FIVE ABSTRACTION RULES
TO REMOVE TRANSITIONS
WHILE PRESERVING
COMPOSITIONAL SYNTHESIS RESULTS

Sahar Mohajerani, Robi Malik, Martin Fabian

Technical Report: R003/2013
January 20, 2014

©Sahar Mohajerani, Robi Malik, Martin Fabian
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 [15] 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 [6,8,15,18]. Compositional methods seek to avoid large state spaces using abstraction, and have been used in verification [2,5] and synthesis [6,13,14]. 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 [4, 16]. Observation equivalence [12] and conflict equivalence [11] are well-known abstraction methods for nonblocking verification [5], but for synthesis these abstractions can only be applied in combination with unobservable events [9, 17], which limits their applicability.

Recently, frameworks for compositional synthesis based on abstractions of nondeterministic automata have been proposed [6, 13, 14], 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 often uses observation equivalence for abstraction, which allows for transition removal using the transitive reduction [3], but observation equivalence does not necessarily preserve synthesis results [14]. Supervision equivalence [6] allows for transition removal, but relies on additional state labels that make some desirable abstractions impossible. The methods [13, 14] 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 [12], 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 \( \Sigma \). For the purpose of supervisory control, this alphabet is partitioned into the set \( \Sigma_c \) of controllable events and the set \( \Sigma_u \) of uncontrollable events. Controllable events can be disabled by a supervisor, while uncontrollable events occur spontaneously, and are prefixed by an exclamation mark (!) in this paper. The special termination event \( \omega \in \Sigma_c \) denotes completion of a task, and does not appear anywhere else but to mark such completions.

\( \Sigma^* \) is the set of all finite traces of events from \( \Sigma \), including the empty trace \( \varepsilon \). 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 \( P_\Omega: \Sigma^* \rightarrow \Omega^* \) is the operation that removes from traces \( s \in \Sigma^* \) all events not in \( \Omega \).

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 \( G = (\Sigma, Q, \rightarrow, Q^0) \), where \( \Sigma \) is a finite set of events, \( Q \) is a finite set of states, \( \rightarrow \subseteq Q \times \Sigma \times Q \) is the state transition relation, and \( Q^0 \in Q \) is the set of initial states.

The transition relation is written in infix notation \( x \xrightarrow{\sigma} y \), and is extended to traces in \( \Sigma^* \) by letting \( x \xrightarrow{s} x \) for all \( x \in Q \), and \( x \xrightarrow{s} z \) if \( x \xrightarrow{s} y \) and \( y \xrightarrow{\sigma} z \) for some \( y \in Q \). Furthermore, \( x \xrightarrow{s} \) means \( x \xrightarrow{\sigma} y \) for some \( y \in Q \), and \( x \rightarrow y \) means \( x \xrightarrow{s} y \) for some \( s \in \Sigma^* \). For an alphabet \( \Omega \subseteq \Sigma \), the notation \( x \xrightarrow{\Omega} y \) means \( x \xrightarrow{s} y \)
for some $\sigma \in \Omega$, and $G \xrightarrow{\sigma} x$ means $q^o \xrightarrow{\sigma} x$ for some $q^o \in Q^o$. The language of automaton $G$ is $L(G) = \{ s \in \Sigma^* \mid G \xrightarrow{s} \}$. Finally, $G$ is deterministic, if $|Q^o| \leq 1$, 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 $\omega$ 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 $Q^\omega$ are shown shaded in the figures of this paper instead of explicitly showing $\omega$-transitions.

When multiple automata are brought together to interact, lock-step synchronisation in the style of [7] is used.

Definition 2 Let $G_1 = \langle \Sigma_1, Q_1, \rightarrow_1, Q^i_1 \rangle$ and $G_2 = \langle \Sigma_2, Q_2, \rightarrow_2, Q^i_2 \rangle$ be two automata. The synchronous composition of $G_1$ and $G_2$ is

$$G_1 \parallel G_2 = \langle \Sigma_1 \cup \Sigma_2, Q_1 \times Q_2, \rightarrow, Q^i_1 \times Q^i_2 \rangle$$

where

$$(x, y) \xrightarrow{\sigma} (x', y')$$ if $\sigma \in \Sigma_1 \cap \Sigma_2$, $x \xrightarrow{\sigma_1} x'$, $y \xrightarrow{\sigma_2} y'$;

$$(x, y) \xrightarrow{\sigma} (x', y)$$ if $\sigma \in \Sigma_1 \setminus \Sigma_2$, $x \xrightarrow{\sigma_1} x'$;

$$(x, y) \xrightarrow{\sigma} (x, y')$$ if $\sigma \in \Sigma_2 \setminus \Sigma_1$, $y \xrightarrow{\sigma_2} y'$.

2.3 Supervisory Control Theory

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

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

Definition 4 An automaton $G = \langle \Sigma, Q, \rightarrow, Q^o \rangle$ is nonblocking if, for every state $x \in Q$ and every trace $s \in (\Sigma \setminus \{\omega\})^*$ such that $Q^o \xrightarrow{s} x$, there exists a trace $t \in \Sigma^*$ such that $x \xrightarrow{ts}$. Two automata $G_1$ and $G_2$ are nonconflicting if $G_1 \parallel G_2$ is nonblocking.
For a plant \( G \) and specification \( K \), it is shown in [15] that there exists a least restrictive controllable sublanguage

\[ \text{sup} C_G (L(K)) \subseteq L(K) \]  

such that \( \text{sup} C_G (L(K)) \) is controllable with respect to \( G \) and nonblocking, and this language can be computed using a fixpoint 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** The restriction of \( G = \langle \Sigma, Q, \rightarrow, Q^o \rangle \) to \( X \subseteq Q \) is \( G|_X = \langle \Sigma, Q, \rightarrow|_X, Q^o \cap X \rangle \) where \( \rightarrow|_X = \{ (x, \sigma, y) \in \rightarrow \mid x, y \in X \} \).

**Definition 6** [10] Let \( G = \langle \Sigma, Q, \rightarrow, Q^o \rangle \) be an automaton. The synthesis step operator \( \Theta_G : 2^Q \rightarrow 2^Q \) for \( G \) is defined by

\[
\Theta_G^\text{cont}(X) = \{ x \in X \mid x \Sigma \rightarrow y \text{ implies } y \in X \} ;
\]

\[
\Theta_G^\text{nonb}(X) = \{ x \in X \mid x \xrightarrow{t} |_X \text{ for some } t \in \Sigma^* \} .
\]

**Theorem 1** [10] Let \( G = \langle \Sigma, Q, \rightarrow, Q^o \rangle \) be a deterministic automaton. \( \text{sup} C_N(G) \) is the least restrictive subautomaton of \( G \) that is controllable with respect to \( G \) and nonblocking.

The synthesis operator \( \text{sup} C_N \) performs synthesis for a plant automaton \( G \). A simple transformation [6] 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 \( \text{sup} C_N(G) \) or a language \( L(\text{sup} C_N(G)) \), 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

\[ S: \Sigma^* \rightarrow \{0, 1\} . \]
If $S(s\sigma) = 0$ for some $s \in \Sigma^*$ and $\sigma \in \Sigma_c$ then the supervisor disables the controllable event $\sigma$ after observing trace $s$, otherwise it enables $\sigma$. This results in the following closed-loop behaviour $\mathcal{L}(S/G)$ of the plant $G$ under the control of supervisor $S$:

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

A supervisor can be constructed naturally from a language $L \subseteq \Sigma^*$, by letting $S_L(s) = 1$ if and only if $s \in L$. For such a supervisor to be feasible, $L$ must be controllable [15].

### 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,

$$\mathcal{G} = \{G_1, G_2, \ldots, G_n\} . \quad (7)$$

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, the set of local events is denoted by $\Upsilon$, 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 automata $G$ and $T$ in figure 1 with $\Sigma_u = \Upsilon = \{!\gamma\}$. 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 $S_H = \sup CN(H \parallel T)$ disables event $\alpha$ 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 8 A synthesis pair is a pair $(G; D)$, where

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

Synthesis pairs are a variant of synthesis triples [13] 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 $G$ to be controlled and the redirection map $D$, which maps each input trace $s$ accepted by the original plant before all abstractions, to a trace accepted by the current abstracted plant $G$. A solution to the abstracted synthesis problem $G$ can be interpreted as a supervisor for the original plant by taking the redirection map into account.

Definition 9 For every synthesis pair $(G; D)$, define the represented supervisor map $S_{(G; D)}: \Sigma^* \rightarrow \{0, 1\}$ as follows:

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

Compositional synthesis starts by converting a control problem such as (7) into a synthesis pair $(G_0; id)$ where $G_0 = \{G_1, G_2, \ldots, G_n\}$ and $id: \Sigma^* \rightarrow \Sigma^*$ is the identity map, i.e, $id(s) = s$ for all $s \in \Sigma^*$. This initial synthesis pair is repeatedly abstracted such that the supervisor obtained from the abstraction remains a solution for the original problem. To ensure this property, each new synthesis pair needs to be synthesis equivalent to the previous pair.

Definition 10 Two synthesis pairs $(G_1; D_1)$ and $(G_2; D_2)$ are called synthesis equivalent with respect to plant $G$, written $(G_2; D_2) \simeq_{synth,G} (G_1; D_1)$, if $L(S_{(G_1; D_1)}/G) =$
\( \mathcal{L}(S(G_2; D_2)/G) \). Furthermore, \((G_1; D_1)\) and \((G_2; D_2)\) are synthesis equivalent, written \((G_2; D_2) \simeq_{\text{synth}} (G_1; D_1)\), if \((G_2; D_2) \simeq_{\text{synth}, G} (G_1; D_1)\) for every automaton \(G\).

Compositional synthesis terminates once \(G = \{H\}\) 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 \(G_0 = \{G_1, \ldots, G_n\}\) be a set of automata, and let \((G_k; D_k)\) be a synthesis pair such that \((G_0; \text{id}) \simeq_{\text{synth}, G_0} (G_k; D_k)\). Then
\[
\mathcal{L}(S(G_k; D_k)/G_0) = \mathcal{L}(\sup \mathcal{C}\mathcal{N}(G_0)).
\]

**Proof.** For \((G_0; \text{id})\) it follows from definition 9 that
\[
S_{(G_0; \text{id})}(s) = \begin{cases} 
1, & \text{if } s \in \mathcal{L}(\sup \mathcal{C}\mathcal{N}(G_0)) \\
0, & \text{otherwise}. 
\end{cases}
\]
By (6), it follows that \(\mathcal{L}(S(G_0; \text{id})/G_0) = \{ s \in G_0 \mid s \in \mathcal{L}(\sup \mathcal{C}\mathcal{N}(G_0)) \}\), which implies \(\mathcal{L}(S(G_0; \text{id})/G_0) = \mathcal{L}(\sup \mathcal{C}\mathcal{N}(G_0))\). Then it follows from definition 10 that \(\mathcal{L}(S(G_k; D_k)/G_0) = \mathcal{L}(S(G_0; \text{id})/G_0) = \mathcal{L}(\sup \mathcal{C}\mathcal{N}(G_0))\). \(\square\)

## 4 Transition-Wise Synthesis Equivalence

Several methods are known to abstract synthesis pairs such that the number of states is reduced [6, 14]. 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 [14], is augmented by a transition-based concept in definition 12.

**Definition 11** Let \(G = \langle \Sigma, Q, \rightarrow_G, Q^o \rangle\) and \(H = \langle \Sigma, Q, \rightarrow_H, Q^o \rangle\) be two automata. \(H\) is a state-wise synthesis abstraction of \(G\) with respect to \(\Upsilon \subseteq \Sigma\), if it holds for all automata \(T\) with \(\Sigma_T \cap \Upsilon = \emptyset\) that \(\hat{\Theta}_G{T} \subseteq \hat{\Theta}_H{T}\).

**Definition 12** Let \(G = \langle \Sigma, Q, \rightarrow_G, Q^o \rangle\) and \(H = \langle \Sigma, Q, \rightarrow_H, Q^o \rangle\) be two automata. \(H\) is a transition-wise synthesis abstraction of \(G\) with respect to \(\Upsilon \subseteq \Sigma\) if for every transition \(x \xrightarrow{\sigma} y\) there exist \(t, u \in \Upsilon^*\) such that:

(i) \(x \xrightarrow{tP_{\sigma}(\sigma)} y\)
(ii) for all automata $T$ such that $\Sigma_T \cap \Upsilon = \emptyset$ and all transitions $(x, x_T) \xrightarrow{\sigma_T} |_{\hat{\Theta}_G|T}$

$(y, y_T)$ of $\text{supC}(G \mid T)$ it holds that $(x, x_T) \xrightarrow{tP_u(x_T)u} |_{\hat{\Theta}_H|T} (y, y_T)$.

**Definition 13** Two automata $G$ and $H$ are state-wise (or transition-wise) synthesis equivalent with respect to $\Upsilon$, if $G$ is a state-wise (or transition-wise) synthesis abstraction of $H$ with respect to $\Upsilon$ and $H$ is a state-wise (or transition-wise) synthesis abstraction of $G$ with respect to $\Upsilon$.

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 $H = \langle \Sigma, Q, \rightarrow_H, Q^0 \rangle$ be a transition-wise synthesis abstraction of $G = \langle \Sigma, Q, \rightarrow_G, Q^0 \rangle$ 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 $\Sigma = \Omega \cup \Upsilon$, and let $T = \langle \Sigma_T, Q_T, \rightarrow_T, Q_T^0 \rangle$ 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 $(x, x_T) \in X^{n+1}_{H|T} = \Theta_H|T(X^n_{H|T}) = \Theta_{H|T}^{\text{cont}}(X^n_{H|T}) \cap \Theta_{H|T}^{\text{nonb}}(X^n_{H|T})$.

To see that $(x, x_T) \in \Theta_{H|T}^{\text{cont}}(X^n_{H|T})$, let $v \in \Sigma_u$ such that $(x, x_T) \xrightarrow{v} |_{H|T} (y, y_T)$. From $\rightarrow_H \subseteq \rightarrow_G$, it follows that $(x, x_T) \xrightarrow{v} |_{G|T} (y, y_T)$. Since $(x, x_T) \in \hat{\Theta}_G|T$ and $v \in \Sigma_u$, it follows that $(x, x_T) \xrightarrow{v} |_{\hat{\Theta}_G|T} (y, y_T)$. By definition 13, there exist $t, u \in \Upsilon^*$ such that $(x, x_T) \xrightarrow{tP_u(x_T)u} |_{\hat{\Theta}_H|T} (y, y_T)$. This implies $(y, y_T) \in \hat{\Theta}_H|T \subseteq \Theta^n_{H|T}(Q \times Q_T) = X^n_{H|T}$. As $v$ and $(y, y_T)$ were chosen arbitrarily, it follows that $(x, x_T) \in \Theta_{H|T}^{\text{cont}}(X^n_{H|T})$.

Furthermore, to see that $(x, x_T) \in \Theta_{H|T}^{\text{nonb}}(X^n_{H|T})$, 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 $(x, x_T) \in \Theta_{H|T}^{\text{nonb}}(X^n_{H|T})$. \hfill $\square$

To preserve transition-wise synthesis equivalence after removal of a transition, definition 12 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
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 $D : \Sigma^* \rightarrow \Sigma^*$ is constructed where $D(\alpha s) = \gamma \alpha s$ for all $s \in \Sigma^*$ and $D(s) = s$ for all $s$ such that $\alpha$ 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 $(\mathcal{G}; \text{id})$ is synthesis equivalent to $(\mathcal{H}; D)$ where $\mathcal{H} = \{H, T\}$. Although the supervisor $S_H = \text{supCN}(H \parallel T)$ obtained for $\mathcal{H}$ cannot directly be used to control the original plant $\mathcal{G}$, this becomes possible in combination with the redirection map $D$. As $D(\alpha) = \gamma \alpha \in \mathcal{L}(\text{supCN}(H \parallel T))$, the supervisor computed for $(\mathcal{H}, D)$ will enable the controllable event $\alpha$ 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 14** Let $\Sigma_1 \subseteq \Sigma_2$. The extension of a prefix-preserving map $D_1 : \Sigma_1^* \rightarrow \Sigma_1^*$ is $D_2 : \Sigma_2^* \rightarrow \Sigma_2^*$, defined by

$$D_2(\varepsilon) = D_1(\varepsilon)$$

$$D_2(s\sigma) = \begin{cases} D_2(s)t' & \text{if } \sigma \in \Sigma_1, \, D_1(P_{\Sigma_1}(s)) = s', \\
&D_2(s) & \text{if } \sigma \notin \Sigma_1. \end{cases}$$

A redirection map $D_1$ is extended by copying the additional events without change at the appropriate position into the output stream. The extension $D_2$ is well-defined if $D_1$ 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 $D_1$ and $D_2$ are both denoted by $D$.

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 15** Let \( G \) and \( H \) be two automata. A map \( \mathcal{D} : \Sigma^* \to \Sigma^* \) 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

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

(13)

**Proposition 5** Let \( G = \langle \Sigma, Q \rightarrow G, Q^0 \rangle \) and \( H = \langle \Sigma, Q \rightarrow H, Q^0 \rangle \) 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 \( \Upsilon \).

**Proof.** Let \( \Sigma = \Omega \cup \Upsilon \). Since \( G \) and \( H \) are transition-wise synthesis equivalent, for every transition \( x \xrightarrow{\sigma} y \) there exists a trace \( d(x, \sigma) = tP_\Sigma(\sigma)u \) where \( t, u \in \Upsilon^* \) satisfy the conditions (i) and (ii) in Definition 12. Then construct the prefix-preserving map \( \mathcal{D} : \Sigma^* \to \Sigma^* \) as follows:

\[
\mathcal{D}(\varepsilon) = \varepsilon
\]

(14)

\[
\mathcal{D}(t\sigma) = \begin{cases} 
\mathcal{D}(t)d(x, \sigma), & \text{if } G \xrightarrow{t} x \xrightarrow{\sigma} \\
\mathcal{D}(t)\sigma, & \text{otherwise}
\end{cases}
\]

(15)

Now let \( T = \langle \Sigma_T, Q_T \rightarrow T, Q_T^0 \rangle \) 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:

\[
\mathcal{D}(\varepsilon) = \varepsilon
\]

(16)

\[
\mathcal{D}(t\sigma) = \begin{cases} 
\mathcal{D}(t)d(x, \sigma), & \text{if } G \xrightarrow{P_\Sigma(t)} x \xrightarrow{P_\Sigma(\sigma)} \\
\mathcal{D}(t)\sigma, & \text{otherwise}
\end{cases}
\]

(17)

Note that condition (i) in definition 12 ensures that \( \mathcal{D}(P_\Sigma(s)) \) is a trace in \( L(G) \) implies \( P_\Sigma(s) \in L(G) \). Furthermore, for all \( s = \sigma_1 \cdots \sigma_n \) such that \( P_\Sigma(s) \in L(G) \),

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

(18)

where \( G \xrightarrow{P_\Sigma(\sigma_1 \cdots \sigma_k)} x_k \). It remains to be confirmed that \( \mathcal{D} \) satisfies Definition 15. Therefore, let \( s = \sigma_1 \cdots \sigma_n \in (\Sigma \cup \Sigma_T)^* \).

First assume that \( \sup \mathcal{CN}(G \parallel T) \xrightarrow{s} (x_0, x_T) \). Then there exists a path \( G \parallel T \xrightarrow{\varepsilon} (x_0, x_0^T) \xrightarrow{\sigma_1} (x_1, x_1^T) \xrightarrow{\sigma_2} (x_2, x_2^T) \cdots x_n \xrightarrow{\sigma_n} (x_n, x_n^T) = (x, x_T) \).
Consider some $k \in \{1, \ldots, n\}$. If $\sigma_k \in \Sigma$, then since $(x_{k-1}, x_{k-1}^T) \overset{\kappa}{\rightarrow} (x_k, x_k^T)$ it follows by definition 12 that $(x_{k-1}, x_{k-1}^T) \overset{\kappa}{\rightarrow} (x_k, x_k^T)$. If $\sigma_k \in \Sigma_T \setminus \Sigma$, then $d(x_{k-1}, \sigma_k) = \sigma_k$ and $(x_{k-1}, x_{k-1}^T) \overset{\kappa}{\rightarrow} (x_k, x_k^T)$ as $(x_{k-1}, x_{k-1}^T), (x_k, x_k^T) \in \hat{\Theta}_{G\parallel T} = \hat{\Theta}_{H\parallel T}$ since $G$ and $H$ are state-wise synthesis equivalent. Combining these paths for $k = 1, \ldots, n$ gives $H \parallel T \overset{e}{\rightarrow} (x_0, x_0^T) \overset{\kappa}{\rightarrow} (x_1, x_1^T) \overset{\kappa}{\rightarrow} (x_n, x_n^T)$, and this implies by (18) that $\supCN(H \parallel T) \overset{D(s)}{\rightarrow} (x_n, x_n^T) = (x, x^T)$.

Conversely assume that $\supCN(H \parallel T) \overset{D(s)}{\rightarrow} (x, x^T)$. Note that $\mathcal{D}(\mathcal{P}_\Sigma(s)) = \mathcal{P}_\Sigma(\mathcal{D}(s)) \in \mathcal{L}(H) \subseteq \mathcal{L}(G)$, which implies $\mathcal{P}_\Sigma(s) \in \mathcal{L}(G)$. By (18), there exists a path $H \parallel T \overset{e}{\rightarrow} (x_0, x_0^T) \overset{\kappa}{\rightarrow} (x_1, x_1^T) \overset{\kappa}{\rightarrow} (x_n, x_n^T)$ 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 $x_{k-1} \overset{\kappa}{\rightarrow} G x_k$, and since $T$ does not synchronise on the events introduced by $d$, this implies $(x_{k-1}, x_{k-1}^T) \overset{\kappa}{\rightarrow} (x_k, x_k^T)$. Then, given $(x_{k-1}, x_{k-1}^T), (x_k, x_k^T) \in \hat{\Theta}_{H\parallel T} = \hat{\Theta}_{G\parallel T}$, it follows that $(x_{k-1}, x_{k-1}^T) \overset{\kappa}{\rightarrow} (x_k, x_k^T)$. Otherwise, if $\sigma_k \in \Sigma_T \setminus \Sigma$, then $d(x_{k-1}, \sigma_k) = \sigma_k$ and it follows from $\hat{\Theta}_{H\parallel T} = \hat{\Theta}_{G\parallel T}$ that $(x_{k-1}, x_{k-1}^T) \overset{\kappa}{\rightarrow} (x_k, x_k^T)$. Combining these transitions for $k = 1, \ldots, n$ gives $G \parallel T \overset{e}{\rightarrow} (x_0, x_0^T) \overset{\kappa}{\rightarrow} (x_n, x_n^T)$, i.e., $\supCN(G \parallel T) \overset{e}{\rightarrow} (x_n, x_n^T) = (x, x^T)$.

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

**Proposition 6** Let $G = \{G_1, \ldots, G_n\}$ and let $H = \{H_1, G_2, \ldots, G_n\}$ where $G_i = (\Sigma_i, Q_i, \cdot \rightarrow_i, Q_i^o)$, and let $D_1 : \Sigma_1^* \rightarrow \Sigma_1^*$ be a synthesis-preserving redirection map from $G_1$ to $H_1$ with respect to $\Upsilon \subseteq \Sigma_1$ such that $\Upsilon \cap \Sigma_1 = \cdots = \Upsilon \cap \Sigma_n = \emptyset$. Then $(G; \mathcal{D}) \overset{\sim}{\Rightarrow} \text{synth}(H; D_1 \circ \mathcal{D})$ for every prefix-preserving map $D$.

**Proof.** Let $S_1 = S_{G_1; D}$ and $S_2 = S_{G_2; D_1 \circ \mathcal{D}}$, let $T = G_2 \parallel \cdots \parallel G_n$, and let $G$ be an automaton. It is to be shown that $(G; \mathcal{D}) \overset{\sim}{\Rightarrow} \text{synth,} G (H; D_1 \circ \mathcal{D})$ based on definition 10.

First, let $s \in \mathcal{L}(S_1/G)$. This means $s \in \mathcal{L}(G)$ and $\mathcal{D}(s) \in \mathcal{L}(\supCN(G_1 \parallel T))$. Since $\mathcal{D}(s) \in \mathcal{L}(\supCN(G_1 \parallel T))$, it follows that $G_1 \parallel T \overset{\mathcal{D}(s)}{\rightarrow} \hat{\Theta}_{G_1 \parallel T}$. Since $D_1$ is a synthesis-preserving redirection map, it follows by definition 15 that $H_1 \parallel T \overset{D_1(\mathcal{D}(s))}{\rightarrow} \hat{\Theta}_{H_1 \parallel T}$, which implies $D_1(\mathcal{D}(s)) \in \mathcal{L}(\supCN(H_1 \parallel T))$. Since also $s \in \mathcal{L}(G)$ and $\mathcal{D}(s) \in \mathcal{L}(\supCN(G_1 \parallel T))$, it follows that $H_1 \parallel T \overset{\mathcal{D}(s)}{\rightarrow} \hat{\Theta}_{H_1 \parallel T}$. Therefore, $(G; \mathcal{D}) \overset{\sim}{\Rightarrow} \text{synth,} G (H; D_1 \circ \mathcal{D})$.
\[ \mathcal{L}(G) \], it follows that \( s \in \mathcal{L}(S_2/G) \).

Conversely, let \( s \in \mathcal{L}(S_2/G) \). This means \( \mathcal{D}_1(\mathcal{D}(s)) \in \mathcal{L}(\supCN(H_1 \parallel T)) \) and \( s \in \mathcal{L}(G) \). Since \( \mathcal{D}_1(\mathcal{D}(s)) \in \mathcal{L}(\supCN(H_1 \parallel T)) \), it follows that \( \supCN(H_1 \parallel T) \xrightarrow{\mathcal{D}_1(\mathcal{D}(s))} \). Since \( \mathcal{D}_1 \) is a synthesis-preserving redirection map, it follows by definition 15 that \( \supCN(G_1 \parallel T) \xrightarrow{\mathcal{D}(s)} \), which implies \( \mathcal{D}(s) \in \mathcal{L}(\supCN(G_1 \parallel T)) \). Since also \( s \in \mathcal{L}(G) \), it follows that \( s \in \mathcal{L}(S_1/G) \). □

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 \( \Upsilon \) such that \( (G; \mathcal{D}) \simeq_{\text{synth}} (H; \mathcal{D}_1 \circ \mathcal{D}) \).

**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 [12]. It can be implemented by simple algorithms, and its application in compositional verification can substantially reduce the state space [5]. The idea of observation equivalence is to identify and merge states with the same future behaviour.

**Definition 16** Let \( G = \langle \Sigma, Q_G, \rightarrow_G, Q^0_G \rangle \) and \( H = \langle \Sigma, Q_H, \rightarrow_H, Q^0_H \rangle \) be two automata with \( \Sigma = \Omega \cup \Upsilon \). Then \( G \) and \( H \) are observation equivalent with respect
to $\Upsilon$, 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 $x_G \xrightarrow{\sigma} y_G$, then there exist $t, u \in \Upsilon^*$ such that $x_H \xrightarrow{t\sigma u} y_H$;
- if $x_G \approx x_H$ and $x_H \xrightarrow{\sigma} y_H$, then there exist $t, u \in \Upsilon^*$ such that $x_G \xrightarrow{t\sigma u} y_G$;
- for each $q_G^\circ \in Q_G^\circ$ there exists $q_H^\circ \in Q_H^\circ$ such that $q_G^\circ \approx q_H^\circ$, and vice versa.

Observation equivalence is tested based on the transitive closure of the local event transitions [1]. 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 [3] if the automaton contains a matching redirection path.

**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 $\Sigma = \Omega \cup \Upsilon$ and $\rightarrow_H \subseteq \rightarrow_G$. Automaton $H$ is a result of observation equivalence redundant transition removal from $G$ with respect to $\Upsilon$, if for all transitions $x \xrightarrow{\sigma} y$ there exist $t, u \in \Upsilon^*$ such that $x \xrightarrow{t\sigma u} y_H$.

**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 [3, 12]. Unfortunately, this does not include synthesis equivalence [14].

**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 $\beta$, 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 $\beta$ 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 18** Let $G = \langle \Sigma, Q, \rightarrow_G, Q_0 \rangle$ and $H = \langle \Sigma, Q, \rightarrow_H, Q_0 \rangle$ be two automata with $\Sigma = \Omega \cup \Upsilon$ and $\rightarrow_H \subseteq \rightarrow_G$. Automaton $H$ is a result of *uncontrollable redundant transition removal* from $G$ with respect to $\Upsilon$, if the following conditions hold for all transitions $x \xrightarrow{\sigma} y$.

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

(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 18 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 $H = \langle \Sigma, Q, \rightarrow_H, Q_0 \rangle$ be a result of uncontrollable redundant transition removal from $G = \langle \Sigma, Q, \rightarrow_G, Q_0 \rangle$ with respect to $\Upsilon \subseteq \Sigma$. Then $G$ and $H$ are state-wise synthesis equivalent with respect to $\Upsilon$.

**Proof.** Let $\Sigma = \Omega \cup \Upsilon$, and let $T = \langle \Sigma_T, Q_T, \rightarrow, Q_T^0 \rangle$ 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 $\hat{\Theta}_G\|T \subseteq X^*_H = \Theta^n_{H\|T}(Q \times Q_T)$.

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

15
Inductive step. Assume \( \hat{\Theta}_{G||T} \subseteq X^n_H \) for some \( n \geq 0 \), and let \( (x, x_T) \in \hat{\Theta}_{G||T} \). It remains to be shown that \( (x, x_T) \in X^{n+1}_H = \Theta_H||T(X^n_H) = \Theta^\text{cont}_H||T(X^n_H) \cap \Theta^\text{nonb}_H||T(X^n_H) \).

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

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

\[
(x, x_T) = (x_0, x_T^0) \xrightarrow{\sigma_1} \hat{\Theta}_{G||T} \cdots \xrightarrow{\sigma_k} \hat{\Theta}_{G||T} \xrightarrow{\omega} \hat{\Theta}_{G||T} (x_{k+1}, x_{k+1}^T). \tag{19}
\]

Consider a transition \( (x_{l-1}, x_{l-1}^T) \xrightarrow{\sigma_l} H||T (x_l, x_l^T) \) 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 18 there exist traces \( t_l, u_l \in \pi \) such that \( x_{l-1} \xrightarrow{t_lP_\Sigma(\sigma_l)u_l} x_l \). Since \( \Sigma_T \cap \Upsilon = \emptyset \), it follows that \( (x_{l-1}, x_{l-1}^T) \xrightarrow{t_lP_\Sigma(\sigma_l)u_l} G||T (x_l, x_l^T) \), and since \( (x_{l-1}, x_{l-1}^T) \in \hat{\Theta}_{G||T} \) and \( t_lP_\Sigma(\sigma_l)u_l \in \Sigma^*_u \), it follows by controllability that \( (x_{l-1}, x_{l-1}^T) \xrightarrow{t_lP_\Sigma(\sigma_l)u_l} G||T (x_l, x_l^T) \).

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

\[
(x, x_T) = (x_0, x_T^0) \xrightarrow{t_1P_\Sigma(\sigma_1)u_1} H||T |X^n_H \cdots \xrightarrow{t_kP_\Sigma(\sigma_k)u_k} H||T |X^n_H (x_{k+1}, x_{k+1}^T),
\]

which implies \( (x, x_T) \in \Theta^\text{nonb}_H||T(X^n_H) \).

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

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

Base case. Clearly \( \hat{\Theta}_{H||T} \subseteq Q \times Q_T = \Theta^0_G||T(Q \times Q_T) = X^0_G \).
Lemma 10  Let $H \subseteq X^n_G$ be an uncontrollable redundant transition, and by definition 18 there exist $t, u \in \Sigma_u$ and $(x, x_T) \not\in H_T (y, y_T)$. If $u \not\in \Sigma$ or $x \not\in \mathcal{G}_H y$, then clearly $(x, x_T) \not\in H_T (y, y_T)$, and since $(x, x_T) \in \hat{\mathcal{H}}_H T$ and $u \in \Sigma_u$, it follows by controllability that $(y, y_T) \in \hat{\mathcal{H}}_H T$. Otherwise $x \mapsto_G y$ is an uncontrollable redundant transition, and by definition 18 there exist $t, u \in (\Sigma_u \cap \Upsilon)^*$ such that $(x, x_T) \not\in H_T (y, y_T)$. Since $\Sigma_T \cap \Upsilon = \emptyset$, it follows that $(x, x_T) \not\in H_T (y, y_T)$, and since $tP_{\Sigma}(u) \not\in \Sigma_u$ and $(x, x_T) \in \hat{\mathcal{H}}_H T$ it follows by controllability that $(y, y_T) \in \hat{\mathcal{H}}_H T$. In both cases by inductive assumption $(y, y_T) \in \hat{\mathcal{H}}_H T \subseteq X^n_G$, and since $u \in \Sigma_u$ 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 $(x, x_T) \in \hat{\mathcal{H}}_H T$, there exists a path $(x, x_T) \not\in \hat{\mathcal{H}}_H T \subseteq X^n_G$. Hence, $(x, x_T) \in \Theta^\text{nonb}_{G|T}(X^n_G)$.

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

Lemma 10  Let $H = \langle \Sigma, Q, \rightarrow_H, Q^0 \rangle$ be a result of uncontrollable redundant transition removal from $G = \langle \Sigma, Q, \rightarrow_G, Q^0 \rangle$ with respect to $\Upsilon \subseteq \Sigma$. Then $G$ and $H$ are transition-wise synthesis equivalent with respect to $\Upsilon$.

Proof.  It must be shown that $G$ is a transition-wise synthesis abstraction of $H$ and vice versa. Condition (i) in definition 13 follows immediately from definition 18. To show condition (ii), let $\Sigma = \Omega \cup \Upsilon$, and let $T = \langle \Sigma_T, Q, \rightarrow, Q^0 \rangle$ be an automaton such that $\Sigma_T \cap \Upsilon = \emptyset$.

First, let $(x, x_T) \not\in \hat{\mathcal{H}}_H T (y, y_T)$. By lemma 9 it holds that $(x, x_T) \not\in \hat{\mathcal{H}}_H T$. If $\sigma \not\in \Sigma$ or $x \not\in \mathcal{G}_H y$, then clearly $(x, x_T) \not\in H_T (y, y_T)$, which implies $(x, x_T) \not\in \hat{\mathcal{H}}_H T (y, y_T)$, and $(x, x_T) \not\in H_T (y, y_T)$. Otherwise $x \not\in \mathcal{G}_H y$ is an uncontrollable redundant transition, and by definition 18 there exist $t, u \in (\Sigma_u \cap \Upsilon)^*$ such that $x \not\in \mathcal{H}_T (y, y_T)$. Since $\Sigma_T \cap \Upsilon = \emptyset$, it follows that $(x, x_T) \not\in H_T (y, y_T)$, and since $(x, x_T) \in \hat{\mathcal{H}}_H T$ and $tP_{\Sigma}(u) \not\in \Sigma_u$, it follows by controllability that $(x, x_T) \not\in \hat{\mathcal{H}}_H T (y, y_T)$. Thus, in both cases, there exist $t, u \in \Upsilon^*$ such that $(x, x_T) \not\in \hat{\mathcal{H}}_H T (y, y_T)$.}

17
Conversely, let \((x, x^T) \overset{\sigma}{\rightarrow}_{\hat{\Theta}_{H \parallel T}} (y, y^T)\). Since \(\rightarrow_H \subseteq \rightarrow_G\), it follows that \((x, x^T) \overset{\sigma}{\rightarrow}_{G \parallel T} (y, y^T)\). Also \((x, x^T), (y, y^T) \in \hat{\Theta}_{H \parallel T} = \hat{\Theta}_{G \parallel T}\) by lemma 9, which implies \((x, x^T) \overset{\sigma}{\rightarrow}_{\hat{\Theta}_{G \parallel T}} (y, y^T)\). Then let \(t = \varepsilon\) and \(u = P_\Upsilon(\sigma)\), and it follows that \((x, x^T) \overset{tP_\Upsilon(\sigma)u}{\rightarrow}_{\hat{\Theta}_{G \parallel T}} (y, y^T)\).

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

**Theorem 11** Let \(H = \langle \Sigma, Q, \rightarrow_H, Q^0 \rangle\) be a result of uncontrollable redundant transition removal from \(G = \langle \Sigma, Q, \rightarrow_G, Q^0 \rangle\) with respect to \(\Upsilon \subseteq \Sigma\). Then \(G\) and \(H\) are state-wise and transition-wise synthesis equivalent with respect to \(\Upsilon\).

**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 \(q_0 \overset{\alpha}{\rightarrow} q_2\) is observation equivalence redundant because of \(q_0 \overset{\gamma}{\rightarrow} q_2\). Its removal results in \(H\). In both \(G\) and \(H\), the controllable event \(\gamma\) must be disabled in the initial state to prevent blocking via the uncontrollable event \(!\upsilon\). However, after disabling \(\gamma\), termination is no longer possible in \(H\), yet it remains possible in \(G\) via \(q_0 \overset{\alpha}{\rightarrow} q_2\). The test \(T\) demonstrates that \(H\) is not a state-wise synthesis abstraction of \(G\).

In example 4, the redirection path \(q_0 \overset{\gamma}{\rightarrow} q_2\) 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 $!v$ is the only uncontrollable event. Transition $q_0 \overset{\alpha}{\to} q_2$ is observation equivalence redundant because of $q_0 \overset{\gamma}{\to} q_2$, and its removal results in $H$. In $H \parallel T$, the controllable event $\gamma$ must be disabled to prevent blocking via the uncontrollable event $!v$. By disabling $\gamma$, state $(q_2, q_T^1)$ becomes unreachable in $\supCN(H \parallel T)$, but it remains reachable in $\supCN(G \parallel T)$. 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 $\Sigma_u = \Upsilon = \{!\mu, !v\}$. Transition $q_0 \overset{\alpha}{\to} q_3$ is observation equivalence redundant because $q_0 \overset{!\mu}{\alpha}!\mu q_3$. Let $H_1$ be the result of removing the transition $q_0 \overset{\alpha}{\to} q_3$. In both $G_1$ and $H_1$, the controllable transition $q_1 \overset{\alpha}{\to} q_2$ must be disabled to avert blocking via the uncontrollable event $!v$. Removing this transition makes $q_3$ unreachable in $\supCN(H_1 \parallel T)$, but it remains reachable in $\supCN(G_1 \parallel T)$. The test $T$ demonstrates that $G_1$ and $H_1$ are not transition-wise synthesis equivalent as $G_1$ is not a transition-wise synthesis abstraction of $H_1$.

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 19** Let $G = \langle \Sigma, Q, \rightarrow_G, Q^0 \rangle$ and $H = \langle \Sigma, Q, \rightarrow_H, Q^0 \rangle$ be two automata with $\Sigma = \Omega \cup \Upsilon$ and $\rightarrow_H \subseteq \rightarrow_G$. Automaton $H$ is a result of controllable prefix-redundant transition removal from $G$ with respect to $\Upsilon$, if the following conditions hold for all transitions $x \rightarrow_H y$.

1. If $\sigma \in \Sigma_u$ then $x \rightarrow_H y$.

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

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 $H = \langle \Sigma, Q, \rightarrow_H, Q^0 \rangle$ be a result of controllable prefix-redundant transition removal from $G = \langle \Sigma, Q, \rightarrow_G, Q^0 \rangle$ with respect to $\Upsilon \subseteq \Sigma$. Then $G$ and $H$ are state-wise synthesis equivalent with respect to $\Upsilon$.

**Proof.** Let $\Sigma = \Omega \cup \Upsilon$, and let $T = \langle \Sigma_T, Q_T, \rightarrow, Q^0 \rangle$ 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 $\hat{\Theta}_{G||T} \subseteq X^n_H = \Theta_{H||T}^n(Q \times Q_T)$.

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

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

To see that $(x, x_T) \in \Theta_{H||T}^\text{cont}(X^n_H)$, let $\upsilon \in \Sigma_u$ and $(x, x_T) \overset{\upsilon}{\implies}_H (y, y_T)$. Since $\rightarrow_H \subseteq \rightarrow_G$, it follows that $(x, x_T) \overset{\upsilon}{\implies}_G (y, y_T)$. 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 $(x, x_T) \in \Theta_{H||T}^\text{cont}(X^n_H)$.

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

\[
(x, x_T) = (x_0, x_T^0) \overset{\sigma_1}{\implies}_{\hat{\Theta}_{G||T}} \cdots \overset{\sigma_k}{\implies}_{\hat{\Theta}_{G||T}} (x_k, x_T^k) \overset{\omega}{\implies}_{\hat{\Theta}_{G||T}} (x_{k+1}, x_T^{k+1})
\]

(20)
Consider a transition \((x_{l-1}, x_{l-1}^T) \xrightarrow{\sigma_l} H (x_l, x_l^T)\) in (20). If \(\sigma_l \notin \Sigma\) or \(x_{l-1} \xrightarrow{\sigma_l} H x_l\), then clearly \((x_{l-1}, x_{l-1}^T) \xrightarrow{\sigma_l} H (x_l, x_l^T)\), and by inductive assumption it follows that \((x_{l-1}, x_{l-1}^T), (x_l, x_l^T) \in \hat{\Theta}_{G \parallel T} \subseteq X_H^n\), i.e.,

\((x_{l-1}, x_{l-1}^T) \xrightarrow{\sigma_l} H \parallel T \mid X_H^n (x_l, x_l^T)\).

Otherwise \(x_{l-1} \xrightarrow{\sigma_l} G x_l\) is a controllable prefix-redundant transition, and by definition 19 there exists \(t_l \in (\Upsilon \cap \Sigma_u)^*\) such that \(x_{l-1} \xrightarrow{t_l} y_{l-1} \xrightarrow{P_l(\sigma_l)} H x_l\). Since \(\Sigma_T \cap \Upsilon = \emptyset\), it follows that

\((x_{l-1}, x_{l-1}^T) \xrightarrow{t_l} H \parallel T (y_{l-1}, x_{l-1}^T) \xrightarrow{P_l(\sigma_l)} H \parallel T (x_l, x_l^T)\).

Since \((x_{l-1}, x_{l-1}^T) \in \hat{\Theta}_{G \parallel T}\) and \(t_l \in \Sigma_u^*\) and \(\rightarrow_H \subseteq \rightarrow_G\), it follows that \((x_{l-1}, x_{l-1}^T) \xrightarrow{t_l} \hat{\Theta}_{G \parallel T} (y_{l-1}, x_{l-1}^T)\).

Since also \((x_{l-1}, x_{l-1}^T) \in \hat{\Theta}_{G \parallel T} \subseteq X_H\) by inductive assumption, it follows that \((x_{l-1}, x_{l-1}^T) \xrightarrow{t_l} H \parallel T \mid X_H^n (y_{l-1}, x_{l-1}^T) \xrightarrow{P_l(\sigma_l)} H \parallel T \mid X_H^n (x_l, x_l^T)\). 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

\[ (x, x_T) = (x_0, x_0^T) \xrightarrow{t_1 P_l(\sigma_1)} H \parallel T \mid X_H^n \xrightarrow{\cdots} \xrightarrow{t_k P_l(\sigma_k)} H \parallel T \mid X_H^n \xrightarrow{t_{k+1} \omega} H \parallel T \mid X_H^n (x_{k+1}, x_{k+1}^T), \]

which implies \((x, x_T) \in \Theta_G^{\parallel T}(X_H^n)\).

It has been shown that \((x, x_T) \in \Theta_G^{\parallel T}(X_H^n) \cap \Theta_G^{\parallel T}(X_H^n) = X_H^{n+1}\).

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

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

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

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

Next it is shown that \((x, x_T) \in \Theta_{G \parallel T}^{\parallel T}(X_G^n)\). Since \((x, x_T) \in \hat{\Theta}_{H \parallel T}\), there
exists a path \((x, x_T) \xrightarrow{t_\omega} \hat{\Theta}_{H \parallel T}\). Since \(\rightarrow_H \subseteq \rightarrow_G\), it follows by inductive assumption that \((x, x_T) \xrightarrow{t_\omega} X^n_G\). Hence, \((x, x_T) \in \Theta^n_{G \parallel T}(X^n_G)\).

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

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

**Proof.** It must be shown that \(G\) is a transition-wise synthesis abstraction of \(H\) and vice versa. Condition (i) in definition 13 follows immediately from definition 19. To show condition (ii), let \(\Sigma = \Omega \cup \Upsilon\), and let \(T = \langle \Sigma_T, Q, \rightarrow, Q^o \rangle\) be an automaton such that \(\Sigma_T \cap \Upsilon = \emptyset\).

First, let \((x, x_T) \xrightarrow{\sigma} \hat{\Theta}_{G \parallel T} (y, y_T)\). By lemma 12 it holds that \((x, x_T), (y, y_T) \in \hat{\Theta}_{H \parallel T}\). If \(\sigma \notin \Sigma\) or \(x \xrightarrow{\sigma_H} y\), then clearly \((x, x_T) \xrightarrow{\sigma_H} T_H (y, y_T)\), which implies \((x, x_T) \xrightarrow{\sigma} \hat{\Theta}_{H \parallel T} (y, y_T)\) and \((x, x_T) \xrightarrow{P_T(\sigma)P_{\Upsilon}(\sigma)} \hat{\Theta}_{H \parallel T} (y, y_T)\). Otherwise \(x \xrightarrow{\sigma} y\) is a controllable prefix-redundant transition, and by definition 19 there exists \(t \in (\Sigma_u \cap \Upsilon)^*\) such that \(x \xrightarrow{\Sigma_u} z \xrightarrow{P_{\Upsilon}(t)} H\). Since \(\Sigma_T \cap \Upsilon = \emptyset\), it follows that \((x, x_T) \xrightarrow{\Sigma_u} \hat{\Theta}_{H \parallel T} (z, x_T) \xrightarrow{P_{\Upsilon}(t)} H (y, y_T)\), and since \((x, x_T) \in \hat{\Theta}_{H \parallel T}\) and \(t \in \Sigma_u\), it follows by controllability that \((x, x_T) \xrightarrow{tP_{\Upsilon}(t)} \hat{\Theta}_{H \parallel T} (y, y_T)\). Then let \(u = \varepsilon\), and in both cases there exist \(t, u \in \Upsilon^*\) such that \((x, x_T) \xrightarrow{tP_{\Upsilon}(t)u} \hat{\Theta}_{H \parallel T} (y, y_T)\).

Conversely, let \((x, x_T) \xrightarrow{\sigma} \hat{\Theta}_{H \parallel T} (y, y_T)\). Since \(\rightarrow_H \subseteq \rightarrow_G\), it follows that \((x, x_T) \xrightarrow{\sigma} G (y, y_T)\). Also \((x, x_T), (y, y_T) \in \hat{\Theta}_{H \parallel T} = \hat{\Theta}_{G \parallel T}\) by lemma 12, which implies \((x, x_T) \xrightarrow{\sigma} \hat{\Theta}_{G \parallel T} (y, y_T)\). Then let \(t = \varepsilon\) and \(u = P_H(\sigma)\), and it follows that \((x, x_T) \xrightarrow{tP_H(\sigma)u} \hat{\Theta}_{G \parallel T} (y, y_T)\).

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

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

**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 !\( \nu \)-transition in state \( q_2 \). Automaton \( G_3 \) also has the observation equivalence redundant transition \( q_0 \xrightarrow{\mu \alpha} q_3 \) and an !\( \nu \)-transition enabled after \( \alpha \) on the redirection path \( q_0 \xrightarrow{\mu \alpha \nu} q_3 \). Yet, in this case, the !\( \nu \)-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 \( q_2 \xrightarrow{\nu} \). 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 20 Let \( G = \langle \Sigma, Q, \rightarrow, Q^o \rangle \) be an automaton and \( \Upsilon \subseteq \Sigma \). A path

\[
x_0 \xrightarrow{\sigma_1} x_1 \xrightarrow{\sigma_2} \cdots \xrightarrow{\sigma_k} x_k
\]

is a weakly controllable \( \Upsilon \)-path if \( \sigma_1, \ldots, \sigma_k \in \Upsilon \) and for all uncontrollable transitions \( x_l \xrightarrow{\nu} y \) with \( 0 \leq l < k \) and \( \nu \in \Sigma_u \) it holds that \( \nu \in \Upsilon \) 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 21 Let \( G = \langle \Sigma, Q, \rightarrow, Q^o \rangle \) and \( H = \langle \Sigma, Q, \rightarrow_H, Q^o \rangle \) be two automata with \( \Sigma = \Omega \cup \Upsilon \) and \( \rightarrow_H \subseteq \rightarrow_G \). Automaton \( H \) is a result of controllable suffix-redundant transition removal from \( G \) with respect to \( \Upsilon \), if the following conditions hold for all transitions \( x \xrightarrow{\sigma} y \) from \( G \).

(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 \( x \xrightarrow{P_1(\sigma)}_H z \xrightarrow{u}_H y \), and \( z \xrightarrow{u}_G y \) is a weakly controllable \( \Upsilon \)-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 \( \Upsilon \)-paths.

**Lemma 15** Let \( G = \langle \Sigma, Q_G, \rightarrow_G, Q^0_G \rangle \) and \( T = \langle \Sigma_T, Q_T, \rightarrow_T, Q^0_T \rangle \) be automata, and let \( \Upsilon \subseteq \Sigma \setminus \Sigma_T \). Furthermore, let \( x \xrightarrow{s} y \) be a weakly controllable \( \Upsilon \)-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
\[
(x, x^T) = (x_0, x^T) \xrightarrow{\sigma_1} G||T (x_1, x^T) \xrightarrow{\sigma_2} G||T \cdots \xrightarrow{\sigma_k} G||T (x_k, x^T) = (y, x^T). 
\] (23)

It remains to be shown that this path is in \( \hat{\Theta}_{G||T} \). Let \( Y_T = \Sigma_u \cap (\Sigma_T \setminus \Sigma) \) and
\[
Y_T = \{ y^T \in Q_T \mid x^T \xrightarrow{u} y^T \text{ for some } u \in (\Upsilon_u^T)^* \}. 
\] (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 \((x_j, y^T) \in \Theta_{G||T}^\text{cont}(X^n) \), let \( u \in \Sigma_u \) and \((x_j, y^T) \xrightarrow{u} G||T (z, z^T) \). If \( z \in \Sigma \), then since \( x_0 \xrightarrow{s} G x_k \) is a weakly controllable \( \Upsilon \)-path, it must hold that \( u \in \Upsilon \) and \( x_j \xrightarrow{u} G x_j \) for some \( 0 \leq l \leq k \). This implies \( y^T = z^T \) and \((x_j, y^T) \xrightarrow{u} G||T (x_l, y^T) = (x_l, y^T) \in X^n \) by inductive assumption. If \( u \notin \Sigma \), then \( u \in \Sigma_T \setminus \Sigma \) and \( z = x_j \) and \( y_T \xrightarrow{u_T} z^T \). 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 \( u \in \Sigma_u \), it follows that \((x_j, y^T) \in \Theta_{G||T}^\text{cont}(X^n) \).

Next, it is shown that \((x_j, y^T) \in \Theta_{G||T}^\text{nonb}(X^n) \). As \( \sigma_j+1, \ldots, \sigma_k \in \Upsilon \) and \( \Sigma_T \cap \Upsilon = \emptyset \), it holds by inductive assumption that,
\[
(x_j, y^T) \xrightarrow{\sigma_{j+1}}_{X^n} \cdots \xrightarrow{\sigma_k}_{X^n} (x_k, y^T) = (y, y^T). 
\] (25)

Since \( y^T \in Y_T \), there exists \( u \in (\Upsilon_u^T)^* \) such that \( x^T \xrightarrow{u} y^T \), and this implies \((y, x^T) \xrightarrow{u} G||T (y, x^T) \). 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 \(\tau \in \Sigma^*\) such that \((y, y^T) \xrightarrow{\tau} \hat{\Theta}_{G|T}\), and as \(\hat{\Theta}_{G|T} \subseteq X^n\) it follows that

\[(x_j, y^T) \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_{h}} X^n (x_k, y^T) = (y, y^T) \xrightarrow{\tau} X^n . \quad (26)\]

This implies \((x_j, y^T) \in \Theta_{H||T}^{nonb}(X^n)\).

\[\text{Lemma 16} \quad \text{Let} \quad H = \langle \Sigma, Q, \rightarrow_H, Q^0 \rangle \quad \text{be a result of controllable suffix-redundant transition removal from} \quad G = \langle \Sigma, Q, \rightarrow_G, Q^0 \rangle \quad \text{with respect to} \quad \Upsilon \subseteq \Sigma. \quad \text{Then} \quad G \quad \text{and} \quad H \quad \text{are state-wise synthesis equivalent with respect to} \quad \Upsilon.\]

\[\text{Proof.} \quad \text{Let} \quad \Sigma = \Omega \cup \Upsilon, \quad \text{and let} \quad T = \langle \Sigma_T, Q, \rightarrow, Q^0 \rangle \quad \text{be an automaton such that} \quad \Sigma_T \cap \Upsilon = \emptyset. \quad \text{It is to be shown that} \quad \hat{\Theta}_{G||T} = \hat{\Theta}_{H||T}.\]

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

\[\text{Base case.} \quad \text{Clearly} \quad \hat{\Theta}_{G||T} \subseteq Q \times Q_T = \Theta_{H||T}(Q \times Q_T) = X^n_H.\]

\[\text{Inductive step.} \quad \text{Assume} \quad \hat{\Theta}_{G||T} \subseteq X^n_H \quad \text{for some} \quad n \geq 0, \quad \text{and let} \quad (x, x_T) \in \hat{\Theta}_{G||T}. \quad \text{It remains to be shown that} \quad (x, x_T) \in X^{n+1}_H = \Theta_{H||T}(X^n_H) = \Theta_{H||T}(X^n_H) \cap \Theta_{H||T}^{nonb}(X^n_H).\]

To see that \((x, x_T) \in \Theta_{H||T}^{non}(X^n_H)\), let \(v \in \Sigma_n\) and \((x, x_T) \xrightarrow{v} \hat{\Theta}_{H||T} (y, y_T)\).

Since \(\rightarrow_H \subseteq \rightarrow_G\), it follows that \((x, x_T) \xrightarrow{v} \hat{\Theta}_{G||T} (y, y_T)\). Since \((x, x_T) \in \hat{\Theta}_{G||T}\) and \(v \in \Sigma_n\), it follows by controllability and by inductive assumption that \((y, y_T) \in \hat{\Theta}_{G||T} \subseteq X^n_H\), and since \(v \in \Sigma_n\) was chosen arbitrarily, it follows that \((x, x_T) \in \Theta_{H||T}(X^n_H)\).

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

\[(x, x_T) = (x_0, x_0^T) \xrightarrow{\sigma_1} \hat{\Theta}_{G||T} (x_k, x_k^T) \xrightarrow{\tau} \hat{\Theta}_{G||T} (x_{k+1}, x_{k+1}^T). \quad (27)\]

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 \(x_{l-1} \xrightarrow{\sigma_l} \hat{\Theta}_{G||T} x_l\), then clearly \((x_{l-1}, x_{l-1}^T) \xrightarrow{\sigma_l} \hat{\Theta}_{H||T} (x_l, x_l^T)\), and by inductive assumption \((x_l, x_l^T) \in \hat{\Theta}_{G||T} \subseteq X^n_H\), i.e., \((x_{l-1}, x_{l-1}^T) \xrightarrow{\sigma_l} X^n_H (x_l, x_l^T)\) and \((x_{l-1}, x_{l-1}^T) \xrightarrow{P_T(\sigma_l)P_H(\sigma_l)} X^n_H (x_l, x_l^T)\). Otherwise \(x_{l-1} \xrightarrow{\sigma_l} \hat{\Theta}_{G||T} x_l\) is a controllable suffix-redundant transition, and by definition 21 there exists \(u \in \Upsilon^*\)
such that \( x_{l-1} \xrightarrow{P_1(\sigma_l)} H \xrightarrow{u_l} x_l \) where \( z_l \xrightarrow{u_l} G x_l \) is a weakly controllable \( \Upsilon \)-path. Since \( \Sigma_T \cap \Upsilon = \emptyset \), it follows that \( (x_{l-1}, x_{l-1}^T) \xrightarrow{P_1(\sigma_l)} H || T \xrightarrow{u_l} (x_l, x_l^T) \). Since \( (x_l, x_l^T) \in \hat{\Theta}_{||T} \) it follows by lemma 15 that \( (z_l, x_l^T) \xrightarrow{u_l} \hat{\Theta}_{||T} (x_l, x_l^T) \). Since also \( (x_{l-1}, x_{l-1}^T) \in \hat{\Theta}_{||T} \) follows that \( (x_{l-1}, x_{l-1}^T) \xrightarrow{P_1(\sigma_l)} \hat{\Theta}_{||T} (x_l, x_l^T) \), and thus \( (x_{l-1}, x_{l-1}^T) \xrightarrow{P_1(\sigma_l)} H || T (x_l, x_l^T) \) by inductive assumption. Combining these paths for all transitions in (27) gives traces \( u_1, \ldots, u_k \in \Upsilon^* \) such that

\[
(x, x_T) = (x_0, x_0^T) \xrightarrow{P_1(\sigma_1) u_1} (x_1, x_1^T) \cdots \xrightarrow{P_1(\sigma_k) u_k} (x_k, x_k^T) \xrightarrow{\omega} (x_{k+1}, x_{k+1}^T),
\]

which implies \( (x, x_T) \in \Theta^\text{conf}_{||T}(X^n_T) \cap \Theta^\text{nonb}_{||T}(X^n_T) = X^n_{||T}. \)

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

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

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

To see that \( (x, x_T) \in \Theta_{||T}^\text{cont}(X^n_G) \), let \( u \in \Sigma_u \) and \( (x, x_T) \xrightarrow{u} (y, y_T) \).

If \( u \notin \Sigma \), then clearly \( (x, x_T) \xrightarrow{u} (y, y_T) \). Otherwise, since \( u \in \Sigma_u \), the transition \( x \xrightarrow{u} y \) cannot be controllable suffix-redundant, which also implies \( (x, x_T) \xrightarrow{u} (y, y_T) \). Since \( (x, x_T) \in \hat{\Theta}_{||T} \) and \( u \in \Sigma_u \), it follows that \( (y, y_T) \in \hat{\Theta}_{||T} \subseteq X^n_G \) by inductive assumption, and thus \( (x, x_T) \in \Theta_{||T}^\text{cont}(X^n_G) \).

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

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

**Lemma 17** Let \( H = (\Sigma, Q, \rightarrow_H, Q^0) \) be a result of controllable suffix-redundant transition removal from \( G = (\Sigma, Q, \rightarrow_G, Q^0) \) with respect to \( \Upsilon \subseteq \Sigma \). Then \( G \) and \( H \) are transition-wise synthesis equivalent with respect to \( \hat{\Upsilon} \).
Proof. It must be shown that \( G \) is a transition-wise synthesis abstraction of \( H \) and vice versa. Condition (i) in definition 13 follows immediately from definition 21. To show condition (ii), let \( \Sigma = \Omega \cup \Upsilon \), and let \( T = (\Sigma_T, Q, \rightarrow, Q^0) \) be an automaton such that \( \Sigma_T \cap \Upsilon = \emptyset \).

First, let \( (x, x^T) \xrightarrow{\sigma}_{H} (y, y^T) \). By lemma 16 it holds that \( (x, x^T), (y, y^T) \in \hat{\Theta}_{H||T} \). Since \( \sigma \notin \Sigma \) 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 \( (x, x^T) \xrightarrow{P_H(\sigma)P_{\Upsilon}(\sigma)}_{\hat{\Theta}_{H||T}} (y, y^T) \). Otherwise \( x \xrightarrow{\sigma}_G y \) is a controllable suffix-redundant transition, and by definition 21, there exists \( u \in \Upsilon^* \) such that \( x \xrightarrow{P_{\Upsilon}(\sigma)}_H z \xrightarrow{u}_H y \) where \( z \xrightarrow{u}_G y \) is a weakly controllable \( \Upsilon \)-path.

Since \( \Sigma_T \cap \Upsilon = \emptyset \), it follows that \( (x, x^T) \xrightarrow{P_{\Upsilon}(\sigma)}_{H||T} (z, y^T) \xrightarrow{u}_{H||T} (y, y^T) \). 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 \( (x, x^T) \xrightarrow{P_{\Upsilon}(\sigma)u}_{\hat{\Theta}_{G||T}} (y, y^T) \) and thus \( (x, x^T) \xrightarrow{tP_{\Upsilon}(\sigma)u}_{\hat{\Theta}_{H||T}} (y, y^T) \). Thus, in both cases, there exist \( t = \varepsilon \) and \( u \in \Upsilon^* \) such that \( (x, x^T) \xrightarrow{tP_{\Upsilon}(\sigma)u}_{\hat{\Theta}_{H||T}} (y, y^T) \).

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 \( (x, x^T) \xrightarrow{tP_{\Upsilon}(\sigma)u}_{\hat{\Theta}_{G||T}} (y, y^T) \).

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

Theorem 18 Let \( H = (\Sigma, Q, \rightarrow_H, Q^0) \) be a result of controllable suffix-redundant transition removal from \( G = (\Sigma, Q, \rightarrow_G, Q^0) \) with respect to \( \Upsilon \subseteq \Sigma \). Then \( G \) and \( H \) are state-wise and transition-wise synthesis equivalent with respect to \( \Upsilon \).

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 22 Let \( G = (\Sigma, Q, \rightarrow_G, Q^0) \) and \( H = (\Sigma, Q, \rightarrow_H, Q^0) \) be two automata with \( \Sigma = \Omega \cup \Upsilon \) and \( \rightarrow_H \subseteq \rightarrow_G \). Automaton \( H \) is a result of controllable redundant transition removal from \( G \) with respect to \( \Upsilon \), if the following conditions hold for all transitions \( x \xrightarrow{\sigma}_G y \).

(i) If \( \sigma \in \Sigma_u \) then \( x \xrightarrow{\sigma}_H y \).
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 \( y \xrightarrow{u} G y \) is a weakly controllable \( \Upsilon \)-path.

**Theorem 19** Let \( H = \langle \Sigma, Q, \rightarrow_H, Q^0 \rangle \) be a result of controllable redundant transition removal from \( G = \langle \Sigma, Q, \rightarrow_G, Q^0 \rangle \) with respect to \( \Upsilon \subseteq \Sigma \). Then \( G \) and \( H \) are state-wise and transition-wise synthesis equivalent with respect to \( \Upsilon \).

**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 \( \rightarrow_G = \rightarrow_H \cup \{x, \sigma, y\} \) where \( x \xrightarrow{\sigma} y \) is a controllable redundant transition.

As \( x \xrightarrow{\sigma} y \) is a controllable redundant transition, there exists a redirection path \( x \xrightarrow{t} z_1 \xrightarrow{P_\Omega(\sigma)} H z_2 \xrightarrow{t} H y \) where \( t \in (\Sigma_u \cap \Upsilon)^* \) and \( z_2 \xrightarrow{\sigma} y \) is a weakly controllable \( \Upsilon \)-path. Consider automata \( G' = \langle \Sigma, Q, \rightarrow_{G'}, Q^0 \rangle \) with \( \rightarrow_{G'} = \rightarrow_G \cup \{x, \sigma, z_2\} \) and \( H' = \langle \Sigma, Q, \rightarrow_{H'}, Q^0 \rangle \) with \( \rightarrow_{H'} = \rightarrow_H \cup \{x, \sigma, z_2\} \).

Since \( x \xrightarrow{\sigma} z_1 \xrightarrow{P_\Omega(\sigma)} G z_2 \), 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 \( x \xrightarrow{\sigma} z_2 \xrightarrow{t} H' y \), it holds that \( x \xrightarrow{\sigma} H' y \) 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* [13] 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 23** Let \( G = \langle \Sigma, Q, \rightarrow_G, Q^0 \rangle \) and \( H = \langle \Sigma, Q, \rightarrow_H, Q^0 \rangle \) be two automata with \( \Sigma = \Omega \cup \Upsilon \) and \( \rightarrow_H \subseteq \rightarrow_G \). Automaton \( H \) is a result of *local selfloop removal* from \( G \) with respect to \( \Upsilon \), if for all transitions \( x \xrightarrow{\sigma} y \) 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 $H = (\Sigma, Q, \to_H, Q^0)$ be a result of local selfloop removal from $G = (\Sigma, Q, \to_G, Q^0)$ with respect to $\Upsilon \subseteq \Sigma$. Then $G$ and $H$ are state-wise and transition-wise synthesis equivalent with respect to $\Upsilon$.

**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 $\to_G = \to_H \cup (x, \sigma, x)$ 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 $t \Pi_1(\sigma) u = \varepsilon$ 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 $t \Pi_3(\sigma) = \varepsilon$ and $x \xrightarrow{\varepsilon} G x$, so $x \xrightarrow{\sigma} G x$ is a controllable prefix-redundant transition. The claim follows from theorem 14. \qed

**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 $B_1$ ($l f_1$), and then they go to $M_2$ ($s_2$) and are placed into $B_2$ ($l f_2$). 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 $B_1$ ($l r e$). At any time, $M_1$ and $B_1$ can be reset by the controllable event $r s$. Figure 6 shows the system layout and the automata model. Events $l f_1, l f_2, l f_3, l f_4$ and $l r e$ are uncontrollable, all other events are controllable.

Compositional synthesis starts with the pair $(G_0; \id)$ where $G_0 = \{M_1, M_2, M_3, M_4, B_1, B_2\}$. The first step is to calculate the composition $B_1 || M_1$ shown in figure 7. Now $l f_1, r s$, and $s_1$ are local events, which makes $q_0 \xrightarrow{r s} q_0$ a local selfloop and $q_2 \xrightarrow{r s} q_0$ a controllable prefix-redundant transition with redirection path $q_2 \xrightarrow{r s} q_3 \xrightarrow{r s} q_0$. Removal of these transitions results in $H_1$. The modified synthesis pair is $(G_1; D_1)$ where $G_1 = \{H_1, M_2, M_3, M_4, B_2\}$ and $D_1$ is a synthesis preserving redirection map that redirects $q_2 \xrightarrow{r s} q_0$ and $q_0 \xrightarrow{r s} q_0$ via $q_2 \xrightarrow{r s} q_3 \xrightarrow{r s} q_0$ and $q_0 \xrightarrow{r s} q_0$, respectively.

Next, $B_2 || M_3$ is computed, shown in figure 7. This makes $l f_3$ and $s_3$ local events, and $q_3 \xrightarrow{l f_3} \bot$ becomes an uncontrollable redundant transition with redirection path $q_3 \xrightarrow{l f_3} q_1 \xrightarrow{l f_2} \bot$. The new synthesis pair is $(G_2; D_2 \circ D_1)$ where $G_2 = \{H_1, M_2, M_4, H_2\}$ and $D_2$ is a synthesis preserving redirection map which
Figure 6: Manufacturing system example.

Figure 7: Some subsystems of the manufacturing example. The transitions to be removed are marked by double-line strike-through.
redirects $q_3 \xrightarrow{\ell_2} \bot$ via $q_3 \xrightarrow{\ell_3} q_1 \xrightarrow{\ell_2} \bot$.

The final synthesis step to compute $\supCN(G_2)$ explores the state space of $G_2$ 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 [6, 14] and transitions, as well as renaming [13] to remove nondeterminism.

References


