Working Paper Series
ISSN 1177-777X

FIVE ABSTRACTION RULES TO REMOVE TRANSITIONS WHILE PRESERVING COMPOSITIONAL SYNTHESIS RESULTS

Sahar Mohajerani, Robi Malik, Martin Fabian

Working Paper: 01/2012
March 13, 2012

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

FIVE ABSTRACTION RULES TO REMOVE TRANSITIONS WHILE PRESERVING COMPOSITIONAL SYNTHESIS RESULTS

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

March 13, 2012

Abstract

This working paper investigates under which conditions *transitions* can be removed from an automaton while preserving important synthesis properties. The work is part of a framework for *compositional synthesis* of least restrictive controllable and nonblocking supervisors for modular discrete event systems. The method for transition removal complements previous results, which are largely focused on state merging. Issues concerning transition removal in synthesis are discussed, and *redirection maps* are introduced to enable a supervisor to process an event, even though the corresponding transition is no longer present in the model. Based on the results, different tech-

niques are proposed to remove controllable and uncontrollable transitions, and an example shows the potential of the method for practical problems.

1 Introduction

Supervisory control theory [16] provides a general framework to compute least restrictive strategies to control a given *plant* such that its behaviour satisfies a given *specification*. Synthesis for systems with a large number of components is impeded by an inherent complexity problem known as *state-space explosion*. A lot of research has been devoted to overcome the state-space explosion problem, and also to find more comprehensible supervisors [7, 9, 16, 19].

Compositional methods seek to avoid large state spaces using *abstraction* and have been used in verification [1, 3, 6] and synthesis [7, 14, 15]. In a system with a large number of components, it is often possible to simplify individual components before composing them with the rest of the system, achieving significant performance improvements. Several ways to simplify components have been investigated in recent years.

Natural projection is a standard and effective way to compute abstractions, although strong restrictions need to be imposed to ensure the preservation of synthesis results [5, 17]. *Observation equivalence* [13] and *conflict equivalence* [12] are well-known abstraction methods for nonblocking verification [6], but for synthesis these abstractions can only be applied in combination with unobservable events [10, 18], which limits their applicability.

Recently, frameworks for compositional synthesis based on abstractions of nondeterministic automata have been proposed [7, 14, 15], in some cases showing substantial reduction of the number of states encountered during synthesis. This working paper seeks to enhance these methods by providing means to remove *transitions*. This is important, because for large systems, the number of transitions may exceed the number of states by several orders of magnitude.

Compositional verification typically includes observation equivalence abstraction, which allows for transition removal using the transitive reduction [4], but observation equivalence does not necessarily preserve synthesis results [15]. *Supervision equivalence* [7] allows for transition removal, but relies on additional state labels that make some desirable abstractions impossible. The methods [14, 15] avoid *event hiding* that may cause problems in synthesis abstraction, but these approaches make it difficult to remove transitions.

This working paper proposes some concrete means to identify transitions that are redundant for the purpose of synthesis. These methods are based on observation equivalence [13], but are more restrictive because of the need to preserve

synthesis results. It is also shown how to restore the removed transitions to enable a synthesised supervisor to make control decisions based on a model with removed transitions.

This working paper is organised as follows. After the preliminaries in section 2, a framework to support transition removal in compositional synthesis is presented in section 3. In section 4, a sufficient condition for transition-removing abstraction is described, and in section 5, concrete methods to remove transitions are given and proven to be sound. Finally, section 6 demonstrates transition removal using a practical example, and section 7 adds some concluding remarks.

2 Preliminaries

2.1 Events and Languages

The behaviour of discrete event systems is described using events and languages. *Events* represent incidents that cause transitions from one state to another and are taken from a finite alphabet Σ. For the purpose of supervisory control, this alphabet is partitioned into the set Σ_c of *controllable* events and the set Σ_u of *uncontrollable* events. Controllable events can be disabled by a supervisor, while uncontrollable events occur spontaneously. The special *termination event* $\omega \in \Sigma_c$ denotes completion of a task.

Σ^* is the set of all finite traces of events from Σ, including the *empty trace* ε. 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 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 Finite-State Automata

Discrete event systems are typically modelled as deterministic automata, but nondeterministic automata may be obtained as intermediate results from 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 $Q^\circ \in Q$ is the set of *initial states*.

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 $s \in \Sigma^*$. For an alphabet $\Omega \subseteq \Sigma$, the notation $x \xrightarrow{\Omega} y$ means $x \xrightarrow{\sigma} y$

for some $\sigma \in \Omega$, and $G \xrightarrow{s} x$ means $q^\circ \xrightarrow{s} x$ for some $q^\circ \in Q^\circ$. The *language* of automaton G is $\mathcal{L}(G) = \{\, s \in \Sigma^* \mid G \xrightarrow{s} \}$. Finally, G is *deterministic*, if $|Q^\circ| \leq 1$, and $x \xrightarrow{\sigma} y_1$ and $x \xrightarrow{\sigma} y_2$ always implies $y_1 = y_2$.

A special requirement is that states reached by the termination event ω do not have any outgoing transitions, i.e., if $x \xrightarrow{\omega} y$ then there does not exist $\sigma \in \Sigma$ such that $y \xrightarrow{\sigma}$. This ensures that the termination event, if it occurs, is always the final event of any trace. 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 shown shaded in the figures of this paper instead of explicitly showing ω-transitions.

When multiple automata are brought together to interact, lock-step synchronisation in the style of [8] is used.

Definition 2 Let $G_1 = \langle \Sigma_1, Q_1, \rightarrow_1, Q_1^\circ \rangle$ and $G_2 = \langle \Sigma_2, Q_2, \rightarrow_2, Q_2^\circ \rangle$ be two automata. The *synchronous composition* of G_1 and G_2 is

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

where

$$\begin{array}{l} (x,y) \xrightarrow{\sigma} (x',y') \text{ if } \sigma \in \Sigma_1 \cap \Sigma_2,\ x \xrightarrow{\sigma}_1 x',\ y \xrightarrow{\sigma}_2 y'\,; \\ (x,y) \xrightarrow{\sigma} (x',y) \text{ if } \sigma \in \Sigma_1 \setminus \Sigma_2,\ x \xrightarrow{\sigma}_1 x'\,; \\ (x,y) \xrightarrow{\sigma} (x,y') \text{ if } \sigma \in \Sigma_2 \setminus \Sigma_1,\ y \xrightarrow{\sigma}_2 y'\,. \end{array}$$

2.3 Supervisory Control Theory

Given *plant* and *specification* automata, the *supervisory control theory* [16] 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 this supervisor are *controllability* and *nonblocking*.

Definition 3 Specification $K = \langle \Sigma, Q_K, \rightarrow_K, Q_K^\circ \rangle$ is *controllable* with respect to plant $G = \langle \Sigma, Q_G, \rightarrow_G, Q_G^\circ \rangle$ if, for every trace $s \in \Sigma^*$, every state $x \in Q_K$, and every uncontrollable event $\upsilon \in \Sigma_u$ such that $K \xrightarrow{s} x$ and $G \xrightarrow{s\upsilon}$, it holds that $x \xrightarrow{\upsilon}_K$.

Definition 4 Let $G = \langle \Sigma, Q, \rightarrow, Q^\circ \rangle$. A state $x \in Q$ is called *reachable* in G if $G \rightarrow x$, and *coreachable* if $x \xrightarrow{s\omega}$ for some $s \in \Sigma^*$. G is called *nonblocking* if every reachable state is coreachable.

For a plant G and specification K, it is shown in [16] that there exists a *least restrictive* controllable sublanguage

$$\sup\mathcal{C}_G(K) \subseteq \mathcal{L}(K) \tag{2}$$

such that $\sup\mathcal{C}_G(K)$ is controllable with respect to G and nonblocking, and this language can be computed using a fix-point iteration. This result can be reformulated in automata form, using an iteration on the state set. The synthesis result for an automaton G is obtained by restricting G to a maximal set of controllable and nonblocking states.

Definition 5 [11] Let $G = \langle \Sigma, Q, \rightarrow, Q^\circ \rangle$ be an automaton. The *synthesis step* operator $\Theta_G \colon 2^Q \rightarrow 2^Q$ for G is defined by $\Theta_G(X) = \Theta_G^{\mathrm{cont}}(X) \cap \Theta_G^{\mathrm{nonb}}(X)$, where

$$\Theta_G^{\mathrm{cont}}(X) = \{\, x \in X \mid x \stackrel{\Sigma_u}{\rightarrow} y \text{ implies } y \in X \,\} ; \tag{3}$$

$$\Theta_G^{\mathrm{nonb}}(X) = \{\, x \in X \mid x \stackrel{t\omega}{\rightarrow}_{|X} \text{ for some } t \in \Sigma^* \,\} . \tag{4}$$

Theorem 1 [11] Let $G = \langle \Sigma, Q, \rightarrow, Q^\circ \rangle$. The synthesis step operator Θ_G has a greatest fix-point $\mathrm{gfp}\Theta_G = \hat{\Theta}_G \subseteq Q$. If the state set Q is finite, then the sequence $X^0 = Q$, $X^{i+1} = \Theta_G(X^i)$ reaches this fix-point in a finite number of steps, i.e., $\hat{\Theta}_G = X^n$ for some $n \geq 0$.

Definition 6 The *synthesis result* for $G = \langle \Sigma, Q, \rightarrow, Q^\circ \rangle$ is $\sup\mathcal{CN}(G) = G_{|\hat{\Theta}_G}$, where $G_{|X} = \langle \Sigma, Q, \rightarrow_{|X}, Q^\circ \cap X \rangle$ with $\rightarrow_{|X} = \{\, (x, \sigma, y) \in \rightarrow \mid x, y \in X \,\}$ denotes the *restriction* of G to $X \subseteq Q$.

Theorem 2 Let $G = \langle \Sigma, Q, \rightarrow, Q^\circ \rangle$ be a deterministic automaton. $\sup\mathcal{CN}(G)$ is the least restrictive subautomaton of G that is controllable with respect to G and nonblocking.

The synthesis operator $\sup\mathcal{CN}$ performs synthesis for a plant automaton G. A simple transformation [7] exists to transform problems that also involve specifications into the plant-only control problems considered in this working paper.

The result of the synthesis operator is an automaton $\sup\mathcal{CN}(G)$ or a language $\mathcal{L}(\sup\mathcal{CN}(G))$, which describes the behaviour of a controlled system. In practice this is implemented as a *supervisor* that decides which controllable events are to be enabled or disabled in a given state. In this paper, a supervisor is a map

$$\mathcal{S} \colon \Sigma^* \rightarrow \{0, 1\} . \tag{5}$$

If $\mathcal{S}(s\sigma) = 0$ for some $s \in \Sigma^*$ and $\sigma \in \Sigma_c$ then the supervisor disables the controllable event σ after observing trace s, otherwise it enables σ. This results in the following *closed-loop behaviour* $\mathcal{L}(\mathcal{S}/G)$ of the plant G under the control of supervisor $\mathcal{S}$:

$$\mathcal{L}(\mathcal{S}/G) = \{\, s \in \mathcal{L}(G) \mid \mathcal{S}(s) = 1 \,\} . \tag{6}$$

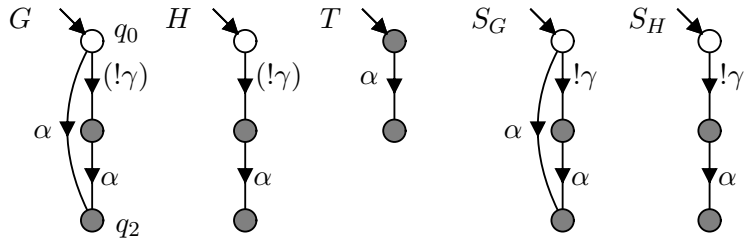

Figure 1: Example of transition removal.

A supervisor can be constructed naturally from a language $L \subseteq \Sigma^*$, by letting $\mathcal{S}_L(s) = 1$ if and only if $s \in L$. For such a supervisor to be feasible, L must be controllable [16].

3 Compositional Synthesis

Many supervisory control problems can be presented as a set of interacting components. Then the synthesis problem consists of finding the least restrictive controllable and nonblocking supervisor for a set of plants,

$$\mathcal{G} = \{G_1, G_2, \ldots, G_n\} \,. \tag{7}$$

Compositional synthesis exploits the modularity of such systems and avoids building the complete synchronous product. Individual components G_i are simplified and replaced by smaller abstractions H_i. Synchronous composition is computed step by step, abstracting again the intermediate results. Eventually the abstractions result in a single automaton H, the abstract description of the system (7). Once found, H is used instead of the original system to calculate a synthesis result that leads to a solution for the original synthesis problem (7).

Individual components G_i typically contain events that do not appear in any other component G_j with $j \neq i$. These events are called *local events*. In the following, local events are denoted by the set Υ, and $\Omega = \Sigma \setminus \Upsilon$ denotes the non-local or *shared* events. Local events are helpful to find abstractions and are parenthesised in the figures.

This paper focuses on abstractions that remove transitions from an automaton. This leads to a problem, because it is no longer obvious how to construct a supervisor from such an abstraction. After removal of transitions it is not clear how a supervisor can enact control over the events labelling the removed transitions.

Example 1 Consider the modular system $\mathcal{G} = \{G, T\}$ in figure 1 with $\Sigma_u = \{!\gamma\}$ where $!\gamma$ is the only local event. Automaton H is obtained by removing $q_0 \xrightarrow{\alpha} q_2$.

Although H is an appropriate abstraction of G, as explained below in example 2, the supervisor $S_H = \sup\mathcal{CN}(H \parallel T)$ disables event α in the initial state, and therefore is not a least restrictive supervisor for $G \parallel T$.

To solve this problem, the models (7) are augmented by a *redirection map* that contains the information needed to finally implement a supervisor.

Definition 7 A *synthesis pair* is a pair $(\mathcal{G}; \mathcal{D})$, where

- $\mathcal{G} = \{G_1, G_2, \ldots, G_n\}$ is a set of uncontrolled plant automata;
- $\mathcal{D}\colon \Sigma^* \to \Sigma^*$ is a prefix-preserving *redirection map*, i.e., a map such that $s \sqsubseteq t$ implies $\mathcal{D}(s) \sqsubseteq \mathcal{D}(t)$.

Synthesis pairs are a variant of *synthesis triples* [14] that collect all the information needed for the transition-based abstractions considered in this working paper. The compositional synthesis algorithm manipulates synthesis pairs. Each pair represents a partially solved synthesis problem, consisting of the plant model $\mathcal{G}$ to be controlled and the redirection map $\mathcal{D}$, which maps each input trace s accepted by the original plant before all abstractions, to a trace accepted by the current abstracted plant $\mathcal{G}$. A solution to the abstracted synthesis problem $\mathcal{G}$ can be interpreted as a supervisor for the original plant by taking the redirection map into account.

Definition 8 For every synthesis pair $(\mathcal{G}; \mathcal{D})$, define the represented supervisor map $\mathcal{S}_{(\mathcal{G};\mathcal{D})}\colon \Sigma^* \to \{0, 1\}$ as follows:

$$\mathcal{S}_{(\mathcal{G};\mathcal{D})}(s) = \begin{cases} 1, & \text{if } \mathcal{D}(s) \in \mathcal{L}(\sup\mathcal{CN}(\mathcal{G})); \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Compositional synthesis starts by converting a control problem such as (7) into the synthesis pair $(\mathcal{G}_0; \mathrm{id})$ where $\mathcal{G}_0 = \{G_1, G_2, \ldots, G_n\}$ and $\mathrm{id}\colon \Sigma^* \to \Sigma^*$ is the identity map, i.e, $\mathrm{id}(s) = s$. This initial synthesis pair is repeatedly abstracted in such a way that the supervisor obtained from the abstraction remains a solution for the original synthesis problem. To ensure this property, each new synthesis pair needs to be *synthesis equivalent* to the previous pair.

Definition 9 Two synthesis pairs $(\mathcal{G}_1; \mathcal{D}_1)$ and $(\mathcal{G}_2; \mathcal{D}_2)$ are *synthesis equivalent* with respect to plant G, written $(\mathcal{G}_2; \mathcal{D}_2) \simeq_{\mathrm{synth},G} (\mathcal{G}_1; \mathcal{D}_1)$, if $\mathcal{L}(\mathcal{S}_{(\mathcal{G}_1;\mathcal{D}_1)}/G) = \mathcal{L}(\mathcal{S}_{(\mathcal{G}_2;\mathcal{D}_2)}/G)$. Furthermore, $(\mathcal{G}_1; \mathcal{D}_1)$ and $(\mathcal{G}_2; \mathcal{D}_2)$ are synthesis equivalent, written $(\mathcal{G}_2; \mathcal{D}_2) \simeq_{\mathrm{synth}} (\mathcal{G}_1; \mathcal{D}_1)$, if $(\mathcal{G}_2; \mathcal{D}_2) \simeq_{\mathrm{synth},G} (\mathcal{G}_1; \mathcal{D}_1)$ for every automaton G.

Compositional synthesis terminates once $\mathcal{G} = \{H\}$ consists of a single automaton representing the abstracted system description. The following result confirms that the closed-loop behaviour obtained in the end is equal to a solution for the original synthesis problem.

Proposition 3 Let $\mathcal{G}_0 = \{G_1, \ldots, G_n\}$ be a set of automata. Let $(\mathcal{G}_k; \mathcal{D}_k)$ be a synthesis pair such that $(\mathcal{G}_0; \mathrm{id}) \simeq_{\mathrm{synth}, \mathcal{G}_0} (\mathcal{G}_k; \mathcal{D}_k)$. Then

$$\mathcal{L}(\mathcal{S}_{(\mathcal{G}_k;\mathcal{D}_k)}/\mathcal{G}_0) = \mathcal{L}(\mathrm{sup}\mathcal{CN}(\mathcal{G}_0)) \; . \tag{9}$$

Proof. For $(\mathcal{G}_0; \mathrm{id})$ it follows from definition 8 that

$$\mathcal{S}_{(\mathcal{G}_0;\mathrm{id})}(s) = \begin{cases} 1, & \text{if } s \in \mathcal{L}(\mathrm{sup}\mathcal{CN}(\mathcal{G}_0)) \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

By (6), it follows that $\mathcal{L}(\mathcal{S}_{(\mathcal{G}_0;\mathrm{id})}/\mathcal{G}_0) = \{\, s \in \mathcal{G}_0 \mid s \in \mathcal{L}(\mathrm{sup}\mathcal{CN}(\mathcal{G}_0)) \,\}$, which implies $\mathcal{L}(\mathcal{S}_{(\mathcal{G}_0;\mathrm{id})}/\mathcal{G}_0) = \mathcal{L}(\mathrm{sup}\mathcal{CN}(\mathcal{G}_0))$. Then it follows from definition 9 that $\mathcal{L}(\mathcal{S}_{(\mathcal{G}_k;\mathcal{D}_k)}/\mathcal{G}_0) = \mathcal{L}(\mathcal{S}_{(\mathcal{G}_0;\mathrm{id})}/\mathcal{G}_0)) = \mathcal{L}(\mathrm{sup}\mathcal{CN}(\mathcal{G}_0))$. □

4 Transition-Wise Synthesis Equivalence

Several methods are known to abstract synthesis pairs such that the number of states is reduced [7,15]. The abstractions are performed by manipulating the states and transitions of individual automata, such that synthesis equivalence is preserved. To allow for transition removal, state-wise synthesis abstraction, which is a special case of a definition from [15], is augmented by a transition-based concept in definition 11.

Definition 10 Let $G = \langle \Sigma, Q, \rightarrow_G, Q^\circ \rangle$ and $H = \langle \Sigma, Q, \rightarrow_H, Q^\circ \rangle$ be two automata. H is a *state-wise synthesis abstraction* of G with respect to $\Upsilon \subseteq \Sigma$, if it holds for all automata T such that $\Sigma_T \cap \Upsilon = \emptyset$ that $\hat{\Theta}_{G\|T} \subseteq \hat{\Theta}_{H\|T}$.

Definition 11 Let $G = \langle \Sigma, Q, \rightarrow_G, Q^\circ \rangle$ and $H = \langle \Sigma, Q, \rightarrow_H, Q^\circ \rangle$ be two automata. H is a *transition-wise synthesis abstraction* of G with respect to $\Upsilon \subseteq \Sigma$ if for every transition $x \xrightarrow{\sigma}_G y$ there exist $t, u \in \Upsilon^*$ such that:

(i) $x \xrightarrow{tP_\Omega(\sigma)u}_H y$;

(ii) for all automata T such that $\Sigma_T \cap \Upsilon = \emptyset$ and all transitions $(x, x_T) \xrightarrow[|\hat{\Theta}_{G\|T}]{\sigma} (y, y_T)$ of $\mathrm{sup}\mathcal{CN}(G \parallel T)$ it holds that $(x, x_T) \xrightarrow[|\hat{\Theta}_{H\|T}]{tP_\Omega(\sigma)u} (y, y_T)$.

Definition 12 Two automata G and H are state-wise (or transition-wise) *synthesis equivalent* with respect to Υ, if G is a state-wise (or transition-wise) synthesis abstraction of H with respect to Υ and H is a state-wise (or transition-wise) synthesis abstraction of G with respect to Υ.

Although closely related, state-wise and transition-wise synthesis equivalence are independent concepts. If an abstraction is obtained by transition removal, as considered in this working paper, then transition-wise synthesis abstraction implies state-wise synthesis abstraction in only one direction.

Lemma 4 Let $H = \langle \Sigma, Q, \rightarrow_H, Q^\circ \rangle$ be a transition-wise synthesis abstraction of $G = \langle \Sigma, Q, \rightarrow_G, Q^\circ \rangle$ with respect to $\Upsilon \subseteq \Sigma$ such that $\rightarrow_H \subseteq \rightarrow_G$. Then H is a state-wise synthesis abstraction of G.

Proof. Let $\Sigma = \Omega \dot\cup \Upsilon$, and let $T = \langle \Sigma_T, Q_T, \rightarrow_T, Q_T^\circ \rangle$ be an automaton such that $\Sigma_T \cap \Upsilon = \emptyset$. To prove that $\hat\Theta_{G\|T} \subseteq \hat\Theta_{H\|T}$, it is shown by induction on $n \geq 0$ that $\hat\Theta_{G\|T} \subseteq X^n_{H\|T} = \Theta^n_{H\|T}(Q \times Q_T)$.

Base case. $n = 0$. Clearly $\hat\Theta_{G\|T} \subseteq Q \times Q_T = \Theta^0_{H\|T}(Q \times Q_T) = X^0_{H\|T}$.

Inductive step. Let $(x, x_T) \in \hat\Theta_{G\|T}$ for some $n \geq 0$. It is to be shown that $(x, x_T) \in X^{n+1}_{H\|T} = \Theta_{H\|T}(X^n_{H\|T}) = \Theta^{\mathrm{cont}}_{H\|T}(X^n_{H\|T}) \cap \Theta^{\mathrm{nonb}}_{H\|T}(X^n_{H\|T})$.

To see that $(x, x_T) \in \Theta^{\mathrm{cont}}_{H\|T}(X^n_{H\|T})$, let $\upsilon \in \Sigma_u$ such that $(x, x_T) \xrightarrow{\upsilon}_{H\|T} (y, y_T)$. From $\rightarrow_H \subseteq \rightarrow_G$, it follows that $(x, x_T) \xrightarrow{\upsilon}_{G\|T} (y, y_T)$. Since $(x, x_T) \in \hat\Theta_{G\|T}$ and $\upsilon \in \Sigma_u$, it follows that $(x, x_T) \xrightarrow{\upsilon}_{|\hat\Theta_{G\|T}} (y, y_T)$. By definition 12, there exist $t, u \in \Upsilon^*$ such that $(x, x_T) \xrightarrow{tP_\Omega(\upsilon)u}_{|\hat\Theta_{H\|T}} (y, y_T)$. This implies $(y, y_T) \in \hat\Theta_{H\|T} \subseteq \Theta^n_{H\|T}(Q \times Q_T) = X^n_{H\|T}$. As υ and (y, y_T) were chosen arbitrarily, it follows that $(x, x_T) \in \Theta^{\mathrm{cont}}_{H\|T}(X^n_{H\|T})$.

Furthermore, to see that $(x, x_T) \in \Theta^{\mathrm{nonb}}_{H\|T}(X^n_{H\|T})$, note that $(x, x_T) \in \hat\Theta_{G\|T}$ means $(x, x_T) \xrightarrow{t\omega}_{|\hat\Theta_{G\|T}}$ for some $t \in \Sigma^*$. By inductive assumption, it follows that $(x, x_T) \xrightarrow{t\omega}_{|X^n_{H\|T}}$, which by definition implies $(x, x_T) \in \Theta^{\mathrm{nonb}}_{H\|T}(X^n_{H\|T})$. □

To preserve transition-wise synthesis equivalence after removal of a transition, definition 11 requires the existence of a so-called *redirection path* that links the source and target states of the removed transition. A redirection path for transition $x \xrightarrow{\sigma} y$ with respect to Υ is a path $x \xrightarrow{tP_\Omega(\sigma)u} y$ such that $t, u \in \Upsilon^*$. Using these paths, the redirection map is constructed to replace the removed transitions by the matching redirection paths. This enables the supervisor to make control decisions about the removed transitions.

Example 2 Consider again the automata in figure 1. Transition $q_0 \xrightarrow{\alpha} q_2$ can be removed from G, producing the state-wise and transition-wise synthesis equivalent automaton H. From this abstraction, a redirection map $\mathcal{D}\colon \Sigma^* \to \Sigma^*$ is constructed where $\mathcal{D}(\alpha s) = !\gamma\alpha s$ for all $s \in \Sigma^*$ and $\mathcal{D}(s) = s$ for all s such that α is not a prefix of s.

If G in figure 1 is placed in a larger system, say $\mathcal{G} = \{G, T\}$, then the synthesis pair $(\mathcal{G}; \mathrm{id})$ is synthesis equivalent to $(\mathcal{H}; \mathcal{D})$ where $\mathcal{H} = \{H, T\}$. Although the supervisor $S_H = \sup\mathcal{CN}(H \parallel T)$ obtained for $\mathcal{H}$ cannot directly be used to control the original plant $\mathcal{G}$, this becomes possible in combination with the redirection map $\mathcal{D}$. As $\mathcal{D}(\alpha) = !\gamma\alpha \in \mathcal{L}(\sup\mathcal{CN}(H \parallel T))$, the supervisor computed for $(\mathcal{H}, \mathcal{D})$ will enable the controllable event α in the initial state, in the same way as a supervisor computed for the original system $\mathcal{G}$.

It is shown in the following that a redirection map as shown in example 2 can be constructed in all cases where transition removal applied to a component results in a state-wise and transition-wise synthesis equivalent abstraction. First, for a redirection map constructed for individual automata to be used in the context of a synthesis pair, it must be extended to the complete alphabet.

Definition 13 Let $\Sigma_1 \subseteq \Sigma_2$. The *extension* of a prefix-preserving map $\mathcal{D}_1\colon \Sigma_1^* \to \Sigma_1^*$ is $\mathcal{D}_2\colon \Sigma_2^* \to \Sigma_2^*$, defined by

$$\mathcal{D}_2(\varepsilon) = \mathcal{D}_1(\varepsilon) \tag{11}$$

$$\mathcal{D}_2(s\sigma) = \begin{cases} \mathcal{D}_2(s)t' & \text{if } \sigma \in \Sigma_1,\ \mathcal{D}_1(P_{\Sigma_1}(s)) = s', \\ & \text{and } \mathcal{D}_1(P_{\Sigma_1}(s\sigma)) = s't'; \\ \mathcal{D}_2(s)\sigma & \text{if } \sigma \notin \Sigma_1. \end{cases} \tag{12}$$

A redirection map $\mathcal{D}_1$ is extended by copying the additional events without change at the appropriate position into the output stream. The extension $\mathcal{D}_2$ is well-defined if $\mathcal{D}_1$ is a prefix-preserving map. In the following, if the alphabets are clear from the context, a prefix-preserving map is identified with its extension, and $\mathcal{D}_1$ and $\mathcal{D}_2$ are both denoted by $\mathcal{D}$.

For a redirection map to form a synthesis equivalent pair, it must satisfy the following property of being synthesis-preserving, which is closely related to state-wise and transition-wise synthesis equivalence. A map satisfying this requirement can be constructed in all cases where a component is replaced by a state-wise and transition-wise synthesis equivalent abstraction resulting from transition removal.

Definition 14 Let G and H be two automata. A map $\mathcal{D}\colon \Sigma^* \to \Sigma^*$ is called a *synthesis-preserving redirection map* from G to H with respect to $\Upsilon \subseteq \Sigma$ if for all

automata T such that $\Sigma_T \cap \Upsilon = \emptyset$ and for all $s \in (\Sigma \cup \Sigma_T)^*$, it holds that

$$\mathrm{sup}\mathcal{CN}(G \parallel T) \xrightarrow{s} (x, x_T) \quad \text{if and only if} \quad \mathrm{sup}\mathcal{CN}(H \parallel T) \xrightarrow{\mathcal{D}(s)} (x, x_T)\ . \tag{13}$$

Proposition 5 Let $G = \langle \Sigma, Q, \rightarrow_G, Q^\circ \rangle$ and $H = \langle \Sigma, Q, \rightarrow_H, Q^\circ \rangle$ be state-wise and transition-wise synthesis equivalent with respect to $\Upsilon \subseteq \Sigma$, and let $\rightarrow_H \subseteq \rightarrow_G$. Then there exists a synthesis-preserving redirection map from G to H with respect to Υ.

Proof. Let $\Sigma = \Omega \,\dot\cup\, \Upsilon$. Since G and H are transition-wise synthesis equivalent, for every transition $x \xrightarrow{\sigma} y$ there exists a trace $d(x, \sigma) = tP_\Omega(\sigma)u$ where $t, u \in \Upsilon^*$ satisfy the conditions (i) and (ii) in Definition 11. Then construct the prefix-preserving map $\mathcal{D}\colon \Sigma^* \to \Sigma^*$ as follows:

$$\mathcal{D}(\varepsilon) = \varepsilon \tag{14}$$

$$\mathcal{D}(t\sigma) = \begin{cases} \mathcal{D}(t)d(x,\sigma), & \text{if } G \xrightarrow{t} x \xrightarrow{\sigma}; \\ \mathcal{D}(t)\sigma, & \text{otherwise}\ . \end{cases} \tag{15}$$

Now let $T = \langle \Sigma_T, Q_T, \rightarrow_T, Q_T^\circ \rangle$ be an automaton such that $\Sigma_T \cap \Upsilon = \emptyset$. Then d is extended to $\Sigma \cup \Sigma_T$ by letting $d(x, \sigma) = \sigma$ for all $\sigma \in \Sigma_T \setminus \Sigma$, and the extension of $\mathcal{D}$ to $(\Sigma \cup \Sigma_T)^*$ is given by:

$$\mathcal{D}(\varepsilon) = \varepsilon \tag{16}$$

$$\mathcal{D}(t\sigma) = \begin{cases} \mathcal{D}(t)d(x,\sigma), & \text{if } G \xrightarrow{P_\Sigma(t)} x \xrightarrow{P_\Sigma(\sigma)}; \\ \mathcal{D}(t)\sigma, & \text{otherwise}\ . \end{cases} \tag{17}$$

Note that condition (i) in definition 11 ensures that $\mathcal{D}(P_\Sigma(s)) \in \mathcal{L}(G)$ implies $P_\Sigma(s) \in \mathcal{L}(G)$. Furthermore, for all $s = \sigma_1 \cdots \sigma_n$ such that $P_\Sigma(s) \in \mathcal{L}(G)$,

$$\mathcal{D}(s) = d(x_0, \sigma_1)d(x_1, \sigma_2) \cdots d(x_{n-1}, \sigma_n) \tag{18}$$

where $G \xrightarrow{P_\Sigma(\sigma_1 \cdots \sigma_k)} x_k$. It remains to be confirmed that $\mathcal{D}$ satisfies definition 14. Therefore, let $s = \sigma_1 \cdots \sigma_n \in (\Sigma \cup \Sigma_T)^*$.

First assume that $\mathrm{sup}\mathcal{CN}(G \parallel T) \xrightarrow{s} (x, x^T)$. Then there exists a path $G \parallel T \xrightarrow{\varepsilon} (x_0, x_0^T) \xrightarrow{\sigma_1}_{|\hat\Theta_{G\parallel T}} (x_1, x_1^T) \xrightarrow{\sigma_2}_{|\hat\Theta_{G\parallel T}} \cdots \xrightarrow{\sigma_n}_{|\hat\Theta_{G\parallel T}} (x_n, x_n^T) = (x, x^T)$. Consider some $k = 1, \ldots, n$. If $\sigma_k \in \Sigma$, then since $(x_{k-1}, x_{k-1}^T) \xrightarrow{\sigma_k}_{|\hat\Theta_{G\parallel T}} (x_k, x_k^T)$ it follows by definition 11 that $(x_{k-1}, x_{k-1}^T) \xrightarrow{d(x_{k-1},\sigma_k)}_{|\hat\Theta_{H\parallel T}} (x_k, x_k^T)$. If $\sigma_k \in \Sigma_T \setminus \Sigma$, then $d(x_{k-1}, \sigma_k) = \sigma_k$ and $(x_{k-1}, x_{k-1}^T) \xrightarrow{\sigma_k}_{|\hat\Theta_{H\parallel T}} (x_k, x_k^T)$ as

$(x_{k-1}, x_{k-1}^T), (x_k, x_k^T) \in \hat{\Theta}_{G\|T} = \hat{\Theta}_{H\|T}$ since G and H are state-wise synthesis equivalent. Combining these paths for $k = 1, \ldots, n$ gives $H \parallel T \xrightarrow{\varepsilon} (x_0, x_0^T) \xrightarrow[|\hat{\Theta}_{H\|T}]{d(x_0,\sigma_1)} (x_1, x_1^T) \xrightarrow[|\hat{\Theta}_{H\|T}]{d(x_1,\sigma_2)} \cdots \xrightarrow[|\hat{\Theta}_{H\|T}]{d(x_{n-1},\sigma_n)} (x_n, x_n^T)$, and this implies by (18) that $\sup\mathcal{CN}(H \parallel T) \xrightarrow{\mathcal{D}(s)} (x_n, x_n^T) = (x, x^T)$.

Conversely assume that $\sup\mathcal{CN}(H \parallel T) \xrightarrow{\mathcal{D}(s)} (x, x^T)$. Note that $\mathcal{D}(P_\Sigma(s)) = P_\Sigma(\mathcal{D}(s)) \in \mathcal{L}(H) \subseteq \mathcal{L}(G)$, which implies $P_\Sigma(s) \in \mathcal{L}(G)$. By (18), there exists a path $H \parallel T \xrightarrow{\varepsilon} (x_0, x_0^T) \xrightarrow[|\hat{\Theta}_{H\|T}]{d(x_0,\sigma_1)} (x_1, x_1^T) \xrightarrow[|\hat{\Theta}_{H\|T}]{d(x_1,\sigma_2)} \cdots \xrightarrow[|\hat{\Theta}_{H\|T}]{d(x_{n-1},\sigma_n)} (x_n, x_n^T) = (x, x^T)$ such that $\mathcal{D}(s) = d(x_0, \sigma_1) \cdots d(x_{n-1}, \sigma_n)$. Consider $k = 1, \ldots, n$. If $\sigma_k \in \Sigma$, then $x_{k-1} \xrightarrow{\sigma_k}_G x_k$, and since T does not synchronise on the events introduced by d, this implies $(x_{k-1}, x_{k-1}^T) \xrightarrow{\sigma_k}_{G\|T} (x_k, x_k^T)$. Then, given $(x_{k-1}, x_{k-1}^T), (x_k, x_k^T) \in \hat{\Theta}_{H\|T} = \hat{\Theta}_{G\|T}$, it follows that $(x_{k-1}, x_{k-1}^T) \xrightarrow[|\hat{\Theta}_{G\|T}]{\sigma_k} (x_k, x_k^T)$. Otherwise, if $\sigma_k \in \Sigma_T \setminus \Sigma$, then $d(x_{k-1}, \sigma_k) = \sigma_k$ and it follows from $\hat{\Theta}_{H\|T} = \hat{\Theta}_{G\|T}$ that $(x_{k-1}, x_{k-1}^T) \xrightarrow[|\hat{\Theta}_{G\|T}]{\sigma_k} (x_k, x_k^T)$. Combining these transitions for $k = 1, \ldots, n$ gives $G \parallel T \xrightarrow{\varepsilon} (x_0, x_0^T) \xrightarrow[|\hat{\Theta}_{G\|T}]{\sigma_1} \cdots \xrightarrow[|\hat{\Theta}_{G\|T}]{\sigma_n} (x_n, x_n^T)$, i.e., $\sup\mathcal{CN}(G \parallel T) \xrightarrow{s} (x_n, x_n^T) = (x, x^T)$. □

The following proposition confirms that a synthesis-preserving redirection map can be used to construct a synthesis equivalent pair.

Proposition 6 Let $\mathcal{G} = \{G_1, \ldots, G_n\}$ and let $\mathcal{H} = \{H_1, G_2, \ldots, G_n\}$ where $G_i = \langle \Sigma_i, Q_i, \rightarrow_i, Q_i^\circ \rangle$, and let $\mathcal{D}_1 \colon \Sigma_1^* \to \Sigma_1^*$ be a synthesis-preserving redirection map from G_1 to H_1 with respect to $\Upsilon \subseteq \Sigma_1$ such that $\Upsilon \cap \Sigma_2 = \cdots = \Upsilon \cap \Sigma_n = \emptyset$. Then $(\mathcal{G}; \mathcal{D}) \simeq_{\text{synth}} (\mathcal{H}; \mathcal{D}_1 \circ \mathcal{D})$ for every prefix-preserving map $\mathcal{D}$.

Proof. Let $\mathcal{S}_1 = \mathcal{S}_{(\mathcal{G}_1;\mathcal{D})}$ and $\mathcal{S}_2 = \mathcal{S}_{(\mathcal{G}_2;\mathcal{D}_1\circ\mathcal{D})}$, let $T = G_2 \parallel \cdots \parallel G_n$, and let G be an automaton. It is to be shown that $(\mathcal{G}; \mathcal{D}) \simeq_{\text{synth},G} (\mathcal{H}; \mathcal{D}_1 \circ \mathcal{D})$ based on definition 9.

First, let $s \in \mathcal{L}(\mathcal{S}_1/G)$. This means $s \in \mathcal{L}(G)$ and $\mathcal{D}(s) \in \mathcal{L}(\sup\mathcal{CN}(G_1 \| T))$. Since $\mathcal{D}(s) \in \mathcal{L}(\sup\mathcal{CN}(G_1 \parallel T))$, it follows that $G_1 \parallel T \xrightarrow[|\hat{\Theta}_{G_1\|T}]{\mathcal{D}(s)}$. Since $\mathcal{D}_1$ is a synthesis-preserving redirection map, it follows by definition 14 that $H_1 \parallel T \xrightarrow[|\hat{\Theta}_{H_1\|T}]{\mathcal{D}_1(\mathcal{D}(s))}$, which implies $\mathcal{D}_1(\mathcal{D}(s)) \in \mathcal{L}(\sup\mathcal{CN}(H_1 \parallel T))$. Since also $s \in \mathcal{L}(G)$, it follows that $s \in \mathcal{L}(\mathcal{S}_2/G)$.

Conversely, let $s \in \mathcal{L}(\mathcal{S}_2/G)$. This means $\mathcal{D}_1(\mathcal{D}(s)) \in \mathcal{L}(\sup\mathcal{CN}(H_1 \parallel T))$ and $s \in \mathcal{L}(G)$. Since $\mathcal{D}_1(\mathcal{D}(s)) \in \mathcal{L}(\sup\mathcal{CN}(H_1 \| T))$, it follows that $\sup\mathcal{CN}(H_1 \| T) \xrightarrow{\mathcal{D}_1(\mathcal{D}(s))}$. Since $\mathcal{D}_1$ is a synthesis-preserving redirection map, it follows by

definition 14 that $\mathrm{sup}\mathcal{CN}(G_1 \| T) \xrightarrow{\mathcal{D}(s)}$, which implies $\mathcal{D}(s) \in \mathcal{L}(\mathrm{sup}\mathcal{CN}(G_1 \| T))$. Since also $s \in \mathcal{L}(G)$, it follows that $s \in \mathcal{L}(\mathcal{S}_1/G)$. □

After removing some transition from a component $G_i \in \mathcal{G}$, by proposition 5 it is possible to construct a synthesis-preserving redirection map, provided that state-wise and transition-wise synthesis equivalence are satisfied. By proposition 6 this results in a synthesis equivalent pair. The following theorem combines these results and shows that synthesis results can always be preserved when replacing a component by a state-wise and transition-wise synthesis equivalent abstraction resulting from transition removal.

Theorem 7 Let $\mathcal{G} = \{G_1, \ldots, G_n\}$ and $\mathcal{H} = \{H_1, G_2, \ldots, G_n\}$ such that G_1 and H_1 are state-wise and transition-wise synthesis equivalent with respect to $\Upsilon \subseteq \Sigma_1$ such that $\Upsilon \cap \Sigma_2 = \cdots = \Upsilon \cap \Sigma_n = \emptyset$ and $\rightarrow_{H_1} \subseteq \rightarrow_{G_1}$. Then there exists a synthesis-preserving redirection map $\mathcal{D}_1$ from G_1 to H_1 with respect to Υ such that $(\mathcal{G}; \mathcal{D}) \simeq_{\mathrm{synth}} (\mathcal{H}; \mathcal{D}_1 \circ \mathcal{D})$.

Proof. This follows directly from proposition 5 and proposition 6. □

5 Transition Removal Abstraction

According to theorem 7, synthesis results are preserved if transition removal in a component results in a state-wise and transition-wise synthesis equivalent abstraction. This section proposes some concrete methods to construct such abstractions, based on the idea of observation equivalence.

5.1 Observation Equivalence

Observation equivalence or *weak bisimilarity* is a well-known general abstraction method for nondeterministic automata [13]. It can be implemented by simple algorithms, and its application in compositional verification can substantially reduce the state space [6]. The idea of observation equivalence is to identify and merge states with the same future behaviour.

Definition 15 Let $G = \langle \Sigma, Q_G, \rightarrow_G, Q_G^\circ \rangle$ and $H = \langle \Sigma, Q_H, \rightarrow_H, Q_H^\circ \rangle$ be two automata with $\Sigma = \Omega \dot{\cup} \Upsilon$. Then G and H are *observation equivalent* with respect to Υ, written $G \approx H$, if there exists an observation equivalence relation $\approx \subseteq Q_G \times Q_H$, i.e., a relation such that

- if $x_G \approx x_H$ and $x_G \xrightarrow{\sigma}_G y_G$, then there exist $t, u \in \Upsilon^*$ such that $x_H \xrightarrow{tP_\Omega(\sigma)u}_H y_H$;

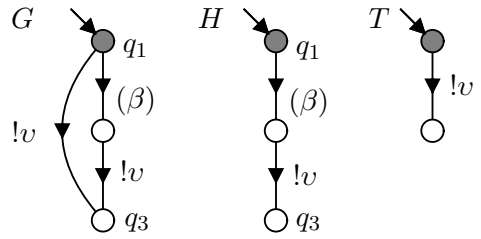

Figure 2: H is observation equivalent to G, but not a synthesis abstraction.

- if $x_G \approx x_H$ and $x_H \xrightarrow{\sigma}_H y_H$, then there exist $t, u \in \Upsilon^*$ such that $x_G \xrightarrow{tP_\Omega(\sigma)u}_G y_G$;
- for each $q^\circ_G \in Q^\circ_G$ there exists $q^\circ_H \in Q^\circ_H$ such that $q^\circ_G \approx q^\circ_H$, and vice versa.

Observation equivalence is tested based on the transitive closure of the local event transitions [2]. The number of transitions can be substantially reduced by considering only the transitive reduction. More precisely, a transition $x \xrightarrow{\sigma} y$ is *observation equivalence redundant* and can be removed [4] if the automaton contains a matching redirection path.

Definition 16 Let $G = \langle \Sigma, Q, \rightarrow_G, Q^\circ \rangle$ and $H = \langle \Sigma, Q, \rightarrow_H, Q^\circ \rangle$ be two automata with $\Sigma = \Omega \dot\cup \Upsilon$ and $\rightarrow_H \subseteq \rightarrow_G$. Automaton H is a result of *observation equivalence redundant transition removal* from G with respect to Υ, if for all transitions $x \xrightarrow{\sigma}_G y$ there exist $t, u \in \Upsilon^*$ such that $x \xrightarrow{tP_\Omega(\sigma)u}_H y$.

Proposition 8 Let $G = \langle \Sigma, Q, \rightarrow_G, Q^\circ \rangle$, and let $H = \langle \Sigma, Q, \rightarrow_H, Q^\circ \rangle$ be a result of observation equivalence redundant transition removal from G with respect to $\Upsilon \subseteq \Sigma$. Then it holds that $G \approx H$.

Observation equivalence redundant transitions can be removed while preserving observation equivalence, which in turn ensures preservation of most temporal logic properties [4, 13]. Unfortunately, this does not include synthesis equivalence [15].

Example 3 Consider automata G, H, and T in figure 2. The uncontrollable transition $q_1 \xrightarrow{!\upsilon} q_3$ is observation equivalence redundant with respect to $\Upsilon = \{\beta\}$. Removing it produces H. In G and H, the uncontrollable event $!\upsilon$ leads to the blocking state q_3. With H, blocking can be prevented by disabling β, leaving only the initial state. But with G, the uncontrollable transition $q_1 \xrightarrow{!\upsilon} q_3$ produces an empty synthesis result. The test T demonstrates that G and H are not state-wise synthesis equivalent since G is not a state-wise synthesis abstraction of H.

This counterexample shows that in general synthesis equivalence is not preserved by removing observation equivalence redundant transitions, so extra restrictions need to be imposed.

5.2 Uncontrollable Redundant Transitions

In example 3, if the local event β was uncontrollable, then the resultant abstraction H would be a transition-wise synthesis abstraction of G. This suggests to interpret an uncontrollable transition as redundant if the local transitions used in the redirection path are also uncontrollable.

Definition 17 Let $G = \langle \Sigma, Q, \rightarrow_G, Q^\circ \rangle$ and $H = \langle \Sigma, Q, \rightarrow_H, Q^\circ \rangle$ be two automata with $\Sigma = \Omega \dot\cup \Upsilon$ and $\rightarrow_H \subseteq \rightarrow_G$. Automaton H is a result of *uncontrollable redundant transition removal* from G with respect to Υ, if the following conditions hold for all transitions $x \xrightarrow{\sigma}_G y$.

(i) If $\sigma \in \Sigma_c$ then $x \xrightarrow{\sigma}_H y$.

(ii) If $\sigma \in \Sigma_u$ then there exist $t, u \in (\Upsilon \cap \Sigma_u)^*$ such that $x \xrightarrow{tP_\Omega(\sigma)u}_H y$.

The transitions present in $\rightarrow_G$ but not in $\rightarrow_H$ in definition 17 are called *uncontrollable redundant* transitions. These transitions can be removed while producing a synthesis equivalent abstraction.

To prove the viability of uncontrollable redundant transition removal, it is shown in the following two lemmas that the method always yields a state-wise and transition-wise synthesis abstraction. Then it follows by theorem 7 that a redirection map can be constructed to give a synthesis equivalent pair.

Lemma 9 Let $H = \langle \Sigma, Q, \rightarrow_H, Q^\circ \rangle$ be a result of uncontrollable redundant transition removal from $G = \langle \Sigma, Q, \rightarrow_G, Q^\circ \rangle$ with respect to $\Upsilon \subseteq \Sigma$. Then G and H are state-wise synthesis equivalent with respect to Υ.

Proof. Let $\Sigma = \Omega \dot\cup \Upsilon$, and let $T = \langle \Sigma_T, Q, \rightarrow, Q^\circ \rangle$ be an automaton such that $\Sigma_T \cap \Upsilon = \emptyset$. It is to be shown that $\hat\Theta_{G\|T} = \hat\Theta_{H\|T}$.

(i) Firstly, to see that $\hat\Theta_{G\|T} \subseteq \hat\Theta_{H\|T}$, it is shown by induction on $n \geq 0$ that $\hat\Theta_{G\|T} \subseteq X_H^n = \Theta_{H\|T}^n(Q \times Q_T)$.

Base case. Clearly $\hat\Theta_{G\|T} \subseteq Q \times Q_T = \Theta_{H\|T}^0(Q \times Q_T) = X_H^0$.

Inductive step. Assume $\hat\Theta_{G\|T} \subseteq X_H^n$ for some $n \geq 0$, and let $(x, x_T) \in \hat\Theta_{G\|T}$. It remains to be shown that $(x, x_T) \in X_H^{n+1} = \Theta_{H\|T}(X_H^n) = \Theta_{H\|T}^{\mathrm{cont}}(X_H^n) \cap \Theta_{H\|T}^{\mathrm{nonb}}(X_H^n)$.

To see that $(x, x_T) \in \Theta^{\mathrm{cont}}_{H\|T}(X^n_H)$, let $\upsilon \in \Sigma_u$ and $(x, x_T) \xrightarrow{\upsilon}_{H\|T} (y, y_T)$. Since $\rightarrow_H \subseteq \rightarrow_G$, it follows that $(x, x_T) \xrightarrow{\upsilon}_{G\|T} (y, y_T)$. Since $(x, x_T) \in \hat{\Theta}_{G\|T}$ and $\upsilon \in \Sigma_u$, it follows by controllability and by inductive assumption that $(y, y_T) \in \hat{\Theta}_{G\|T} \subseteq X^n_H$, and since $\upsilon \in \Sigma_u$ was chosen arbitrarily, it follows that $(x, x_T) \in \Theta^{\mathrm{cont}}_{H\|T}(X^n_H)$.

Next it is shown that $(x, x_T) \in \Theta^{\mathrm{nonb}}_{H\|T}(X^n_H)$. Since $(x, x_T) \in \hat{\Theta}_{G\|T}$, there exists a path

$$(x, x_T) = (x_0, x_0^T) \xrightarrow{\sigma_1}_{|\hat{\Theta}_{G\|T}} \cdots \xrightarrow{\sigma_k}_{|\hat{\Theta}_{G\|T}} (x_k, x_k^T) \xrightarrow{\omega}_{|\hat{\Theta}_{G\|T}} (x_{k+1}, x_{k+1}^T)\,. \tag{19}$$

Consider a transition $(x_{l-1}, x_{l-1}^T) \xrightarrow{\sigma_l}_{|\hat{\Theta}_{G\|T}} (x_l, x_l^T)$ in (19). If $\sigma_l \notin \Sigma$ or $x_{l-1} \xrightarrow{\sigma_l}_H x_l$, then clearly $(x_{l-1}, x_{l-1}^T) \xrightarrow{\sigma_l}_{H\|T} (x_l, x_l^T)$, and by inductive assumption it follows that $(x_{l-1}, x_{l-1}^T), (x_l, x_l^T) \in \hat{\Theta}_{G\|T} \subseteq X^n_H$, i.e., $(x_{l-1}, x_{l-1}^T) \xrightarrow{\sigma_l}_{H\|T\,|X^n_H} (x_l, x_l^T)$. Otherwise $x_{l-1} \xrightarrow{\sigma_l}_G x_l$ is an uncontrollable redundant transition, and by definition 17 there exist traces $t_l, u_l \in (\Sigma_u \cap \Upsilon)^*$ such that $x_{l-1} \xrightarrow{t_l P_\Omega(\sigma_l) u_l}_G x_l$. Since $\Sigma_T \cap \Upsilon = \emptyset$, it follows that $(x_{l-1}, x_{l-1}^T) \xrightarrow{t_l P_\Omega(\sigma_l) u_l}_{G\|T} (x_l, x_l^T)$, and since $(x_{l-1}, x_{l-1}^T) \in \hat{\Theta}_{G\|T}$ and $t_l P_\Omega(\sigma_l) u_l \in \Sigma_u^*$, it follows by controllability that $(x_{l-1}, x_{l-1}^T) \xrightarrow{t_l P_\Omega(\sigma_l) u_l}_{|\hat{\Theta}_{G\|T}} (x_l, x_l^T)$. Then by inductive assumption $(x_{l-1}, x_{l-1}^T) \xrightarrow{t_l P_\Omega(\sigma_l) u_l}_{|X^n_H} (x_l, x_l^T)$. Combining these paths for all transitions in (19) gives traces $t_1, u_1, \ldots, t_k, u_k, t_{k+1} \in (\Sigma_u \cap \Upsilon)^*$ such that

$$\begin{aligned}(x, x_T) = (x_0, x_0^T) &\xrightarrow{t_1 P_\Omega(\sigma_1) u_1}_{H\|T\,|X^n_H} \cdots \xrightarrow{t_k P_\Omega(\sigma_k) u_k}_{H\|T\,|X^n_H} \\ &(x_k, x_k^T) \xrightarrow{t_{k+1}\omega}_{H\|T\,|X^n_H} (x_{k+1}, x_{k+1}^T)\,,\end{aligned}$$

which implies $(x, x_T) \in \Theta^{\mathrm{nonb}}_{H\|T}(X^n_H)$.

It has been shown that $(x, x_T) \in \Theta^{\mathrm{cont}}_{H\|T}(X^n_H) \cap \Theta^{\mathrm{nonb}}_{H\|T}(X^n_H) = X^{n+1}_H$.

(ii) Conversely, to see that $\hat{\Theta}_{H\|T} \subseteq \hat{\Theta}_{G\|T}$, it is shown by induction on $n \geq 0$ that $\hat{\Theta}_{H\|T} \subseteq X^n_G = \Theta^n_{G\|T}(Q \times Q_T)$.

Base case. Clearly $\hat{\Theta}_{H\|T} \subseteq Q \times Q_T = \Theta^0_{G\|T}(Q \times Q_T) = X^0_G$.

Inductive step. Assume $\hat{\Theta}_{H\|T} \subseteq X^n_G$ for some $n \geq 0$, and let $(x, x_T) \in \hat{\Theta}_{H\|T}$. It remains to be shown that $(x, x_T) \in X^{n+1}_G = \Theta_{G\|T}(X^n_G) = \Theta^{\mathrm{cont}}_{G\|T}(X^n_G) \cap \Theta^{\mathrm{nonb}}_{G\|T}(X^n_G)$.

To see that $(x, x_T) \in \Theta^{\text{cont}}_{G\|T}(X^n_G)$, let $\upsilon \in \Sigma_u$ and $(x, x_T) \xrightarrow{\upsilon}_{G\|T} (y, y_T)$. If $\upsilon \notin \Sigma$ or $x \xrightarrow{\upsilon}_H y$, then clearly $(x, x^T) \xrightarrow{\upsilon}_{H\|T} (y, y^T)$, and since $(x, x^T) \in \hat{\Theta}_{H\|T}$ and $\upsilon \in \Sigma_u$, it follows by controllability that $(y, y^T) \in \hat{\Theta}_{H\|T}$. Otherwise $x \xrightarrow{\upsilon}_G y$ is an uncontrollable redundant transition, and by definition 17 there exist $t, u \in (\Sigma_u \cap \Upsilon)^*$ such that $x \xrightarrow{tP_\Omega(\upsilon)u}_H y$. Since $\Sigma_T \cap \Upsilon = \emptyset$, it follows that $(x, x^T) \xrightarrow{tP_\Omega(\upsilon)u}_{H\|T} (y, y^T)$, and since $tP_\Omega(\upsilon)u \in \Sigma^*_u$ and $(x, x^T) \in \hat{\Theta}_{H\|T}$ it follows by controllability that $(y, y^T) \in \hat{\Theta}_{H\|T}$. In both cases by inductive assumption $(y, y^T) \in \hat{\Theta}_{H\|T} \subseteq X^n_G$, and since $\upsilon \in \Sigma_u$ was chosen arbitrarily, it follows that $(x, x^T) \in \Theta^{\text{cont}}_{G\|T}(X^n_G)$.

Next it is shown that $(x, x_T) \in \Theta^{\text{nonb}}_{G\|T}(X^n_G)$. Since $(x, x_T) \in \hat{\Theta}_{H\|T}$, there exists a path $(x, x_T) \xrightarrow{t\omega}_{|\hat{\Theta}_{H\|T}}$. Since $\rightarrow_H \subseteq \rightarrow_G$, it follows by inductive assumption that $(x, x_T) \xrightarrow{t\omega}_{|X^n_G}$. Hence, $(x, x_T) \in \Theta^{\text{nonb}}_{G\|T}(X^n_G)$.

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

Lemma 10 Let $H = \langle \Sigma, Q, \rightarrow_H, Q^\circ \rangle$ be a result of uncontrollable redundant transition removal from $G = \langle \Sigma, Q, \rightarrow_G, Q^\circ \rangle$ with respect to $\Upsilon \subseteq \Sigma$. Then G and H are transition-wise synthesis equivalent with respect to Υ.

Proof. It must be shown that G is a transition-wise synthesis abstraction of H and vice versa. Condition (i) in definition 12 follows immediately from definition 17. To show condition (ii), let $\Sigma = \Omega \dot\cup \Upsilon$, and let $T = \langle \Sigma_T, Q, \rightarrow, Q^\circ \rangle$ be an automaton such that $\Sigma_T \cap \Upsilon = \emptyset$.

First, let $(x, x^T) \xrightarrow{\sigma}_{|\hat{\Theta}_{G\|T}} (y, y^T)$. By lemma 9 it holds that $(x, x^T), (y, y^T) \in \hat{\Theta}_{H\|T}$. If $\sigma \notin \Sigma$ or $x \xrightarrow{\sigma}_H y$, then clearly $(x, x^T) \xrightarrow{\sigma}_{H\|T} (y, y^T)$, which implies $(x, x^T) \xrightarrow{\sigma}_{|\hat{\Theta}_{H\|T}} (y, y^T)$ and $(x, x^T) \xrightarrow{P_\Upsilon(\sigma)P_\Omega(\sigma)}_{|\hat{\Theta}_{H\|T}} (y, y^T)$. Otherwise $x \xrightarrow{\sigma}_G y$ is an uncontrollable redundant transition, and by definition 17 there exist $t, u \in (\Sigma_u \cap \Upsilon)^*$ such that $x \xrightarrow{tP_\Omega(\upsilon)u}_H y$. Since $\Sigma_T \cap \Upsilon = \emptyset$, it follows that $(x, x^T) \xrightarrow{tP_\Omega(\sigma)u}_{H\|T} (y, y^T)$, and since $(x, x^T) \in \hat{\Theta}_{H\|T}$ and $tP_\Omega(\sigma)u \in \Sigma^*_u$, it follows by controllability that $(x, x^T) \xrightarrow{tP_\Omega(\sigma)u}_{|\hat{\Theta}_{H\|T}} (y, y^T)$. Thus, in both cases, there exist $t, u \in \Upsilon^*$ such that $(x, x^T) \xrightarrow{tP_\Omega(\sigma)u}_{|\hat{\Theta}_{H\|T}} (y, y^T)$.

Conversely, let $(x, x^T) \xrightarrow{\sigma}_{|\hat{\Theta}_{H\|T}} (y, y^T)$. Since $\rightarrow_H \subseteq \rightarrow_G$, it follows that $(x, x^T) \xrightarrow{\sigma}_{G\|T} (y, y^T)$. Also $(x, x^T), (y, y^T) \in \hat{\Theta}_{H\|T} = \hat{\Theta}_{G\|T}$ by lemma 9,

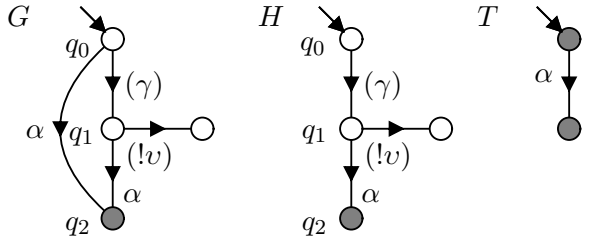

Figure 3: H is observation equivalent to G, but not a synthesis abstraction.

which implies $(x, x^T) \xrightarrow{\sigma}_{|\hat{\Theta}_{G\|T}} (y, y^T)$. Then let $t = \varepsilon$ and $u = P_\Upsilon(\sigma)$, and it follows that $(x, x^T) \xrightarrow{tP_\Omega(\sigma)u}_{|\hat{\Theta}_{G\|T}} (y, y^T)$.

Thus, G and H are transition-wise synthesis equivalent. □

Theorem 11 Let $H = \langle \Sigma, Q, \rightarrow_H, Q^\circ \rangle$ be a result of uncontrollable redundant transition removal from $G = \langle \Sigma, Q, \rightarrow_G, Q^\circ \rangle$ with respect to $\Upsilon \subseteq \Sigma$. Then G and H are state-wise and transition-wise synthesis equivalent with respect to Υ.

Proof. Follows directly from lemma 9 and lemma 10. □

5.3 Controllable Redundant Transitions

For uncontrollable events, an uncontrollable redirection path guarantees transition-wise synthesis equivalence. Unfortunately this idea does not work for controllable events.

Example 4 Consider automaton G in figure 3 where $\Upsilon = \{\gamma, !\upsilon\}$ and $!\upsilon$ is the only uncontrollable event. Transition $q_0 \xrightarrow{\alpha} q_2$ is observation equivalence redundant because of $q_0 \xrightarrow{\gamma\alpha} q_2$. Its removal results in H. In both G and H, the controllable event γ must be disabled in the initial state to prevent blocking via the uncontrollable event $!\upsilon$. However, after disabling γ, termination is no longer possible in H, yet it remains possible in G via $q_0 \xrightarrow{\alpha} q_2$. The test T demonstrates that H is not a state-wise synthesis abstraction of G.

In example 4, the redirection path $q_0 \xrightarrow{\gamma\alpha} q_2$ contains the state q_1, which is unsafe due to its outgoing uncontrollable $!\upsilon$-transition. This suggests to disallow redirection paths with uncontrollable events enabled along them. However, the following example shows that this is not enough.

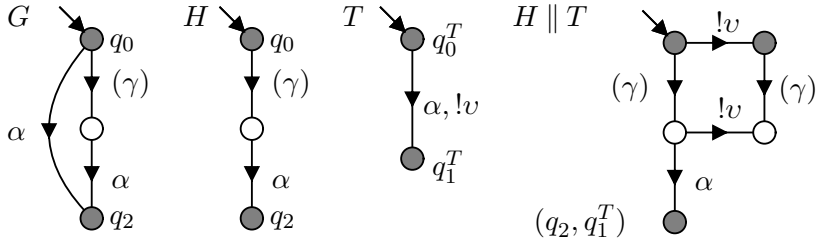

Figure 4: H is observation equivalent to G, but not a synthesis abstraction.

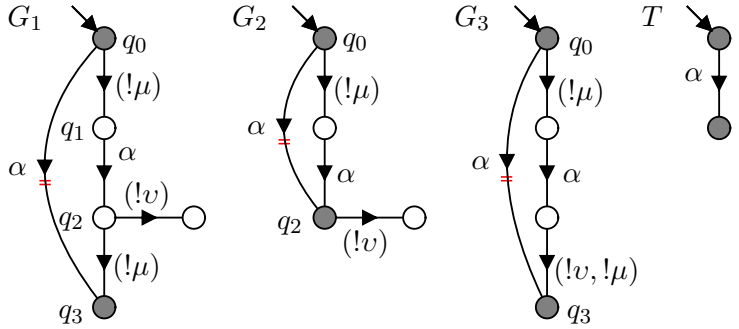

Figure 5: Different redirection paths after the event of a removed transition. The transitions to be removed are marked by double-line strike-through.

Example 5 Consider automata G and T in figure 4 where $\Upsilon = \{\gamma\}$ and $!\upsilon$ is the only uncontrollable event. Transition $q_0 \xrightarrow{\alpha} q_2$ is observation equivalence redundant because of $q_0 \xrightarrow{\gamma\alpha} q_2$, and its removal results in H. In $H \parallel T$, the controllable event γ must be disabled to prevent blocking via the uncontrollable event $!\upsilon$. By disabling γ, state (q_2, q_1^T) becomes unreachable in $\mathrm{sup}\mathcal{CN}(H \parallel T)$, but it remains reachable in $\mathrm{sup}\mathcal{CN}(G \parallel T)$. The test T demonstrates that G and H are not transition-wise synthesis equivalent as G is not a transition-wise synthesis abstraction of H.

The situation in examples 4 and 5 can be avoided by not allowing any controllable events on a redirection path except for the event of the removed transition. However, the following counterexample reveals that one more condition is needed to guarantee a correct abstraction.

Example 6 Consider automaton G_1 in figure 5 where $\Sigma_u = \Upsilon = \{!\mu, !\upsilon\}$. Transition $q_0 \xrightarrow{\alpha} q_3$ is observation equivalence redundant because $q_0 \xrightarrow{!\mu\alpha!\mu} q_3$. Let H_1 be the result of removing the transition $q_0 \xrightarrow{\alpha} q_3$. In both G_1 and H_1, the control-

lable transition $q_1 \xrightarrow{\alpha} q_2$ must be disabled to avert blocking via the uncontrollable event $!\upsilon$. Removing this transition makes q_3 unreachable in $\sup\mathcal{CN}(H \parallel T)$, but it remains reachable in $\sup\mathcal{CN}(G \parallel T)$. The test T demonstrates that G and H are not transition-wise synthesis equivalent as G is not a transition-wise synthesis abstraction of H.

Example 6 shows that there is a problem with uncontrollable local events *after* the event of a removed transition on a redirection path. The problem disappears if there are no further events after the removed event, as in automaton G_2 in figure 5. This leads to the idea of *controllable prefix-redundant* transition removal.

Definition 18 Let $G = \langle \Sigma, Q, \rightarrow_G, Q^\circ \rangle$ and $H = \langle \Sigma, Q, \rightarrow_H, Q^\circ \rangle$ be two automata with $\Sigma = \Omega \dot\cup \Upsilon$ and $\rightarrow_H \subseteq \rightarrow_G$. Automaton H is a result of *controllable prefix-redundant transition removal* from G with respect to Υ, if the following conditions hold for all transitions $x \xrightarrow{\sigma}_G y$.

(i) If $\sigma \in \Sigma_u$ then $x \xrightarrow{\sigma}_H y$.

(ii) If $\sigma \in \Sigma_c$ then there exists $t \in (\Upsilon \cap \Sigma_u)^*$ such that $x \xrightarrow{tP_\Omega(\sigma)}_H y$.

To prove the viability of controllable prefix-redundant transition removal, it is again shown that the method always yields a state-wise and transition-wise synthesis abstraction.

Lemma 12 Let $H = \langle \Sigma, Q, \rightarrow_H, Q^\circ \rangle$ be a result of controllable prefix-redundant transition removal from $G = \langle \Sigma, Q, \rightarrow_G, Q^\circ \rangle$ with respect to $\Upsilon \subseteq \Sigma$. Then G and H are state-wise synthesis equivalent with respect to Υ.

Proof. Let $\Sigma = \Omega \dot\cup \Upsilon$, and let $T = \langle \Sigma_T, Q, \rightarrow, Q^\circ \rangle$ be an automaton such that $\Sigma_T \cap \Upsilon = \emptyset$. It is to be shown that $\hat\Theta_{G\|T} = \hat\Theta_{H\|T}$.

(i) Firstly, to see that $\hat\Theta_{G\|T} \subseteq \hat\Theta_{H\|T}$, it is shown by induction on $n \geq 0$ that $\hat\Theta_{G\|T} \subseteq X_H^n = \Theta_{H\|T}^n(Q \times Q_T)$.

Base case. Clearly $\hat\Theta_{G\|T} \subseteq Q \times Q_T = \Theta_{H\|T}^0(Q \times Q_T) = X_H^0$.

Inductive step. Assume $\hat\Theta_{G\|T} \subseteq X_H^n$ for some $n \geq 0$, and let $(x, x_T) \in \hat\Theta_{G\|T}$. It remains to be shown that $(x, x_T) \in X_H^{n+1} = \Theta_{H\|T}(X_H^n) = \Theta_{H\|T}^{\mathrm{cont}}(X_H^n) \cap \Theta_{H\|T}^{\mathrm{nonb}}(X_H^n)$.

To see that $(x, x_T) \in \Theta_{H\|T}^{\mathrm{cont}}(X_H^n)$, let $\upsilon \in \Sigma_u$ and $(x, x_T) \xrightarrow{\upsilon}_{H\|T} (y, y_T)$. Since $\rightarrow_H \subseteq \rightarrow_G$, it follows that $(x, x_T) \xrightarrow{\upsilon}_{G\|T} (y, y_T)$. Since $(x, x_T) \in$

$\hat{\Theta}_{G\|T}$ and $\upsilon \in \Sigma_u$, it follows by controllability and by inductive assumption that $(y, y_T) \in \hat{\Theta}_{G\|T} \subseteq X_H^n$, and since $\upsilon \in \Sigma_u$ was chosen arbitrarily, it follows that $(x, x_T) \in \Theta_{H\|T}^{\mathrm{cont}}(X_H^n)$.

Next it is shown that $(x, x_T) \in \Theta_{H\|T}^{\mathrm{nonb}}(X_H^n)$. Since $(x, x_T) \in \hat{\Theta}_{G\|T}$, there exists a path

$$(x, x_T) = (x_0, x_0^T) \xrightarrow{\sigma_1}_{|\hat{\Theta}_{G\|T}} \cdots \xrightarrow{\sigma_k}_{|\hat{\Theta}_{G\|T}} (x_k, x_k^T) \xrightarrow{\omega}_{|\hat{\Theta}_{G\|T}} (x_{k+1}, x_{k+1}^T)\,. \tag{20}$$

Consider a transition $(x_{l-1}, x_{l-1}^T) \xrightarrow{\sigma_l}_{|\hat{\Theta}_{G\|T}} (x_l, x_l^T)$ in (20). If $\sigma_l \notin \Sigma$ or $x_{l-1} \xrightarrow{\sigma_l}_H x_l$, then clearly $(x_{l-1}, x_{l-1}^T) \xrightarrow{\sigma_l}_{H\|T} (x_l, x_l^T)$, and by inductive assumption it follows that $(x_{l-1}, x_{l-1}^T), (x_l, x_l^T) \in \hat{\Theta}_{G\|T} \subseteq X_H^n$, i.e., $(x_{l-1}, x_{l-1}^T) \xrightarrow{\sigma_l}_{H\|T\,|X_H^n} (x_l, x_l^T)$. Otherwise $x_{l-1} \xrightarrow{\sigma_l}_G x_l$ is a controllable prefix-redundant transition, and by definition 18 there exists $t_l \in (\Upsilon \cap \Sigma_u)^*$ such that $x_{l-1} \xrightarrow{t_l}_H y_{l-1} \xrightarrow{P_\Omega(\sigma_l)}_H x_l$. Since $\Sigma_T \cap \Upsilon = \emptyset$, it follows that $(x_{l-1}, x_{l-1}^T) \xrightarrow{t_l}_{H\|T} (y_{l-1}, x_{l-1}^T) \xrightarrow{P_\Omega(\sigma_l)}_{H\|T} (x_l, x_l^T)$. Since $(x_{l-1}, x_{l-1}^T) \in \hat{\Theta}_{G\|T}$ and $t_l \in \Sigma_u^*$ and $\rightarrow_H \subseteq \rightarrow_G$, it follows that $(x_{l-1}, x_{l-1}^T) \xrightarrow{t_l}_{|\hat{\Theta}_{G\|T}} (y_{l-1}, x_{l-1}^T)$. Since also $(x_{l-1}, x_{l-1}^T) \in \hat{\Theta}_{G\|T} \subseteq X_H^n$ by inductive assumption, it follows that $(x_{l-1}, x_{l-1}^T) \xrightarrow{t_l}_{H\|T\,|X_H^n} (y_{l-1}, x_{l-1}^T) \xrightarrow{P_\Omega(\sigma_l)}_{H\|T\,|X_H^n} (x_l, x_l^T)$. Combining these paths for all the transitions in (20) gives traces $t_1, \ldots, t_k, t_{k+1} \in (\Sigma_u \cap \Upsilon)^*$ such that

$$\begin{aligned}(x, x_T) = (x_0, x_0^T) &\xrightarrow{t_1 P_\Omega(\sigma_1)}_{H\|T\,|X_H^n} \cdots \xrightarrow{t_k P_\Omega(\sigma_k)}_{H\|T\,|X_H^n} (x_k, x_k^T) \\ &\xrightarrow{t_{k+1}\omega}_{H\|T\,|X_H^n} (x_{k+1}, x_{k+1}^T)\,,\end{aligned} \tag{21}$$

which implies $(x, x_T) \in \Theta_{H\|T}^{\mathrm{nonb}}(X_H^n)$.

It has been shown that $(x, x_T) \in \Theta_{H\|T}^{\mathrm{cont}}(X_H^n) \cap \Theta_{H\|T}^{\mathrm{nonb}}(X_H^n) = X_H^{n+1}$.

(ii) Conversely, to see that $\hat{\Theta}_{H\|T} \subseteq \hat{\Theta}_{G\|T}$, it is shown by induction on $n \geq 0$ that $\hat{\Theta}_{H\|T} \subseteq X_G^n = \Theta_{G\|T}^n(Q \times Q_T)$.

Base case. Clearly $\hat{\Theta}_{H\|T} \subseteq Q \times Q_T = \Theta_{G\|T}^0(Q \times Q_T) = X_G^0$.

Inductive step. Assume $\hat{\Theta}_{H\|T} \subseteq X_G^n$ for some $n \geq 0$, and let $(x, x_T) \in \hat{\Theta}_{H\|T}$. It remains to be shown that $(x, x_T) \in X_G^{n+1} = \Theta_{G\|T}(X_G^n) = \Theta_{G\|T}^{\mathrm{cont}}(X_G^n) \cap \Theta_{G\|T}^{\mathrm{nonb}}(X_G^n)$.

To see that $(x, x_T) \in \Theta^{\mathrm{cont}}_{G\|T}(X^n_G)$, let $\upsilon \in \Sigma_u$ and $(x, x_T) \xrightarrow{\upsilon}_{G\|T} (y, y_T)$. If $\upsilon \notin \Sigma$, then clearly $(x, x_T) \xrightarrow{\upsilon}_{H\|T} (y, y_T)$. Otherwise, since $\upsilon \in \Sigma_u$, the transition $x \xrightarrow{\upsilon}_G y$ cannot be controllable prefix-redundant, which also implies $(x, x_T) \xrightarrow{\upsilon}_{H\|T} (y, y_T)$. Since $(x, x_T) \in \hat{\Theta}_{H\|T}$ and $\upsilon \in \Sigma_u$, it follows that $(y, y_T) \in \hat{\Theta}_{H\|T} \subseteq X^n_G$ by inductive assumption, and thus $(x, x_T) \in \Theta^{\mathrm{cont}}_{G\|T}(X^n_G)$.

Next it is shown that $(x, x_T) \in \Theta^{\mathrm{nonb}}_{G\|T}(X^n_G)$. Since $(x, x_T) \in \hat{\Theta}_{H\|T}$, there exists a path $(x, x_T) \xrightarrow{t\omega}_{|\hat{\Theta}_{H\|T}}$. Since $\rightarrow_H \subseteq \rightarrow_G$, it follows by inductive assumption that $(x, x_T) \xrightarrow{t\omega}_{|X^n_G}$. Hence, $(x, x_T) \in \Theta^{\mathrm{nonb}}_{G\|T}(X^n_G)$.

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

Lemma 13 Let $H = \langle \Sigma, Q, \rightarrow_H, Q^\circ \rangle$ be a result of controllable prefix-redundant transition removal from $G = \langle \Sigma, Q, \rightarrow_G, Q^\circ \rangle$ with respect to $\Upsilon \subseteq \Sigma$. Then G and H are transition-wise synthesis equivalent with respect to Υ.

Proof. It must be shown that G is a transition-wise synthesis abstraction of H and vice versa. Condition (i) in definition 12 follows immediately from definition 18. To show condition (ii), let $\Sigma = \Omega \dot{\cup} \Upsilon$, and let $T = \langle \Sigma_T, Q, \rightarrow, Q^\circ \rangle$ be an automaton such that $\Sigma_T \cap \Upsilon = \emptyset$.

First, let $(x, x^T) \xrightarrow{\sigma}_{|\hat{\Theta}_{G\|T}} (y, y^T)$. By lemma 12 it holds that $(x, x^T), (y, y^T) \in \hat{\Theta}_{H\|T}$. If $\sigma \notin \Sigma$ or $x \xrightarrow{\sigma}_H y$, then clearly $(x, x^T) \xrightarrow{\sigma}_{H\|T} (y, y^T)$, which implies $(x, x^T) \xrightarrow{\sigma}_{|\hat{\Theta}_{H\|T}} (y, y^T)$ and $(x, x^T) \xrightarrow{P_\Upsilon(\sigma)P_\Omega(\sigma)}_{|\hat{\Theta}_{H\|T}} (y, y^T)$. Otherwise $x \xrightarrow{\sigma}_G y$ is a controllable prefix-redundant transition, and by definition 18 there exists $t \in (\Sigma_u \cap \Upsilon)^*$ such that $x \xrightarrow{t}_H z \xrightarrow{P_\Omega(\upsilon)}_H y$. Since $\Sigma_T \cap \Upsilon = \emptyset$, it follows that $(x, x^T) \xrightarrow{t}_{H\|T} (z, x^T) \xrightarrow{P_\Omega(\sigma)}_{H\|T} (y, y^T)$, and since $(x, x^T) \in \hat{\Theta}_{H\|T}$ and $t \in \Sigma^*_u$, it follows by controllability that $(x, x^T) \xrightarrow{tP_\Omega(\sigma)}_{|\hat{\Theta}_{H\|T}} (y, y^T)$. Then let $u = \varepsilon$, and in both cases there exist $t, u \in \Upsilon^*$ such that $(x, x^T) \xrightarrow{tP_\Omega(\sigma)u}_{|\hat{\Theta}_{H\|T}} (y, y^T)$.

Conversely, let $(x, x^T) \xrightarrow{\sigma}_{|\hat{\Theta}_{H\|T}} (y, y^T)$. Since $\rightarrow_H \subseteq \rightarrow_G$, it follows that $(x, x^T) \xrightarrow{\sigma}_{G\|T} (y, y^T)$. Also $(x, x^T), (y, y^T) \in \hat{\Theta}_{H\|T} = \hat{\Theta}_{G\|T}$ by lemma 12, which implies $(x, x^T) \xrightarrow{\sigma}_{|\hat{\Theta}_{G\|T}} (y, y^T)$. Then let $t = \varepsilon$ and $u = P_\Upsilon(\sigma)$, and it follows that $(x, x^T) \xrightarrow{tP_\Omega(\sigma)u}_{|\hat{\Theta}_{G\|T}} (y, y^T)$.

Thus, G and H are transition-wise synthesis equivalent. □

Theorem 14 Let $H = \langle \Sigma, Q, \rightarrow_H, Q^\circ \rangle$ be a result of controllable prefix-redundant transition removal from $G = \langle \Sigma, Q, \rightarrow_G, Q^\circ \rangle$ with respect to $\Upsilon \subseteq \Sigma$. Then G and H are state-wise and transition-wise synthesis equivalent with respect to Υ.

Proof. Follows directly from lemma 12 and lemma 13. □

Controllable prefix-redundant transition removal only allows for local events *before* the event of a removed transition. Local events after this event can also be considered by adding additional requirements.

Example 7 As shown in example 6, removal of the transition $q_0 \xrightarrow{\alpha} q_3$ in G_1 in figure 5 does not ensure synthesis abstraction because of the uncontrollable $!\upsilon$-transition in state q_2. Automaton G_3 also has the observation equivalence redundant transition $q_0 \xrightarrow{\alpha} q_3$ and an $!\upsilon$-transition enabled after α on the redirection path $q_0 \xrightarrow{!\mu\alpha!\mu} q_3$. Yet, in this case, the $!\upsilon$-transition does not lead to a blocking state, and the removal of $q_0 \xrightarrow{\alpha} q_3$ results in a state-wise and transition-wise synthesis equivalent automaton.

Automata G_1 and G_3 in figure 5 differ in the target state of $q_2 \xrightarrow{!\upsilon}$. This suggests to allow uncontrollable events in the second part of a redirection provided that they are local and lead to a target state on the redirection path.

Definition 19 Let $G = \langle \Sigma, Q, \rightarrow, Q^\circ \rangle$ be an automaton and $\Upsilon \subseteq \Sigma$. A path

$$x_0 \xrightarrow{\sigma_1} x_1 \xrightarrow{\sigma_2} \cdots \xrightarrow{\sigma_k} x_k \tag{22}$$

is a *weakly controllable Υ-path* if $\sigma_1, \ldots, \sigma_k \in \Upsilon$ and for all uncontrollable transitions $x_l \xrightarrow{\upsilon} y$ with $0 \leq l < k$ and $\upsilon \in \Sigma_u$ it holds that $\upsilon \in \Upsilon$ and $y = x_j$ for some $0 \leq j \leq k$.

A weakly controllable path consists of only local transitions, and furthermore all uncontrollable transitions enabled along this path must use local events and lead to states along the path. Imposing this condition on the redirection path gives the condition for a *controllable suffix-redundant transition*, which is sufficient for synthesis equivalence.

Definition 20 Let $G = \langle \Sigma, Q, \rightarrow_G, Q^\circ \rangle$ and $H = \langle \Sigma, Q, \rightarrow_H, Q^\circ \rangle$ be two automata with $\Sigma = \Omega \dot\cup \Upsilon$ and $\rightarrow_H \subseteq \rightarrow_G$. Automaton H is a result of *controllable suffix-redundant transition removal* from G with respect to Υ, if the following conditions hold for all transitions $x \xrightarrow{\sigma}_G y$.

(i) If $\sigma \in \Sigma_u$ then $x \xrightarrow{\sigma}_H y$.

(ii) If $\sigma \in \Sigma_c$ then there exists $u \in \Upsilon^*$ such that $x \xrightarrow{P_\Omega(\sigma)}_H z \xrightarrow{u}_H y$, and $z \xrightarrow{u}_G y$ is a weakly controllable Υ-path.

In controllable prefix-redundant transition removal, there may be uncontrollable events in all states along the redirection path, but there may be no local events after the event of the removed transition. In suffix-redundant transition removal, all uncontrollable events enabled along the redirection path must be local and lead to a state along the redirection path.

It is again shown that controllable suffix-redundant transition removal always yields a state-wise and transition-wise synthesis abstraction. Before that, lemma 15 establishes a key property of weakly controllable Υ-paths.

Lemma 15 Let $G = \langle \Sigma, Q_G, \rightarrow_G, Q_G^\circ \rangle$ and $T = \langle \Sigma_T, Q_T, \rightarrow_T, Q_T^\circ \rangle$ be automata, and let $\Upsilon \subseteq \Sigma \setminus \Sigma_T$. Furthermore, let $x \xrightarrow{s}_G y$ be a weakly controllable Υ-path. Then for all $x^T \in Q_T$ such that $(y, x^T) \in \hat{\Theta}_{G\|T}$ it holds that $(x, x^T) \xrightarrow{s}_{|\hat{\Theta}_{G\|T}} (y, x^T)$.

Proof. Let $s = \sigma_1 \cdots \sigma_k$. As $s \in \Upsilon^*$ and $\Sigma_T \cap \Upsilon = \emptyset$, there exist states $x_0, \ldots, x_k \in Q$ such that

$$(x, x^T) = (x_0, x^T) \xrightarrow{\sigma_1}_{G\|T} (x_1, x^T) \xrightarrow{\sigma_2}_{G\|T} \cdots \xrightarrow{\sigma_k}_{G\|T} (x_k, x^T) = (y, x^T) \,. \quad (23)$$

It remains to be shown that this path is in $\hat{\Theta}_{G\|T}$. Let $\Upsilon_u^T = \Sigma_u \cap (\Sigma_T \setminus \Sigma)$ and

$$Y_T = \{\, y^T \in Q_T \mid x^T \xrightarrow{u}_T y^T \text{ for some } u \in (\Upsilon_u^T)^* \,\} \,. \quad (24)$$

It is shown by induction on $n \geq 0$ that for all $0 \leq j \leq k$ and for all $y^T \in Y_T$ it holds that $(x_j, y^T) \in X^n = \Theta^n_{G\|T}(Q \times Q_T)$. As $x^T \in Y_T$, this will imply $(x, x^T) \xrightarrow{s}_{|\hat{\Theta}_{G\|T}} (y, x^T)$.

Base case. $n = 0$. Clearly $(x_j, y^T) \in Q \times Q_T = \Theta^0_{G\|T}(Q \times Q_T) = X^0$.

Inductive step. Let $0 \leq j \leq k$ and $y^T \in Y_T$. It must be shown that $(x_j, y^T) \in X^{n+1} = \Theta_{G\|T}(X^n) = \Theta^{\text{cont}}_{G\|T}(X^n) \cap \Theta^{\text{nonb}}_{G\|T}(X^n)$.

To see that $(x_j, y^T) \in \Theta^{\text{cont}}_{G\|T}(X^n)$, let $\upsilon \in \Sigma_u$ and $(x_j, y^T) \xrightarrow{\upsilon}_{G\|T} (z, z^T)$. If $\upsilon \in \Sigma$, then since $x_0 \xrightarrow{s}_G x_k$ is a weakly controllable Υ-path, it must hold that $\upsilon \in \Upsilon$ and $x_j \xrightarrow{\upsilon}_G z = x_l$ for some $0 \leq l \leq k$. This implies $y^T = z^T$ and $(x_j, y^T) \xrightarrow{\upsilon}_{G\|T} (z, z^T) = (x_l, y^T) \in X^n$ by inductive assumption. If $\upsilon \notin \Sigma$, then $\upsilon \in \Sigma_T \setminus \Sigma$ and $z = x_j$ and $y_T \xrightarrow{\upsilon}_T z^T$. Then clearly $z^T \in Y_T$ and $(z, z^T) = (x_j, z^T) \in X^n$ by inductive assumption. As this can be shown for all $\upsilon \in \Sigma_u$, it follows that $(x_j, y^T) \in \Theta^{\text{cont}}_{G\|T}(X^n)$.

Next, it is shown that $(x_j, y^T) \in \Theta^{\mathrm{nonb}}_{G\|T}(X^n)$. As $\sigma_{j+1}, \ldots, \sigma_k \in \Upsilon$ and $\Sigma_T \cap \Upsilon = \emptyset$, it holds by inductive assumption that,

$$(x_j, y^T) \xrightarrow{\sigma_{j+1}}_{|X^n} \cdots \xrightarrow{\sigma_k}_{|X^n} (x_k, y^T) = (y, y^T)\,. \tag{25}$$

Since $y^T \in Y_T$, there exists $u \in (\Upsilon_u^T)^*$ such that $x^T \xrightarrow{u}_T y^T$, and this implies $(y, x^T) \xrightarrow{u}_{G\|T} (y, y_T)$. Since $(y, x^T) \in \hat{\Theta}_{G\|T}$ by assumption and $u \in \Sigma_u^*$, it follows that $(y, y^T) \in \hat{\Theta}_{G\|T}$. Then there exists $t \in \Sigma^*$ such that $(y, y^T) \xrightarrow{t\omega}_{|\hat{\Theta}_{G\|T}}$, and as $\hat{\Theta}_{G\|T} \subseteq X^n$ it follows that

$$(x_j, y^T) \xrightarrow{\sigma_{j+1}}_{|X^n} \cdots \xrightarrow{\sigma_k}_{|X^n} (x_k, y^T) = (y, y^T) \xrightarrow{t\omega}_{|X^n}\,. \tag{26}$$

This implies $(x_j, y^T) \in \Theta^{\mathrm{nonb}}_{G\|T}(X^n)$. □

Lemma 16 Let $H = \langle \Sigma, Q, \rightarrow_H, Q^\circ \rangle$ be a result of controllable suffix-redundant transition removal from $G = \langle \Sigma, Q, \rightarrow_G, Q^\circ \rangle$ with respect to $\Upsilon \subseteq \Sigma$. Then G and H are state-wise synthesis equivalent with respect to Υ.

Proof. Let $\Sigma = \Omega \,\dot\cup\, \Upsilon$, and let $T = \langle \Sigma_T, Q, \rightarrow, Q^\circ \rangle$ be an automaton such that $\Sigma_T \cap \Upsilon = \emptyset$. It is to be shown that $\hat{\Theta}_{G\|T} = \hat{\Theta}_{H\|T}$.

(i) Firstly, to see that $\hat{\Theta}_{G\|T} \subseteq \hat{\Theta}_{H\|T}$, it is shown by induction on $n \geq 0$ that $\hat{\Theta}_{G\|T} \subseteq X_H^n = \Theta^n_{H\|T}(Q \times Q_T)$.

Base case. Clearly $\hat{\Theta}_{G\|T} \subseteq Q \times Q_T = \Theta^0_{H\|T}(Q \times Q_T) = X_H^0$.

Inductive step. Assume $\hat{\Theta}_{G\|T} \subseteq X_H^n$ for some $n \geq 0$, and let $(x, x_T) \in \hat{\Theta}_{G\|T}$. It remains to be shown that $(x, x_T) \in X_H^{n+1} = \Theta_{H\|T}(X_H^n) = \Theta^{\mathrm{cont}}_{H\|T}(X_H^n) \cap \Theta^{\mathrm{nonb}}_{H\|T}(X_H^n)$.

To see that $(x, x_T) \in \Theta^{\mathrm{cont}}_{H\|T}(X_H^n)$, let $\upsilon \in \Sigma_u$ and $(x, x_T) \xrightarrow{\upsilon}_{H\|T} (y, y_T)$. Since $\rightarrow_H \subseteq \rightarrow_G$, it follows that $(x, x_T) \xrightarrow{\upsilon}_{G\|T} (y, y_T)$. Since $(x, x_T) \in \hat{\Theta}_{G\|T}$ and $\upsilon \in \Sigma_u$, it follows by controllability and by inductive assumption that $(y, y_T) \in \hat{\Theta}_{G\|T} \subseteq X_H^n$, and since $\upsilon \in \Sigma_u$ was chosen arbitrarily, it follows that $(x, x_T) \in \Theta^{\mathrm{cont}}_{H\|T}(X_H^n)$.

Next, it is shown that $(x, x_T) \in \Theta^{\mathrm{nonb}}_{H\|T}(X_H^n)$. Since $(x, x_T) \in \hat{\Theta}_{G\|T}$, there exists a path

$$(x, x_T) = (x_0, x_0^T) \xrightarrow{\sigma_1}_{|\hat{\Theta}_{G\|T}} \cdots \xrightarrow{\sigma_k}_{|\hat{\Theta}_{G\|T}} (x_k, x_k^T) \xrightarrow{\omega}_{|\hat{\Theta}_{G\|T}} (x_{k+1}, x_{k+1}^T)\,. \tag{27}$$

Consider a transition $(x_{l-1}, x_{l-1}^T) \xrightarrow{\sigma_l}_{|\hat{\Theta}_{G\|T}} (x_l, x_l^T)$ in (27). If $\sigma_l \notin \Sigma$ or $x_{l-1} \xrightarrow{\sigma_l}_H x_l$, then clearly $(x_{l-1}, x_{l-1}^T) \xrightarrow{\sigma_l}_{H\|T} (x_l, x_l^T)$, and by inductive assumption $(x_l, x_l^T) \in \hat{\Theta}_{G\|T} \subseteq X_H^n$, i.e., $(x_{l-1}, x_{l-1}^T) \xrightarrow{\sigma_l}_{|X_H^n} (x_l, x_l^T)$ and $(x_{l-1}, x_{l-1}^T) \xrightarrow{P_\Omega(\sigma_l)P_\Upsilon(\sigma_l)}_{|X_H^n} (x_l, x_l^T)$. Otherwise $x_{l-1} \xrightarrow{\sigma_l}_G x_l$ is a controllable suffix-redundant transition, and by definition 20 there exists $u \in \Upsilon^*$ such that $x_{l-1} \xrightarrow{P_\Omega(\sigma_l)}_H z_l \xrightarrow{u}_H x_l$ where $z_l \xrightarrow{u}_G x_l$ is a weakly controllable Υ-path. Since $\Sigma_T \cap \Upsilon = \emptyset$, it follows that $(x_{l-1}, x_{l-1}^T) \xrightarrow{P_\Omega(\sigma_l)}_{H\|T} (z_l, x_l^T) \xrightarrow{u}_{H\|T} (x_l, x_l^T)$. Since $(x_l, x_l^T) \in \hat{\Theta}_{G\|T}$ it follows by lemma 15 that $(z_l, x_l^T) \xrightarrow{u}_{|\hat{\Theta}_{G\|T}} (x_l, x_l^T)$. Since also $(x_{l-1}, x_{l-1}^T) \in \hat{\Theta}_{G\|T}$ it follows that $(x_{l-1}, x_{l-1}^T) \xrightarrow{P_\Omega(\sigma_l)u}_{|\hat{\Theta}_{G\|T}} (x_l, x_l^T)$, and thus $(x_{l-1}, x_{l-1}^T) \xrightarrow{P_\Omega(\sigma_l)u}_{|X_H^n} (x_l, x_l^T)$ by inductive assumption. Combining these paths for all transitions in (27) gives traces $u_1, \ldots, u_k \in \Upsilon^*$ such that

$$(x, x_T) = (x_0, x_0^T) \xrightarrow{P_\Omega(\sigma_1)u_1}_{|X_H^n} \cdots \xrightarrow{P_\Omega(\sigma_k)u_k}_{|X_H^n} (x_k, x_k^T) \xrightarrow{\omega}_{|X_H^n} (x_{k+1}, x_{k+1}^T), \tag{28}$$

which implies $(x, x_T) \in \Theta^{\mathrm{nonb}}_{H\|T}(X_H^n)$.

It has been shown that $(x, x_T) \in \Theta^{\mathrm{cont}}_{H\|T}(X_H^n) \cap \Theta^{\mathrm{nonb}}_{H\|T}(X_H^n) = X_H^{n+1}$.

(ii) Conversely, to see that $\hat{\Theta}_{H\|T} \subseteq \hat{\Theta}_{G\|T}$, it is shown by induction on $n \geq 0$ that $\hat{\Theta}_{H\|T} \subseteq X_G^n = \Theta^n_{G\|T}(Q \times Q_T)$.

Base case. Clearly $\hat{\Theta}_{H\|T} \subseteq Q \times Q_T = \Theta^0_{G\|T}(Q \times Q_T) = X_G^0$.

Inductive step. Assume $\hat{\Theta}_{H\|T} \subseteq X_G^n$ for some $n \geq 0$, and let $(x, x_T) \in \hat{\Theta}_{H\|T}$. It remains to be shown that $(x, x_T) \in X_G^{n+1} = \Theta_{G\|T}(X_G^n) = \Theta^{\mathrm{cont}}_{G\|T}(X_G^n) \cap \Theta^{\mathrm{nonb}}_{G\|T}(X_G^n)$.

To see that $(x, x_T) \in \Theta^{\mathrm{cont}}_{G\|T}(X_G^n)$, let $\upsilon \in \Sigma_u$ and $(x, x_T) \xrightarrow{\upsilon}_{G\|T} (y, y_T)$. If $\upsilon \notin \Sigma$, then clearly $(x, x_T) \xrightarrow{\upsilon}_{H\|T} (y, y_T)$. Otherwise, since $\upsilon \in \Sigma_u$, the transition $x \xrightarrow{\upsilon}_G y$ cannot be controllable suffix-redundant, which also implies $(x, x_T) \xrightarrow{\upsilon}_{H\|T} (y, y_T)$. Since $(x, x_T) \in \hat{\Theta}_{H\|T}$ and $\upsilon \in \Sigma_u$, it follows that $(y, y_T) \in \hat{\Theta}_{H\|T} \subseteq X_G^n$ by inductive assumption, and thus $(x, x_T) \in \Theta^{\mathrm{cont}}_{G\|T}(X_G^n)$.

Next it is shown that $(x, x_T) \in \Theta^{\mathrm{nonb}}_{G\|T}(X_G^n)$. Since $(x, x_T) \in \hat{\Theta}_{H\|T}$, there exists a path $(x, x_T) \xrightarrow{t\omega}_{|\hat{\Theta}_{H\|T}}$. Since $\rightarrow_H \subseteq \rightarrow_G$, it follows by inductive assumption that $(x, x_T) \xrightarrow{t\omega}_{|X_G^n}$. Hence, $(x, x_T) \in \Theta^{\mathrm{nonb}}_{G\|T}(X_G^n)$.

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

Lemma 17 Let $H = \langle \Sigma, Q, \rightarrow_H, Q^\circ \rangle$ be a result of controllable suffix-redundant transition removal from $G = \langle \Sigma, Q, \rightarrow_G, Q^\circ \rangle$ with respect to $\Upsilon \subseteq \Sigma$. Then G and H are transition-wise synthesis equivalent with respect to Υ.

Proof. It must be shown that G is a transition-wise synthesis abstraction of H and vice versa. Condition (i) in definition 12 follows immediately from definition 20. To show condition (ii), let $\Sigma = \Omega \dot\cup \Upsilon$, and let $T = \langle \Sigma_T, Q, \rightarrow, Q^\circ \rangle$ be an automaton such that $\Sigma_T \cap \Upsilon = \emptyset$.

First, let $(x, x^T) \xrightarrow{\sigma}_{|\hat\Theta_{G\|T}} (y, y^T)$. By lemma 16 it holds that $(x, x^T), (y, y^T) \in \hat\Theta_{H\|T}$. If $\sigma \notin \Sigma$ or $x \xrightarrow{\sigma}_H y$, then clearly $(x, x^T) \xrightarrow{\sigma}_{H\|T} (y, y^T)$, which implies $(x, x^T) \xrightarrow{\sigma}_{|\hat\Theta_{H\|T}} (y, y^T)$ and $(x, x^T) \xrightarrow{P_\Omega(\sigma)P_\Upsilon(\sigma)}_{|\hat\Theta_{H\|T}} (y, y^T)$. Otherwise $x \xrightarrow{\sigma}_G y$ is a controllable suffix-redundant transition, and by definition 20, there exists $u \in \Upsilon^*$ such that $x \xrightarrow{P_\Omega(\sigma)}_H z \xrightarrow{u}_H y$ where $z \xrightarrow{u}_G y$ is a weakly controllable Υ-path. Since $\Sigma_T \cap \Upsilon = \emptyset$, it follows that $(x, x^T) \xrightarrow{P_\Omega(\sigma)}_{H\|T} (z, y^T) \xrightarrow{u}_{H\|T} (y, y^T)$. Since $(y, y^T) \in \hat\Theta_{G\|T}$ it follows by lemma 15 that $(z, y^T) \xrightarrow{u}_{|\hat\Theta_{G\|T}} (y, y^T)$. Since also $(x, x^T) \in \hat\Theta_{G\|T}$ it follows that $(x, x^T) \xrightarrow{P_\Omega(\sigma)u}_{|\hat\Theta_{G\|T}} (y, y^T)$ and thus $(x, x^T) \xrightarrow{P_\Omega(\sigma)u}_{|\hat\Theta_{H\|T}} (y, y^T)$. Thus, in both cases, there exist $t = \varepsilon$ and $u \in \Upsilon^*$ such that $(x, x^T) \xrightarrow{tP_\Omega(\sigma)u}_{|\hat\Theta_{H\|T}} (y, y^T)$.

Conversely, let $(x, x^T) \xrightarrow{\sigma}_{|\hat\Theta_{H\|T}} (y, y^T)$. Since $\rightarrow_H \subseteq \rightarrow_G$, it follows that $(x, x^T) \xrightarrow{\sigma}_{G\|T} (y, y^T)$. Also $(x, x^T), (y, y^T) \in \hat\Theta_{H\|T} = \hat\Theta_{G\|T}$ by lemma 16, which implies $(x, x^T) \xrightarrow{\sigma}_{\hat\Theta_{G\|T}} (y, y^T)$. Then let $t = \varepsilon$ and $u = P_\Upsilon(\sigma)$, and it follows that $(x, x^T) \xrightarrow{tP_\Omega(\sigma)u}_{|\hat\Theta_{G\|T}} (y, y^T)$.

Thus, G and H are transition-wise synthesis equivalent. $\square$

Theorem 18 Let $H = \langle \Sigma, Q, \rightarrow_H, Q^\circ \rangle$ be a result of controllable suffix-redundant transition removal from $G = \langle \Sigma, Q, \rightarrow_G, Q^\circ \rangle$ with respect to $\Upsilon \subseteq \Sigma$. Then G and H are state-wise and transition-wise synthesis equivalent with respect to Υ.

Proof. Follows directly from lemma 16 and lemma 17. $\square$

Both controllable prefix-redundant and controllable suffix-redundant transition removal preserve synthesis equivalence. These conditions can be combined to allow sequences of local events before *and* after a removed transition.

Definition 21 Let $G = \langle \Sigma, Q, \rightarrow_G, Q^\circ \rangle$ and $H = \langle \Sigma, Q, \rightarrow_H, Q^\circ \rangle$ be two automata with $\Sigma = \Omega \dot\cup \Upsilon$ and $\rightarrow_H \subseteq \rightarrow_G$. Automaton H is a result of *controllable redundant transition removal* from G with respect to Υ, if the following conditions hold for all transitions $x \xrightarrow{\sigma}_G y$.

(i) If $\sigma \in \Sigma_u$ then $x \xrightarrow{\sigma}_H y$.

(ii) If $\sigma \in \Sigma_c$ then there exist $t \in (\Upsilon \cap \Sigma_u)^*$ and $u \in \Upsilon^*$ such that $x \xrightarrow{tP_\Omega(\sigma)}_H z \xrightarrow{u}_H y$, and $z \xrightarrow{u}_G y$ is a weakly controllable Υ-path.

Theorem 19 Let $H = \langle \Sigma, Q, \rightarrow_H, Q^\circ \rangle$ be a result of controllable redundant transition removal from $G = \langle \Sigma, Q, \rightarrow_G, Q^\circ \rangle$ with respect to $\Upsilon \subseteq \Sigma$. Then G and H are state-wise and transition-wise synthesis equivalent with respect to Υ.

Proof. It is enough to show that the removal of a single controllable redundant transition results in a state-wise and transition-wise synthesis equivalent automaton. The rest of the claim follows by induction. Therefore let $\rightarrow_G = \rightarrow_H \dot\cup (x, \sigma, y)$ where $x \xrightarrow{\sigma}_G y$ is a controllable redundant transition.

As $x \xrightarrow{\sigma}_G y$ is a controllable redundant transition, there exists a redirection path $x \xrightarrow{t}_H z_1 \xrightarrow{P_\Omega(\sigma)}_H z_2 \xrightarrow{t}_H y$ where $t \in (\Sigma_u \cap \Upsilon)^*$ and $z_2 \xrightarrow{t}_G y$ is a weakly controllable Υ-path. Consider automata $G' = \langle \Sigma, Q, \rightarrow_{G'}, Q^\circ \rangle$ with $\rightarrow_{G'} = \rightarrow_G \cup (x, \sigma, z_2)$ and $H' = \langle \Sigma, Q, \rightarrow_{H'}, Q^\circ \rangle$ with $\rightarrow_{H'} = \rightarrow_H \cup (x, \sigma, z_2)$. Since $x \xrightarrow{t}_G z_1 \xrightarrow{P_\Omega(\sigma)}_G z_2$, the transition $x \xrightarrow{\sigma} z_2$ is controllable prefix-redundant in G'. Therefore, G is a result of controllable prefix-redundant transition removal from G', and likewise H is a result of controllable prefix-redundant transition removal from H'. Furthermore, as $x \xrightarrow{\sigma}_{G'} z_2 \xrightarrow{t}_{G'} y$, it holds that $x \xrightarrow{\sigma}_{G'} y$ is a controllable suffix-redundant transition, and H' is a result of controllable suffix-redundant transition removal from G'. Then the claim follows from theorem 14 and theorem 18. □

5.4 Local Selfloop Removal

Selfloop removal [14] is a synthesis-preserving abstraction that removes events from a system as soon as they only appear in selfloops in *all* components. Transition-wise synthesis equivalence leads to a modified version of this abstraction, which allows the removal of *local* selfloops, i.e., the removal of transitions $x \xrightarrow{\sigma} x$ where $\sigma \in \Upsilon$ is a local event.

Definition 22 Let $G = \langle \Sigma, Q, \rightarrow_G, Q^\circ \rangle$ and $H = \langle \Sigma, Q, \rightarrow_H, Q^\circ \rangle$ be two automata with $\Sigma = \Omega \dot\cup \Upsilon$ and $\rightarrow_H \subseteq \rightarrow_G$. Automaton H is a result of *local selfloop*

removal from G with respect to Υ, if for all transitions $x \xrightarrow{\sigma}_G y$ such that $\sigma \in \Omega$ or $x \neq y$ it holds that $x \xrightarrow{\sigma}_H y$.

Local selfloop removal can be considered as a special case of controllable or uncontrollable redundant transition removal, by considering empty sequences of local events in the redirection path.

Theorem 20 Let $H = \langle \Sigma, Q, \rightarrow_H, Q^\circ \rangle$ be a result of local selfloop removal from $G = \langle \Sigma, Q, \rightarrow_G, Q^\circ \rangle$ with respect to $\Upsilon \subseteq \Sigma$. Then G and H are state-wise and transition-wise synthesis equivalent with respect to Υ.

Proof. It is enough to show that the removal of a single local selfloop results in a state-wise and transition-wise synthesis equivalent automaton. The rest of the claim follows by induction. Therefore let $\rightarrow_G = \rightarrow_H \dot\cup\, (x, \sigma, x)$ where $x \xrightarrow{\sigma}_G x$ is a local selfloop.

If $\sigma \in \Sigma_u$ then let $t = u = \varepsilon \in \Sigma_u^*$. Given $\sigma \in \Upsilon$, it follows that $tP_\Omega(\sigma)u = \varepsilon$ and $x \xrightarrow{\varepsilon}_G x$, so $x \xrightarrow{\sigma}_G x$ is an uncontrollable redundant transition. The claim follows from theorem 11.

If $\sigma \in \Sigma_c$ then let $t = \varepsilon \in \Sigma_u^*$. Given $\sigma \in \Upsilon$, it follows that $tP_\Omega(\sigma) = \varepsilon$ and $x \xrightarrow{\varepsilon}_G x$, so $x \xrightarrow{\sigma}_G x$ is a controllable prefix-redundant transition. The claim follows from theorem 14. □

6 Example

In this section, the proposed synthesis procedure is applied to a manufacturing system. The model consists of four machines M_1, M_2, M_3, and M_4, linked by two buffers B_1 and B_2. Workpieces are first processed by M_1 (s_1) and then placed into B_1 ($!f_1$), then they go to M_2 (s_2) and are placed into B_2 ($!f_2$). From B_2, the workpieces either go to M_3 for final processing (s_3) or to M_4 (s_4) for additional processing. However, M_4 has a fault that occasionally sends a workpiece back to B_1 ($!re$). At any time, M_1 and B_1 can be reset by the controllable event *rs*. Figure 6 shows the system layout and the automata model. Events $!f_1$, $!f_2$, $!f_3$, $!f_4$ and $!re$ are uncontrollable, all other events are controllable.

Compositional synthesis starts with the pair $(\mathcal{G}_0; \mathrm{id})$ where $\mathcal{G}_0 = \{M_1, M_2, M_3, M_4, B_1, B_2\}$. The first step is to calculate the composition $B_1 \parallel M_1$ shown in figure 7. Now $!f_1$, rs, and s_1 are local events, which makes $q_0 \xrightarrow{rs} q_0$ a local selfloop and $q_2 \xrightarrow{rs} q_0$ a controllable prefix-redundant transition with redirection path $q_2 \xrightarrow{!f_1} q_3 \xrightarrow{rs} q_0$. Removal of these transitions results in H_1. The modified synthesis pair is $(\mathcal{G}_1; \mathcal{D}_1)$ where $\mathcal{G}_1 = \{H_1, M_2, M_3, M_4, B_2\}$ and $\mathcal{D}_1$ is a synthesis

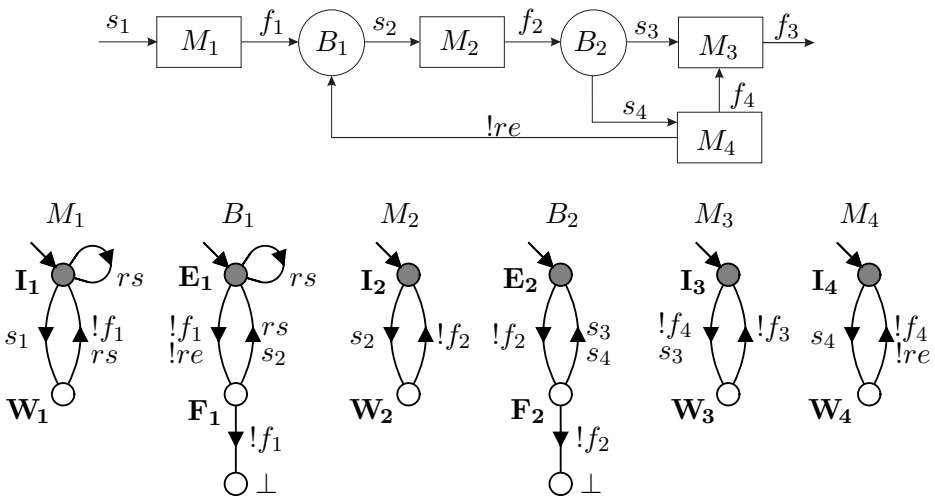

Figure 6: Manufacturing system example.

preserving redirection map that redirects $q_2 \xrightarrow{rs} q_0$ and $q_0 \xrightarrow{rs} q_0$ via $q_2 \xrightarrow{!f_1} q_3 \xrightarrow{rs} q_0$ and $q_0 \xrightarrow{\varepsilon} q_0$, respectively.

Next, $B_2 \parallel M_3$ is computed, shown in figure 7. This makes $!f_3$ and s_3 local events, and $q_3 \xrightarrow{!f_2} \perp$ becomes an uncontrollable redundant transition with redirection path $q_3 \xrightarrow{!f_3} q_1 \xrightarrow{!f_2} \perp$. The new synthesis pair is $(\mathcal{G}_2; \mathcal{D}_2 \circ \mathcal{D}_1)$ where $\mathcal{G}_2 = \{H_1, M_2, M_4, H_2\}$ and $\mathcal{D}_2$ is a synthesis preserving redirection map which redirects $q_3 \xrightarrow{!f_2} \perp$ via $q_3 \xrightarrow{!f_3} q_1 \xrightarrow{!f_2} \perp$.

The final synthesis step to compute $\sup\mathcal{CN}(\mathcal{G}_2)$ explores the state space of $\mathcal{G}_2$

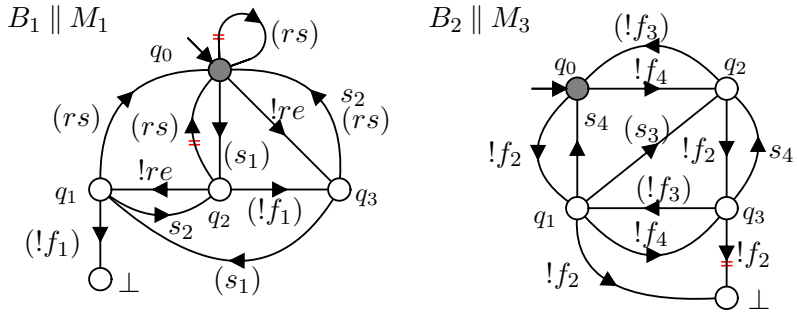

Figure 7: Some subsystems of the manufacturing example. The transitions to be removed are marked by double-line strike-through.

which has 100 states and 290 transitions. This is in contrast to standard monolithic synthesis, which explores the same state space using 340 transitions. Both the final monolithic and compositional supervisor have 26 states. However, the compositional supervisor has 63 transitions, while the monolithic supervisor has 81 transitions.

These improvements have been achieved by removing just three transitions from the model. More savings are likely in larger contexts, particularly in combination with state-removing abstraction rules.

7 Conclusions

It has been shown under which conditions transitions can be removed from an automaton while preserving compositional synthesis results. Different techniques to remove controllable and uncontrollable transitions have been presented, and a practical example has demonstrated how the number of transitions is reduced. The methods proposed in this paper are not intended to be used in isolation, but they will be combined with other synthesis-preserving abstraction methods. In the future, the authors plan to develop a framework for compositional synthesis that combines abstractions that remove states [7, 15] and transitions, as well as renaming [14] to remove nondeterminism.

References

[1] A. Aziz, V. Singhal, G. M. Swamy, and R. K. Brayton. Minimizing interacting finite state machines: A compositional approach to language containment. In *Proceedings of International Conference on Computer Design*, 1994.

[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] E. M. Clarke, D. E. Long, and K. L. McMillan. Compositional model checking. In *Proceedings of 5th IEEE Symposium on Logic in Computer Science*, pages 353–362, 1989.

[4] Jaana Eloranta. Minimizing the number of transitions with respect to observation equivalence. *BIT*, 31(4):397–419, 1991.

[5] Lei Feng and W. M. Wonham. Computationally efficient supervisor design: Abstraction and modularity. In *Proceedings of the 8th International Workshop on Discrete Event Systems, WODES'06*, pages 3–8, Ann Arbor, MI, USA, July 2006.

[6] Hugo Flordal and Robi Malik. Compositional verification in supervisory control. *SIAM Journal of Control and Optimization*, 48(3):1914–1938, 2009.

[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] Ryan J. Leduc, Bertil A. Brandin, Mark Lawford, and W. M. Wonham. Hierarchical interface-based supervisory control—part I: Serial case. *IEEE Transactions on Automatic Control*, 50(9):1322–1335, September 2005.

[10] 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.

[11] 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.

[12] Robi Malik, David Streader, and Steve Reeves. Fair testing revisited: A process-algebraic characterisation of conflicts. In Farn Wang, editor, *Proceedings of 2nd International Symposium on Automated Technology for Verification and Analysis, ATVA 2004*, volume 3299 of *LNCS*, pages 120–134, Taipei, Taiwan, October–November 2004. Springer-Verlag.

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

[14] 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.

[15] 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.

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

[17] 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.

[18] 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.

[19] K. C. Wong and W. M. Wonham. Modular control and coordination of discrete-event systems. *Discrete Event Dynamic Systems: Theory and Applications*, 8(3):247–297, October 1998.