in Q$ and $x_1^T, x_2^T \in Q_T$ and $\sigma \in \Sigma \cup \Sigma_T$ such that $(x_1, x_1^T) \xrightarrow{\sigma} (x_2, x_2^T)$, $([x_1], x_1^T) \xrightarrow{\sigma}_{|\hat{\Theta}_{G/\sim\|T}} ([x_2], x_2^T)$, and $x_1 \sim y_1$. Consider three cases.

(i) If $\sigma \notin \Sigma$, then $\sigma \neq \omega$ and $\sigma \in \Sigma_T \setminus \Sigma \subseteq \Omega$ and $x_1 = x_2$ and $x_1^T \xrightarrow{\sigma} x_2^T$. Given $([x_1], x_1^T) \xrightarrow{\sigma}_{|\hat{\Theta}_{G/\sim\|T}} ([x_2], x_2^T)$, it follows that $([y_1], x_1^T) = ([x_1], x_1^T) \in \hat{\Theta}_{G/\sim\|T}$ and $([y_1], x_2^T) = ([x_1], x_2^T) = ([x_2], x_2^T) \in \hat{\Theta}_{G/\sim\|T}$, and therefore $(y_1, x_1^T), (y_1, x_2^T) \in X$ by assumption. This implies that $(y_1, x_1^T) \xrightarrow{P_\Omega(\sigma)}_{|X} (y_1, x_2^T)$.

(ii) If $\sigma \in \Sigma \cap \Sigma_\mathrm{u}$, then $x_1 \overset{\sigma}{\Rightarrow}_\mathrm{u} x_2$, and since $x_1 \sim y_1$ and $\sim$ is stable with respect to $\overset{\sigma}{\Rightarrow}_\mathrm{u}$, there exists $y_2 \in Q$ such that $y_1 \overset{\sigma}{\Rightarrow}_\mathrm{u} y_2$. Thus, $y_1 \xrightarrow{t_1 P_\Omega(\sigma) t_2} y_2$ for some $t_1, t_2 \in (\Upsilon \cap \Sigma_\mathrm{u})^*$. Let $r \sqsubseteq t_1 P_\Omega(\sigma) t_2$ such that $y_1 \xrightarrow{r} z$. Then $[x_1] = [y_1] \xrightarrow{r} [z]$, and since $\Sigma_T \cap \Upsilon = \emptyset$, it follows that $([x_1], x_1^T) \xrightarrow{r} ([z], x_d^T)$ for some $d \in \{1, 2\}$. Since $r \in \Sigma_\mathrm{u}^*$ and $([x_1], x_1^T) \in \hat{\Theta}_{G/\sim\|T}$, it follows that $([z], x_d^T) \in \hat{\Theta}_{G/\sim\|T}$. This implies $(z, x_d^T) \in X$ by assump-

tion. This argument holds for all prefixes $r \sqsubseteq t_1 P_\Omega(\sigma) t_2$, and therefore $(y_1, x_1^T) \xrightarrow{t_1 P_\Omega(\sigma) t_2}_{|X} (y_2, x_2^T)$.

(iii) If $\sigma \in \Sigma \cap \Sigma_c$, then $x_1 \overset{\sigma}{\Rightarrow}_{\text{wsoe}} x_2$ or $x_1 \overset{\Upsilon}{\Rightarrow}_{\text{wsoe}} x_2$, and since $x_1 \sim y_1$ and $\sim$ is stable with respect to these relations, there exists $y_2 \sim x_2$ such that $y_1 \overset{\sigma}{\Rightarrow}_{\text{wsoe}} y_2$ or $y_1 \overset{\Upsilon}{\Rightarrow}_{\text{wsoe}} y_2$. That is, there exists a path $y_1 = z_0 \xrightarrow{\tau_1} \cdots \xrightarrow{\tau_k} z_k \xrightarrow{P_\Omega(\sigma)} z_{k+1} \xrightarrow{\tau_{k+1}} \cdots \xrightarrow{\tau_{l-1}} z_l = y_2$ such that $x_2 \sim y_2$ and $\tau_1, \ldots, \tau_{l-1} \in \Upsilon$. The first part of this path satisfies the conditions for $z_0 \overset{\sigma}{\Rightarrow}_{\text{soe}} z_{k+1}$ or $z_0 \overset{\Upsilon}{\Rightarrow}_{\text{soe}} z_{k+1}$ in definition 13, and the second part satisfies the conditions for $z_{k+1} \overset{\Upsilon}{\Rightarrow}_{c} z_l$ in definition 14. Since $\tau_1, \ldots, \tau_{l-1} \in \Upsilon$ and $\Sigma_T \cap \Upsilon = \emptyset$, it holds that

$$\begin{aligned}(y_1, x_1^T) = (z_0, x_1^T) \xrightarrow{\tau_1} \cdots \xrightarrow{\tau_k} (z_k, x_1^T) \xrightarrow{P_\Omega(\sigma)} \\ (z_{k+1}, x_2^T) \xrightarrow{\tau_{k+1}} \cdots \xrightarrow{\tau_{l-1}} (z_l, x_2^T) = (y_2, x_2^T)\end{aligned} \tag{11}$$

It follows that

$$\begin{aligned}([z_0], x_1^T) \xrightarrow{\tau_1} \cdots \xrightarrow{\tau_k} ([z_k], x_1^T) \xrightarrow{P_\Omega(\sigma)} \\ ([z_{k+1}], x_2^T) \xrightarrow{\tau_{k+1}} \cdots \xrightarrow{\tau_{l-1}} ([z_l], x_2^T)\,.\end{aligned} \tag{12}$$

It is shown in the following that this path also exists in the restriction of $G/\!\sim \,\|\, T$ to $\hat{\Theta}_{G/\sim \| T}$.

For the first part of the path (12), it is shown by induction on i that $([z_i], x_1^T) \in \hat{\Theta}_{G/\sim \| T}$, for $i = 0, \ldots, k$ if $\sigma \in \Omega$, and for $i = 0, \ldots, k-1$ if $\sigma \in \Upsilon$.

Base case. For $i = 0$, it follows by assumption that $([z_0], x_1^T) = ([y_1], x_1^T) = ([x_1], x_1^T) \in \hat{\Theta}_{G/\sim \| T}$.

Inductive step. Assume the claim holds for some $i \geq 0$, i.e., $([z_i], x_1^T) \in \hat{\Theta}_{G/\sim \| T}$. It must be shown that $([z_{i+1}], x_1^T) \in \hat{\Theta}_{G/\sim \| T}$. There are two possibilities for $\tau_{i+1} \in \Upsilon$:

a) $\tau_{i+1} \in \Sigma_c$. In this case, it follows from $z_0 \overset{\sigma}{\Rightarrow}_{\text{soe}} z_{k+1}$ or $z_0 \overset{\Upsilon}{\Rightarrow}_{\text{soe}} z_{k+1}$ by definition 13 that $z_{i+1} \sim x_1$, and thus $([z_{i+1}], x_1^T) = ([x_1], x_1^T) \in \hat{\Theta}_{G/\sim \| T}$ by assumption.

b) $\tau_{i+1} \in \Sigma_u$. As $(z_i, x_1^T) \xrightarrow{\tau_{i+1}} (z_{i+1}, x_1^T)$, it holds that $([z_i], x_1^T) \xrightarrow{\tau_{i+1}} ([z_{i+1}], x_1^T)$, and $([z_i], x_1^T) \in \hat{\Theta}_{G/\sim \| T}$ by inductive assumption. Then $([z_{i+1}], x_1^T) \in \hat{\Theta}_{G/\sim \| T}$ because $\tau_{i+1} \in \Sigma_u$.

If $\sigma = \omega$, the second part of the path (12) is empty and the claim follows. Otherwise note that by assumption,

$$([x_2], x_2^T) \in \hat{\Theta}_{G/\sim\|T} \,. \tag{13}$$

It is shown that $([z_i], x_2^T) \in \hat{\Theta}_{G/\sim\|T}$ for $k < i < l$. Let $\Upsilon_{\mathrm{u}}^T = \Sigma_{\mathrm{u}} \cap (\Sigma_T \setminus \Sigma)$ and

$$Y^T = \{\, y^T \in Q_T \mid x_2^T \xrightarrow{u}_T y^T \text{ for some } u \in (\Upsilon_{\mathrm{u}}^T)^* \,\} \,. \tag{14}$$

As $x_2^T \in Y^T$, it is enough to show that $([z_i], y^T) \in \hat{\Theta}_{G/\sim\|T}$ for all $y^T \in Y^T$. It is shown by induction on $n \geq 0$ that for all $k < i < l$ and for all $y^T \in Y^T$ it holds that $([z_i], y^T) \in \tilde{X}^n = \Theta^n_{G/\sim\|T}(Q/\sim \times Q_T)$.

Base case. $n = 0$. Clearly $([z_i], y^T) \in Q/\sim \times Q_T = \Theta^0_{G/\sim\|T}(Q/\sim \times Q_T) = \tilde{X}^0$.

Inductive step. Let $k < i < l$ and $y^T \in Y^T$. It must be shown that $([z_i], y^T) \in \tilde{X}^{n+1} = \Theta_{G/\sim\|T}(\tilde{X}^n) = \Theta^{\mathrm{cont}}_{G\|T}(\tilde{X}^n) \cap \Theta^{\mathrm{nonb}}_{G\|T}(\tilde{X}^n)$.

To see that $([z_i], y^T) \in \Theta^{\mathrm{cont}}_{G\|T}(\tilde{X}^n)$, let $\upsilon \in \Sigma_{\mathrm{u}}$ and $([z_i], y^T) \xrightarrow{\upsilon}_{G/\sim\|T} ([z], z^T)$. Consider three cases.

a) $\upsilon \in \Sigma \cap \Upsilon$. In this case $y^T = z^T$ and $[z_i] \xrightarrow{\upsilon} [z]$, so there exist $z_i' \sim z_i$ and $z' \sim z$ such that $z_i' \xrightarrow{\upsilon} z'$ and thus $z_i' \overset{\upsilon}{\Rightarrow}_{\mathrm{u}} z'$. As $z_i \sim z_i'$ and $\sim$ is stable with respect to $\overset{\upsilon}{\Rightarrow}_{\mathrm{u}}$, there exists $z'' \sim z'$ such that $z_i \overset{\upsilon}{\Rightarrow}_{\mathrm{u}} z''$. As $\upsilon \in \Sigma_{\mathrm{u}} \cap \Upsilon$, this means $z_i \xrightarrow{u} z''$ for some $u \in (\Sigma_{\mathrm{u}} \cap \Upsilon)^*$. As z_i is on a path $z_{k+1} \overset{\Upsilon}{\Rightarrow}_{\mathrm{c}} z_l$, it follows from definition 14 that $z'' \sim z_j$ for some $k < j \leq l$. If $j < l$, then by inductive assumption $([z], z^T) = ([z'], z^T) = ([z''], z^T) = ([z_j], z^T) \in \tilde{X}^n$. If $j = l$, then note that $([x_2], x_2^T) \xrightarrow{u} ([x_2], z^T)$ for some $u \in (\Upsilon_{\mathrm{u}}^T)^*$ as $z^T = y^T \in Y^T$, and given (13) it follows that $([y_2], z^T) = ([x_2], z^T) \in \hat{\Theta}_{G/\sim\|T}$. Then $([z], z^T) = ([z'], z^T) = ([z''], z^T) = ([z_l], z^T) = ([y_2], z^T) \in \hat{\Theta}_{G/\sim\|T} \subseteq \tilde{X}^n$.

b) $\upsilon \in \Sigma \cap \Omega$. In this case $[z_i] \xrightarrow{\upsilon} [z]$, so there exist $z_i' \sim z_i$ and $z' \sim z$ such that $z_i' \xrightarrow{\upsilon} z'$, and thus $z_i' \overset{\upsilon}{\Rightarrow}_{\mathrm{u}} z'$. As $z_i \sim z_i'$ and $\sim$ is stable with respect to $\overset{\upsilon}{\Rightarrow}_{\mathrm{u}}$, there exists $z'' \sim z'$ such that $z_i \overset{\upsilon}{\Rightarrow}_{\mathrm{u}} z''$. As z_i is on a path $z_{k+1} \overset{\Upsilon}{\Rightarrow}_{\mathrm{c}} z_l = y_2 \sim x_2$, it follows from definition 14 that $x_2 \overset{\upsilon}{\Rightarrow}_{\mathrm{u}} z_2''$ for some $z_2'' \sim z'' \sim z' \sim z$. Then since $y^T \in Y^T$ and by definition of $\Rightarrow_{\mathrm{u}}$, there exist $u \in (\Upsilon_{\mathrm{u}}^T)^*$ and $u_1, u_2 \in (\Sigma_{\mathrm{u}} \cap \Upsilon)^*$ such that $([x_2], x_2^T) \xrightarrow{u}_{G/\sim\|T} ([x_2], y^T) \xrightarrow{u_1 \upsilon u_2}_{G/\sim\|T} ([z_2''], z^T)$. Given (13), it follows that $([z], z^T) = ([z_2''], z^T) \in \hat{\Theta}_{G/\sim\|T} \subseteq \tilde{X}^n$.

c) $\upsilon \notin \Sigma$. In this case, $\upsilon \in \Sigma_T \setminus \Sigma$ and $[z_i] = [z]$ and $y^T \xrightarrow{\upsilon}_T z^T$. Then clearly $z^T \in Y^T$ and $([z], z^T) = ([z_i], z^T) \in \tilde{X}^n$ by inductive assumption.

Thus $([z], z^T) \in \tilde{X}^n$ can be shown for all $\upsilon \in \Sigma_{\mathrm{u}}$, and it follows that $([z_i], y^T) \in \Theta^{\mathrm{cont}}_{G/\sim\|T}(\tilde{X}^n)$.

Next, it is shown that $([z_i], y^T) \in \Theta^{\mathrm{nonb}}_{G/\sim\|T}(\tilde{X}^n)$. As $\tau_{k+1}, \ldots, \tau_l \in \Upsilon$ and $\Sigma_T \cap \Upsilon = \emptyset$, it holds by inductive assumption that,

$$([z_{k+1}], y^T) \xrightarrow{\tau_{k+1}}_{|\tilde{X}^n} \cdots \xrightarrow{\tau_k}_{|\tilde{X}^n} ([z_l], y^T) \,. \tag{15}$$

Since $y^T \in Y^T$, there exists $u \in (\Upsilon_{\mathrm{u}}^T)^*$ such that $x_2^T \xrightarrow{u}_T y^T$, and this implies $([x_2], x_2^T) = ([z_l], x_2^T) \xrightarrow{u}_{G/\sim\|T} ([z_l], y^T)$. Since $u \in \Sigma_{\mathrm{u}}^*$, it follows by (13) that $([z_l], y^T) \in \hat{\Theta}_{G/\sim\|T}$. Then there exists $t \in \Sigma^*$ such that $([z_l], y^T) \xrightarrow{t\omega}_{|\hat{\Theta}_{G/\sim\|T}}$. Thus

$$([z_i], y^T) \xrightarrow{\tau_{i+1}}_{|\tilde{X}^n} \cdots \xrightarrow{\tau_k}_{|\tilde{X}^n} ([z_l], y^T) \xrightarrow{t\omega}_{|\tilde{X}^n} \,. \tag{16}$$

This implies $([z_i], y^T) \in \Theta^{\mathrm{nonb}}_{G/\sim\|T}(\tilde{X}^n)$.

It has been shown that all states $([z_i], x_d^T)$ on the path (12) are in $\hat{\Theta}_{G/\sim\|T}$, except for the last state when $\sigma = \omega$. This implies by assumption $(z_i, x_d^T) \in X$ for all states on the path (11), except for the last state when $\sigma = \omega$. Therefore, $(y_1, x_1^T) \xrightarrow{t_1 P_\Omega(\sigma) t_2}_{|X} (y_2, x_2^T)$. □

Proposition 6 Let $\sim$ be a weak synthesis observation equivalence on $G = \langle \Sigma, Q, \rightarrow, Q^\circ \rangle$ with respect to $\Upsilon \subseteq \Sigma$. Then $\sim$ is a state-wise synthesis equivalence on G with respect to Υ.

Proof. Let $T = \langle \Sigma_T, Q_T, \rightarrow_T, Q_T^\circ \rangle$ with $\Sigma_T \cap \Upsilon = \emptyset$ and $\Sigma \cup \Sigma_T = \Omega \,\dot\cup\, \Upsilon$. The conditions of state-wise synthesis equivalence in definition 15 must be confirmed.

(i) It is shown by induction on $n \geq 0$ that $(x, x_T) \in \hat{\Theta}_{G\|T}$ implies $([x], x_T) \in \tilde{X}^n = \Theta^n_{G/\sim\|T}(Q/\sim \times Q_T)$.

Base case. $([x], x_T) \in Q/\sim \times Q_T = \Theta^0_{G/\sim\|T}(Q/\sim \times Q_T) = \tilde{X}^0$.

Inductive step. Assume the claim holds for some $n \geq 0$, i.e., if $(x, x_T) \in \hat{\Theta}_{G\|T}$ then $([x], x_T) \in \tilde{X}^n$. Now let $(x, x_T) \in \hat{\Theta}_{G\|T}$. It must be shown that $([x], x_T) \in \tilde{X}^{n+1} = \Theta_{G/\sim\|T}(\tilde{X}^n) = \Theta^{\mathrm{cont}}_{G/\sim\|T}(\tilde{X}^n) \cap \Theta^{\mathrm{nonb}}_{G/\sim\|T}(\tilde{X}^n)$.

To see that $([x], x_T) \in \Theta^{\text{cont}}_{G/\sim\|T}(\tilde{X}^n)$, let $\upsilon \in \Sigma_{\text{u}}$ and $([x], x_T) \xrightarrow{\upsilon} ([y], y_T)$. Consider two cases.

a) $\upsilon \notin \Sigma$. In this case, $[x] = [y]$ and $(x, x_T) \xrightarrow{\upsilon} (x, y_T)$, and it follows from $(x, x_T) \in \hat{\Theta}_{G\|T}$ and $\upsilon \in \Sigma_{\text{u}}$ that $(x, y_T) \in \hat{\Theta}_{G\|T}$. Then by inductive assumption $([y], y_T) = ([x], y_T) \in \tilde{X}^n$.

b) $\upsilon \in \Sigma$. In this case, there exist $x' \in [x]$ and $y' \in [y]$ such that $x' \xrightarrow{\upsilon} y'$. Thus $x' \overset{\upsilon}{\Rightarrow}_{\text{u}} y'$, and since $\sim$ is stable with respect to $\overset{\upsilon}{\Rightarrow}_{\text{u}}$, there exists $y'' \sim y'$ such that $x \overset{\upsilon}{\Rightarrow}_{\text{u}} y''$. Then $(x, x_T) \xrightarrow{t_1 P_\Omega(\upsilon) t_2} (y'', y_T)$ for some $t_1, t_2 \in (\Upsilon \cap \Sigma_{\text{u}})^*$. Since $(x, x_T) \in \hat{\Theta}_{G\|T}$ and $t_1 P_\Omega(\upsilon) t_2 \in \Sigma^*_{\text{u}}$, it follows that $(y'', y_T) \in \hat{\Theta}_{G\|T}$. Therefore by inductive assumption $([y], y_T) = ([y'], y_T) = ([y''], y_T) \in \tilde{X}^n$.

Thus $([y], y_T) \in \tilde{X}^n$ can be shown for all $\upsilon \in \Sigma_{\text{u}}$, and it follows that $([x], x_T) \in \Theta^{\text{cont}}_{G/\sim\|T}(\tilde{X}^n)$.

Next, it is shown that $([x], x_T) \in \Theta^{\text{nonb}}_{G/\sim\|T}(\tilde{X}^n)$. Since $(x, x_T) \in \hat{\Theta}_{G\|T}$, there exists a path

$$(x, x_T) = (x_0, x_0^T) \xrightarrow{\sigma_1}_{|\hat{\Theta}_{G\|T}} \cdots \xrightarrow{\sigma_k}_{|\hat{\Theta}_{G\|T}} (x_k, x_k^T) \xrightarrow{\omega}_{|\hat{\Theta}_{G\|T}} (x_{k+1}, x_{k+1}^T) \,.$$

Then $(x_l, x_l^T) \in \hat{\Theta}_{G\|T}$ for $l = 0, \ldots, k$. By inductive assumption, it follows that $([x_l], x_l^T) \in \tilde{X}^n$ for $l = 0, \ldots, k$. Thus,

$$([x], x_T) = ([x_0], x_0^T) \xrightarrow{\sigma_1}_{|\tilde{X}^n} \cdots \xrightarrow{\sigma_k}_{|\tilde{X}^n} ([x_k], x_k^T) \xrightarrow{\omega}_{|\tilde{X}^n} ([x_{k+1}], x_{k+1}^T) \,,$$

which implies $([x], x_T) \in \Theta^{\text{nonb}}_{G/\sim\|T}(\tilde{X}^n)$.

Thus, it has been shown that $([x], x_T) \in \Theta^{\text{cont}}_{G/\sim\|T}(\tilde{X}^n) \cap \Theta^{\text{nonb}}_{G/\sim\|T}(\tilde{X}^n) = \tilde{X}^{n+1}$.

(ii) Now it is shown by induction on $n \geq 0$ that $([x], x_T) \in \hat{\Theta}_{G/\sim\|T}$ implies $(x, x_T) \in X^n = \Theta^n_{G\|T}(Q \times Q_T)$.

Base case. $(x, x_T) \in Q \times Q_T = \Theta^0_{G\|T}(Q \times Q_T) = X^0$.

Inductive step. Assume the statement holds for $n \geq 0$, i.e, if $([x], x_T) \in \hat{\Theta}_{G/\sim\|T}$ then $(x, x_T) \in X^n$. Let $([x], x_T) \in \hat{\Theta}_{G/\sim\|T}$. It must be shown that $(x, x_T) \in X^{n+1} = \Theta_{G\|T}(X^n) = \Theta^{\text{cont}}_{G\|T}(X^n) \cap \Theta^{\text{nonb}}_{G\|T}(X^n)$.

To see that $(x, x_T) \in \Theta^{\text{cont}}_{G\|T}(X^n)$, let $\upsilon \in \Sigma_{\text{u}}$ and $(x, x_T) \xrightarrow{\upsilon} (y, y_T)$. This implies $([x], x_T) \xrightarrow{\upsilon} ([y], y_T)$. Since $([x], x_T) \in \hat{\Theta}_{G/\sim\|T}$ and $\upsilon \in \Sigma_{\text{u}}$, it

follows that $([y], y_T) \in \hat{\Theta}_{G/\sim\|T}$. Then by inductive assumption $(y, y_T) \in X^n$, and thus $(x, x_T) \in \Theta^{\text{cont}}_{G\|T}(X^n)$.

Next it is shown that $(x, x_T) \in \Theta^{\text{nonb}}_{G\|T}(X^n)$. Since $([x], x_T) \in \hat{\Theta}_{G/\sim\|T}$, there exists a path

$$\begin{aligned}([x], x_T) = ([x_0], x_0^T) &\xrightarrow{\sigma_1}_{|\hat{\Theta}_{G/\sim\|T}} \cdots \xrightarrow{\sigma_k}_{|\hat{\Theta}_{G/\sim\|T}} \\ &([x_k], x_k^T) \xrightarrow{\omega}_{|\hat{\Theta}_{G/\sim\|T}} ([x_{k+1}], x_{k+1}^T)\,. \end{aligned} \tag{17}$$

Consider the first transition in (17). Since $[x_0] \xrightarrow{P_\Sigma(\sigma_1)} [x_1]$, there exists $x_0' \in [x_0]$ and $x_1' \in [x_1]$ such that $x_0' \xrightarrow{P_\Sigma(\sigma_1)} x_1'$. The conditions of lemma 5 apply to this transition: by inductive assumption, X^n can be used as the set X in the lemma, and $([x_0'], x_0^T) = ([x_0], x_0^T) \in \hat{\Theta}_{G/\sim\|T}$, $([x_1'], x_1^T) = ([x_1], x_1^T) \in \hat{\Theta}_{G/\sim\|T}$ or $\sigma_1 = \omega$, $(x_0', x_0^T) \xrightarrow{\sigma_1} (x_1', x_1^T)$, and $x_0' \sim x_0$. So there exist $t_1, u_1 \in \Upsilon^*$ and $x_1'' \in Q$ such that $(x_0, x_0^T) \xrightarrow{t_1 P_\Omega(\sigma_1) u_1}_{|X^n} (x_1'', x_1^T)$ and $x_1' \sim x_1''$.

Since $x_1'' \in [x_1'] = [x_1]$, the same logic also applies to the second transition in (17). Therefore, there exist $t_2, u_2 \in \Upsilon^*$ and $x_2'' \in Q$ such that $(x_1'', x_1^T) \xrightarrow{t_2 P_\Omega(\sigma_2) u_2}_{|X^n} (x_2'', x_2^T)$ and $x_2 \sim x_2' \sim x_2''$. By induction, it follows that there exist $t_1, u_1, \ldots, t_k, u_k, t_{k+1} \in \Upsilon^*$ and $x_1'', \ldots, x_k'' \in Q$ such that

$$\begin{aligned}(x, x_T) = (x_0, x_0^T) &\xrightarrow{t_1 P_\Omega(\sigma_1) u_1}_{|X^n} (x_1'', x_1^T) \xrightarrow{t_2 P_\Omega(\sigma_2) u_2}_{|X^n} \cdots \\ &\xrightarrow{t_k P_\Omega(\sigma_k) u_k}_{|X^n} (x_k'', x_k^T) \xrightarrow{t_{k+1}\omega}_{|X^n}\,. \end{aligned} \tag{18}$$

Therefore, $(x, x_T) \in \Theta^{\text{nonb}}_{G\|T}(X^n)$.

Thus, it has been shown that $(x, x_T) \in \Theta^{\text{cont}}_{G\|T}(X^n) \cap \Theta^{\text{nonb}}_{G\|T}(X^n) = X^{n+1}$. □

B Synthesis Observation Equivalence

This appendix contains a proof that synthesis observation equivalence is a special case of weak synthesis observation equivalence, so all results about weak synthesis observation equivalence shown in appendix A also apply to synthesis observation equivalence. Theorem 8 shows the main result of this section, which states that every weak synthesis observation equivalence also is a synthesis observation equivalence. The proof uses a lemma about the uncontrollable transitions outgoing from states along a path $x \overset{\Upsilon}{\Rightarrow}_{\text{soe}} y$.

Lemma 7 Let $G = \langle \Sigma, Q, \rightarrow, Q^\circ \rangle$ be an automaton, and let $\sim \subseteq Q \times Q$ be stable with respect to $\overset{\upsilon}{\Rightarrow}_{\mathrm{u}}$ for all $\upsilon \in \Sigma_{\mathrm{u}}$. Furthermore, let $\Upsilon \subseteq \Sigma$ and $x \overset{\Upsilon}{\Rightarrow}_{\mathrm{soe}} y$. For every state z on this path, if $z \overset{\upsilon}{\Rightarrow}_{\mathrm{u}} z'$ for some $\upsilon \in \Sigma_{\mathrm{u}}$, then there exists $z'' \in Q$ such that $x \overset{\upsilon}{\Rightarrow}_{\mathrm{u}} z''$ and $z' \sim z''$.

Proof. Write the path $x \overset{\Upsilon}{\Rightarrow}_{\mathrm{soe}} y$ as $x = z_0 \overset{\tau_1}{\rightarrow} \ldots \overset{\tau_k}{\rightarrow} z_k = y$. Let z_j be a state on the path such that $z_j \overset{\upsilon}{\Rightarrow}_{\mathrm{u}} z'$ for some $\upsilon \in \Sigma_{\mathrm{u}}$. We must show that there exists z'' such that $x \overset{\upsilon}{\Rightarrow}_{\mathrm{u}} z''$ and $z' \sim z''$. Let i, $0 \leq i \leq j$, be the greatest index such that $i = 0$ or $\tau_i \in \Sigma_{\mathrm{c}}$. If $i = 0$ then $z_i = z_0 = x$, and if $i \geq 1$, it follows from definition 13 that $z_i \sim x$. Thus, $z_i \sim x$ in both cases. Since $z_i \overset{\tau_{i+1}}{\longrightarrow} \ldots \overset{\tau_j}{\rightarrow} z_j \overset{\upsilon}{\Rightarrow}_{\mathrm{u}} z'$ with $\tau_l \in \Sigma_{\mathrm{u}} \cap \Upsilon$ for $i+1 \leq l \leq j$ it follows that $z_i \overset{\upsilon}{\Rightarrow}_{\mathrm{u}} z'$. Since $\sim$ is stable with respect to $\overset{\upsilon}{\Rightarrow}_{\mathrm{u}}$ there exists $z'' \sim z'$ such that $x \overset{\upsilon}{\Rightarrow}_{\mathrm{u}} z''$. □

Theorem 8 Let $G = \langle \Sigma, Q, \rightarrow, Q^\circ \rangle$ be an automaton and let $\sim$ be a synthesis observation equivalence on G with respect to Υ. Then $\sim$ is a weak synthesis observation equivalence on G with respect to Υ.

Proof. Let $x_1, x_2, y_1 \in Q$ such that $x_1 \sim x_2$ and $x_1 \overset{\Upsilon}{\Rightarrow}_{\mathrm{wsoe}} y_1$ or $x_1 \overset{\sigma}{\Rightarrow}_{\mathrm{wsoe}} y_1$ for some $\sigma \in \Sigma_{\mathrm{c}} \cap \Omega$ or $x_1 \overset{\upsilon}{\Rightarrow}_{\mathrm{u}} y_1$ for some $\upsilon \in \Sigma_{\mathrm{u}}$. It must be shown that there exists y_2 such that $x_2 \overset{\sigma}{\Rightarrow}_{\mathrm{wsoe}} y_2$ or $x_2 \overset{\Upsilon}{\Rightarrow}_{\mathrm{wsoe}} y_2$ or $x_2 \overset{\upsilon}{\Rightarrow}_{\mathrm{u}} y_2$ and $y_1 \sim y_2$.

If $x_1 \overset{\upsilon}{\Rightarrow}_{\mathrm{u}} y_1$ then since $x_1 \sim x_2$ and $\sim$ is stable with respect to $\overset{\upsilon}{\Rightarrow}_{\mathrm{u}}$ it follows that there exists y_2 such that $x_2 \overset{\upsilon}{\Rightarrow}_{\mathrm{u}} y_2$.

$x_1 \overset{\sigma}{\Rightarrow}_{\mathrm{wsoe}} y_1$ or $x_1 \overset{\Upsilon}{\Rightarrow}_{\mathrm{wsoe}} y_1$ means $x_1 \overset{\sigma}{\Rightarrow}_{\mathrm{soe}} q_1 \overset{\Upsilon}{\Rightarrow}_{\mathrm{c}} y_1$ or $x_1 \overset{\Upsilon}{\Rightarrow}_{\mathrm{soe}} q_1 \overset{\Upsilon}{\Rightarrow}_{\mathrm{c}} y_1$ respectively, where $q_1 \overset{\Upsilon}{\Rightarrow}_{\mathrm{c}} y_1$ is a path $q_1 = z_0 \overset{\tau_1}{\rightarrow} \cdots \overset{\tau_k}{\rightarrow} z_k = y_1$ with $\tau_1, \ldots, \tau_k \in \Upsilon$. Since $x_1 \sim x_2$ and $\sim$ is stable with respect to $\overset{\sigma}{\Rightarrow}_{\mathrm{soe}}$ and $\overset{\Upsilon}{\Rightarrow}_{\mathrm{soe}}$, there exists q_2 such that $x_2 \overset{\sigma}{\Rightarrow}_{\mathrm{soe}} q_2$ or $x_2 \overset{\Upsilon}{\Rightarrow}_{\mathrm{soe}} q_2$ and $q_1 \sim q_2$. It is first shown by induction on $i = 0, \ldots, k$ that there exists a path

$$q_2 = z'_0 \Rightarrow z'_1 \Rightarrow \cdots \Rightarrow z'_k = y_2 \tag{19}$$

such that $z'_i \sim z_i$ for all i, and each $z'_i \Rightarrow z'_{i+1}$ is $z'_i \overset{\tau_{i+1}}{\Longrightarrow}_{\mathrm{u}} z'_{i+1}$ if $\tau_{i+1} \in \Sigma_{\mathrm{u}}$ and $z'_i \overset{\Upsilon}{\Rightarrow}_{\mathrm{soe}} z'_{i+1}$ if $\tau_{i+1} \in \Sigma_{\mathrm{c}}$.

Base case. For $i = 0$, the claim clearly holds as $z'_0 = q_2 \sim q_1 = z_0$.

Inductive step. Assume the path up to z'_i with $z_i \sim z'_i$ has been constructed for some i. To obtain z'_{i+1} consider two cases. If $\tau_{i+1} \in \Sigma_{\mathrm{u}}$, then since $\sim$ is stable with respect to $\overset{\tau_{i+1}}{\Longrightarrow}_{\mathrm{u}}$, from $z_i \overset{\tau_{i+1}}{\longrightarrow} z_{i+1}$ it follows that there exists z'_{i+1} such that $z'_i \overset{\tau_{i+1}}{\Longrightarrow}_{\mathrm{u}} z'_{i+1}$ and $z'_{i+1} \sim z_{i+1}$. If $\tau_{i+1} \in \Sigma_{\mathrm{c}}$, then since $\sim$ is stable with respect

to $\overset{\Upsilon}{\Rightarrow}_{\mathrm{soe}}$, from $z_i \xrightarrow{\tau_{i+1}} z_{i+1}$ it follows that there exists z'_{i+1} such that $z'_i \overset{\Upsilon}{\Rightarrow}_{\mathrm{soe}} z'_{i+1}$ and $z'_{i+1} \sim z_{i+1}$.

Now it needs to be shown that $q_2 \overset{\Upsilon}{\Rightarrow}_{\mathrm{c}} y_2$. According to definition 14, the following properties need to be shown for every state z on the path (19).

(i) If $z \overset{u}{\rightarrow} z'$ for some $u \in (\Sigma_{\mathrm{u}} \cap \Upsilon)^*$ then $z' \sim \bar{z}$ for some $\bar{z}$ on the path (19).

(ii) If $z \overset{v}{\Rightarrow}_{\mathrm{u}} z'$ for some $v \in \Sigma_{\mathrm{u}} \cap \Omega$ then there exists $y'_2 \sim z'$ such that $y_2 \overset{v}{\Rightarrow}_{\mathrm{u}} y'_2$.

Let z be such a state on the path (19) and assume it is on the subpath $z'_i \Rightarrow z'_{i+1}$. Then consider two cases.

Case 1: z is on a subpath $z'_i \overset{\tau_{i+1}}{\Longrightarrow}_{\mathrm{u}} z'_{i+1}$. Then $\tau_{i+1} \in \Sigma_{\mathrm{u}} \cap \Upsilon$.

If $z \overset{u}{\rightarrow} z'$ for some $u \in (\Sigma_{\mathrm{u}} \cap \Upsilon)^*$, then clearly $z'_i \overset{\tau_{i+1}}{\Longrightarrow}_{\mathrm{u}} z'$. Since $\sim$ is stable with respect to $\overset{\tau_{i+1}}{\Longrightarrow}_{\mathrm{u}}$, from $z'_i \sim z_i$ it follows that there exists z'' such that $z_i \overset{\tau_{i+1}}{\Longrightarrow}_{\mathrm{u}} z''$ and $z' \sim z''$. Since z_i is on the path $q_1 \overset{\Upsilon}{\Rightarrow}_{\mathrm{c}} y_1$, it follows from definition 14 that $z'' \sim z_j$ for some j. Thus $z' \sim z'' \sim z_j \sim z'_j$, showing (i).

If $z \overset{v}{\Rightarrow}_{\mathrm{u}} z'$ for some $v \in \Sigma_{\mathrm{u}} \cap \Omega$, then $z'_i \overset{v}{\Rightarrow}_{\mathrm{u}} z'$. Since $\sim$ is stable with respect to $\overset{v}{\Rightarrow}_{\mathrm{u}}$, from $z'_i \sim z_i$ it follows that there exists z'' such that $z_i \overset{v}{\Rightarrow}_{\mathrm{u}} z''$ and $z'' \sim z'$. Since z_i is on the path $q_1 \overset{\Upsilon}{\Rightarrow}_{\mathrm{c}} y_1$, by definition 14 there exists y'_1 such that $y_1 \overset{v}{\Rightarrow}_{\mathrm{u}} y'_1$ and $z'' \sim y'_1$. Since $y_1 \sim y_2$ and $\sim$ is stable with respect to $\overset{v}{\Rightarrow}_{\mathrm{u}}$, there exists y'_2 such that $y_2 \overset{v}{\Rightarrow}_{\mathrm{u}} y'_2$ and $y'_2 \sim y'_1 \sim z'' \sim z'$, showing (ii).

Case 2: z is on a subpath $z'_i \overset{\Upsilon}{\Rightarrow}_{\mathrm{soe}} z'_{i+1}$.

If $z \overset{u}{\rightarrow} z'$ for some $u \in (\Sigma_{\mathrm{u}} \cap \Upsilon)^*$, then by lemma 7 there exists z'' such that $z'_i \overset{u}{\Rightarrow}_{\mathrm{u}} z''$ and $z'' \sim z'$. Since $\sim$ is stable with respect to $\overset{u}{\Rightarrow}_{\mathrm{u}}$, from $z_i \sim z'_i$ it follows that there exists $\bar{z}$ such that $z_i \overset{u}{\Rightarrow}_{\mathrm{u}} \bar{z}$ and $\bar{z} \sim z''$. Since z_i is on the path $q_1 \overset{\Upsilon}{\Rightarrow}_{\mathrm{c}} y_1$, it follows from definition 14 that $\bar{z} \sim z_j$ for some j. Thus $z' \sim z'' \sim \bar{z} \sim z_j \sim z'_j$, showing (i).

If $z \overset{v}{\Rightarrow}_{\mathrm{u}} z'$ for some $v \in \Sigma_{\mathrm{u}} \cap \Omega$, then by lemma 7 there exists z'' such that $z'_i \overset{v}{\Rightarrow}_{\mathrm{u}} z''$ and $z' \sim z''$. Since $\sim$ is stable with respect to $\overset{v}{\Rightarrow}_{\mathrm{u}}$ and since $z_i \sim z'_i$, there exists $\bar{z}$ such that $z_i \overset{v}{\Rightarrow}_{\mathrm{u}} \bar{z}$ and $\bar{z} \sim z''$. Since z_i is on the path $q_1 \overset{\Upsilon}{\Rightarrow}_{\mathrm{c}} y_1$, it follows from definition 14 that there exists y'_1 such that $y_1 \overset{v}{\Rightarrow}_{\mathrm{u}} y'_1$ and $\bar{z} \sim y'_1$. Since $y_1 \sim y_2$ and $\sim$ is stable with respect to $\overset{v}{\Rightarrow}_{\mathrm{u}}$, there exists y'_2 such that $y_2 \overset{v}{\Rightarrow}_{\mathrm{u}} y'_2$ and $y'_2 \sim y'_1 \sim \bar{z} \sim z'' \sim z'$, showing (ii).

This completes the proof that $q_2 \overset{\Upsilon}{\Rightarrow}_{\mathrm{c}} y_2$. □

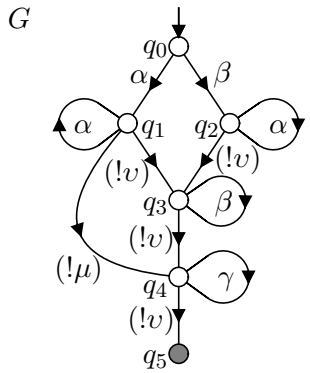

Figure 2: Synthesis observation equivalence does not imply 3-state synthesis observation equivalence.

C Implemented Synthesis Observation Equivalence

This appendix discusses the properties of the implemented variation of weak synthesis observation equivalence used for the experiments in section 5. The implementation differs from true weak synthesis observation equivalence, because checking for equivalence to all states on a $\Rightarrow_{\mathrm{c}}$-path would make the Backward Search (algorithm 3) exponential. To avoid this, the algorithm only compares with three states that are readily accessible at the time of testing. This results in the following variation of synthesis observation equivalence.

Definition 16 Let $G = \langle \Sigma, Q, \rightarrow, Q^\circ \rangle$ be an automaton with $\Sigma = \Omega \,\dot\cup\, \Upsilon$. An equivalence relation $\sim \subseteq Q \times Q$ is a *3-state synthesis observation equivalence* on G with respect to Υ, if $\sim$ is stable with respect to $\overset{\Upsilon}{\Rightarrow}_{\mathrm{wsoe3}}$, to $\overset{\sigma}{\Rightarrow}_{\mathrm{wsoe3}}$ for each $\sigma \in \Sigma_{\mathrm{c}} \cap \Omega$, and to $\overset{\upsilon}{\Rightarrow}_{\mathrm{u}}$ for each $\upsilon \in \Sigma_{\mathrm{u}}$.

- $x \overset{\Upsilon}{\Rightarrow}_{\mathrm{wsoe3}} y$ if $x \overset{\Upsilon}{\Rightarrow}_{\mathrm{soe}} z \overset{\Upsilon}{\Rightarrow}_{\mathrm{c3}} y$ for some $z \in Q$.
- $x \overset{\sigma}{\Rightarrow}_{\mathrm{wsoe3}} y$ if $x \overset{\sigma}{\Rightarrow}_{\mathrm{soe}} z \overset{\Upsilon}{\Rightarrow}_{\mathrm{c3}} y$ for some $z \in Q$.
- $x \overset{\Upsilon}{\Rightarrow}_{\mathrm{c3}} y$ if there exists a path $x = z_0 \overset{\tau_1}{\rightarrow} \cdots \overset{\tau_k}{\rightarrow} z_k = y$ such that $\tau_1, \ldots, \tau_k \in \Upsilon$, and $z_j \overset{u}{\rightarrow} z'$ for $u \in (\Sigma_{\mathrm{u}} \cap \Upsilon)^*$ implies $z' \sim z_j$ or $z' \sim z_{j+1}$ or $z' \sim y$, and $z_j \overset{\upsilon}{\Rightarrow}_{\mathrm{u}} z'$ for $\upsilon \in \Sigma_{\mathrm{u}} \cap \Omega$ implies $y \overset{\upsilon}{\Rightarrow}_{\mathrm{u}} z''$ for some $z'' \sim z'$.

Example 4 Consider automaton G in figure 2 with $\Sigma_{\mathrm{u}} = \Upsilon = \{!\upsilon, !\mu\}$. An equivalence relation $\sim$ such that $q_1 \sim q_2$ is a synthesis observation equivalence and

a weak synthesis observation equivalence, but not a 3-state synthesis observation equivalence relation.

States q_1 and q_2 are synthesis observation equivalent: states q_3, q_4 and q_5 are reachable from both q_1 and q_2 by exactly the same relations $\Rightarrow_{\mathrm{soe}}$ and $\Rightarrow_{\mathrm{u}}$, and in addition it holds that $q_1 \stackrel{\alpha}{\Rightarrow}_{\mathrm{soe}} q_1$ and $q_2 \stackrel{\alpha}{\Rightarrow}_{\mathrm{soe}} q_2$ with $q_1 \sim q_2$.

Also note that $q_1 \stackrel{\alpha}{\Rightarrow}_{\mathrm{wsoe}} q_5$ because $q_1 \stackrel{\alpha}{\rightarrow} q_1 \stackrel{!\upsilon}{\rightarrow} q_3 \stackrel{!\upsilon}{\rightarrow} q_4 \stackrel{!\upsilon}{\rightarrow} q_5$ and the state q_4 reached by $q_1 \stackrel{!\mu}{\rightarrow} q_4$ is on this path. Since also $q_2 \stackrel{\alpha}{\Rightarrow}_{\mathrm{wsoe}} q_5$, states q_1 and q_2 can be weakly synthesis observation equivalent.

However, $q_1 \stackrel{\alpha}{\Rightarrow}_{\mathrm{wsoe3}} q_5$ does not hold, because the state q_4 is not equivalent to q_1, q_3, or q_5. As on the other hand $q_2 \stackrel{\alpha}{\Rightarrow}_{\mathrm{wsoe3}} q_5$, states q_1 and q_2 cannot be 3-state synthesis observation equivalent.

The example shows that 3-state synthesis observation equivalence is different from both synthesis observation equivalence and weak synthesis observation equivalence. Most importantly, synthesis observation equivalence does not imply 3-state synthesis observation equivalence, although the experiments suggest that 3-state synthesis observation equivalence usually is coarser in practice.

On the other hand, it is true that 3-state synthesis observation equivalence implies weak synthesis observation equivalence, so by theorem 3, 3-state synthesis observation equivalence also produces correct abstractions.

Theorem 9 Let $G = \langle \Sigma, Q, \rightarrow, Q^\circ \rangle$ be an automaton and let $\sim$ be a 3-state synthesis observation equivalence on G with respect to Υ. Then $\sim$ is a weak synthesis observation equivalence on G with respect to Υ.

Proof. Let $x_1, x_2, y_1 \in Q$ such that $x_1 \sim x_2$ and $x_1 \stackrel{\Upsilon}{\Rightarrow}_{\mathrm{wsoe}} y_1$ or $x_1 \stackrel{\sigma}{\Rightarrow}_{\mathrm{wsoe}} y_1$ for some $\sigma \in \Sigma_{\mathrm{c}} \cap \Omega$ or $x_1 \stackrel{\upsilon}{\Rightarrow}_{\mathrm{u}} y_1$ for some $\upsilon \in \Sigma_{\mathrm{u}}$. It must be shown that there exists y_2 such that $x_2 \stackrel{\sigma}{\Rightarrow}_{\mathrm{wsoe}} y_2$ or $x_2 \stackrel{\Upsilon}{\Rightarrow}_{\mathrm{wsoe}} y_2$ or $x_2 \stackrel{\upsilon}{\Rightarrow}_{\mathrm{u}} y_2$ and $y_1 \sim y_2$.

If $x_1 \stackrel{\upsilon}{\Rightarrow}_{\mathrm{u}} y_1$ then since $x_1 \sim x_2$ and $\sim$ is stable with respect to $\stackrel{\upsilon}{\Rightarrow}_{\mathrm{u}}$ it follows that there exists y_2 such that $x_2 \stackrel{\upsilon}{\Rightarrow}_{\mathrm{u}} y_2$.

$x_1 \stackrel{\sigma}{\Rightarrow}_{\mathrm{wsoe}} y_1$ or $x_1 \stackrel{\Upsilon}{\Rightarrow}_{\mathrm{wsoe}} y_1$ means $x_1 \stackrel{\sigma}{\Rightarrow}_{\mathrm{soe}} q_1 \stackrel{\Upsilon}{\Rightarrow}_{\mathrm{c}} y_1$ or $x_1 \stackrel{\Upsilon}{\Rightarrow}_{\mathrm{soe}} q_1 \stackrel{\Upsilon}{\Rightarrow}_{\mathrm{c}} y_1$ respectively, where $q_1 \stackrel{\Upsilon}{\Rightarrow}_{\mathrm{c}} y_1$ is a path $q_1 = z_0 \stackrel{\tau_1}{\rightarrow} \cdots \stackrel{\tau_k}{\rightarrow} z_k = y_1$ with $\tau_1, \ldots, \tau_k \in \Upsilon$. Then also $x_1 \stackrel{\sigma}{\Rightarrow}_{\mathrm{wsoe3}} q_1$ or $x_1 \stackrel{\Upsilon}{\Rightarrow}_{\mathrm{wsoe3}} q_1$ by definition 16. Since $x_1 \sim x_2$ and $\sim$ is stable with respect to $\stackrel{\sigma}{\Rightarrow}_{\mathrm{wsoe3}}$ and $\stackrel{\Upsilon}{\Rightarrow}_{\mathrm{wsoe3}}$, there exists q_2 such that $x_2 \stackrel{\sigma}{\Rightarrow}_{\mathrm{wsoe3}} q_2$ or $x_2 \stackrel{\Upsilon}{\Rightarrow}_{\mathrm{wsoe3}} q_2$ and $q_1 \sim q_2$. It is first shown by induction on $i = 0, \ldots, k$ that there exists a path

$$q_2 = z'_0 \Rightarrow z'_1 \Rightarrow \cdots \Rightarrow z'_k = y_2 \tag{20}$$

such that $z'_i \sim z_i$ for all i, and each $z'_i \Rightarrow z'_{i+1}$ is $z'_i \stackrel{\tau_{i+1}}{\Longrightarrow}_{\mathrm{u}} z'_{i+1}$ if $\tau_{i+1} \in \Sigma_{\mathrm{u}}$ and $z'_i \stackrel{\Upsilon}{\Rightarrow}_{\mathrm{wsoe3}} z'_{i+1}$ if $\tau_{i+1} \in \Sigma_{\mathrm{c}}$.

Base case. For $i = 0$, the claim clearly holds as $z'_0 = q_2 \sim q_1 = z_0$.

Inductive step. Assume the path up to z'_i with $z_i \sim z'_i$ has been constructed for some i. To obtain z'_{i+1} consider two cases. If $\tau_{i+1} \in \Sigma_{\mathrm{u}}$, then since $\sim$ is stable with respect to $\stackrel{\tau_{i+1}}{\Longrightarrow}_{\mathrm{u}}$, from $z_i \stackrel{\tau_{i+1}}{\longrightarrow} z_{i+1}$ it follows that there exists z'_{i+1} such that $z'_i \stackrel{\tau_{i+1}}{\Longrightarrow}_{\mathrm{u}} z'_{i+1}$ and $z'_{i+1} \sim z_{i+1}$. If $\tau_{i+1} \in \Sigma_{\mathrm{c}}$, then since $\sim$ is stable with respect to $\stackrel{\Upsilon}{\Rightarrow}_{\mathrm{wsoe3}}$, from $z_i \stackrel{\tau_{i+1}}{\longrightarrow} z_{i+1}$ it follows that there exists z'_{i+1} such that $z'_i \stackrel{\Upsilon}{\Rightarrow}_{\mathrm{wsoe3}} z'_{i+1}$ and $z'_{i+1} \sim z_{i+1}$.

This shows the existence of the path (20). As $x_2 \stackrel{\sigma}{\Rightarrow}_{\mathrm{wsoe3}} q_2$ or $x_2 \stackrel{\Upsilon}{\Rightarrow}_{\mathrm{wsoe3}} q_2$, there exists p_2 such that $x_2 \stackrel{\sigma}{\Rightarrow}_{\mathrm{soe}} p_2 \stackrel{\Upsilon}{\Rightarrow}_{\mathrm{c3}} q_2$ or $x_2 \stackrel{\Upsilon}{\Rightarrow}_{\mathrm{soe}} p_2 \stackrel{\Upsilon}{\Rightarrow}_{\mathrm{c3}} q_2$. Then the path $p_2 \stackrel{\Upsilon}{\Rightarrow}_{\mathrm{c3}} q_2 \Rightarrow y_2$ can be written as

$$p_2 = q'_0 \stackrel{\Upsilon}{\Rightarrow}_{\mathrm{c3}} z'_0 \Rightarrow q'_1 \Rightarrow z'_1 \Rightarrow \cdots \Rightarrow q'_k \Rightarrow z'_k = y_2 \ , \tag{21}$$

where each $z'_i \Rightarrow q'_{i+1} \Rightarrow z'_{i+1}$ is $z'_i \stackrel{\tau_{i+1}}{\Longrightarrow}_{\mathrm{u}} q'_{i+1} = z'_{i+1}$ if $\tau_{i+1} \in \Sigma_{\mathrm{u}}$, and $z'_i \stackrel{\Upsilon}{\Rightarrow}_{\mathrm{soe}} q'_{i+1} \stackrel{\Upsilon}{\Rightarrow}_{\mathrm{c3}} z'_{i+1}$ if $\tau_{i+1} \in \Sigma_{\mathrm{c}}$. It remains to be shown that $p_2 \stackrel{\Upsilon}{\Rightarrow}_{\mathrm{c}} y_2$. According to definition 14, the following properties need to be shown for every state z on the path (21).

(i) If $z \stackrel{u}{\rightarrow} z'$ for some $u \in (\Sigma_{\mathrm{u}} \cap \Upsilon)^*$ then $z' \sim \bar{z}$ for some $\bar{z}$ on the path (21).

(ii) If $z \stackrel{\upsilon}{\Rightarrow}_{\mathrm{u}} z'$ for some $\upsilon \in \Sigma_{\mathrm{u}} \cap \Omega$ then there exists $y'_2 \sim z'$ such that $y_2 \stackrel{\upsilon}{\Rightarrow}_{\mathrm{u}} y'_2$.

Let z be a state on the path (21). Consider three cases.

Case 1: z is on a subpath $z'_i \stackrel{\tau_{i+1}}{\Longrightarrow}_{\mathrm{u}} q'_{i+1} = z'_{i+1}$. Then $\tau_{i+1} \in \Sigma_{\mathrm{u}} \cap \Upsilon$.

If $z \stackrel{u}{\rightarrow} z'$ for some $u \in (\Sigma_{\mathrm{u}} \cap \Upsilon)^*$, then clearly $z'_i \stackrel{\tau_{i+1}}{\Longrightarrow}_{\mathrm{u}} z'$. Since $\sim$ is stable with respect to $\stackrel{\tau_{i+1}}{\Longrightarrow}_{\mathrm{u}}$, from $z'_i \sim z_i$ it follows that there exists z'' such that $z_i \stackrel{\tau_{i+1}}{\Longrightarrow}_{\mathrm{u}} z''$ and $z'' \sim z'$. Since z_i is on the path $q_1 \stackrel{\Upsilon}{\Rightarrow}_{\mathrm{c}} y_1$, it follows from definition 14 that $z'' \sim z_j$ for some j. Thus $z' \sim z'' \sim z_j \sim z'_j$, showing (i).

If $z \stackrel{\upsilon}{\Rightarrow}_{\mathrm{u}} z'$ for some $\upsilon \in \Sigma_{\mathrm{u}} \cap \Omega$, then $z'_i \stackrel{\upsilon}{\Rightarrow}_{\mathrm{u}} z'$. Since $\sim$ is stable with respect to $\stackrel{\upsilon}{\Rightarrow}_{\mathrm{u}}$, from $z'_i \sim z_i$ it follows that there exists z'' such that $z_i \stackrel{\upsilon}{\Rightarrow}_{\mathrm{u}} z''$ and $z'' \sim z'$. Since z_i is on the path $q_1 \stackrel{\Upsilon}{\Rightarrow}_{\mathrm{c}} y_1$, by definition 14 there exists y'_1 such that $y_1 \stackrel{\upsilon}{\Rightarrow}_{\mathrm{u}} y'_1$ and $z'' \sim y'_1$. Since $y_1 \sim y_2$ and $\sim$ is stable with respect to $\stackrel{\upsilon}{\Rightarrow}_{\mathrm{u}}$, there exists y'_2 such that $y_2 \stackrel{\upsilon}{\Rightarrow}_{\mathrm{u}} y'_2$ and $y'_2 \sim y'_1 \sim z'' \sim z'$, showing (ii).

Case 2: z is on a subpath $z'_i \stackrel{\Upsilon}{\Rightarrow}_{\mathrm{soe}} q'_{i+1}$.

If $z \xrightarrow{u} z'$ for some $u \in (\Sigma_{\mathrm{u}} \cap \Upsilon)^*$, then by lemma 7 there exists z'' such that $z_i' \overset{u}{\Rightarrow}_{\mathrm{u}} z''$ and $z' \sim z''$. Since $\sim$ is stable with respect to $\overset{u}{\Rightarrow}_{\mathrm{u}}$, from $z_i \sim z_i'$ it follows that there exists $\bar{z}$ such that $z_i \overset{u}{\Rightarrow}_{\mathrm{u}} \bar{z}$ and $\bar{z} \sim z''$. Since z_i is on the path $q_1 \overset{\Upsilon}{\Rightarrow}_{\mathrm{c}} y_1$, it follows from definition 14 that $\bar{z} \sim z_j$ for some j. Thus $z' \sim z'' \sim \bar{z} \sim z_j \sim z_j'$, showing (i).

If $z \overset{v}{\Rightarrow}_{\mathrm{u}} z'$ for some $v \in \Sigma_{\mathrm{u}} \cap \Omega$, then by lemma 7 there exists z'' such that $z_i' \overset{v}{\Rightarrow}_{\mathrm{u}} z''$ and $z' \sim z''$. Since $\sim$ is stable with respect to $\overset{v}{\Rightarrow}_{\mathrm{u}}$ and since $z_i \sim z_i'$, there exists $\bar{z}$ such that $z_i \overset{v}{\Rightarrow}_{\mathrm{u}} \bar{z}$ and $\bar{z} \sim z''$. Since z_i is on the path $q_1 \overset{\Upsilon}{\Rightarrow}_{\mathrm{c}} y_1$, it follows from definition 14 that there exists y_1' such that $y_1 \overset{v}{\Rightarrow}_{\mathrm{u}} y_1'$ and $\bar{z} \sim y_1'$. Since $y_1 \sim y_2$ and $\sim$ is stable with respect to $\overset{v}{\Rightarrow}_{\mathrm{u}}$, there exists y_2' such that $y_2 \overset{v}{\Rightarrow}_{\mathrm{u}} y_2'$ and $y_2' \sim y_1' \sim \bar{z} \sim z'' \sim z'$, showing (ii).

Case 3: z is on a subpath $q_i' \overset{\Upsilon}{\Rightarrow}_{\mathrm{c3}} z_i'$.

If $z \xrightarrow{u} z'$ for some $u \in (\Sigma_{\mathrm{u}} \cap \Upsilon)^*$, then by definition 16 it holds that $z' \sim \bar{z}$ for some $\bar{z}$ on the path $q_i' \overset{\Upsilon}{\Rightarrow}_{\mathrm{c3}} z_i'$. This state $\bar{z}$ clearly is on the path (21), showing (i).

If $z \overset{v}{\Rightarrow}_{\mathrm{u}} z'$ for some $v \in \Sigma_{\mathrm{u}} \cap \Omega$, then by definition 16 there exists $z'' \sim z'$ such that $z_i' \overset{v}{\Rightarrow}_{\mathrm{u}} z''$. Since $\sim$ is stable with respect to $\overset{v}{\Rightarrow}_{\mathrm{u}}$ and since $z_i \sim z_i'$, there exists $\bar{z}$ such that $z_i \overset{v}{\Rightarrow}_{\mathrm{u}} \bar{z}$ and $\bar{z} \sim z''$. Since z_i is on the path $q_1 \overset{\Upsilon}{\Rightarrow}_{\mathrm{c}} y_1$, it follows from definition 14 that there exists y_1' such that $y_1 \overset{v}{\Rightarrow}_{\mathrm{u}} y_1'$ and $\bar{z} \sim y_1'$. Since $y_1 \sim y_2$ and $\sim$ is stable with respect to $\overset{v}{\Rightarrow}_{\mathrm{u}}$, there exists y_2' such that $y_2 \overset{v}{\Rightarrow}_{\mathrm{u}} y_2'$ and $y_2' \sim y_1' \sim \bar{z} \sim z'' \sim z'$, showing (ii).

This completes the proof that $p_2 \overset{\Upsilon}{\Rightarrow}_{\mathrm{c}} y_2$. □