Online Resource 1 - Proofs of theorems in "Compositional coordinator synthesis of extended finite automata"

Martijn A. Goorden, Martin Fabian,
Joanna M. van de Mortel-Fronczak, Michel A. Reniers,
Wan J. Fokkink, and Jacobus E. Rooda

This supplementary document provides a description of all ten abstraction-refinement pairs and the proofs of the theorems mentioned in manuscript Martijn A. Goorden[1], Martin Fabian, Joanna M. van de Mortel-Fronczak, Michel A. Reniers, Wan J. Fokkink, and Jacobus E. Rooda, "Compositional coordinator synthesis of extended finite automata", submitted to Journal of Discrete Event Dynamical Systems. This supplementary material should be read in conjunction with the paper, as mathematical preliminaries are presented in the paper.

Below an overview is given of the abstraction-refinement pairs. The numbers refer to the sections in this document.

Normalization

1. Local normalization
2. Global normalization

CE abstractions

3. FA-based abstractions
4. Partial composition
5. Update simplification
6. Variable unfolding
7. False removal
8. Self-loop removal
9. Event merging
10. Update merging

Remarks The maximally permissive supervisor $\sup CN(G)$ can be calculated by the fixed-point algorithm SSEFA as presented in Ouedraogo et al. [2011]. For convenience, this algorithm is shown in Algorithm 1. It is proven in that paper that $\text{SSEFA}(G) = \sup CN(G)$.

This algorithm uses nonblocking predicates and bad state predicates. Predicates are, like guards, Boolean expressions that evaluate to true or false for a

[1] Corresponding author: Eindhoven University of Technology, m.a.goorden@tue.nl

given valuation, i.e., $\hat{v} \vDash p$ or $\hat{v} \nvDash p$. In Algorithm 1 we use a standard notation for each edge: $e = (o_e, \sigma_e, g_e, u_e, t_e) \in E$, where o_e represents the origin location of edge e, σ_e the event label, g_e the guard, u_e the update, and t_e the terminal location. Furthermore, $N[u_e]$ and $B[u_e]$ represents the substitution of the update expressions in the nonblocking and bad location predicates, respectively. This can be best explained with an example. Let $N \equiv v_1 > v_2 + 2$ be a nonblocking predicate expressing that variable v_1 should be larger than $v_2 + 2$, and $u = \{v_1 \mapsto v_1 + 1, v_2 \mapsto v_2 - v_1\}$ the update that increases the value of v_1 by 1 and sets the new value of v_2 as the difference between the current values of v_2 and v_1. Substituting this update in the predicate N results in $N[u] \equiv v_1 + 1 > v_2 - v_1 + 2$ (which may be simplified into $N[u] \equiv 2 \cdot v_1 > v_2 + 1$).

Algorithm 1 Supervisory Synthesis for EFA (SSEFA)

```
Require: EFA $G = (L, V, \Sigma, E, l_0, v_0, L_m)$
Ensure: SSEFA($G$) is the supremal controllable and nonblocking subautomaton of $G$, if SSEFA($G$) is nonblocking and controllable
 1: Initialize iterators: $i := 0, j := 0, k := 0$
 2: Initialize guards: $\forall e \in E : g_e^0 = g_e$
 3: Initialize the nonblocking predicate of every location $l \in L$:
        $$N_l^{j,0} = \begin{cases} \mathbf{T}, & \text{if } l \in L_m \\ \mathbf{F}, & \text{if } l \notin L_m \end{cases}$$
 4: Update the nonblocking predicate of every location $l \in L$:
        $$N_l^{j,k+1} = N_l^{j,k} \vee \bigvee_{\{e | o_e = l\}} \left[ g_e^j \wedge N_{t_e}^{j,k}[u_e] \right]$$
 5: if there exists an $l \in L$ such that $N_l^{j,k} \neq N_l^{j,k+1}$ then
 6:     $k := k + 1$
 7:     Go to 4
 8: else
 9:     for all $l \in L$: $N_l^j = N_l^{j,k}$
10:     $k := 0$
11: end if
12: Initialize the bad location predicate of every location $l \in L$:
        $$B_l^{j,0} = \begin{cases} \mathbf{T}, & \text{if } l \in L_f \\ \neg N_l^j, & \text{if } l \notin L_f \text{ and } j = 0 \\ \neg N_l^j \vee B_l^{j-1}, & \text{if } l \notin L_f \text{ and } j > 0 \end{cases}$$
13: Update the bad location predicate of every location $l \in L$:
        $$B_l^{j,i+1} = B_l^{j,i} \vee \bigvee_{\{e | o_e = l, \sigma_e \in \Sigma_u\}} \left[ g_e^j \wedge B_{t_e}^{j,i}[u_e] \right]$$
14: if there exists an $l \in L$ such that $B_l^{j,i} \neq B_l^{j,i+1}$ then
15:     $i := i + 1$
16:     Go to 13
17: else
18:     for all $l \in L$: $B_l^j = B_l^{j,i}$
19:     $i := 0$
20: end if
21: Update the guard of every edge $e \in E$:
        $$g_e^{j+1} = \begin{cases} g_e^j \wedge \neg B_{t_e}^j[u_e], & \text{if } \sigma \in \Sigma_c \\ g_e^j, & \text{if } \sigma \in \Sigma_u \end{cases}$$
22: if there exists an $l \in L$ such that $g_e^{j+1} \neq g_e^j$ then
23:     $j := j + 1$
24:     Go to 3
25: else
26:     Stop
27: end if
```

1 Local normalization

Local normalization makes sure that within a single automaton, transitions labeled with the same event have the same effect on the valuations of variables, i.e., each event can be associated with a guard and update. When an automaton is locally normalized, guards and updates are no longer solely associated with transitions, but are also associated with the event.

Lemma 1. *Let E be a deterministic EFA and let $\rho : \Sigma' \to \Sigma$ be a renaming function. Create F such that $\rho(F) = E$. Then $SSEFA(E) = \rho(SSEFA(F))$.*

Proof. From the definition of renaming, it follows that E and F have the same location set, same set of variables, same initial location, same initial valuation, and same set of marked locations. Furthermore, as $\rho(F) = E$, for each edge $e_E = (l_1, \sigma, g, u, l_2) \in E_E$ there exists an edge $e_F \in E_F$ in F such that $e_F = (l_1, \mu, g, u, l_2)$ and $\rho(\mu) = \sigma$, and for each edge $e_F = (l_1, \mu, g, u, l_2) \in E_F$ there exists an edge $e_E \in E_E$ in E such that $e_E = (l_1, \sigma, g, u, l_2)$ and $\sigma = \rho(\mu)$.

Now, consider SSEFA, Algorithm 1. First, for any EFA A, A and SSEFA(A) only differ in the set of edges; all other elements of the EFA tuple are the same. Second, as renaming preserves by definition the controllability status of an event, for each iteration j, the nonblocking predicates and the bad location predicates do not depend on the event name of edges. Therefore, these predicates are the same for E and F. This shows us that, eventually, for every pair of edges $e_E = (l_1, \sigma, g, u, l_2) \in E_E$ and $e_F = (l_1, \mu, g, u, l_2) \in E_F$ with $\sigma = \rho(\mu)$, the fixed-point guard g^* for e_E and e_F are the same, i.e., $(l_1, \sigma, g^*, u, l_2)$ is an edge in SSEFA(E) if and only if (l_1, μ, g^*, u, l_2) is an edge in SSEFA(F).

Finally, when we apply renaming on SSEFA(F) it follows from the definition that (l_1, μ, g^*, u, l_2) is an edge in SSEFA(F) if and only if $(l_1, \rho(\mu), g^*, u, l_2) = (l_1, \sigma, g^*, u, l_2)$ is an edge in ρ(SSEFA(F)). Furthermore, the alphabet of E is the same as ρ(SSEFA(F)). Combining this with the conclusion of the previous paragraph, we can conclude that $(l_1, \sigma, g^*, u, l_2)$ is an edge in SSEFA(E) if and only if $(l_1, \sigma, g^*, u, l_2)$ is an edge in ρ(SSEFA(F)). This concludes the proof. □

Lemma 2. *Let $\mathcal{E} = \{E^1, \ldots, E^n\}$ be a deterministic EFA system, and let $\rho : \Sigma' \to \Sigma_E$ be a renaming function such that $\mathcal{F} = \{F^1, \rho^{-1}(E^2), \ldots, \rho^{-1}(E^n)\}$, $\rho(F^1) = E^1$, and F^1 is a normalized EFA. Then $\mathcal{E} = \rho(\mathcal{F})$.*

Proof. From the definition of renaming and inverse renaming it follows that for any EFA G it holds that $\rho(\rho^{-1}(G)) = G$. As renaming applied on an EFA system is defined as applying renaming on the individual EFAs, it follows that $\rho(\mathcal{F}) = \{\rho(F^1), \rho(\rho^{-1}(E^2)), \ldots, \rho(\rho^{-1}(E^n))\} = \{E^1, E^2, \ldots, E^n\} = \mathcal{E}$. This concludes the proof. □

Theorem 1. *Let $(\mathcal{E}, \xi_1)$ be a coordinator tuple with $\mathcal{E} = \{E^1, \ldots, E^n\}$ a deterministic EFA system, and let $\rho : \Sigma' \to \Sigma_{\mathcal{E}}$ be a renaming function such that $\mathcal{F} = \{F^1, \rho^{-1}(E^2), \ldots, \rho^{-1}(E^n)\}$, $\rho(F^1) = E^1$, and F^1 is a normalized EFA. Then refinement function $\xi(\mathcal{G}) = \rho(\mathcal{G})$ for any EFA $\mathcal{G}$ ensures that $(\mathcal{E}, \xi_1) \simeq_{co} (\mathcal{F}, \xi_1 \circ \xi)$.*

Proof. From the definition of Ξ and the construction of ξ, it follows directly that $\xi \in \Xi$. Therefore, $\xi_1 \circ \xi \in \Xi$ and $(\mathcal{F}, \xi_1 \circ \xi)$ is a coordinator tuple.

Now we show that the two coordinator tuples are coordinator equivalent. From Lemma 2 it follows that $\mathcal{E} = \rho(\mathcal{F})$. Therefore, from Lemma 1 it follows that $\sup CN(\mathcal{E}) = \text{SSEFA}(\mathcal{E}) = \rho(\text{SSEFA}(\mathcal{F})) = \rho(\sup CN(\mathcal{F}))$. Therefore, $\mathcal{L}(\xi_1(\sup CN(\mathcal{E}))) = \mathcal{L}(\xi_1(\rho(\sup CN(\mathcal{F})))) = \mathcal{L}(\xi_1(\xi(\sup CN(\mathcal{F}))))$. □

2 Global normalization

Definition 1. *Let $A = (L_A, \Sigma_A, V_A, E_A, l_{0,A}, \hat{v}_{0,A}, L_{m,A})$ and $B = (L_B, \Sigma_B, V_B, E_B, l_{0,B}, \hat{v}_{0,B}, L_{m,B})$ be two EFAs. A and B are said to be logically equivalent with respect to variable set V, written $A \Leftrightarrow_V B$, if $L_A = L_B$, $\Sigma_A = \Sigma_B$, $V_A = V_B$, $l_{0,A} = l_{0,B}$, $\hat{v}_{0,A} = \hat{v}_{0,B}$, and $L_{m,A} = L_{m,B}$, and $e_A = (l_1, \sigma, g_A, u, l_2) \in E_A$ is an edge in A if and only if $e_B = (l_1, \sigma, g_B, u, l_2) \in E_B$ is an edge in B such that $g_A \Leftrightarrow_V g_B$.*

Lemma 3. *Let E and F be two deterministic EFAs such that $E \Leftrightarrow_V F$. Then SSEFA$(E) \Leftrightarrow_V$ SSEFA(F).*

Proof. As E and F are logically equivalent, and Algorithm 1 only alters guards on edges, it follows that SSEFA(E) and SSEFA(F) have the same location set, alphabet, variables, initial location, initial valuation, and set of marked locations, and $(l_1, \sigma, g_E, u, l_2)$ is an edge in SSEFA(E) if and only if $(l_1, \sigma, g_F, u, l_2)$ is an edge in SSEFA(F). It remains to be proven that $g_E \Leftrightarrow_V g_F$.

In the remainder of this proof, we use the notation ${}^E x$ to refer to usage of some symbol x in EFA E, while ${}^F x$ refers to the usage of some symbol x in EFA F.

Consider the first iteration of Algorithm 1, i.e., $j = 0$. From Line 2 we can observe that for all edges $e = (l_1, \sigma, *, u, l_2)$ with $(l_1, \sigma, {}^E g, u, l_2) \in E$ and $(l_1, \sigma, {}^F g, u, l_2) \in F$ it holds that ${}^E g_e^0 = {}^E g$ and ${}^F g_e^0 = {}^F g$. Therefore, ${}^E g_e^0 \Leftrightarrow_V {}^F g_e^0$.

Continuing with the nonblocking predicates, we observe that the initial nonblocking predicate for each location as defined in Line 3 does not depend on any guard. Therefore, for all locations $l \in L$ it holds that ${}^E N_l^{0,0} = {}^F N_l^{0,0}$. It then follows from Line 4 that for all locations $l \in L$: ${}^E N_l^{0,k+1} \Leftrightarrow_V {}^F N_l^{0,k+1}$. Thus, for all locations $l \in L$ we can conclude that ${}^E N_l^0 \Leftrightarrow_V {}^F N_l^0$.

Continuing with the bad location predicates, we observe from Line 12 that for all locations $l \in L$ it holds that initially ${}^E B_l^{0,0} \Leftrightarrow_V {}^F B_l^{0,0}$. It then follows from Line 13 that for all locations $l \in L$: ${}^E B_l^{0,k+1} \Leftrightarrow_V {}^F B_l^{0,k+1}$. Thus, for all locations $l \in L$ we can conclude that ${}^E B_l^0 \Leftrightarrow_V {}^F B_l^0$.

Finally, continuing with the update of the guards in Line 21, we can conclude that for all edges e it holds that ${}^E g_e^1 \Leftrightarrow_V {}^F g_e^1$.

When the algorithm goes back to Line 3 for the next iteration, we can repeat the argumentation above for $j > 0$ to conclude after each iteration that ${}^E g_e^j \Leftrightarrow_V {}^F g_e^j$. Therefore, when the fixed-point is reached after n iterations, it follows that for all edges e it holds that ${}^{E}g_e^n \Leftrightarrow_V {}^{F}g_e^n$. This concludes the proof. □

Lemma 4. *Let E and F be two deterministic EFAs such that $E \Leftrightarrow_V F$. Then, for any EFA T it holds that $U(E \parallel T) = U(F \parallel T)$.*

Proof. Clearly, $U(E \parallel T)$ and $U(F \parallel T)$ have the same state set, alphabet, initial location, and marked locations. It remains to be proven that they have the same transitions. Because of symmetry of E and F in the lemma it is enough to show that, if $((l_1^E, l_1^T), \hat{v}_1) \xrightarrow{\sigma} ((l_2^E, l_2^T), \hat{v}_2)$ is a transition in $U(E \parallel T)$, then $((l_1^E, l_1^T), \hat{v}_1) \xrightarrow{\sigma} ((l_2^E, l_2^T), \hat{v}_2)$ is a transition in $U(F \parallel T)$.

Assume that $((l_1^E, l_1^T), \hat{v}_1) \xrightarrow{\sigma} ((l_2^E, l_2^T), \hat{v}_2)$ in $U(E \parallel T)$. By the definition of state space this means that $(l_1^E, l_1^T) \xrightarrow{\sigma,g,u} (l_2^E, l_2^T)$ in $E \parallel T$ such that $g[\hat{v}_1] = \mathbf{T}$ and $\hat{v}_2(v) = \hat{v}_1(u(v)))$. Consider three cases for σ.

- $\sigma \in \Sigma \cup \Sigma^T$. Then by the definition of synchronous composition it follows that $l_1^E \xrightarrow{\sigma,g^E,u^E} l_2^E$ in E, $l_1^T \xrightarrow{\sigma,g^T,u^T} l_2^T$ in T, $g = g^E \wedge g^T$, and $u = u^E \oplus u^T$. As F is logically equivalent to E, it follows that $l_1^F \xrightarrow{\sigma,g^F,u^F} l_2^F$ in F where $l_1^F = l_1^E, l_2^F = l_2^E, u^F = u^E$, and $g^F \Leftrightarrow_V g^E$. Finally, as $g[\hat{v}_1] = \mathbf{T}$ and $g = g^E \wedge g^T$, it follows that $g^E[\hat{v}_1] = \mathbf{T}$ and $g^T[\hat{v}_1] = \mathbf{T}$. Together with $g^F \Leftrightarrow_V g^E$ it holds that $g^F[\hat{v}_1] = \mathbf{T}$.

- $\sigma \in \Sigma \setminus \Sigma^T$. Then by the definition of synchronous composition it follows that $l_1^E \xrightarrow{\sigma,g^E,u^E} l_2^E$ in E, $l_1^T = l_2^T$, $g = g^E$, and $u = u^E$. As F is logically equivalent to E, it follows that $l_1^F \xrightarrow{\sigma,g^F,u^F} l_2^F$ in F where $l_1^F = l_1^E, l_2^F = l_2^E, u^F = u^E$, and $g^F \Leftrightarrow_V g^E$. Finally, as $g[\hat{v}_1] = \mathbf{T}, g = g^E$, and $g^F \Leftrightarrow_V g^E$ it holds that $g^F[\hat{v}_1] = \mathbf{T}$.

- $\sigma \in \Sigma^T \setminus \Sigma$. Then by the definition of synchronous composition it follows that $l_1^T \xrightarrow{\sigma,g^T,u^T} l_2^T$ in T, $l_1^E = l_2^E$, $g = g^T$, and $u = u^T$. As F is logically equivalent to E, it follows that there is no transition in F, i.e., $l_1^F = l_1^E$ and $l_2^F = l_2^E$.

Using the definition of the synchronous product on the situations described above, it follows that $(l_1^F, l_1^T) \xrightarrow{\sigma,g,u} (l_2^F, l_2^T)$ in $F \parallel T$ such that $g[\hat{v}_1] = \mathbf{T}$ and $\hat{v}_2(v) = \hat{v}_1(u(v))$. Therefore, $((l_1^F, l_1^T), \hat{v}_1) \xrightarrow{\sigma} ((l_2^F, l_2^T), \hat{v}_2)$ is a transition in $U(F \parallel T)$, which can be rewritten as $((l_1^E, l_1^T), \hat{v}_1) \xrightarrow{\sigma} ((l_2^E, l_2^T), \hat{v}_2)$ with the observations above. □

Lemma 5. *Let E and F be two deterministic EFAs with shared alphabet Σ such that $E \Leftrightarrow_V F$, and $\rho : \Sigma \to \Sigma'$ a renaming function. Then $\rho(E) \Leftrightarrow_V \rho(F)$.*

Proof. From the definition of renaming and that $E \Leftrightarrow_V F$, it follows that $\rho(E)$ and $\rho(F)$ have the same location set, alphabet, variables, initial location, initial valuation, and set of marked locations.

As $E \Leftrightarrow_V F$, it follows that $(l_1, \sigma, g^E, u, l_2)$ is an edge in E if and only if $(l_1, \sigma, g^F, u, l_2)$ is an edge in F and $g^E \Leftrightarrow_V g^F$. Therefore, after applying the renaming function ρ, we know that $(l_1, \rho(\sigma), g^E, u, l_2)$ is an edge in $\rho(E)$ if and only if $(l_1, \rho(\sigma), g^F, u, l_2)$ is an edge in $\rho(F)$ and $g^E \Leftrightarrow_V g^F$. This concludes the proof. □

Lemma 6. *Let E and F be two deterministic EFAs with shared alphabet Σ such that $E \Leftrightarrow_V F$, and $\rho : \Sigma' \to \Sigma$ a renaming function. Then, it holds that $\rho^{-1}(E) \Leftrightarrow_V \rho^{-1}(F)$.*

Proof. From the definition of inverse renaming and that $E \Leftrightarrow_V F$, it follows that $\rho^{-1}(E)$ and $\rho^{-1}(F)$ have the same location set, alphabet, variables, initial location, initial valuation, and set of marked locations.

As $E \Leftrightarrow_V F$ it follows that $(l_1, \sigma, g^E, u, l_2)$ is an edge in E if and only if $(l_1, \sigma, g^F, u, l_2)$ is an edge in F and $g^E \Leftrightarrow_V g^F$. Therefore, after applying the inverse renaming function ρ^{-1}, we know for all $\mu \in \rho^{-1}(\sigma)$ that (l_1, μ, g^E, u, l_2) is an edge in $\rho(E)$ if and only if (l_1, μ, g^F, u, l_2) is an edge in $\rho(F)$ and $g^E \Leftrightarrow_V g^F$. This concludes the proof. □

Lemma 7. *Let E and F be two deterministic EFAs with shared alphabet Σ such that $E \Leftrightarrow_V F$. Then, for any EFA T it holds that $E \parallel T \Leftrightarrow_V F \parallel T$.*

Proof. From the definition of $E \Leftrightarrow_V F$, it follows that E and F have the same location set, alphabet, variables, initial location, initial valuation, and set of marked locations. Furthermore, from the definition of synchronous product it follows that $E \parallel T$ and $F \parallel T$ have the same location set, alphabet, variables, initial location, initial valuation, and set of marked locations.

As $E \Leftrightarrow_V F$ it follows that $(l_1, \sigma, g^E, u, l_2)$ is an edge in E if and only if $(l_1, \sigma, g^F, u, l_2)$ is an edge in F and $g^E \Leftrightarrow_V g^F$. Consider three cases for event σ.

- $\sigma \in \Sigma \cup \Sigma_T$. In this case $((l_1, x_1), \sigma, g^E \wedge g^T, u \oplus u^T, (l_2, x_2))$ is an edge in $E \parallel T$ if and only if $(l_1, \sigma, g^E, u, l_2)$ is an edge in E and $(x_1, \sigma, g^T, u^T, x_2)$ is an edge in T. Similarly, $((l_1, x_1), \sigma, g^F \wedge g^T, u \oplus u^T, (l_2, x_2))$ is an edge in $F \parallel T$ if and only if $(l_1, \sigma, g^F, u, l_2)$ is an edge in F and $(x_1, \sigma, g^T, u^T, x_2)$ is an edge in T. Observe that $g^E \wedge g^T \Leftrightarrow_V g^F \wedge g^T$,
- $\sigma \in \Sigma \setminus \Sigma_T$. In this case $((l_1, x_1), \sigma, g^E, u, (l_2, x_2))$ is an edge in $E \parallel T$ if and only if $(l_1, \sigma, g^E, u, l_2)$ is an edge in E and $x_1 = x_2$. Similarly, $((l_1, x_1), \sigma, g^F, u, (l_2, x_2))$ is an edge in $F \parallel T$ if and only if $(l_1, \sigma, g^F, u, l_2)$ is an edge in F and $x_1 = x_2$
- $\sigma \in \Sigma_T \setminus \Sigma$. In this case $((l_1, x_1), \sigma, g^T, u^T, (l_2, x_2))$ is an edge in $E \parallel T$ if and only if $(x_1, \sigma, g^T, u^T, x_2)$ is an edge in T and $l_1 = l_2$. Similarly, $((l_1, x_1), \sigma, g^T, u^T, (l_2, x_2))$ is an edge in $F \parallel T$ if and only if $(x_1, \sigma, g^T, u^T, x_2)$ is an edge in T and $l_1 = l_2$.

Combining the observations above, we can conclude that $((l_1, x_1), \sigma, g^{ET}, u \oplus u^T, (l_2, x_2))$ is an edge in $E \parallel T$ if and only if $((l_1, x_1), \sigma, g^{FT}, u \oplus u^T, (l_2, x_2))$ is an edge in $F \parallel T$ and $g^{ET} \Leftrightarrow_V g^{FT}$. This concludes the proof. □

Lemma 8. *Let E and F be two deterministic EFAs with shared alphabet Σ such that $E \Leftrightarrow_V F$, and $\xi \in \Xi$ an abstraction function. Then $\xi(E) \Leftrightarrow_V \xi(F)$.*

Proof. This lemma is proven by induction on the structure of ξ. Denote $\xi = \xi_m \circ \ldots \circ \xi_1$. Assume that $\xi_i \circ \ldots \circ \xi_1(E) \Leftrightarrow_V (\xi_i \circ \ldots \circ \xi_1)(F)$ with $i \in [0 \ldots m-1]$ and $\xi_0 = \mathrm{id}$. Consider the following four cases for ξ_{i+1}.

- ξ_{i+1} is the identity function. It follows immediately that $\xi_{i+1} \circ \xi_i \circ \ldots \circ \xi_1(E) \Leftrightarrow_V \xi_{i+1} \circ \xi_i \circ \ldots \circ \xi_1(F)$.
- ξ_{i+1} is a renaming. From Lemma 5 it follows that $\xi_{i+1} \circ \xi_i \circ \ldots \circ \xi_1(E) \Leftrightarrow_V \xi_{i+1} \circ \xi_i \circ \ldots \circ \xi_1(F)$.
- ξ_{i+1} is a renaming in synchronous composition with the original EFA system. From Lemma 5 and 7 it follows that $\xi_{i+1} \circ \xi_i \circ \ldots \circ \xi_1(E) \Leftrightarrow_V \xi_{i+1} \circ \xi_i \circ \ldots \circ \xi_1(F)$.
- ξ_{i+1} is an inverse renaming in synchronous composition with the original EFA system. From Lemma 6 and 7 it follows that $\xi_{i+1} \circ \xi_i \circ \ldots \circ \xi_1(E) \Leftrightarrow_V \xi_{i+1} \circ \xi_i \circ \ldots \circ \xi_1(F)$.

This concludes the proof. □

Lemma 9. *Let $\mathcal{E} = \{E^1, \ldots, E^n\}$ be a deterministic EFA system, where each individual EFA $E^i \in \mathcal{E}$ is locally normalized. Construct the normalized form of $\mathcal{E}$ as $\mathcal{F} = \mathcal{N}(\mathcal{E}) = \{\mathcal{N}(E^1), \ldots, \mathcal{N}(E^n)\}$. Then $\parallel \mathcal{E} \Leftrightarrow_V \parallel \mathcal{F}$.*

Proof. From the definition of the globally normalization function $\mathcal{N}$, it follows for each index i that E^i and F^i have the same set of locations, same alphabet, same variable set, same initial location, same initial valuation, and same set of marked locations. Therefore, the synchronous products $\parallel \mathcal{E}$ and $\parallel \mathcal{F}$ also have the same set of locations, same alphabet, same variable set, same initial location, same initial valuation, and same set of marked locations.

Furthermore, it holds that there is an edge $(l_1^i, \sigma, g_\sigma^i, u, l_2^i)$ in E^i if and only if there is an edge $(l_1^i, \sigma, g_\sigma, u, l_2^i)$ in F^i where $g_\sigma = \bigwedge_{i:\sigma\in\Sigma^i} g_\sigma^i$ and g_σ^i the guard associated with event σ in normalized EFA E^i. Therefore, in the synchronous product we have an edge $((l_1^1, \ldots, l_1^n), \sigma, \bigwedge_{i:\sigma\in\Sigma^i} g_\sigma^i, u, (l_2^1, \ldots, l_2^n))$ in $\parallel \mathcal{E}$ if and only if $((l_1^1, \ldots, l_1^n), \sigma, \bigwedge_{i:\sigma\in\Sigma^i} g_\sigma, u, (l_2^1, \ldots, l_2^n))$ is an edge in $\parallel \mathcal{F}$. Now it follows immediately that $\bigwedge_{i:\sigma\in\Sigma^i} g_\sigma^i \Leftrightarrow_V \bigwedge_{i:\sigma\in\Sigma^i} g_\sigma$. This concludes the proof. □

Theorem 2. *Let $(\mathcal{E}, \xi_1)$ be a coordinator tuple with $\mathcal{E} = \{E^1, \ldots, E^n\}$ a deterministic EFA system, where each individual EFA $E^i \in \mathcal{E}$ locally normalized. Construct the normalized form of $\mathcal{E}$ as $\mathcal{F} = \mathcal{N}(\mathcal{E}) = \{\mathcal{N}(E^1), \ldots, \mathcal{N}(E^n)\}$. Then refinement function $\xi = \mathrm{id}$ ensures that $(\mathcal{E}, \xi_1) \simeq_{co} (\mathcal{F}, \xi_1 \circ \xi)$.*

Proof. From the definition of Ξ and the construction of ξ, it follows directly that $\xi \in \Xi$. Therefore, $\xi_1 \circ \xi \in \Xi$ and $(\mathcal{F}, \xi_1 \circ \xi)$ is a coordinator tuple.

Now we show that the two coordinator tuples are coordinator equivalent. Again, from Lemma 9 it follows that $\parallel \mathcal{E} \Leftrightarrow_V \parallel \mathcal{F}$. From Lemma 3 it follows that

$\sup CN(\mathcal{E}) = \mathrm{SSEFA}(\mathcal{E}) \Leftrightarrow_V \mathrm{SSEFA}(\mathcal{F}) = \sup CN(\mathcal{F})$. And from Lemma 8 it follows that $\xi_1(\sup CN(\mathcal{E})) \Leftrightarrow_V \xi_1(\sup CN(\mathcal{F}))$. Thus, from Lemma 4 it follows that $U(\xi_1(\sup CN(\mathcal{E}))) = U(\xi_1(\sup CN(\mathcal{F})))$. We can now observe that

$$\begin{aligned}\mathcal{L}(\xi_1(\sup CN(\mathcal{E}))) &= \mathcal{L}(U(\xi_1(\sup CN(\mathcal{E})))) \\ &= \mathcal{L}(U(\xi_1(\sup CN(\mathcal{F})))) \\ &= \mathcal{L}(\xi_1(\sup CN(\mathcal{F}))) \\ &= \mathcal{L}(\xi_1(\xi(\sup CN(\mathcal{F}))))\end{aligned}$$

This concludes the proof. □

3 FA-based abstractions

3.1 Description of the abstraction

For FA-based abstractions, we use the following notions of conflict equivalence and synthesis abstraction. The notion of conflict equivalence is from Flordal and Malik [2009], while the notion of synthesis abstraction is from Mohajerani et al. [2011].

Definition 2 (FA conflict equivalence). *Two FAs E and F are called conflict equivalent with respect to a set of local events Γ, denoted with $E \simeq_{conf,\Gamma} F$, if for any FA T with alphabet Σ_T and $\Gamma \cap \Sigma_T = \emptyset$ it holds that $E \parallel T$ is nonblocking if and only if $F \parallel T$ is nonblocking.*

Definition 3 (FA synthesis abstraction). *FA F is called a synthesis abstraction of FA E with respect to a set of local events Γ, denoted with $E \lesssim_{synth,\Gamma} F$, if for any FA T with alphabet Σ_T and $\Gamma \cap \Sigma_T = \emptyset$ it holds that $\mathcal{L}(E \parallel T \parallel \sup CN(E \parallel T)) = \mathcal{L}(E \parallel T \parallel \sup CN(F \parallel T))$.*

Let $E = (L, \Sigma, \rightarrow, l_0, L_m)$ be an FA and $\sim \subseteq L \times L$ an equivalence relation. Given an equivalence relation $\sim$ on L, the equivalence class of a location $l \in L$ is $[l] = \{l' \mid (l, l') \in \sim\}$, and $L/\!\sim = \{[l] \mid l \in L\}$ is the set of all equivalence classes modulo $\sim$. The quotient automaton of E, denoted with $E/\!\sim$, is given by $E/\!\sim = (L/\!\sim, \Sigma, \rightarrow/\!\sim, [l_0], L_m/\!\sim)$ where $\rightarrow/\!\sim = \{([l_1], \sigma, [l_2]) \mid (l_1, \sigma, l_2) \in \rightarrow\}$.

Furthermore, to apply FA-based abstractions, we need the notion of local events (as several FA-based abstractions heavily rely on local events) and a mechanism to transform an EFA to an FA.

In an FA system, an event is considered to be local in A if it only appears in the alphabet of A and not in the alphabet of other FAs. In the context of EFA systems, considering the alphabets is insufficient to determine local events. An EFA may influence (or be influenced by) another EFA through variables. In the work of Mohajerani et al. [2016], an event is considered to be local in an EFA system if it only appears in the alphabet of a single EFA *and* it has no dependencies and effect on variables, i.e., on transitions labeled with a local event the guards are true and the updates do not alter the valuation of variables.

An FA may be obtained from an EFA by calculating its state space, see Definition 2 of the paper. Unfortunately, this operation suffers from the state-space explosion problem. For a normalized EFA, it is also possible to create an FA by simply disregarding all guards and updates, called the FA-form of an EFA, as in a normalized system all transitions labeled with the same event have the same guard and update. Note that the FA form of an EFA may be different from its state space. More information can be found in Mohajerani et al. [2016]. The following definition introduces the FA form of an EFA.

Definition 4 (FA form)**.** *Let* $E = (L, V, \Sigma, \rightarrow, l_0, \hat{v}_0, L_m)$ *be an EFA. The FA form of* E *is the FA* $\phi(E) = (L, \Sigma, \rightarrow_\phi, l_0, L_m)$*, where* $\rightarrow_\phi = \{(x, \sigma, y) \mid (x, \sigma, g, u, y) \in \rightarrow\}$*.*

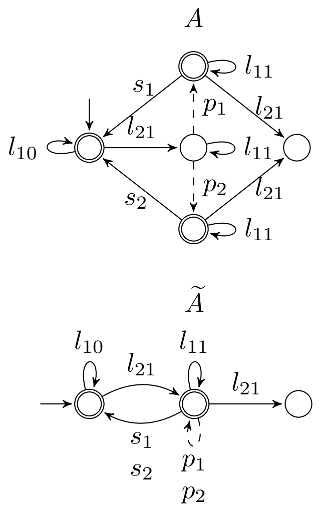
The conversion from an EFA to its FA form does not suffer from the state-space explosion problem, and at the same time makes it possible to perform FA-based abstractions.

The following theorem shows the result for FA-based abstractions. The proof of this theorem can be found in Section 3.2 of this supplementary material.

Theorem 3 (FA-based abstractions)**.** *Let* $(\mathcal{E}, \xi_1)$ *be a coordinator tuple with* $\mathcal{E} = \{E^1, E^2, \ldots, E^n\}$ *a normalized EFA system,* $\sim \subseteq L^1 \times L^1$ *an equivalence relation, and* $\Gamma \subseteq \Sigma^1$ *such that* $(\Sigma^2 \cup \ldots \cup \Sigma^n) \cap \Gamma = \emptyset$ *and* $g_\sigma \equiv \boldsymbol{T}$ *and* $\hat{v}(v) = \hat{v}(u_\sigma(v))$ *for all* $\sigma \in \Gamma, v \in V$*, and* $\hat{v} \in \mathrm{Val}(V)$*. Let* $\mathcal{F} = \{F^1, E^2, \ldots, E^n\}$ *be a normalized EFA system such that* $\phi(F^1) = \phi(E^1) /\!\sim$*,* $\phi(E^1) \simeq_{conf,\Gamma} \phi(F^1)$*, and* $\phi(E^1) \lesssim_{synth,\Gamma} \phi(F^1)$*. Then refinement function* $\xi(\mathcal{G}) = \mathrm{id}(\mathcal{G}) \parallel \mathcal{E}$ *for any EFA system* $\mathcal{G}$ *ensures that* $(\mathcal{E}, \xi_1) \simeq_{co} (\mathcal{F}, \xi_1 \circ \xi)$*.*

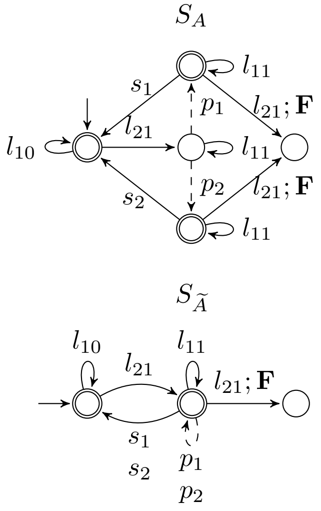
Compared to compositional nonblocking verification, Theorem 3 requires that the FA-based abstraction is not only conflict equivalent, but also a synthesis abstraction. Furthermore, this theorem requires that the abstraction can be performed by creating an equivalence relation on the location set and then calculating the quotient automaton. Several FA-based abstractions fit this requirement, see Flordal and Malik [2009], Mohajerani et al. [2011, 2014a]. In general, quotient automata allow for more behavior than the original automaton. Therefore, the coordinator synthesized for the quotient automaton may contain more behavior than the coordinator for the original automaton. This difference can be taken away by synchronizing the abstracted coordinator with the original automaton.

Example. EFA A as shown in Figure 1 is abstracted into $\widetilde{A}$, also shown in Figure 1, with the FA-based abstraction called *weak synthesis observation equivalence*. The coordinators synthesized from A and $\widetilde{A}$ are shown in Figure 2. Observe that the language of $\mathcal{L}(S_A) \subset \mathcal{L}(S_{\widetilde{A}})$: in the quotient automaton $S_{\widetilde{A}}$,



Event	Guard and update
l_{10}	$v_1 < 2; v_1 := v_1 + 1$
l_{11}	$v_1 < 1; v_1 := v_1 + 1$
l_{21}	$0 < v_1; v_1 := v_1 - 1$
l_{22}	$0 < v_1; v_1 := v_1 - 1$
p_1	$\mathbf{T}$
p_2	$\mathbf{T}$
s_1	$\mathbf{T}$
s_2	$\mathbf{T}$

Fig. 1: EFA A and abstracted EFA $\widetilde{A}$ obtained with the FA-based abstraction called *weak synthesis observation equivalence*.



Event	Guard and update
l_{10}	$v_1 < 2; v_1 := v_1 + 1$
l_{11}	$v_1 < 1; v_1 := v_1 + 1$
l_{21}	$0 < v_1; v_1 := v_1 - 1$
p_1	$\mathbf{T}$
p_2	$\mathbf{T}$
s_1	$\mathbf{T}$
s_2	$\mathbf{T}$

Fig. 2: The coordinators S_A and $S_{\widetilde{A}}$ calculated from A and $\widetilde{A}$, respectively, both shown in Figure 1. In the table the original normalized guards and updates are displayed, while the guards strengthened by supervisor synthesis are displayed in the automaton itself.

for example, a sequence of consecutive p_1 events is possible after string l_{21}, which is not possible in the original automaton S_A. Nevertheless, it holds that $\mathcal{L}(S_A) = \mathcal{L}(S_{\widetilde{A}} \parallel A)$, i.e., the language of the original coordinator is the same as the language of the abstracted coordinator in synchronous composition with the original EFA.

3.2 Proof

Definition 5. *Two FAs E and F are said to be bisimilar, denoted with $E \underline{\leftrightarrow} F$, if there exists a bisimulation relation $R \subseteq L^E \times L^F$ such that $(l_0^E, l_0^F) \in R$. Furthermore, a relation $R \subseteq L^E \times L^F$ is called a bisimulation relation if the following holds for any tuple $(l_1^E, l_1^F) \in R$:*

- *if $l_1^E \xrightarrow{\sigma} l_2^E$ in E, then $l_1^F \xrightarrow{\sigma} l_2^F$ in F with $(l_2^E, l_2^F) \in R$,*
- *if $l_1^F \xrightarrow{\sigma} l_2^F$ in F, then $l_1^E \xrightarrow{\sigma} l_2^E$ in E with $(l_2^E, l_2^F) \in R$,*
- *if $l_1^E \in L_m^E$, then $l_1^F \in L_m^F$, and*
- *if $l_1^F \in L_m^F$, then $l_1^E \in L_m^E$.*

Definition 6. *Two EFAs E and F with same variable set V are said to be valuation bisimilar, denoted with $E \underline{\leftrightarrow}_V F$, if there exists a valuation bisimulation relation $R \subseteq L^E \times L^F \times \mathrm{Val}(V)$ such that $(l_0^E, l_0^F, \hat{v}_0) \in R$. Furthermore, a relation $R \subseteq L^E \times L^F \times \mathrm{Val}(V)$ is called a valuation bisimulation relation if the following holds for any triple $(l_1^E, l_1^F, \hat{v}) \in R$:*

- *if $(l_1^E, \hat{v}) \xrightarrow{\sigma} (l_2^E, \hat{w})$ in $U(E)$, then $(l_1^F, \hat{v}) \xrightarrow{\sigma} (l_2^F, \hat{w})$ in $U(F)$ with $(l_2^E, l_2^F, \hat{w}) \in R$,*
- *if $(l_1^F, \hat{v}) \xrightarrow{\sigma} (l_2^F, \hat{w})$ in $U(F)$, then $(l_1^E, \hat{v}) \xrightarrow{\sigma} (l_2^E, \hat{w})$ in $U(E)$ with $(l_2^E, l_2^F, \hat{w}) \in R$,*
- *if $l_1^E \in L_m^E$, then $l_1^F \in L_m^F$, and*
- *if $l_1^F \in L_m^F$, then $l_1^E \in L_m^E$.*

For EFA E we denote with $\sup \mathit{CN}(E)$ the nonblocking, controllable, and maximally permissive EFA supervisor. For FA E we denote with $\sup \mathit{CN}_F(E)$ the nonblocking, controllable, and maximally permissive FA supervisor.

Lemma 10. *Let E be an FA and $\sim$ an equivalence relation on E such that $E \simeq_{\mathit{conf},\Gamma} E/\!\sim$ and $E \lesssim_{\mathit{synth},\Gamma} E/\!\sim$. Then for any FA T it holds that $\sup \mathit{CN}_F(E \parallel T) \underline{\leftrightarrow} \sup \mathit{CN}_F(E/\!\sim \parallel T) \parallel E \parallel T$.*

Proof. Let $E = (L^E, \Sigma^E, \rightarrow^E, l_0^E, L_m^E)$ and $T = (L^T, \Sigma^T, \rightarrow^T, l_0^T, L_m^T)$. First, observe that the initial location of $E \parallel T$ and $\sup \mathit{CN}_F(E \parallel T)$ is (l_0^E, l_0^T). From the construction of $E/\!\sim$ it follows that the initial location of $E/\!\sim \parallel T$ and $\sup \mathit{CN}_F(E/\!\sim \parallel T)$ is $([l_0^E], l_0^T)$. Therefore, the initial location of $\sup \mathit{CN}_F(E/\!\sim \parallel T) \parallel E/\!\sim \parallel T$ is $([l_0^E], l_0^T, l_0^E, l_0^T)$.

Construct $R \subseteq (L^E \times L^T) \times (L^E/{\sim}, L^T, L^E, L^T)$ as

$$R = \{((e,t),([e],t,e,t)) \mid l \in L^E, t \in L^T, (e,t) \in \mathrm{Reach}(\sup CN_F(E \parallel T))\},$$

where $\mathrm{Reach}(F)$ denotes the reachable states of FA F. We will show that R is a bisimulation relation.

Let $((e,t),([e],t,e,t)) \in R$.

- Let $(e,t) \xrightarrow{\sigma} (e',t')$ in $\sup CN_F(E \parallel T)$. As $(e,t) \in \mathrm{Reach}(\sup CN_F(E \parallel T))$, it holds that there exists a string s such that $(l_0^E, l_0^T) \xrightarrow{s} (e,t)$, i.e., $s \in \mathcal{L}(\sup CN_F(E \parallel T))$. As $E \lesssim_{\text{synth}} E/{\sim}$, it follows that $s \in \mathcal{L}(\sup CN_F(E/{\sim} \parallel T) \parallel E \parallel T)$. Furthermore, $s \in \mathcal{L}(E \parallel T)$. Combining this with the construction of $E/{\sim}$, it follows that $([l_0^E], l_0^T, l_0^E, l_0^T) \xrightarrow{s} ([e],t,e,t)$. As $(e,t) \xrightarrow{\sigma} (e',t')$ in $\sup CN_F(E \parallel T)$, it follows that $(e,t) \xrightarrow{\sigma} (e',t')$ in $E \parallel T$. Consider three cases for σ.
  - $\sigma \in \Sigma^E \cap \Sigma^T$. From the definition of synchronous product it follows that $e \xrightarrow{\sigma} e'$ in E and $t \xrightarrow{\sigma} t'$ in T. From the construction of $E/{\sim}$ it follows that $[e] \xrightarrow{\sigma} [e']$ in $E/{\sim}$.
  - $\sigma \in \Sigma^E \setminus \Sigma^T$. From the definition of synchronous product it follows that $e \xrightarrow{\sigma} e'$ in E and $t = t'$. From the construction of $E/{\sim}$ it follows that $[e] \xrightarrow{\sigma} [e']$ in $E/{\sim}$.
  - $\sigma \in \Sigma^T \setminus \Sigma^E$. From the definition of synchronous product it follows that $t \xrightarrow{\sigma} t'$ in T and $e = e'$. From the construction of $E/{\sim}$ it follows that $[e] = [e']$ in $E/{\sim}$.

  In all three cases it follows using synchronous product again that $([e],t) \xrightarrow{\sigma} ([e'],t')$ in $E/{\sim} \parallel T$. Furthermore, as $(e,t) \xrightarrow{\sigma} (e',t')$ in $\sup CN_F(E \parallel T)$, it also follows that $s\sigma \in \mathcal{L}(\sup CN(E \parallel T)) = \mathcal{L}(\sup CN_F(E/{\sim} \parallel T) \parallel E \parallel T)$. This implies that $([e],t,e,t) \xrightarrow{\sigma} ([e''],t'',e'',t'')$ in $\sup CN_F(E/{\sim} \parallel T) \parallel E \parallel T$ for some $e'' \in L^E$ and $t'' \in L^T$. Combining above observations it must hold that $e'' = e'$ and $t'' = t'$. Therefore, $([e],t,e,t) \xrightarrow{\sigma} ([e'],t',e',t')$ in $\sup CN_F(E/{\sim} \parallel T) \parallel E \parallel T$. Finally, observe that $((e',t'),([e'],t',e',t')) \in R$ by the construction of R.

- Let $([e],t,e,t) \xrightarrow{\sigma} ([e'],t',e',t')$ in $\sup CN_F(E/{\sim} \parallel T) \parallel E \parallel T$. Consider three cases for σ.
  - $\sigma \in \Sigma^E \cap \Sigma^T$. From the definition of synchronous product it follows that $([e],t) \xrightarrow{\sigma} ([e'],t)$ in $\sup CN_F(E/{\sim} \parallel T)$, $e \xrightarrow{\sigma} e'$ in E, and $t \xrightarrow{\sigma} t'$ in T.
  - $\sigma \in \Sigma^E \setminus \Sigma^T$. From the definition of synchronous product it follows that $([e],t) \xrightarrow{\sigma} ([e'],t)$ in $\sup CN_F(E/{\sim} \parallel T)$, $e \xrightarrow{\sigma} e'$ in E, and $t = t'$.
  - $\sigma \in \Sigma^T \setminus \Sigma^E$. From the definition of synchronous product it follows that $([e],t) \xrightarrow{\sigma} ([e'],t)$ in $\sup CN_F(E/{\sim} \parallel T)$, $e = e'$ in E, and $t \xrightarrow{\sigma} t'$ in T.

In all three cases it follows that $(e,t) \xrightarrow{\sigma} (e',t')$ in $E \parallel T$. As $s\sigma \in \mathcal{L}(\sup CN_F(E\ /{\sim} \parallel T) \parallel E \parallel T) = \mathcal{L}(\sup CN_F(E \parallel T))$, it follows that $(e,t) \xrightarrow{\sigma} (e',t')$ in $\sup CN_F(E \parallel T)$. Finally, observe that $((e',t'),([e'],t',e',t')) \in R$ by the construction of R.

- Let $(e,t) \in L^E_m \times L^T_m$ in $\sup CN_F(E \parallel T)$. As both supervisor synthesis and quotient automaton do not alter the marking of locations, it follows that $(e,t) \in L^E_m \times L^T_m$ in $E \parallel T$ and in $E\ /{\sim} \parallel T$. Therefore, $([e],t,e,t) \in L^E_m\ /{\sim} \times L^T_m \times L^E_m \times L^T_m$ in $\sup CN_F(E\ /{\sim} \parallel T) \parallel E \parallel T$.
- Let $([e],t,e,t) \in L^E_m\ /{\sim} \times L^T_m \times L^E_m \times L^T_m$ in $\sup CN_F(E\ /{\sim} \parallel T) \parallel E \parallel T$. As both supervisor synthesis and quotient automaton do not alter the marking of locations, it follows that $(e,t) \in L^E_m \times L^T_m$ in $E \parallel T$, and thus $(e,t) \in L^E_m \times L^T_m$ in $\sup CN_F(E \parallel T)$.

This shows that R is a bisimulation relation. Finally, as the initial locations are in this bisimulation relation (by the construction of it), it follows that $\sup CN_F(E \parallel T) \underline{\leftrightarrow} \sup CN_F(E\ /{\sim} \parallel T) \parallel E \parallel T$. This concludes the proof. □

Lemma 11. *Let $\mathcal{E} = \{E^1, E^2, \ldots, E^n\}$ be a normalized EFA system. Then $(l^1_1, \ldots, l^n_1, \hat{v}) \xrightarrow{\sigma} (l^1_2, \ldots, l^n_2, \hat{w})$ is an edge in $\sup CN_F(\phi(E^1) \parallel \ldots \parallel \phi(E^n) \parallel V_\mathcal{E})$ if and only if $(l^1_1, \ldots, l^n_1) \xrightarrow{\sigma, g^*, u} (l^1_2, \ldots, l^n_2)$ is an edge in $\sup CN(E^1 \parallel \ldots \parallel E^n)$ with $g^*[\hat{v}] = \boldsymbol{T}$ and $\hat{w}(v) = \hat{v}(u(v))$, where $(l^1_1, \ldots, l^n_1, \hat{v})$ is a reachable location in $\sup CN_F(\phi(E^1) \parallel \ldots \parallel \phi(E^n) \parallel V_\mathcal{E})$ and $(l^1_1, \ldots, l^n_1)$ a reachable location in $\sup CN(E^1 \parallel \ldots \parallel E^n)$.*

Proof. First, Lemma 13 of Mohajerani et al. [2016] states that $U(\mathcal{E}) = \phi(E^1) \parallel \ldots \parallel \phi(E^n)$. Therefore, it follows that $\mathcal{L}(\sup CN(\mathcal{E})) = \mathcal{L}(\sup CN_F(U(\mathcal{E}))) = \mathcal{L}(\sup CN_F(\phi(E^1) \parallel \ldots \parallel \phi(E^n)))$.

If $\mathcal{L}(\sup CN(\mathcal{E})) = \emptyset$, then trivially no edge is enabled in $\sup CN(\mathcal{E})$ and $\sup CN_F(\phi(E^1) \parallel \ldots \parallel \phi(E^n))$. In the case that $\mathcal{L}(\sup CN(\mathcal{E})) \neq \emptyset$, we will prove the lemma by induction on the length $i = [0,k]$ of string $s = \sigma_1 \ldots \sigma_k \in \mathcal{L}(\sup CN(\mathcal{E}))$.

Base case. Let $i = 0$. As both $\sup CN(\mathcal{E})$ and $\sup CN_F(\phi(E^1) \parallel \ldots \parallel \phi(E^n))$ are automata, it holds that the empty string is included in their languages. Furthermore, the state reached after performing the empty string is the initial state. For $\sup CN(\mathcal{E})$ the initial state is the same as $\mathcal{E}$, which is initial location $(l^1_0, \ldots, l^n_0)$ together with initial valuation $\hat{v}_0$; for $\sup CN_F(\phi(E^1) \parallel \ldots \parallel \phi(E^n))$ the initial state is the same as $\phi(E^1) \parallel \ldots \parallel \phi(E^n)$, which is $(l^1_0, \ldots, l^n_0, \hat{v}_0)$ by following the definitions of ϕ and $V_\mathcal{E}$.

Inductive step. Assume as induction hypothesis that string $s_i = \sigma_1 \ldots \sigma_i \in \mathcal{L}(\sup CN(\mathcal{E}))$ and $\sup CN(\mathcal{E})$ reached location $(l^1_1, \ldots, l^n_1)$ together with valuation $\hat{v}$ while $\sup CN_F(\phi(E^1) \parallel \ldots \parallel \phi(E^n))$ reached location $(l^1_1, \ldots, l^n_1, \hat{v})$. As $s \in \mathcal{L}(\sup CN(\mathcal{E}))$, it follows that $s_{i+1} = \sigma_i \ldots \sigma_i \sigma_{i+1} \in \mathcal{L}(\sup CN(\mathcal{E}))$. Therefore, there exists an edge $(l^1_1, \ldots, l^n_1) \xrightarrow{\sigma, g^*, u} (l^1_{2'}, \ldots, l^n_{2'})$ in $\sup CN(\mathcal{E})$ with $g^*[\hat{v}] = \mathbf{T}$ and $\hat{w}'(v) = \hat{v}(u(v))$, and there exists an edge $(l^1_1, \ldots, l^n_1, \hat{v}) \xrightarrow{\sigma} (l^1_{2''}, \ldots, l^n_{2''}, \hat{w}'')$ in $\sup CN_F(\phi(E^1) \parallel \ldots \parallel \phi(E^n) \parallel V_\mathcal{E})$. From the construction of the supervisors, it follows that $(l_1^1, \dots, l_1^n) \xrightarrow{\sigma,g,u} (l_{2'}^1, \dots, l_{2'}^n)$ is an edge in $\mathcal{E}$ with $g[\hat{v}] = \mathbf{T}$ and $\hat{w}'(v) = \hat{v}(u(v))$, and $(l_1^1, \dots, l_1^n, \hat{v}) \xrightarrow{\sigma} (l_{2''}^1, \dots, l_{2''}^n, \hat{w}'')$ is an edge in $\phi(E^1) \parallel \dots \parallel \phi(E^n) \parallel V_{\mathcal{E}}$. Following the definitions of ϕ and $V_{\mathcal{E}}$ and that $\mathcal{E}$ is deterministic, we can conclude that $l_{2'}^i = l_{2''}^i = l_2^i$ for all $i \in [1, n]$ and $\hat{w}' = \hat{w}'' = \hat{w}$. Therefore, $(l_1^1, \dots, l_1^n, \hat{v}) \xrightarrow{\sigma} (l_2^1, \dots, l_2^n, \hat{w})$ is an edge in $\sup CN_F(\phi(E^1) \parallel \dots \parallel \phi(E^n) \parallel V_{\mathcal{E}})$ and $(l_1^1, \dots, l_1^n) \xrightarrow{\sigma,g^*,u} (l_2^1, \dots, l_2^n)$ is an edge in $\sup CN(E^1 \parallel \dots \parallel E^n)$ with $g^*[\hat{v}] = \mathbf{T}$ and $\hat{w}(v) = \hat{v}(u(v))$, and $\sup CN(\mathcal{E})$ reached location $(l_2^1, \dots, l_2^n)$ together with valuation $\hat{w}$ while $\sup CN_F(\phi(E^1) \parallel \dots \parallel \phi(E^n))$ reached location $(l_2^1, \dots, l_2^n, \hat{w})$.

As string s is chosen arbitrarily, it follows that $(l_1^1, \dots, l_1^n, \hat{v}) \xrightarrow{\sigma} (l_2^1, \dots, l_2^n, \hat{w})$ is an edge in $\sup CN_F(\phi(E^1) \parallel \dots \parallel \phi(E^n) \parallel V_{\mathcal{E}})$ if and only if $(l_1^1, \dots, l_1^n) \xrightarrow{\sigma,g^*,u}$ $(l_2^1, \dots, l_2^n)$ is an edge in $\sup CN(E^1 \parallel \dots \parallel E^n)$ with $g^*[\hat{v}] = \mathbf{T}$ and $\hat{w}(v) = \hat{v}(u(v))$. This concludes the proof. □

Lemma 12. *Let E and F be two deterministic normalized EFAs with the same alphabet Σ and same variable set V. Furthermore, $E \underline{\leftrightarrow}_V F$. Then for any renaming function $\rho : \Sigma \to \Sigma'$ it holds that $\rho(E) \underline{\leftrightarrow}_V \rho(F)$.*

Proof. As $E \underline{\leftrightarrow}_V F$, there exists a valuation bisimulation relation $R \subseteq Q^E \times Q^F \times \mathrm{Val}(V)$ such that $(l_0^E, l_0^F, \hat{v}_0) \in R$. We show that R is also a valuation bisimulation relation for $\rho(E)$ and $\rho(F)$. Observe that renaming does not change the location set, variable set, initial location, initial valuation, and marked states.

Given a triple $(l_1^E, l_1^F, \hat{v}) \in R$, consider the following four cases.

- If $(l_1^E, \hat{v}) \xrightarrow{\mu} (l_2^E, \hat{w})$ in $U(\rho(E))$, then there exists an event $\sigma \in \rho^{-1}(\mu)$ such that $(l_1^E, \hat{v}) \xrightarrow{\sigma} (l_2^E, \hat{w})$ in $U(E)$. As $E \underline{\leftrightarrow}_V F$, it follows that $(l_1^F, \hat{v}) \xrightarrow{\sigma} (l_2^F, \hat{w})$ in $U(F)$ and $(l_2^E, l_2^F, \hat{w}) \in R$. Applying renaming results in $(l_1^F, \hat{v}) \xrightarrow{\mu} (l_2^F, \hat{w})$ in $U(F)$ as $\rho(\sigma) = \mu$.

- If $(l_1^F, \hat{v}) \xrightarrow{\mu} (l_2^F, \hat{w})$ in $U(\rho(F))$, then there exists an event $\sigma \in \rho^{-1}(\mu)$ such that $(l_1^F, \hat{v}) \xrightarrow{\sigma} (l_2^F, \hat{w})$ in $U(F)$. As $E \underline{\leftrightarrow}_V F$, it follows that $(l_1^E, \hat{v}) \xrightarrow{\sigma}$ $(l_2^E, \hat{w})$ in $U(E)$ and $(l_2^E, l_2^F, \hat{w}) \in R$. Applying renaming results in $(l_1^E, \hat{v})$ $\xrightarrow{\mu} (l_2^E, \hat{w})$ in $U(E)$ as $\rho(\sigma) = \mu$.

- If $l_1^E \in L_m^{\rho(E)}$, then $l_1^E \in L_m^E$ as renaming does not alter the marking of states. As $E \underline{\leftrightarrow}_V F$, it follows that $l_1^F \in L_m^F$ and, therefore, $l_1^F \in L_m^{\rho(F)}$.

- If $l_1^F \in L_m^{\rho(F)}$, then $l_1^F \in L_m^F$ as renaming does not alter the marking of states. As $E \underline{\leftrightarrow}_V F$, it follows that $l_1^E \in L_m^E$ and, therefore, $l_1^E \in L_m^{\rho(E)}$.

Therefore, we can conclude that $\rho(E) \underline{\leftrightarrow}_V \rho(F)$ as $(l_0^{\rho(E)}, l_0^{\rho(F)}, \hat{v}_0) = (l_0^E, l_0^F, \hat{v}_0)$ $\in R$. □

Lemma 13. *Let E and F be two deterministic normalized EFAs with the same alphabet Σ and same variable set V. Furthermore, $E \underline{\leftrightarrow}_V F$. Then for any renaming function $\rho : \Sigma' \to \Sigma$ it holds that $\rho^{-1}(E) \underline{\leftrightarrow}_V \rho^{-1}(F)$.*

Proof. As $E \underline{\leftrightarrow}_V F$, there exists a valuation bisimulation relation $R \subseteq Q^E \times Q^F \times \mathrm{Val}(V)$ such that $(l_0^E, l_0^F, \hat{v}_0) \in R$. We show that R is also a valuation bisimulation relation for $\rho^{-1}(E)$ and $\rho^{-1}(F)$. Observe that renaming does not change the location set, variable set, initial location, initial valuation, and marked states.

Given a triple $(l_1^E, l_1^F, \hat{v}) \in R$, consider the following four cases.

- If $(l_1^E, \hat{v}) \xrightarrow{\mu} (l_2^E, \hat{w})$ in $U(\rho^{-1}(E))$, then $(l_1^E, \hat{v}) \xrightarrow{\sigma} (l_2^E, \hat{w})$ in $U(E)$ where $\rho(\mu) = \sigma$. As $E \underline{\leftrightarrow}_V F$, it follows that $(l_1^F, \hat{v}) \xrightarrow{\sigma} (l_2^F, \hat{w})$ in $U(F)$ and $(l_2^E, l_2^F, \hat{w}) \in R$. Applying inverse renaming results in $(l_1^F, \hat{v}) \xrightarrow{\mu} (l_2^F, \hat{w})$ in $U(F)$ as $\mu \in \rho^{-1}(\sigma) = \rho^{-1}(\rho(\mu))$.

- If $(l_1^F, \hat{v}) \xrightarrow{\mu} (l_2^F, \hat{w})$ in $U(\rho^{-1}(F))$, then $(l_1^F, \hat{v}) \xrightarrow{\sigma} (l_2^F, \hat{w})$ in $U(F)$ where $\rho(\mu) = \sigma$. As $E \underline{\leftrightarrow}_V F$, it follows that $(l_1^E, \hat{v}) \xrightarrow{\sigma} (l_2^E, \hat{w})$ in $U(E)$ and $(l_2^E, l_2^F, \hat{w}) \in R$. Applying inverse renaming results in $(l_1^E, \hat{v}) \xrightarrow{\mu} (l_2^E, \hat{w})$ in $U(E)$ as $\mu \in \rho^{-1}(\sigma) = \rho^{-1}(\rho(\mu))$.

- If $l_1^E \in L_m^{\rho^{-1}(E)}$, then $l_1^E \in L_m^E$ as renaming does not alter the marking of states. As $E \underline{\leftrightarrow}_V F$, it follows that $l_1^F \in L_m^F$ and, therefore, $l_1^F \in L_m^{\rho^{-1}(F)}$.

- If $l_1^F \in L_m^{\rho^{-1}(F)}$, then $l_1^F \in L_m^F$ as renaming does not alter the marking of states. As $E \underline{\leftrightarrow}_V F$, it follows that $l_1^E \in L_m^E$ and, therefore, $l_1^E \in L_m^{\rho^{-1}(E)}$.

Therefore, we can conclude that $\rho^{-1}(E) \underline{\leftrightarrow}_V \rho^{-1}(F)$ as $(l_0^{\rho^{-1}(E)}, l_0^{\rho^{-1}(F)}, \hat{v}_0) = (l_0^E, l_0^F, \hat{v}_0) \in R$. □

Lemma 14. *Let E and F be two deterministic normalized EFAs with the same alphabet Σ and same variable set V. Furthermore, $E \underline{\leftrightarrow}_V F$. Then for any deterministic EFA T it holds that $E \parallel T \underline{\leftrightarrow}_V F \parallel T$.*

Proof. As $E \underline{\leftrightarrow}_V F$, there exists a valuation relation $R \subseteq L^E \times L^F \times \mathrm{Val}(V)$ such that $(l_0^E, l_0^F, \hat{v}_0) \in R$. We construct a new valuation relation $R_T \subseteq (L^E \times L^T) \times (L^F \times L^T) \times \mathrm{Val}(V)$ inductively such that $((l_0^E, l_0^T), (l_0^F, l_0^T)), \hat{v}_0) \in R^T$.

Start with adding $((l_0^E, l_0^T), (l_0^F, l_0^T)), \hat{v}_0) \in R^T$. Given a triple $((l_1^E, l_1^T), (l_1^F, l_1^T)), \hat{v}) \in R^T$, consider the following four cases.

- If $((l_1^E, l_1^T), \hat{v}) \xrightarrow{\sigma} ((l_2^E, l_2^T), \hat{w})$ in $U(E \parallel T)$, then $(l_1^E, l_1^T) \xrightarrow{\sigma,g,u} (l_2^E, l_2^T)$ in $E \parallel T$ with $g[\hat{v}, \hat{w}] = \mathbf{T}$ and $\hat{w}(v) = \hat{v}(u(v))$. Consider three cases for σ.
  - If $\sigma \in \Sigma^E \cap \Sigma^T$, then from the definition of synchronous product it follows that $l_1^E \xrightarrow{\sigma,g^E,u^E} l_2^E$ in E, $l_1^T \xrightarrow{\sigma,g^T,u^T} l_2^T$ in T, $g = g^E \wedge g^T$, $u = u^E \oplus u^T$. As $g[\hat{v}, \hat{w}] = \mathbf{T}$ and $\hat{w}(v) = \hat{v}(u(v))$, it follows that $g^E[\hat{v}, \hat{w}] = \mathbf{T}$ and $\hat{w}(v) = \hat{v}(u^E(v))$ and thus $(l_1^E, \hat{v}) \xrightarrow{\sigma} (l_2^E, \hat{w})$ in $U(E)$. As $E \underline{\leftrightarrow}_V F$, it follows that $(l_1^F, \hat{v}) \xrightarrow{\sigma} (l_2^F, \hat{w})$ in $U(F)$. This implies that $l_1^F \xrightarrow{\sigma,g^F,u^F} l_2^F$ in F with $g^F[\hat{v}, \hat{w}] = \mathbf{T}$ and $\hat{w}(v) = \hat{v}(u(v))$. Using the definition of synchronous product again we can conclude that $(l_1^F, l_1^T) \xrightarrow{\sigma,g,u} (l_2^F, l_2^T)$ in $F \parallel T$.

  - If $\sigma \in \Sigma^E \setminus \Sigma^T$, then from the definition of synchronous product it follows that $l_1^E \xrightarrow{\sigma,g,u} l_2^E$ in E and $l_1^T = l_2^T$ in T. Thus, $(l_1^E, \hat{v}) \xrightarrow{\sigma} (l_2^E, \hat{w})$ in $U(\rho(E))$ and, as $E \underline{\leftrightarrow}_V F$, it follows that $(l_1^F, \hat{v}) \xrightarrow{\sigma} (l_2^F, \hat{w})$ in $U(F)$. This implies that $l_1^F \xrightarrow{\sigma,g,u} l_2^F$ in F with $g[\hat{v}, \hat{w}] = \mathbf{T}$ and $\hat{w}(v) = \hat{v}(u(v))$. Using the definition of synchronous product again we can conclude that $(l_1^F, l_1^T) \xrightarrow{\sigma,g,u} (l_2^F, l_2^T)$ in $F \parallel T$.
  - If $\sigma \in \Sigma^T \setminus \Sigma^E$, then from the definition of synchronous product it follows that $l_1^E = l_2^E$ in E and $l_1^T \xrightarrow{\sigma,g,u} l_2^T$ in T. As $E \underline{\leftrightarrow}_V F$, it follows that $\sigma \notin \Sigma^F$. Using the definition of synchronous product again we can conclude that $(l_1^F, l_1^T) \xrightarrow{\sigma,g,u} (l_2^F, l_2^T)$ in $F \parallel T$.

  In all three cases we have established that $(l_1^F, l_1^T) \xrightarrow{\sigma,g,u} (l_2^F, l_2^T)$ in $F \parallel T$ with $g[\hat{v}, \hat{w}] = \mathbf{T}$ and $\hat{w}(v) = \hat{v}(u(v))$. Therefore, $((l_1^F, l_1^T), \hat{v}) \xrightarrow{\sigma} ((l_2^F, l_2^T), \hat{w})$ in $U(F \parallel T)$ and we add $((l_2^E, l_2^T), (l_2^F, l_2^T)), \hat{w}) \in R^T$.

- Because of symmetry in E and F, we can show with the same reasoning as above that if $((l_1^F, l_1^T), \hat{v}) \xrightarrow{\sigma} ((l_2^F, l_2^T), \hat{w})$ in $U(F \parallel T)$, then $((l_1^E, l_1^T), \hat{v}) \xrightarrow{\sigma} ((l_2^E, l_2^T), \hat{w})$ in $U(E \parallel T)$. Therefore, we add $((l_2^E, l_2^T), (l_2^F, l_2^T)), \hat{w}) \in R^T$.

- If $(l_1^E, l_1^T) \in L_m^{E\parallel T}$, then from the definition of synchronous product it follows that $l_1^E \in L_m^E$ and $l_1^T \in L_m^T$. As $E \underline{\leftrightarrow}_V F$, it follows that $l_1^F \in L_m^F$. Using the definition of synchronous product again, we have that $(l_1^F, l_1^T) \in L_m^{F\parallel T}$.

- Because of symmetry in E and F, we can show with the same reasoning as above that if $(l_1^F, l_1^T) \in L_m^{F\parallel T}$, then $(l_1^E, l_1^T) \in L_m^{E\parallel T}$.

This concludes the proof. □

Lemma 15. *Let E and F be two deterministic normalized EFAs with the same alphabet Σ and same variable set V. Furthermore, $E \underline{\leftrightarrow}_V F$. Then $\mathcal{L}(E) = \mathcal{L}(F)$.*

Proof. As $E \underline{\leftrightarrow}_V F$, there exists a valuation bisimulation relation $R \subseteq L^E \times L^F \times \mathrm{Val}(V)$ such that $(l_0^E, l_0^F, \hat{v}_0) \in R$. We will show by induction on the length i of string $s = \sigma_1 \cdots \sigma_n$ with $n \in \mathbb{N}$ that $s \in \mathcal{L}(E)$ if and only if $s \in \mathcal{L}(F)$.

Base case. For $i = 0$, it holds that $s_0 = \varepsilon \in \mathcal{L}(E)$ and $s_0 \in \mathcal{L}(F)$ as both E and F are automata. Furthermore, $(l_0^E, \hat{v}_0) = \delta^E((l_0^E, \hat{v}_0), s_0)$ in $U(E)$ and $(l_0^F, \hat{v}_0) = \delta^F((l_0^F, \hat{v}_0), s_0)$ in $U(F)$.

Inductive step. Assume that $s_i = \sigma_1 \cdots \sigma_i \in \mathcal{L}(E)$ if and only if $s_i \in \mathcal{L}(F)$, and that $(l_i^E, l_i^F, \hat{v}_i) \in R$ where $(l_i^E, \hat{v}_i) = \delta^E((l_0^E, \hat{v}_0), s_i)$ in $U(E)$ and $(l_i^F, \hat{v}_i) = \delta^E((l_0^F, \hat{v}_0), s_i)$ in $U(F)$. As $(l_i^E, l_i^F, \hat{v}_i) \in R$, it follows directly that $(l_i^E, \hat{v}_i) \xrightarrow{\sigma_{i+1}} (l_{i+1}^E, \hat{v}_{i+1})$ if and only if $(l_i^F, \hat{v}_i) \xrightarrow{\sigma_{i+1}} (l_{i+1}^F, \hat{v}_{i+1})$. Therefore, $s_i\sigma_{i+1} \in \mathcal{L}(E)$ if and only if $s_i\sigma_{i+1} \in \mathcal{L}(F)$. Furthermore, $(l_i^E, \hat{v}_i) \xrightarrow{\sigma_{i+1}} (l_{i+1}^E, \hat{v}_{i+1})$ if and only if $(l_{i+1}^E, l_{i+1}^F, \hat{v}_{i+1}) \in R$ where $(l_{i+1}^E, \hat{v}_{i+1}) = \delta^E((l_0^E, \hat{v}_0), s_{i+1})$ in $U(E)$ and $(l_{i+1}^F, \hat{v}_{i+1}) = \delta^E((l_0^F, \hat{v}_0), s_{i+1})$ in $U(F)$.

Finally, as string s is chosen arbitrarily, it follows that $\mathcal{L}(E) = \mathcal{L}(F)$. This concludes the proof. □

Lemma 16. *Let E and F be two deterministic normalized EFAs with the same alphabet Σ and same variable set V. Furthermore, $E \underline{\leftrightarrow}_V F$ and $\xi \in \Xi$ a refinement function. Then $\mathcal{L}(\xi(E)) = \mathcal{L}(\xi(F))$.*

Proof. This lemma is proven by induction on the structure of ξ. Denote $\xi = \xi_m \circ \ldots \circ \xi_1$. First, we show that $\xi(E) \underline{\leftrightarrow}_V \xi(F)$.

Base case. It follows from the assumption of the lemma that $E \underline{\leftrightarrow}_V F$.

Inductive step. Assume that $\xi_i \circ \ldots \circ \xi_1(E) \underline{\leftrightarrow}_V \xi_i \circ \ldots \circ \xi_1(F)$. Consider the following cases for ξ_{i+1}.

- ξ_{i+1} is the identity function. It follows immediately that $\xi_{i+1} \circ \xi_i \circ \ldots \circ \xi_1(E) \underline{\leftrightarrow}_V \xi_{i+1} \circ \xi_i \circ \ldots \circ \xi_1(F)$.
- ξ_{i+1} is a renaming. From Lemma 12 it follows that $\xi_{i+1} \circ \xi_i \circ \ldots \circ \xi_1$ $(E) \underline{\leftrightarrow}_V \xi_{i+1} \circ \xi_i \circ \ldots \circ \xi_1(F)$.
- ξ_{i+1} is a renaming in synchronous composition with the original EFA system. From Lemmas 12 and 14 it follows that $\xi_{i+1} \circ \xi_i \circ \ldots \circ \xi_1(E)$ $\underline{\leftrightarrow}_V \xi_{i+1} \circ \xi_i \circ \ldots \circ \xi_1(F)$.
- ξ_{i+1} is an inverse renaming in synchronous composition with the original EFA system. From Lemmas 13 and 14 it follows that $\xi_{i+1} \circ \xi_i \circ \ldots \circ \xi_1(E) \underline{\leftrightarrow}_V \xi_{i+1} \circ \xi_i \circ \ldots \circ \xi_1(F)$.

We can conclude that $\xi(E) \underline{\leftrightarrow}_V \xi(F)$. Finally, from Lemma 15 it follows that $\mathcal{L}(\xi(E)) = \mathcal{L}(\xi(F))$. This concludes the proof. □

Lemma 17. *Let $\mathcal{E} = \{E^1, E^2, \ldots, E^n\}$ be a normalized EFA system and $\Gamma \subseteq \Sigma^1$ such that $(\Sigma^2 \cup \ldots \cup \Sigma^n) \cap \Gamma = \emptyset$ and $g_\sigma \equiv \boldsymbol{T}$ and $\hat{v}(v) = \hat{v}(u_\sigma(v))$ for all $\sigma \in \Gamma, v \in V$, and $\hat{v} \in \mathrm{Val}(V)$. Let $\mathcal{F} = \{F^1, E^2, \ldots, E^n\}$ be a normalized EFA system such that $\sim \subseteq L^1 \times L^1$ is an equivalence relation, $\phi(F^1) = \phi(E^1) /\!\sim$, $\phi(E^1) \simeq_{conf,\Gamma} \phi(F^1)$, and $\phi(E^1) \lesssim_{synth,\Gamma} \phi(F^1)$. Then $\sup CN(\mathcal{E}) \underline{\leftrightarrow}_V \sup CN(\mathcal{F}) \parallel \mathcal{E}$.*

Proof. Events from Γ can be considered to be local in the FA-based abstractions, as the proof of Proposition 5 from Mohajerani et al. [2016] shows that $U(\mathcal{E}) = \phi(E^1) \parallel \ldots \parallel \phi(E^n) \parallel V_{\mathcal{E}} = \phi(E^1) \parallel \ldots \parallel \phi(E^n) \parallel V_{\mathcal{E}|\Sigma\setminus\Gamma}$.

In order to show the valuation bisimarity, we first observe from Lemma 10 it follows that $\sup CN_F(\phi(E^1) \parallel \phi(E^2) \parallel \ldots \parallel \phi(E^n) \parallel V_{\mathcal{E}}) \underline{\leftrightarrow} \sup CN_F(\phi(E^1) /\!\sim \parallel \phi(E^2) \parallel \ldots \parallel \phi(E^n) \parallel V_{\mathcal{E}}) \parallel \phi(E^1) \parallel \phi(E^2) \parallel \ldots \parallel \phi(E^n) \parallel V_{\mathcal{E}}$. Furthermore, we know from Lemma 11 that $(l_1^1, \ldots, l_1^n, \hat{v}) \xrightarrow{\sigma} (l_2^1, \ldots, l_2^n, \hat{w})$ is an edge in $\sup CN_F(\phi(E^1) \parallel \ldots \parallel \phi(E^n) \parallel V_{\mathcal{E}})$ if and only if $(l_1^1, \ldots, l_1^n) \xrightarrow{\sigma, g^*, u} (l_2^1, \ldots, l_2^n)$ is an edge in $\sup CN(E^1 \parallel \ldots \parallel E^n)$ with $g^*[\hat{v}] = \mathbf{T}$ and $\hat{w}(v) = \hat{v}(u(v))$. Finally, from the construction of the supervisor it follows that if $(l_1^1, \ldots, l_1^n, \hat{v}) \xrightarrow{\sigma} (l_2^1, \ldots, l_2^n, \hat{w})$ is an edge in $\sup CN_F(\phi(E^1) \parallel \ldots \parallel \phi(E^n) \parallel V_{\mathcal{E}})$, then $(l_1^1, \ldots, l_1^n, \hat{v}) \xrightarrow{\sigma} (l_2^1, \ldots, l_2^n, \hat{w})$ is also an edge in $\phi(E^1) \parallel \ldots \parallel \phi(E^n) \parallel V_{\mathcal{E}}$.

Combining the above observations we know that $(l_1^1, \ldots, l_1^n) \xrightarrow{\sigma, g^*, u} (l_2^1, \ldots, l_2^n)$ is an edge in $\sup CN(E^1 \parallel \ldots \parallel E^n)$ with $g^*[\hat{v}] = \mathbf{T}$ and $\hat{w}(v) = \hat{v}(u(v))$, if and only if $(l_1^1, \ldots, l_1^n, \hat{v}) \xrightarrow{\sigma} (l_2^1, \ldots, l_2^n, \hat{w})$ is an edge in $\sup CN_F(\phi(E^1) \parallel \ldots \parallel \phi(E^n) \parallel V_{\mathcal{E}})$, if and only if $([l_1^1], l_1^2, \ldots, l_1^n, \hat{v}, l_1^1, \ldots, l_1^n, \hat{v}) \xrightarrow{\sigma} ([l_2^1], l_2^2, \ldots, l_2^n, \hat{w}, l_2^1, \ldots, l_2^n, \hat{w})$ is an edge in $\sup CN_F(\phi(E^1) /\!\sim \parallel \phi(E^2) \parallel \ldots \parallel \phi(E^n) \parallel V_{\mathcal{E}}) \parallel \phi(E^1) \parallel \phi(E^2) \parallel \ldots \parallel \phi(E^n) \parallel V_{\mathcal{E}}$.

In order to show valuation bisimilarity, we construct a valuation bisimulation which includes the initial states of both systems. Let $R \subseteq (L^1 \times \ldots \times L^n) \times (L^1 /\!\sim \times L^2 \times \ldots \times L^n, L^1 \times \ldots \times L^n) \times \mathrm{Val}(V)$ be constructed by

$$R = \{((l^1, \ldots, l^n), ([l^1], l^2, \ldots, l^n, l^1, \ldots, l^n), \hat{v}) \mid \\ (l^1, \ldots, l^n) \in \mathrm{Reach}(\sup CN(\mathcal{E}))\}.$$

Consider the triple $((l_1^1, \ldots, l_1^n), ([l_1^1], l_1^2, \ldots, l_1^n, l_1^1, \ldots, l_1^n), \hat{v}) \in R$.

- If $(l_1^1, \ldots, l_1^n, \hat{v}) \xrightarrow{\sigma} (l_2^1, \ldots, l_2^n, \hat{w})$ is an edge in $U(\sup CN(E^1 \parallel \ldots \parallel E^n))$, it follows that $(l_1^1, \ldots, l_1^n) \xrightarrow{\sigma, g^*, u} (l_2^1, \ldots, l_2^n)$ is an edge in $\sup CN(E^1 \parallel \ldots \parallel E^n)$ with $g^*[\hat{v}] = \mathbf{T}$ and $\hat{w}(v) = \hat{v}(u(v))$. With the previous observation we have that $([l_1^1], l_1^2, \ldots, l_1^n, \hat{v}, l_1^1, \ldots, l_1^n, \hat{v}) \xrightarrow{\sigma} ([l_2^1], l_2^2, \ldots, l_2^n, \hat{w}, l_2^1, \ldots, l_2^n, \hat{w})$ is an edge in $\sup CN_F(\phi(E^1) /\!\sim \parallel \phi(E^2) \parallel \ldots \parallel \phi(E^n) \parallel V_{\mathcal{E}}) \parallel \phi(E^1) \parallel \phi(E^2) \parallel \ldots \parallel \phi(E^n) \parallel V_{\mathcal{E}}$. As the construction of $E^1 /\!\sim$ does not alter the alphabet, it follows from the definition of synchronous product that $([l_1^1], l_1^2, \ldots, l_1^n, \hat{v}) \xrightarrow{\sigma} ([l_2^1], l_2^2, \ldots, l_2^n, \hat{w})$ is an edge in $\sup CN_F(\phi(E^1) /\!\sim \parallel \phi(E^2) \parallel \ldots \parallel \phi(E^n) \parallel V_{\mathcal{E}})$ and $(l_1^1, \ldots, l_1^n, \hat{v}) \xrightarrow{\sigma} (l_2^1, \ldots, l_2^n, \hat{w})$ is an edge in $\phi(E^1) \parallel \phi(E^2) \parallel \ldots \parallel \phi(E^n) \parallel V_{\mathcal{E}}$. If we denote $\phi(F^1) = \phi(E^1) /\!\sim$, we can apply Lemma 11 to find that $([l_1^1], l_1^2, \ldots, l_1^n) \xrightarrow{\sigma, g^*, u} ([l_2^1], l_2^2, \ldots, l_2^n)$ is an edge in $\sup CN(F^1 \parallel E^2 \parallel \ldots \parallel E^n)$ with $g^*[\hat{v}] = \mathbf{T}$ and $\hat{w}(v) = \hat{v}(u(v))$. From the fact that $U(\mathcal{E}) = \phi(E^1) \parallel \phi(E^2) \parallel \ldots \parallel \phi(E^n) \parallel V_{\mathcal{E}}$, it follows that $(l_1^1, \ldots, l_1^n) \xrightarrow{\sigma, g, u} (l_2^1, \ldots, l_2^n)$ is an edge in $E^1 \parallel \ldots \parallel E^n$ with $g[\hat{v}] = \mathbf{T}$ and $\hat{w}(v) = \hat{v}(u(v))$. Using the definition of synchronous product, we conclude that $([l_1^1], l_1^2, \ldots, l_1^n, l_1^1, \ldots, l_1^n) \xrightarrow{\sigma, g^* \wedge g, u} ([l_2^1], l_2^2, \ldots, l_2^n, l_2^1, \ldots, l_2^n)$ is an edge in $\sup CN(F^1 \parallel E^2 \parallel \ldots \parallel E^n) \parallel E^1 \parallel E^2 \parallel \ldots \parallel E^n$ with $(g^* \wedge g)[\hat{v}] = \mathbf{T}$ and $\hat{w}(v) = \hat{v}(u(v))$. This implies that $([l_1^1], l_1^2, \ldots, l_1^n, l_1^1, \ldots, l_1^n, \hat{v}) \xrightarrow{\sigma} ([l_2^1], l_2^2, \ldots, l_2^n, l_2^1, \ldots, l_2^n, \hat{w})$ is an edge in $U(\sup CN(F^1 \parallel E^2 \parallel \ldots \parallel E^n) \parallel E^1 \parallel E^2 \parallel \ldots \parallel E^n)$. Finally, from the construction of R it follows that $((l_2^1, \ldots, l_2^n), ([l_2^1], l_2^2, \ldots, l_2^n, l_2^1, \ldots, l_2^n), \hat{w}) \in R$.

- If $([l_1^1], l_1^2, \ldots, l_1^n, l_1^1, \ldots, l_1^n, \hat{v}) \xrightarrow{\sigma} ([l_2^1], l_2^2, \ldots, l_2^n, l_2^1, \ldots, l_2^n, \hat{w})$ is an edge in $U(\sup CN(F^1 \parallel E^2 \parallel \ldots \parallel E^n) \parallel E^1 \parallel \ldots \parallel E^n)$, then $([l_1^1], l_1^2, \ldots, l_1^n, l_1^1, \ldots, l_1^n) \xrightarrow{\sigma, g, u} ([l_2^1], l_2^2, \ldots, l_2^n, l_2^1, \ldots, l_2^n)$ is an edge in $\sup CN(F^1 \parallel E^2 \parallel \ldots \parallel E^n) \parallel E^1 \parallel \ldots \parallel E^n$ with $g[\hat{v}] = \mathbf{T}$ and $\hat{w}(v) = \hat{v}(u(v))$. As both the construction of $E^1 /\!\sim$ and supervisor synthesis do not alter the alphabet and updates, it follows from the definition of synchronous product that $([l_1^1], l_1^2, \ldots, l_1^n) \xrightarrow{\sigma, g', u} ([l_2^1], l_2^2, \ldots, l_2^n)$ is an edge in $\sup CN(F^1 \parallel$

$E^2 \parallel \ldots \parallel E^n)$, $(l_1^1, \ldots, l_1^n) \xrightarrow{\sigma, g'', u} (l_2^1, \ldots, l_2^n)$ is an edge in $E^1 \parallel \ldots \parallel E^n$, and $g = g' \wedge g''$. Therefore, $g'[\hat{v}] = \mathbf{T}$ and $g''[\hat{v}] = \mathbf{T}$. From Lemma 11 it follows that $([l_1^1], l_1^2, \ldots, l_1^n, \hat{v}) \xrightarrow{\sigma} ([l_2^1], l_2^2, \ldots, l_2^n, \hat{w})$ is an edge in $\sup CN_F(\phi(F^1) \parallel \phi(E^2) \parallel \ldots \parallel \phi(E^n) \parallel V_{\mathcal{E}})$, and from the fact that $U(\mathcal{E}) = \phi(E^1) \parallel \phi(E^2) \parallel \ldots \parallel \phi(E^n) \parallel V_{\mathcal{E}}$, it follows that $(l_1^1, \ldots, l_1^n, \hat{v}) \xrightarrow{\sigma} (l_2^1, \ldots, l_2^n, \hat{w})$ is an edge in $\phi(E^1) \parallel \ldots \parallel \phi(E^n) \parallel V_{\mathcal{E}}$. Using the definition of synchronous composition we get that $([l_1^1], l_1^2, \ldots, l_1^n, \hat{v}, l_1^1, \ldots, l_1^n, \hat{v}) \xrightarrow{\sigma} ([l_2^1], l_2^2, \ldots, l_2^n, \hat{w}, l_2^1, \ldots, l_2^n, \hat{w})$ is an edge in $\sup CN_F(\phi(E^1) /{\sim} \parallel \phi(E^2) \parallel \ldots \parallel \phi(E^n) \parallel V_{\mathcal{E}}) \parallel \phi(E^1) \parallel \phi(E^2) \parallel \ldots \parallel \phi(E^n) \parallel V_{\mathcal{E}}$ where we used $\phi(F^1) = \phi(E^1) /{\sim}$. With the previous observation it follows that $(l_1^1, \ldots, l_1^n) \xrightarrow{\sigma, g^*, u} (l_2^1, \ldots, l_2^n)$ is an edge in $\sup CN(E^1 \parallel \ldots \parallel E^n)$ with $g^*[\hat{v}] = \mathbf{T}$ and $\hat{w}(v) = \hat{v}(u(v))$. Now it follows from the definition of unfolding that $(l_1^1, \ldots, l_1^n, \hat{v}) \xrightarrow{\sigma} (l_2^1, \ldots, l_2^n, \hat{w})$ is an edge in $U(\sup CN(E^1 \parallel \ldots \parallel E^n))$. Finally, from the construction of R it follows that $((l_2^1, \ldots, l_2^n), ([l_2^1], l_2^2, \ldots, l_2^n, l_2^1, \ldots, l_2^n), \hat{w}) \in R$.

- Let $(l_1^1, \ldots, l_1^n)$ be a marked location in $\sup CN(E^1 \parallel \ldots \parallel E^n)$. As both the construction of F^1 and supervisor synthesis do not alter the marking of locations, it follows directly that $(l_1^1, \ldots, l_1^n)$ is a marked location of $E^1 \parallel \ldots \parallel E^n$, $([l_1^1], \ldots, l_1^n)$ is a marked location of $F^1 \parallel \ldots \parallel E^n$, and $([l_1^1], \ldots, l_1^n)$ is a marked location of $\sup CN(F^1 \parallel \ldots \parallel E^n)$. Therefore, it follows that $([l_1^1], l_1^2, \ldots, l_1^n, l_1^1, \ldots, l_1^n)$ is a marked location in $\sup CN(F^1 \parallel E^2 \parallel \ldots \parallel E^n) \parallel E^1 \parallel \ldots \parallel E^n$.

- If $([l_1^1], l_1^2, \ldots, l_1^n, l_1^1, \ldots, l_1^n)$ is a marked location in $\sup CN(F^1 \parallel E^2 \parallel \ldots \parallel E^n) \parallel E^1 \parallel \ldots \parallel E^n$, then $([l_1^1], l_1^2, \ldots, l_1^n)$ is a marked location in $\sup CN(F^1 \parallel E^2 \parallel \ldots \parallel E^n)$ and $(l_1^1, \ldots, l_1^n)$ is a marked location in $E^1 \parallel \ldots \parallel E^n$. As supervisor synthesis does not alter the marking of locations, it follows that $(l_1^1, \ldots, l_1^n)$ is a marked location in $\sup CN(E^1 \parallel \ldots \parallel E^n)$.

This shows that R is a valuation bisimulation relation. Finally, as the initial locations are in this bisimulation relation (by construction of it), it follows that $\sup CN(\mathcal{E}) \leftrightarroweq_V \sup CN(\mathcal{F}) \parallel \mathcal{E}$. This concludes the proof. □

Proof of Theorem 3. From the definition of Ξ and the construction of ξ, it follows directly that $\xi \in \Xi$. Therefore, $\xi_1 \circ \xi \in \Xi$ and $(\mathcal{F}, \xi_1 \circ \xi)$ is a coordinator tuple.

Now we show that the two coordinator tuples are coordinator equivalent.

$$\begin{aligned} \mathcal{L}(\xi_1(\sup CN(\mathcal{E}))) &= \mathcal{L}(\xi_1(\sup CN(\mathcal{F}) \parallel \mathcal{E})) \text{ by Lemmas 17 and 16} \\ &= \mathcal{L}(\xi_1(\xi(\sup CN(\mathcal{F})))). \end{aligned}$$

This concludes the proof. □

4 Partial composition

Theorem 4. *Let $(\mathcal{E}, \xi_1)$ be a coordinator tuple with $\mathcal{E} = \{E^1, \dots, E^n\}$ a deterministic normalized EFA system. Construct $\mathcal{F} = \{E^1 \parallel E^2, E^3, \dots, E^n\}$. Then refinement function $\xi = \mathrm{id}$ ensures that $(\mathcal{E}, \xi_1) \simeq_{co} (\mathcal{F}, \xi_1 \circ \xi)$.*

Proof. From the definition of Ξ and the construction of ξ, it follows directly that $\xi \in \Xi$. Therefore, $\xi_1 \circ \xi \in \Xi$ and $(\mathcal{F}, \xi_1 \circ \xi)$ is a coordinator tuple.

Now we show that the two coordinator tuples are coordinator equivalent. We can derive that

$$\begin{aligned}\mathcal{L}(\xi_1(\sup CN(\mathcal{E}))) &= \mathcal{L}(\xi_1(\sup CN(E^1 \parallel E^2 \parallel \dots \parallel E^n))) \\ &= \mathcal{L}(\xi_1(\sup CN((E^1 \parallel E^2) \parallel \dots \parallel E^n))) \\ &= \mathcal{L}(\xi_1(\sup CN(\mathcal{F})) \\ &= \mathcal{L}(\xi_1(\xi(\sup CN(\mathcal{F}))).\end{aligned}$$

This concludes the proof. □

5 Update simplification

In an EFA, guards may be simplified into equivalent ones without changing the behavior of that EFA. For example, the guard $\mathbf{T} \wedge \mathbf{F}$ is equivalent to $\mathbf{F}$. This rewriting operation is called update simplification in the framework of compositional nonblocking verification of Mohajerani et al. [2016]. In simplifying updates, the notion of logical equivalence is used (taken from Mohajerani et al. [2016]).

Definition 7 (Logical equivalence). *Two predicates $p, q \in \Pi_V$ are said to be logically equivalent with respect to variable set V, denoted by $p \Leftrightarrow q$, if $p[\hat{v}] = q[\hat{v}]$ for all valuations $\hat{v} \in val(V)$.*

The following theorem shows that nothing has to be changed in the coordinator to refine an update simplification abstraction, as the behavior of the system is the same before and after the abstraction.

Theorem 5 (Update simplification). *Let $(\mathcal{E}, \xi_1)$ be a coordinator tuple with $\mathcal{E} = \{E^1, \dots, E^n\}$ a deterministic normalized EFA system. Construct $\mathcal{F} = \{F^1, \dots, F^n\}$ with $F^i = (L^i, \Sigma^i, V, \rightarrow^i_{\mathcal{F}}, l^i_0, \hat{v}_0, L^i_m)$ such that $V = vars(\mathcal{E}) = vars(\mathcal{F})$, $g^{\mathcal{E}}_\sigma \Leftrightarrow g^{\mathcal{F}}_\sigma$ for all $\sigma \in \Sigma_{\mathcal{E}} = \Sigma_{\mathcal{F}}$, and $\rightarrow^i_{\mathcal{F}} = \{(x, \sigma, g^{\mathcal{F}}_\sigma, u, y) \mid (x, \sigma, g^{\mathcal{E}}_\sigma, u, y) \in \rightarrow^i_{\mathcal{E}}\}$. Then refinement function $\xi = \mathrm{id}$ ensures that $(\mathcal{E}, \xi_1) \simeq_{co} (\mathcal{F}, \xi_1 \circ \xi)$.*

Proof. From the definition of Ξ and the construction of ξ, it follows directly that $\xi \in \Xi$. Therefore, $\xi_1 \circ \xi \in \Xi$ and $(\mathcal{F}, \xi_1 \circ \xi)$ is a coordinator tuple.

By construction of $\mathcal{F}$ it follows that $\parallel \mathcal{E} \Leftrightarrow \parallel \mathcal{F}$. Therefore, from Lemma 3 it follows that $\sup CN(\mathcal{E}) = \mathrm{SSEFA}(\parallel \mathcal{E}) \Leftrightarrow \mathrm{SSEFA}(\parallel \mathcal{F}) = \sup CN(\mathcal{F})$. And from

Lemma 8 it follows that $\xi_1(\sup CN(\mathcal{E})) \Leftrightarrow \xi_1(\sup CN(\mathcal{F}))$. Thus, from Lemma 4 it follows that $U(\xi_1(\sup CN(\mathcal{E}))) = U(\xi_1(\sup CN(\mathcal{F})))$.

Then, by rewriting, we can show the following.

$$\begin{aligned}\mathcal{L}(\xi_1(\sup CN(\mathcal{E}))) &= \mathcal{L}(U(\xi_1(\sup CN(\mathcal{E}))))\\ &= \mathcal{L}(U(\xi_1(\sup CN(\mathcal{F}))))\\ &= \mathcal{L}(\xi_1(\sup CN(\mathcal{F})))\\ &= \mathcal{L}(\xi_1(\xi(\sup CN(\mathcal{F}))))\end{aligned}$$

This concludes the proof. □

6 Variable unfolding

Lemma 18. *Let E be a deterministic normalized EFA with variable set V and $z \in V$. Then for each edge $e = (l_1, \sigma, g_\sigma, u_\sigma, l_2)$ in E there exists in $\rho_z(SSEFA(E)_{\setminus z}) \parallel E$ a set of edges $\{((a, l_1, l_1), \sigma, g^*_\sigma[z \mapsto a, z' \mapsto b] \wedge b = u_\sigma(z)[z \mapsto a] \wedge g_\sigma, u_\sigma, (b, l_2, l_2)) \mid a, b \in \mathrm{dom}(z)\}$ if $\sigma \in \Sigma_z$, or a single edge $((a, l_1, l_1), \sigma, g^*_\sigma \wedge g_\sigma, u_\sigma, (a, l_2, l_2))$ with $a \in \mathrm{dom}(z)$ if $\sigma \notin \Sigma_z$.*

Proof. From Algorithm 1 we have that if $e = (l_1, \sigma, g_\sigma, u_\sigma, l_2)$ is an edge in E, then $(l_1, \sigma, g^*_\sigma, u_\sigma, l_2)$ is an edge in $\mathrm{SSEFA}(E)$. From the definition of variable unfolding, it follows that $\mathrm{SSEFA}(E)_{\setminus z} = \{U_{\mathrm{SSEFA}(E)}(z), U_z(\mathrm{SSEFA}(E))\}$. This results for edge e in the set of edges $\{((a, l_1), (\sigma, a, b), g^*_\sigma[z \mapsto a, z' \mapsto b] \wedge b = u_\sigma(z)[z \mapsto a], u_{\sigma\setminus z}, (b, l_2)) \mid a, b \in \mathrm{dom}(z)\}$ if $\sigma \in \Sigma_z$ or in the edge $((a, l_1), (\sigma, a, a), g^*_\sigma, u_{\sigma\setminus z}, (a, l_2))$ with $a \in \mathrm{dom}(z)$ if $\sigma \notin \Sigma_z$. After applying renaming ρ_z, it follows that for edge e we have the set of edges $\{((a, l_1), \sigma, g^*_\sigma[z \mapsto a, z' \mapsto b] \wedge b = u_\sigma(z)[z \mapsto a], u_{\sigma\setminus z}, (b, l_2)) \mid a, b \in \mathrm{dom}(z)\}$ if $\sigma \in \Sigma_z$ or in the edge $((a, l_1), \sigma, g^*_\sigma, u_{\sigma\setminus z}, (a, l_2))$ with $a \in \mathrm{dom}(z)$ if $\sigma \notin \Sigma_z$. Finally, in the system $\rho_z(\mathrm{SSEFA}(E)_{\setminus z}) \parallel E$ we have for edge e in E the set of edges $\{((a, l_1, l_1), \sigma, g^*_\sigma[z \mapsto a, z' \mapsto b] \wedge b = u_\sigma(z)[z \mapsto a] \wedge g_\sigma, u_\sigma, (b, l_2, l_2)) \mid a, b \in \mathrm{dom}(z)\}$ if $\sigma \in \Sigma_z$, or a single edge $((a, l_1, l_1), \sigma, g^*_\sigma \wedge g_\sigma, u_\sigma, (a, l_2, l_2))$ with $a \in \mathrm{dom}(z)$ if $\sigma \notin \Sigma_z$. □

Lemma 19. *Let $\mathcal{E}$ be a deterministic normalized EFA system with variable set V and $z \in V$. Then $SSEFA(\mathcal{E}) \underline{\leftrightarrow}_V \rho_z(SSEFA(\mathcal{E})_{\setminus z}) \parallel \mathcal{E}$.*

Proof. Observe that the initial location of $\mathrm{SSEFA}(\mathcal{E})$ is l_0 and the initial valuation $\hat{v}_0$. Thus, the initial location of $\rho_z(\mathrm{SSEFA}(\mathcal{E})_{\setminus z})$ is $(\hat{v}_0(z), l_0)$ and initial valuation is $\hat{v}_{0\setminus z}$. Therefore, the initial location of $\rho_z(\mathrm{SSEFA}(\mathcal{E})_{\setminus z}) \parallel \mathcal{E}$ is $(\hat{v}_0(z), l_0, l_0)$ and initial valuation is $\hat{v}_0$.

Let the relation R be defined as $R = \{(x, (a, x, x), \hat{v}) \mid x \in L^{\mathcal{E}}, \hat{v} \in \mathrm{Val}(V), a = \hat{v}(z)\}$. We will show that this is a valuation bisimulation relation.

Consider the triple $(x, (a, x, x), \hat{v}) \in R$.

- Let $(x, \hat{v}) \xrightarrow{\sigma} (y, \hat{w})$ be an edge in $U(\mathrm{SSEFA}(\mathcal{E}))$ for some y and $\hat{w}$. It then holds that $x \xrightarrow{\sigma, g^*_\sigma, u} y$ is an edge in $\mathrm{SSEFA}(\mathcal{E})$ with $g^*_\sigma[\hat{v}] = \mathbf{T}$ and $\hat{w}(v) =$

$\hat{v}(u(v))$. From the construction of SSEFA($\mathcal{E}$) it follows that $x \xrightarrow{\sigma, g_\sigma, u} y$ is an edge in $\mathcal{E}$ with $g^*_\sigma \preceq g_\sigma$, i.e., $g_\sigma[\hat{v}] = \mathbf{T}$, and $\hat{w}(v) = \hat{v}(u(v))$. Now consider two cases for σ.

- If $\sigma \in \Sigma_z$, then from Lemma 18 it follows that there exists a set of edges $\{((a, x, x), \sigma, g^*_\sigma[z \mapsto a, z' \mapsto b] \land b = u_\sigma(z)[z \mapsto a] \land g_\sigma, u, (b, y, y)) \mid a, b \in \mathrm{dom}(z)\}$ in $\rho_z(\mathrm{SSEFA}(\mathcal{E})_{\setminus z}) \parallel \mathcal{E}$. As $\mathcal{E}$ is deterministic, there exists at most one pair of values for a, b such that $(g^*_\sigma[z \mapsto a, z' \mapsto b] \land b = u_\sigma(z)[z \mapsto a] \land g_\sigma)[\hat{v}, \hat{w}] = \mathbf{T}$. From the construction of R it follows that $\hat{v}(z) = a$ and, with an unchanged update $\hat{w}(v) = \hat{v}(u(v))$, it follows that $\hat{w}(z) = b$, resulting in $(g^*_\sigma[z \mapsto a, z' \mapsto b] \land b = u_\sigma(z)[z \mapsto a])[\hat{v}, \hat{w}] = \mathbf{T}$. We can conclude that $(a, x, x) \xrightarrow{\sigma, g^*_\sigma[z \mapsto a, z' \mapsto b] \land b = u_\sigma(z)[z \mapsto a] \land g_\sigma, u} (b, y, y)$ is an edge in $\rho_z(\mathrm{SSEFA}(\mathcal{E})_{\setminus z}) \parallel \mathcal{E}$ with $(g^*_\sigma[z \mapsto a, z' \mapsto b] \land b = u_\sigma(z)[z \mapsto a] \land g_\sigma)[\hat{v}, \hat{w}] = \mathbf{T}$ and $\hat{w}(v) = \hat{v}(u(v))$.
- If $\sigma \notin \Sigma_z$, then from Lemma 18 it follows that there exists an edge $(a, x, x) \xrightarrow{\sigma, g^*_\sigma \land g_\sigma, u} (a, y, y))$ for some $a \in \mathrm{dom}(z)$ in $\rho_z(\mathrm{SSEFA}(\mathcal{E})_{\setminus z} \parallel \mathcal{E}$. From the construction of R it follows that $\hat{v}(z) = a$. Furthermore, as $g^*_\sigma[\hat{v}] = \mathbf{T}$ and $g_\sigma[\hat{v}] = \mathbf{T}$, it follows that $(g^*_\sigma \land g_\sigma)[\hat{v}, \hat{w}] = \mathbf{T}$. Finally, as u is unchanged, it follows that $\hat{w}(v) = \hat{v}(u(v))$ still holds.

In both cases, it has been shown that $(a, x, x) \xrightarrow{\sigma, g', u} (b, y, y)$ is an edge in $\rho_z(\mathrm{SSEFA}(\mathcal{E})_{\setminus z}) \parallel \mathcal{E}$ with $g'[\hat{v}, \hat{w}] = \mathbf{T}$ and $\hat{w}(v) = \hat{v}(u(v))$. Calculating the state space, it follows that $(a, x, x, \hat{v}) \xrightarrow{\sigma} (b, y, y, \hat{w})$ is an edge in $U(\rho_z(\mathrm{SSEFA}(\mathcal{E})_{\setminus z}) \parallel \mathcal{E})$ with $\hat{v}(z) = a$ and $\hat{w}(z) = b$. Finally, observe by construction of R that $(y, (b, y, y), \hat{w}) \in R$.

- Let $(a, x, x, \hat{v}) \xrightarrow{\sigma} (b, y, y, \hat{w})$ be an edge in $U(\rho_z(\mathrm{SSEFA}(\mathcal{E})_{\setminus z}) \parallel E)$ with $\hat{v}(z) = a$ and some b, y, and $\hat{w}$. It then holds that $(a, x, x) \xrightarrow{\sigma, g, u} (b, y, y)$ is an edge in $\rho_z(\mathrm{SSEFA}(\mathcal{E})_{\setminus z}) \parallel E$ with $g[\hat{v}, \hat{w}] = \mathbf{T}$ and $\hat{w}(v) = \hat{v}(u(v))$. Consider two cases for σ.

  - If $\sigma \in \Sigma_z$, it follows from the definition of synchronous product and variable unfolding that $g = g^*_\sigma[z \mapsto a, z' \mapsto b] \land b = u_\sigma(z)[z \mapsto a] \land g_\sigma$, $(a, x) \xrightarrow{\sigma, g^*_\sigma[z \mapsto a, z' \mapsto b] \land b = u_\sigma(z)[z \mapsto a], u_{\setminus z}[z \mapsto a]} (b, y)$ is an edge in $\rho_z(\mathrm{SSEFA}(\mathcal{E})_{\setminus z})$, and $x \xrightarrow{\sigma, g_\sigma, u} y$ is an edge in $\mathcal{E}$. Furthermore, from the construction of the normalized variable EFA, it follows that $b = \hat{w}(z) = \hat{v}(u(z))$. Combining this with the fact that $g[\hat{v}, \hat{w}] = \mathbf{T}$ and $a = \hat{v}(z)$, it follows that $(g^*_\sigma[z \mapsto a, z' \mapsto b] \land b = u_\sigma(z)[z \mapsto a])[\hat{v}, \hat{w}] = g^*_\sigma[\hat{v}, \hat{w}] = \mathbf{T}$.
  - If $\sigma \notin \Sigma_z$, it follows from the definition of synchronous product and variable unfolding that $g = g^*_\sigma \land g_\sigma$, $(a, x) \xrightarrow{\sigma, g^*_\sigma, u} (b, y)$ is an edge in $\rho_z(\mathrm{SSEFA}(\mathcal{E})_{\setminus z})$, and $x \xrightarrow{\sigma, g_\sigma, u} y$ is an edge in $\mathcal{E}$. Furthermore, from the construction of the normalized variable EFA, it follows that

$b = a = \hat{w}(z)$. Combining this with the fact that $g[\hat{v}, \hat{w}] = \mathbf{T}$, it follows that $g^*_\sigma[\hat{v}, \hat{w}] = \mathbf{T}$.

As $x \xrightarrow{\sigma, g_\sigma, u} y$ is an edge in $\mathcal{E}$ in both cases above, it follows by construction that $x \xrightarrow{\sigma, g^*_\sigma, u} y$ is an edge in SSEFA($\mathcal{E}$). Furthermore, in both cases we have that $g^*_\sigma[\hat{v}, \hat{w}] = \mathbf{T}$ and $\hat{w}(v) = \hat{v}(u(v))$. Finally, it follows that $(x, \hat{v}) \xrightarrow{\sigma} (y, \hat{w})$ is an edge in $U(\text{SSEFA}(\mathcal{E}))$, and from the construction of R that $(y, (b, y, y), \hat{w}) \in R$.

- Let $x \in L_m$ in SSEFA($\mathcal{E}$). As SSEFA($\mathcal{E}$) is a subautomaton of $\mathcal{E}$, it follows that $x \in L_m$ in $\mathcal{E}$. Furthermore, since renaming and variable unfolding do not change the marking of locations, and it is assumed that all valuations are marked, it follows that $(a, x) \in L_m$ in $\rho_z(\text{SSEFA}(\mathcal{E})_{\setminus z})$. From the definition of synchronous product it follows that $(a, x, x) \in L_m$ in $\rho_z(\text{SSEFA}(\mathcal{E})_{\setminus z}) \parallel \mathcal{E}$.
- Let $(a, x, x) \in L_m$ in $\rho_z(\text{SSEFA}(\mathcal{E})_{\setminus z}) \parallel \mathcal{E}$. It follows from the definition of synchronous product that $(a, x) \in L_m$ in $\rho_z(\text{SSEFA}(\mathcal{E})_{\setminus z})$ and $x \in L_m$ in $\mathcal{E}$. As SSEFA($\mathcal{E}$) is a subautomaton of $\mathcal{E}$, it follows that $x \in L_m$ in SSEFA($\mathcal{E}$).

This shows that R is a valuation bisimulation relation. As the initial locations and valuation are related, i.e., $(l_0, (\hat{v}_0(z), l_0, l_0), \hat{v}_0) \in R$, it follows that SSEFA($\mathcal{E}$) $\underline{\leftrightarrow}_V$ $\rho_z(\text{SSEFA}(\mathcal{E})_{\setminus z}) \parallel \mathcal{E}$. This concludes the proof. □

Lemma 20. *Let $\mathcal{E}$ be a deterministic normalized EFA system with variable set V and $z \in V$. Then $\mathcal{E} \underline{\leftrightarrow}_V \rho_z(\mathcal{E}_{\setminus z}) \parallel \mathcal{E}$.*

Proof. We can follow the proof of Lemma 19, where we replace instances of SSEFA($\mathcal{E}$) with $\mathcal{E}$ and g^*_σ with g_σ to prove this lemma. The reason we can follow the proof exactly is that SSEFA($\mathcal{E}$) is a subautomaton of $\mathcal{E}$ according to the definition of a supervisor. □

Lemma 21. *Let $E = (L, V, \Sigma, \rightarrow, l_0, v_0, L_m)$ be an EFA and $\rho : \Sigma \rightarrow \Sigma'$ a renaming. Then $\mathcal{L}(\rho(E)) = \rho(\mathcal{L}(E))$.*

Proof. In case that $L = \emptyset$, i.e., E is an empty automaton, the claim holds trivially as the language of an empty automaton is by definition the empty set.

Observe that $\mathcal{L}(\rho(E)) = \mathcal{L}(U(\rho(E)))$ and $\rho(\mathcal{L}(E)) = \rho(\mathcal{L}(U(E)))$ from the definition of $\mathcal{L}$ for EFAs. This lemma is proven by showing that $\mathcal{L}(\rho(E)) \subseteq \rho(\mathcal{L}(E))$ and $\rho(\mathcal{L}(E)) \subseteq \mathcal{L}(\rho(E))$.

First, we show that $\mathcal{L}(\rho(E)) \subseteq \rho(\mathcal{L}(E))$. Let $s = \sigma_1\sigma_2 \ldots \sigma_n \in \mathcal{L}(\rho(E))$. This implies that $(l_0, \hat{v}_0) \xrightarrow{\sigma_1} (l_1, \hat{v}_1) \xrightarrow{\sigma_2} \ldots \xrightarrow{\sigma_n} (l_n, \hat{v}_n)$ in $U(\rho(E))$. By induction on the length i of the prefix $s_i = \sigma_1\sigma_2 \ldots \sigma_i$ we show that $s \in \rho(\mathcal{L}(E))$.

Base case. Let $i = 0$ and $s_0 = \varepsilon$. As $\rho(\varepsilon) = \epsilon$ and ε is in the language of any nonempty automaton, it follows directly that $s_0 \in \rho(\mathcal{L}(E))$.

Inductive step. Let $s_i \in \rho(\mathcal{L}(E))$ be the induction hypothesis, i.e., there exists a string $m_i = \mu_1\mu_2 \ldots \mu_i \in \mathcal{L}(E)$ such that $\rho(m_i) = s_i$.

The transition $(l_i, \hat{v}_i) \xrightarrow{\sigma_{i+1}} (l_{i+1}, \hat{v}_{i+1})$ in $U(\rho(E))$ implies that there exists a transition $l_i \xrightarrow{\sigma_{i+1}, g_{i+1}, u_{i+1}} l_{i+1}$ in $\rho(E)$ such that $g_{i+1}[\hat{v}_i] = \mathbf{T}$ and $\hat{v}_{i+1}(v) = \hat{v}_i(u_{i+1}(v))$. From the definition of the renaming ρ it follows that there exists a $\mu_{i+1} \in \Sigma$ such that $\rho(\mu_{i+1}) = \sigma_{i+1}$ and $l_i \xrightarrow{\mu_{i+1}, g_{i+1}, u_{i+1}} l_{i+1}$ in E. Thus, $(l_i, \hat{v}_i) \xrightarrow{\mu_{i+1}} (l_{i+1}, \hat{v}_{i+1})$ is a transition in $U(E)$. With the induction hypothesis it follows that $\mu_1\mu_2 \ldots \mu_i\mu_{i+1} \in \mathcal{L}(U(E))$ and $\rho(\mu_1\mu_2 \ldots \mu_i\mu_{i+1}) = s_i\sigma_{i+1} \in \rho(\mathcal{L}(U(E)))$.

As string s is chosen arbitrarily, it follows that $\mathcal{L}(\rho(E)) \subseteq \rho(\mathcal{L}(E))$.

Secondly, we show that $\rho(\mathcal{L}(E)) \subseteq \mathcal{L}(\rho(E))$. Let $s = \sigma_1\sigma_2 \ldots \sigma_n \in \rho(\mathcal{L}(E))$. From the definition of renaming ρ it follows that there exists a string $m = \mu_1\mu_2 \ldots \mu_n \in \mathcal{L}(E)$ such that $\rho(m) = s$. As $m \in \mathcal{L}(E)$, it implies that path $(l_0, \hat{v}_0) \xrightarrow{\mu_1} (l_1, \hat{v}_1) \xrightarrow{\mu_2} \ldots \xrightarrow{\mu_n} (l_n, \hat{v}_n)$ is in $U(E)$. From the definition of state space, it follows that path $l_0 \xrightarrow{\mu_1, g_1, u_1} l_1 \xrightarrow{\mu_2, g_2, u_2} \ldots \xrightarrow{\mu_n, g_n, u_n} l_n$ is in E, where for each transition i it holds that $g_i[\hat{v}_{i-1}] = \mathbf{T}$ and $\hat{v}_i(v) = \hat{v}_{i+1}(u_i(v))$. Applying renaming ρ on this path results in $l_0 \xrightarrow{\rho(\mu_1), g_1, u_1} l_1 \xrightarrow{\rho(\mu_2), g_2, u_2} \ldots \xrightarrow{\rho(\mu_n), g_n, u_n} l_n$ in $\rho(E)$, where for each transition i it still holds that $g_i[\hat{v}_{i-1}] = \mathbf{T}$ and $\hat{v}_i(v) = \hat{v}_{i+1}(u_i(v))$. Therefore, $(l_0, \hat{v}_0) \xrightarrow{\rho(\mu_1)} (l_1, \hat{v}_1) \xrightarrow{\rho(\mu_2)} \ldots \xrightarrow{\rho(\mu_n)} (l_n, \hat{v}_n)$ is a path in $U(\rho(E))$ and $\rho(\mu_1)\rho(\mu_2) \ldots \rho(\mu_n) = \rho(\mu_1\mu_2 \ldots \mu_n) = \rho(m) = s \in \mathcal{L}(U(\rho(E))) = \mathcal{L}(\rho(E))$. As string s is chosen arbitrarily, it follows that $\rho(\mathcal{L}(E)) \subseteq \mathcal{L}(\rho(E))$. □

Lemma 22. *Let E be a deterministic normalized EFA with variable set V. Let $z \in V$. Then* $(\sup CN(E))_{\setminus z} \Leftrightarrow_{V \setminus \{z\}} \sup CN(E_{\setminus z})$.

Proof. Let $E = (L, \Sigma, V \to, l_0, \check{v}_0, L_m)$. After unfolding variable z it follows that the initial location of $E_{\setminus z} = \| \{U_E(z), U_z(E)\}$ is $(\check{v}_0(z), l_0)$. Combining this with Lemma 15 of Mohajerani et al. [2016], which states that $(a, x, \hat{v}) \xrightarrow{\sigma} (b, y, \hat{w})$ in $\rho_z(U(E_{\setminus z}))$ if and only if $(x, \hat{v} \oplus \{z \mapsto a\}) \xrightarrow{\sigma} (y, \hat{w} \oplus \{z \mapsto b\})$ in $U(E)$, it follows that for any EFA T $\mathcal{L}(T) = \rho_z(\mathcal{L}(T_{\setminus z}))$. Therefore, it follows that

$$\mathcal{L}(\sup CN(E)) = \rho_z(\mathcal{L}((\sup CN(E))_{\setminus z}))$$

As $\sup CN(E)$ results in the maximally permissive supervisor for E, we can replace E by its state-space finite automata or even by its language and then calculate the maximally permissive supervisor based on the finite automata or language, respectively, rather than the EFA representation: $\mathcal{L}(\sup CN(E)) = \mathcal{L}(\sup CN_F(U(E))) = \sup CN_L(\mathcal{L}(E))$ where $\sup CN_F$ and $\sup CN_L$ are based on finite automata and languages, respectively. Furthermore, Lemma 13 of Mohajerani et al. [2014b] shows that renaming and $\sup CN_F$ can be changed, i.e., for any finite automata T and renaming ρ it holds that $\rho(\sup CN_F(T)) =$

$\sup CN_F(\rho(T))$. Therefore, we can show the following.

$$\begin{aligned}
\mathcal{L}(\sup CN(E)) &= \sup CN_L(\mathcal{L}(E)) \\
&= \sup CN_L(\rho_z(\mathcal{L}(E_{\setminus z}))) \\
&= \sup CN_L(\rho_z(\mathcal{L}(U(E_{\setminus z})))) \\
&= \sup CN_L(\mathcal{L}(\rho_z(U(E_{\setminus z})))) \text{ by Lemma 21} \\
&= \mathcal{L}(\sup CN_F(\rho_z(U(E_{\setminus z})))) \\
&= \mathcal{L}(\rho_z(\sup CN_F(U(E_{\setminus z})))) \text{ by Lemma 21} \\
&= \rho_z(\mathcal{L}(\sup CN_F(U(E_{\setminus z})))) \\
&= \rho_z(\sup CN_L(\mathcal{L}(U(E_{\setminus z})))) \\
&= \rho_z(\sup CN_L(\mathcal{L}(E_{\setminus z}))) \\
&= \rho_z(\mathcal{L}(\sup CN(E_{\setminus z})))
\end{aligned}$$

Therefore, we can conclude that $\rho_z(\mathcal{L}((\sup CN(E))_{\setminus z})) = \rho_z(\mathcal{L}(\sup CN(E_{\setminus z})))$ and thus $\mathcal{L}((\sup CN(E))_{\setminus z}) = \mathcal{L}(\sup CN(E_{\setminus z}))$.

Combining the definitions of supervisor and variable unfolding, it follows that $(\sup CN(E))_{\setminus z}$ and $\sup CN(E_{\setminus z})$ have the same location set, alphabet, variable set, initial location, initial valuation, and marked states. Furthermore, the sets of edges are similar: there is an edge $((a,x),\sigma,g_1,u,(b,y))$ in $(\sup CN(E))_{\setminus z}$ if and only if there is an edge $((a,x),\sigma,g_2,u,(b,y))$ in $\sup CN(E_{\setminus z})$. Notice that the only difference is the guards on these edges. Furthermore, as E is deterministic, both $(\sup CN(E))_{\setminus z}$ and $\sup CN(E_{\setminus z})$ are also deterministic.

Combining the above observations, it follows that location $(a,x,\hat{v})$ is reached in $U((\sup CN(E))_{\setminus z})$ if and only if the same state is reached in $U(\sup CN(E_{\setminus z}))$. In other words, $\sup CN(E))_{\setminus z}$ and $\sup CN(E_{\setminus z})$ are synchronized with their locations and valuations. Now assume that some location (a,x) is reached in both $\sup CN(E))_{\setminus z}$ and $\sup CN(E_{\setminus z})$. Consider each valuation $\hat{v}$ and each edge $e_* = ((a,x),\sigma,g_*,u,(b,y))$ where $* = 1$ for $\sup CN(E))_{\setminus z}$ and $* = 2$ for $\sup CN(E_{\setminus z})$. As their languages are the same, it holds that edge e_1 is enabled if and only if e_2 is enabled. Therefore, $g_1[\hat{v}] = \mathbf{T}$ if and only if $g_2[\hat{v}] = \mathbf{T}$. As we have chosen valuation $\hat{v}$ arbitrarily, it follows that $g_1 \Leftrightarrow_{V\setminus z} g_2$. Furthermore, as the location and edge is chosen arbitrarily, it follows that $\sup CN(E))_{\setminus z} \Leftrightarrow_{V\setminus z} \sup CN(E_{\setminus z})$. This concludes the proof. □

Theorem 6. *Let $(\mathcal{E},\xi_1)$ be a coordinator tuple with $\mathcal{E}$ a deterministic normalized EFA system with variable set V and $z \in V$. Then refinement function $\xi(\mathcal{G}) = \rho_z(\mathcal{G}) \parallel \mathcal{E}$ for any EFA system $\mathcal{G}$ ensures that $(\mathcal{E},\xi_1) \simeq_{co} (\mathcal{E}_{\setminus z},\xi_1 \circ \xi)$.*

Proof. From the definition of Ξ and the construction of ξ, it follows directly that $\xi \in \Xi$. Therefore, $\xi_1 \circ \xi \in \Xi$ and $(\mathcal{E}_{\setminus z},\xi_1 \circ \xi)$ is a coordinator tuple.

Now we show that the two coordinator tuples are coordinator equivalent.

By rewriting, we can show the following.

$$\begin{aligned}\mathcal{L}(\xi_1(\xi(\sup CN(\mathcal{E}_{\setminus z})))) &= \mathcal{L}(\xi_1(\xi(\text{SSEFA}(\mathcal{E}_{\setminus z}))))\\ &= \mathcal{L}(U(\xi_1(\xi(\text{SSEFA}(\mathcal{E}_{\setminus z}))))) \text{ by definition of language}\\ &= \mathcal{L}(U(\xi_1(\xi(\text{SSEFA}(\mathcal{E})_{\setminus z})))) \text{ by Lemmas 22 and 4}\\ &= \mathcal{L}(\xi_1(\xi(\text{SSEFA}(\mathcal{E})_{\setminus z})))\\ &= \mathcal{L}(\xi_1(\rho_z(\text{SSEFA}(\mathcal{E})_{\setminus z}) \parallel \mathcal{E}))\\ &= \mathcal{L}(\xi_1(\text{SSEFA}(\mathcal{E}))) \text{ by Lemmas 16 and 19}\\ &= \mathcal{L}(\xi_1(\sup CN(\mathcal{E})))\end{aligned}$$

This concludes the proof. □

7 False removal

7.1 Description of the abstraction

It may happen that after variable unfolding and update simplification, several events have a false guard, i.e., transitions labeled with these events are never enabled. Furthermore, it could be that the synchronous composition of two EFAs may result in having an event in the alphabet, but no transition in the synchronous composition is labeled with this event. In this case, the event is also never executed. These events can be safely removed from the EFA system without altering the behavior of the system.

Events (and transitions labeled with these events) can be removed from an EFA by restricting it to an alphabet $\Sigma' \subseteq \Sigma$ (see Mohajerani et al. [2016]).

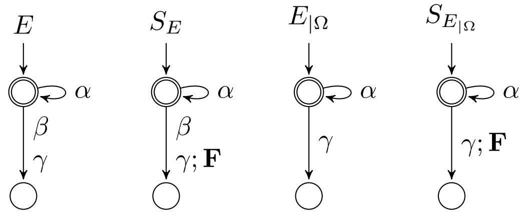
Definition 8 (Restriction)**.** *Let $E = (L, V, \Sigma, \rightarrow, l_0, v_0, L_m)$ be an EFA. The restriction of E with respect to Σ' is $E_{|\Sigma'} = (L, V, \Sigma' \cap \Sigma, \rightarrow_{|\Sigma'}, l_0, v_0, L_m)$ where $\rightarrow_{|\Sigma'} = \{(l_1, \sigma, g, u, l_2) \mid (l_1, \sigma, g, u, l_2) \in \rightarrow, \sigma \in \Sigma'\}$. The restriction of EFA system $\mathcal{E} = \{E^1, \ldots, E^n\}$ with respect to Σ', denoted with $\mathcal{E}_{|\Sigma'}$, is $\mathcal{E}_{|\Sigma'} = \{E^1_{|\Sigma'}, \ldots, E^n_{|\Sigma'}\}$.*

Theorem 7 (False removal)**.** *Let $(\mathcal{E}, \xi_1)$ be a coordinator tuple with $\mathcal{E}$ a deterministic normalized EFA system, and let $\Sigma_{\mathcal{E}} = \Omega \mathbin{\dot\cup} \Lambda$ such that for all $\lambda \in \Lambda$ at least one of the following conditions holds:*

1. *$g_\lambda \equiv \boldsymbol{F}$, or*
2. *there exists an $E^i \in \mathcal{E}$ such that $\lambda \in \Sigma^i$, but there does not exist any transition $x \xrightarrow{\lambda, g_\lambda, u_\lambda} y$ in E^i.*

Then refinement function $\xi = \text{id}$ ensures that $(\mathcal{E}, \xi_1) \simeq_{co} (\mathcal{E}_{|\Omega}, \xi_1 \circ \xi)$.

Theorem 7 shows that when events are removed from the EFA system because they are never enabled, the abstracted coordinator does not need to be



Event	Guard and update
α	**T**
β	**F**
γ	**T**

Fig. 3: Example of false removal and coordinator refinement.

changed. The proof of this theorem can be found in Section 7.2 of this supplementary material.

Example. Figure 3 shows an example of false removal. Event β in E has a false guard and is therefore never enabled. Therefore, this event is removed, resulting in $E_{|\Omega}$. The coordinator for E, automaton S_E in Figure 3, only has to strengthen the guard of γ to prevent the system from reaching the blocking location. The same holds for the abstracted coordinator $S_{E_{|\Omega}}$. The languages of the original and the abstracted coordinators are the same, so no refinement is needed.

7.2 Proof

Lemma 23. *Let $\mathcal{E}$ be a deterministic normalized EFA system, and let $\Sigma_{\mathcal{E}} = \Omega \mathbin{\dot{\cup}} \Lambda$ such that for all $\lambda \in \Lambda$ at least one of the following conditions holds:*

1. *$g_\lambda \equiv \boldsymbol{F}$, or*
2. *there exists an $E^i \in \mathcal{E}$ such that $\lambda \in \Sigma^i$, but there does not exist any transition $x \xrightarrow{\lambda, g_\lambda, u_\lambda} y$ in E^i.*

Then $\mathcal{L}(U(\mathcal{E})) = \mathcal{L}(U(\mathcal{E}_{|\Omega}))$.

Proof. From Lemma 16 of Mohajerani et al. [2016] it follows that $(x_1, \ldots, x_n, \hat{v}) \xrightarrow{\sigma} (y_1, \ldots, y_n, \hat{w})$ in $U(\mathcal{E}_{|\Omega})$ implies $(x_1, \ldots, x_n, \hat{v} \oplus \hat{u}) \xrightarrow{\sigma} (y_1, \ldots, y_n, \hat{w} \oplus \hat{u})$ in $U(\mathcal{E})$ where $\hat{u} \in dom(vars(\mathcal{E}) \setminus vars(\mathcal{E}_{|\Omega}))$.

From Lemma 17 of Mohajerani et al. [2016] it follows that $(x_1, \ldots, x_n, \hat{v}) \xrightarrow{\sigma} (y_1, \ldots, y_n, \hat{w})$ in $U(\mathcal{E})$ implies $(x_1, \ldots, x_n, \hat{v}_{|W}) \xrightarrow{\sigma} (y_1, \ldots, y_n, \hat{w}_{|W})$ in $U(\mathcal{E}_{|\Omega})$ where $W = vars(\mathcal{E}_{|\Omega}))$.

Therefore, each transition in one system can be matched with a transition in the other system. As the initial locations of $U(\mathcal{E})$ and $U(\mathcal{E}_{|\Omega})$ are the same, it follows that $\mathcal{L}(U(\mathcal{E})) = \mathcal{L}(U(\mathcal{E}_{|\Omega}))$. This concludes the proof. □

Lemma 24. *Let E, F be two EFA and $\Omega \subseteq \Sigma^E \cup \Sigma^F$. Then $(E \parallel F)_{|\Omega} = E_{|\Omega} \parallel F_{|\Omega}$.*

Proof. It is clear that $(E \parallel F)_{|\Omega}$ and $E_{|\Omega} \parallel F_{|\Omega}$ have the same location set, variable set, alphabet, initial location, initial valuation, and marked locations. It remains to be proven that they have the same transitions.

Assume that $(l_1^E, l_1^F) \xrightarrow{\sigma,g,u} (l_2^E, l_2^F)$ is a transition in $(E \parallel F)_{|\Omega}$. Clearly, as $\sigma \in \Omega$ it follows that $(l_1^E, l_1^F) \xrightarrow{\sigma,g,u} (l_2^E, l_2^F)$ in $E \parallel F$. Consider three cases for σ.

- $\sigma \in \Sigma^E \cap \Sigma^F$. Then by the definition of synchronous composition it follows that $l_1^E \xrightarrow{\sigma,g^E,u^E} l_2^E$ in E, $l_1^F \xrightarrow{\sigma,g^F,u^F} l_2^F$ in F, $g = g^E \wedge g^F$, and $u = u^E \oplus u^F$. Applying the restriction on E and F, it follows that $l_1^E \xrightarrow{\sigma,g^E,u^E} l_2^E$ in $E_{|\Omega}$, $l_1^F \xrightarrow{\sigma,g^F,u^F} l_2^F$ in $F_{|\Omega}$, and the rest remains unchanged.
- $\sigma \in \Sigma^E \setminus \Sigma^F$. Then by the definition of synchronous composition it follows that $l_1^E \xrightarrow{\sigma,g^E,u^E} l_2^E$ in E, $l_1^F = l_2^F$, $g = g^E$, and $u = u^E$. Applying the restriction on E and F, it follows that $l_1^E \xrightarrow{\sigma,g^E,u^E} l_2^E$ in $E_{|\Omega}$, and the rest remains unchanged.
- $\sigma \in \Sigma^F \setminus \Sigma^E$. Then by the definition of synchronous composition it follows that $l_1^F \xrightarrow{\sigma,g^F,u^F} l_2^F$ in F, $l_1^E = l_2^E$, $g = g^F$, and $u = u^F$. Applying the restriction on F and E, it follows that $l_1^F \xrightarrow{\sigma,g^F,u^F} l_2^F$ in $F_{|\Omega}$, and the rest remains unchanged.

Applying the definition of synchronous composition on the three cases above, it follows that $(l_1^E, l_1^F) \xrightarrow{\sigma,g,u} (l_2^E, l_2^F)$ is a transition in $E_{|\Omega} \parallel F_{|\Omega}$.

Conversely, assume that $(l_1^E, l_1^F) \xrightarrow{\sigma,g,u} (l_2^E, l_2^F)$ is a transition in $E_{|\Omega} \parallel F_{|\Omega}$. Clearly, $\sigma \in \Omega$. Consider three cases for σ.

- $\sigma \in \Sigma^E \cap \Sigma^F$. Then by the definition of synchronous composition it follows that $l_1^E \xrightarrow{\sigma,g^E,u^E} l_2^E$ in $E_{|\Omega}$, $l_1^F \xrightarrow{\sigma,g^F,u^F} l_2^F$ in $F_{|\Omega}$, $g = g^E \wedge g^F$, and $u = u^E \oplus u^F$. As $\sigma \in \Omega$, it follows that $l_1^E \xrightarrow{\sigma,g^E,u^E} l_2^E$ in E, $l_1^F \xrightarrow{\sigma,g^F,u^F} l_2^F$ in F, and the rest remains unchanged.
- $\sigma \in \Sigma^E \setminus \Sigma^F$. Then by the definition of synchronous composition it follows that $l_1^E \xrightarrow{\sigma,g^E,u^E} l_2^E$ in $E_{|\Omega}$, $l_1^F = l_2^F$, $g = g^E$, and $u = u^E$. As $\sigma \in \Omega$, it follows that $l_1^E \xrightarrow{\sigma,g^E,u^E} l_2^E$ in E, and the rest remains unchanged.
- $\sigma \in \Sigma^F \setminus \Sigma^E$. Then by the definition of synchronous composition it follows that $l_1^F \xrightarrow{\sigma,g^F,u^F} l_2^F$ in $F_{|\Omega}$, $l_1^E = l_2^E$, $g = g^F$, and $u = u^F$. As $\sigma \in \Omega$, it follows that $l_1^F \xrightarrow{\sigma,g^F,u^F} l_2^F$ in F, and the rest remains unchanged.

Applying the definition of synchronous composition on the three cases above, it follows that $(l_1^E, l_1^F) \xrightarrow{\sigma,g,u} (l_2^E, l_2^F)$ is a transition in $E \parallel F$. Applying the restriction on $E \parallel F$, it follows that $(l_1^E, l_1^F) \xrightarrow{\sigma,g,u} (l_2^E, l_2^F)$ is a transition in $(E \parallel F)_{|\Omega}$. This concludes the proof. □

Lemma 25. *Let $\mathcal{E}$ be a deterministic normalized EFA system, and let $\Sigma_{\mathcal{E}} = \Omega \mathbin{\dot\cup} \Lambda$ such that for all $\lambda \in \Lambda$ at least one of the following conditions holds:*

1. *$g_\lambda \equiv \boldsymbol{F}$, or*
2. *there exists an $E^i \in \mathcal{E}$ such that $\lambda \in \Sigma^i$, but there does not exist any transition $x \xrightarrow{\lambda, g_\lambda, u_\lambda} y$ in E^i.*

Then SSEFA$(\mathcal{E})_{|\Omega}$ = SSEFA$(\mathcal{E}_{|\Omega})$.

Proof. Observe that the SSEFA algorithm may only change the guards on edges in E, while the restriction operator may only change the alphabet Σ and remove edges from E. As the restriction operator is the only operator that may change the alphabet, the alphabets of SSEFA$(\mathcal{E})_{|\Omega}$ and SSEFA$(\mathcal{E}_{|\Omega})$ are trivially the same. It remains to be proven that the sets of edges are the same. This is proven by showing that for all edges $e \in E$ with $\sigma_e \in \Omega$ it holds that $g_e^{\text{SSEFA}(\mathcal{E})} = g_e^{\text{SSEFA}(\mathcal{E}_{|\Omega})}$ and that for all edges $e \in E$ with $\sigma_e \in \Lambda$ it holds that $g_e^{\text{SSEFA}(\mathcal{E})} = \mathbf{F}$.

Consider the first iteration of Algorithm 1, i.e., $j = 0$. Observe that the initial nonblocking predicate for each location as defined in Line 3 does not depend on any guard. Therefore, these initial nonblocking predicates are the same for SSEFA$(\mathcal{E})$ and SSEFA$(\mathcal{E}_{|\Omega})$. The equation on Line 4 can be rewritten as

$$N_l^{0,k+1} = N_l^{0,k} \vee \bigvee_{\{e|o_e = l, \sigma_e \in \Omega\}} \left[g_e^0 \wedge N_{t_e}^{0,k}[u_e] \right] \vee \bigvee_{\{e|o_e = l, \sigma_e \in \Lambda\}} \left[g_e^0 \wedge N_{t_e}^{0,k}[u_e] \right].$$

Now, using that $g_e^0 = \mathbf{F}$ for all edges $e \in \{e|o_e = l, \sigma_e \in \Lambda\}$, we can rewrite the above equation into

$$N_l^{0,k+1} = N_l^{0,k} \vee \bigvee_{\{e|o_e = l, \sigma_e \in \Omega\}} \left[g_e^0 \wedge N_{t_e}^{0,k}[u_e] \right].$$

Therefore, we can conclude that the nonblocking predicates $N_l^{0,k+1}$ and eventually N_l^0 are the same for SSEFA$(\mathcal{E})$ and SSEFA$(\mathcal{E}_{|\Omega})$.

Moving to Line 12, we observe that the initial bad location predicates do not depend on any guard. Therefore, the initial bad location predicates are the same for SSEFA$(\mathcal{E})$ and SSEFA$(\mathcal{E}_{|\Omega})$. The equation on Line 13 can be rewritten as

$$B_l^{0,i+1} = B_l^{0,i} \vee \bigvee_{\{e|o_e = l, \sigma_e \in \Omega \cap \Sigma_u\}} \left[g_e^0 \wedge B_{t_e}^{0,i}[u_e] \right] \vee \bigvee_{\{e|o_e = l, \sigma_e \in \Lambda \cap \Sigma_u\}} \left[g_e^0 \wedge B_{t_e}^{0,i}[u_e] \right].$$

Now, using that $g_e^0 = \mathbf{F}$ for all edges $e \in \{e|o_e = l, \sigma_e \in \Lambda\}$, we can rewrite the above equation into

$$B_l^{0,i+1} = B_l^{0,i} \vee \bigvee_{\{e|o_e = l, \sigma_e \in \Omega \cap \Sigma_u\}} \left[g_e^0 \wedge B_{t_e}^{0,i}[u_e] \right].$$

Therefore, we can conclude that the bad location predicates $B_l^{0,k+1}$ and eventually B_l^0 are the same for SSEFA($\mathcal{E}$) and SSEFA($\mathcal{E}_{|\Omega}$).

Moving to Line 21, we can now conclude that for all edges $e \in E$ with $\sigma_e \in \Omega$ it holds that g_e^1 is the same for SSEFA($\mathcal{E}$) and SSEFA($\mathcal{E}_{|\Omega}$), and for all edges $e \in E$ with $\sigma_e \in \Lambda$ it holds that $g_e^1 = \mathbf{F}$.

When the algorithm goes back to Line 3 for the next iteration, we can repeat the argumentation above for $j > 0$ to conclude after each iteration that both the nonblocking predicates N_l^j and bad location predicates B_l^j are the same for SSEFA($\mathcal{E}$) and SSEFA($\mathcal{E}_{|\Omega}$), and that for all edges $e \in E$ with $\sigma_e \in \Omega$ it holds that g_e^{j+1} is the same for SSEFA($\mathcal{E}$) and SSEFA($\mathcal{E}_{|\Omega}$), and for all edges $e \in E$ with $\sigma_e \in \Lambda$ it holds that $g_e^{j+1} = \mathbf{F}$. □

Proof of Theorem 7. From the definition of Ξ and the construction of ξ, it follows directly that $\xi \in \Xi$. Therefore, $\xi_1 \circ \xi \in \Xi$ and $(\mathcal{E}_{|\Omega}, \xi_1 \circ \xi)$ is a coordinator tuple.

Now we show that the two coordinator tuples are coordinator equivalent. By rewriting, we can show the following.

$$\begin{aligned}\mathcal{L}(\xi_1(\xi(\sup CN(\mathcal{E}_{|\Omega})))) &= \mathcal{L}(\xi_1(\sup CN(\mathcal{E}_{|\Omega})))\\ &= \mathcal{L}(\xi_1(\sup CN(\mathcal{E})_{|\Omega})) \text{ from Lemma 25}\\ &= \mathcal{L}(\xi_1(\sup CN(\mathcal{E}))) \text{ from Lemma 23}\end{aligned}$$

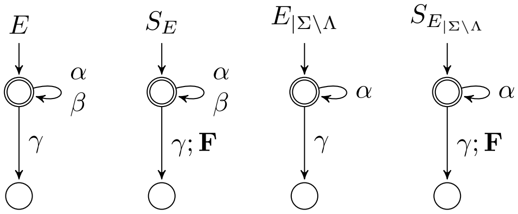
This concludes the proof. □

8 Selfloop removal

8.1 Description of the abstraction

In FAs, selfloops can be safely removed in compositional synthesis, as these selfloops do not change the state of the system, see Mohajerani et al. [2014a]. In EFAs, selfloops do not alter the location, but they may change the valuation and therefore the state of the system. Therefore, events are considered to be selfloop only in EFAs if all transitions labeled with these events cause no location change *and* no valuation change.

The coordinator obtained from the EFA system before selfloop removal and the coordinator obtained from the EFA system after selfloop removal are not entirely the same. The difference is that by removing selfloops, these events are no longer included in the language of the abstracted coordinator, while these events are included in the original coordinator. To refine the abstracted coordinator, i.e., to have the same language as the original coordinator, these removed selfloops need to be placed back at the right locations. This can be achieved by performing the synchronous composition of the abstracted coordinator with the EFA system before selfloop removal. This is summarized in the following theorem. The proof of this theorem can be found in Section 8.2 of this supplementary material.



Event	Guard and update
α	$x = 0; x := x + 1$
β	$x = 1$
γ	$x < 1$

Fig. 4: Example of selfloop removal and coordinator refinement.

Theorem 8 (Selfloop removal)**.** *Let $(\mathcal{E}, \xi_1)$ be a coordinator tuple with $\mathcal{E}$ a deterministic normalized EFA system and let $\Lambda \subseteq \Sigma_{\mathcal{E}}$, where for each $\lambda \in \Lambda$, any transition $(l_1, \lambda, g, u, l_2) \in \rightarrow_{\mathcal{E}}$ implies $l_1 = l_2$ and $\hat{v}_2(v) = \hat{v}_1(u(v)) = \hat{v}_1(v)$ for all $v \in V$ and $\hat{v}_1, \hat{v}_2 \in \mathrm{Val}(V)$. Then refinement function $\xi(\mathcal{G}) = \rho(\mathcal{G}) \parallel \mathcal{E}$ where $\rho = \mathrm{id}$ is the identity renaming function for any EFA system $\mathcal{G}$ ensures that $(\mathcal{E}, \xi_1) \simeq_{co} (\mathcal{E}_{|\Sigma\setminus\Lambda}, \xi_1 \circ \xi)$.*

Example. Figure 4 shows an example of EFA E with selfloops. While both events α and β appear on selfloop transitions, the update of α results in a change in valuation. Therefore, this event cannot be considered to be a selfloop in the EFA setting. Event β does not change the valuation and can thus be considered as a selfloop. The abstracted EFA $E_{|\Sigma\setminus\Lambda}$ is also shown in Figure 4. Both coordinators obtained from the original system and the abstracted system are included in Figure 4, where the guards strengthened by supervisor synthesis are displayed in the automata themselves. As can be seen, both coordinators prevent event γ from happening. Unfortunately, $\mathcal{L}(S_{E_{|\Sigma\setminus\Lambda}}) \subset \mathcal{L}(S_E)$, as event β is no longer possible in the abstracted EFA. However, by taking the synchronous composition of the coordinator and the EFA system before selfloop removal, this language inclusion can be transformed into a language equality, i.e., $\mathcal{L}(S_{E_{|\Sigma\setminus\Lambda}} \parallel E) = \mathcal{L}(S_E)$.

8.2 Proof

Lemma 26. *Let $\mathcal{E}$ a deterministic normalized EFA and let $\Lambda \subseteq \Sigma_{\mathcal{E}}$, where for each $\lambda \in \Lambda$, any transition $(l_1, \lambda, g, u, l_2) \in \rightarrow_{\mathcal{E}}$ implies $l_1 = l_2$ and $\hat{v}_2(v) = \hat{v}_1(u(v)) = \hat{v}_1(v)$ for all $v \in V$ and $\hat{v}_1, \hat{v}_2 \in \mathrm{Val}(V)$. Then SSEFA$(\mathcal{E})_{|\Sigma_{\mathcal{E}}\setminus\Lambda} =$ SSEFA$(\mathcal{E}_{|\Sigma_{\mathcal{E}}\setminus\Lambda})$.*

Proof. First, for notational simplicity, we denote $\Omega = \Sigma \setminus \Lambda$ and $\mathcal{F} = \mathcal{E}_{|\Sigma_{\mathcal{E}}\setminus\Lambda} = \mathcal{E}_{|\Omega}$. Furthermore, in this proof, we use the notation ${}^{\mathcal{E}}x$ to refer to usage of some symbol x in EFA $\mathcal{E}$, while ${}^{\mathcal{F}}x$ refers to the usage of some symbol x in EFA $\mathcal{F}$.

Observe that the SSEFA algorithm may only change the guards on edges in E, while the restriction operator may only change the alphabet Σ and remove edges from E. As the restriction operator is the only operator that may change the alphabet, the alphabets of $\mathrm{SSEFA}(\mathcal{E})_{|\Omega}$ and $\mathrm{SSEFA}(\mathcal{E}_{|\Omega})$ are trivially the same. It remains to be proven that the sets of edges are the same. This is proven by showing that for all edges $e \in E$ with $\sigma_e \in \Omega$ it holds that the fixed-point guard ${}^{\mathcal{E}}g_e^n = {}^{\mathcal{F}}g_e^n$.

Consider the first iteration of Algorithm 1, i.e., $j = 0$. Observe that the initial nonblocking predicate for each location as defined in Line 3 does not depend on any guard. Therefore, these initial nonblocking predicates are the same for SSEFA($\mathcal{E}$) and SSEFA($\mathcal{E}_{|\Omega}$). The equation on Line 4 can be rewritten as

$$N_l^{0,k+1} = N_l^{0,k} \vee \bigvee_{\{e|o_e=l,\sigma_e\in\Omega\}} \left[g_e^0 \wedge N_{t_e}^{0,k}[u_e]\right] \vee \bigvee_{\{e|o_e=l,\sigma_e\in\Lambda\}} \left[g_e^0 \wedge N_{t_e}^{0,k}[u_e]\right].$$

Now, using that $t_e = l$ and $\hat{v}(u_e(v)) = \hat{v}(v)$ for all edges $e \in \{e|o_e = l, \sigma_e \in \Lambda\}$, we have $N_{t_e}^{0,k}[u_e] = N_l^{0,k}$. By using the fact that for any two predicates p and q it holds that $q \vee [p \wedge q] = q$, we can rewrite the above equation into

$$N_l^{0,k+1} = N_l^{0,k} \vee \bigvee_{\{e|o_e=l,\sigma_e\in\Omega\}} \left[g_e^0 \wedge N_{t_e}^{0,k}[u_e]\right].$$

Therefore, we can conclude that the nonblocking predicate ${}^{\mathcal{E}}N_l^{0,k+1} = {}^{\mathcal{F}}N_l^{0,k+1}$ and eventually ${}^{\mathcal{E}}N_l^0 = {}^{\mathcal{F}}N_l^0$.

Moving to Line 12, we observe that the initial bad location predicates do not depend directly on any guard. Therefore, the initial bad location predicates are the same for SSEFA($\mathcal{E}$) and SSEFA($\mathcal{E}_{|\Omega}$). The equation on Line 13 can be rewritten as

$$B_l^{0,i+1} = B_l^{0,i} \vee \bigvee_{\{e|o_e=l,\sigma_e\in\Omega\cap\Sigma_u\}} \left[g_e^0 \wedge B_{t_e}^{0,i}[u_e]\right] \vee \bigvee_{\{e|o_e=l,\sigma_e\in\Lambda\cap\Sigma_u\}} \left[g_e^0 \wedge B_{t_e}^{0,i}[u_e]\right].$$

Now, using again that $t_e = l$ and $\hat{v}(u_e(v)) = \hat{v}(v)$ for all edges $e \in \{e|o_e = l, \sigma_e \in \Lambda\}$, we have $B_{t_e}^{0,k}[u_e] = B_l^{0,k}$. Therefore, we can rewrite the above equation into

$$B_l^{0,i+1} = B_l^{0,i} \vee \bigvee_{\{e|o_e=l,\sigma_e\in\Omega\cap\Sigma_u\}} \left[g_e^0 \wedge B_{t_e}^{0,i}[u_e]\right].$$

Therefore, we can conclude that the bad location predicate ${}^{\mathcal{E}}B_l^{0,k+1} = {}^{\mathcal{F}}B_l^{0,k+1}$ and eventually ${}^{\mathcal{E}}B_l^0 = {}^{\mathcal{F}}B_l^0$.

Moving to Line 21, we can now conclude that for all edges $e \in E$ with $\sigma_e \in \Omega$ it holds that ${}^{\mathcal{E}}g_e^1 = {}^{\mathcal{F}}g_e^1$.

When the algorithm goes back to Line 3 for the next iteration, we can repeat the argumentation above for $j > 0$ to conclude after each iteration that both the nonblocking predicates N_l^j and bad location predicates B_l^j are the same for SSEFA($\mathcal{E}$) and SSEFA($\mathcal{E}_{|\Omega}$), and that for all edges $e \in E$ with $\sigma_e \in \Omega$ it holds that g_e^{j+1} is the same for SSEFA($\mathcal{E}$) and SSEFA($\mathcal{E}_{|\Omega}$). □

Lemma 27. *Let $\mathcal{E}$ a deterministic normalized EFA and let $\Lambda \subseteq \Sigma_{\mathcal{E}}$, where for each $\lambda \in \Lambda$ any transition $(l_1, \lambda, g, u, l_2) \in \rightarrow_{\mathcal{E}}$ implies $l_1 = l_2$ and $\hat{v}_2(v) = \hat{v}_1(u(v)) = \hat{v}_1(v)$ for all $v \in V$ and $\hat{v}_1, \hat{v}_2 \in \mathrm{Val}(V)$. Then $((l_1, l_1, \hat{v}), \sigma, (l_2, l_2, \hat{w}))$ is an edge in $U(SSEFA(\mathcal{E})_{|\Sigma\setminus\Lambda} \parallel \mathcal{E})$ if and only if $((l_1, l_1, \hat{v}), \sigma, (l_2, l_2, \hat{w}))$ is an edge in $U(SSEFA(\mathcal{E}) \parallel \mathcal{E})$.*

Proof. Observe that the only difference between $\text{SSEFA}(\mathcal{E})_{|\Sigma\setminus\Lambda}$ and $\text{SSEFA}(\mathcal{E})$ is the absence of the selfloops in the first one. Therefore, we only have to show for $\sigma \in \Lambda$ that $((l_1, l_1, \hat{v}), \sigma, (l_2, l_2, \hat{w}))$ is an edge in $U(\text{SSEFA}(\mathcal{E})_{|\Sigma\setminus\Lambda} \parallel \mathcal{E})$ if and only if $((l_1, l_1, \hat{v}), \sigma, (l_2, l_2, \hat{w}))$ is an edge in $U(\text{SSEFA}(\mathcal{E}) \parallel \mathcal{E})$.

By taking the synchronous product of $\text{SSEFA}(\mathcal{E})_{|\Sigma\setminus\Lambda}$ and $\mathcal{E}$, these selfloops are placed back. Observe now that the difference between $\text{SSEFA}(\mathcal{E})_{|\Sigma\setminus\Lambda} \parallel \mathcal{E}$ and $\text{SSEFA}(\mathcal{E}) \parallel \mathcal{E}$ is the guards on the selfloops: in $\text{SSEFA}(\mathcal{E})_{|\Sigma\setminus\Lambda} \parallel \mathcal{E}$ each edge e labeled with $\lambda \in \Lambda$ has its original guard g_λ, while in $\text{SSEFA}(\mathcal{E})$ it is the fixed-point guard g^*_e (where we used the fact that $p^*_e \preceq p_\lambda$ and $p^*_e \wedge p_\lambda = p^*_e$). Consider two cases for $\lambda \in \Lambda$.

- If $\lambda \in \Sigma_u$, it follows from Line 21 of Algorithm 1 that $g^*_e = g_\lambda$.
- If $\lambda \in \Sigma_c$, it follows from Line 21 of Algorithm 1 that $g^*_e = g_\lambda \wedge \neg B^*_{t_e}[u_e]$. As $\lambda \in \Lambda$, we know that $t_e = o_e$ and $\hat{v}_2(v) = \hat{v}_1(u(v)) = \hat{v}_1(v)$. Therefore, $g^*_e = g_\lambda \wedge \neg B^*_{o_e}$. According to Lemma 1 of Ouedraogo et al. [2011] it follows that either every state $(l, \hat{v})$ for which $B^*_l[\hat{v}] = \mathbf{T}$ is unreachable in $\text{SSEFA}(\mathcal{E})$ or there exists an initial state $(l_0, \hat{v}_0)$ for which $B^*_{l_0}[\hat{v}_0] = \mathbf{T}$. Applying this lemma, either for each reachable state $(e_o, \hat{w})$ it holds that $B^*_{e_o}[\hat{w}] = \mathbf{F}$ and thus $g^*_e[\hat{w}] = g_\lambda[\hat{w}]$, or from Theorems 2 and 3 of Ouedraogo et al. [2011] we know that $\text{SSEFA}(\mathcal{E})$ is an empty supervisor, and thus $\text{SSEFA}(\mathcal{E})_{|\Sigma\setminus\Lambda}$ is also an empty supervisor.

For both cases we can conclude that for all $\lambda \in \Lambda : ((l, l, \hat{v}), \lambda, (l, l, \hat{v}))$ is an edge in $U(\text{SSEFA}(\mathcal{E})_{|\Sigma\setminus\Lambda} \parallel \mathcal{E})$ if and only if $((l, l, \hat{v}), \lambda, (l, l, \hat{v}))$ is an edge in $U(\text{SSEFA}(\mathcal{E}))$. This concludes the proof. □

Lemma 28. *Let E and F be two deterministic EFAs with shared alphabet Σ and variable set V such that $((l_1, \hat{v}), \sigma, (l_2, \hat{w}))$ is an edge in $U(E)$ if and only if $((l_1, \hat{v}), \sigma, (l_2, \hat{w}))$ is an edge in $U(F)$, and $\rho : \Sigma' \to \Sigma$ a renaming function. Then, for any EFA T it holds that $((l_1, t_1, \hat{p}), \sigma, (l_2, t_2, \hat{q}))$ is an edge in $U(E \parallel T)$ if and only if $((l_1, t_1, \hat{p}), \sigma, (l_2, t_2, \hat{q}))$ is an edge in $U(F \parallel T)$.*

Proof. If $((l_1, \hat{v}), \sigma, (l_2, \hat{w}))$ is an edge in $U(E)$, then $(l_1, \sigma, g^E, u^E, l_2)$ is an edge in E with $g^E[\hat{v}] = \mathbf{T}$ and $\hat{w}(v) = \hat{v}(u^E(v))$; similarly, if $((l_1, \hat{v}), \sigma, (l_2, \hat{w}))$ is an edge in $U(F)$, then $(l_1, \sigma, g^F, u^F, l_2)$ is an edge in F with $g^F[\hat{v}] = \mathbf{T}$ and $\hat{w}(v) = \hat{v}(u^F(v))$. From the assumption that each update in an EFA is well defined for all variables, we can replace u^E and u^F by u with $\hat{v}(u(v)) = \hat{w}(v)$. Consider three cases for event σ.

- $\sigma \in \Sigma \cup \Sigma_T$. In this case $((l_1, x_1), \sigma, g^E \wedge g^T, u \oplus u^T, (l_1, x_2))$ is an edge in $E \parallel T$ if and only if $(l_1, \sigma, g^E, u, l_2)$ is an edge in E and $(x_1, \sigma, g^T, u^T, x_2)$ is an edge in T. Similarly, $((l_1, x_1), \sigma, g^F \wedge g^T, u \oplus u^T, (l_1, x_2))$ is an edge in $F \parallel T$ if and only if $(l_1, \sigma, g^F, u, l_2)$ is an edge in E and $(x_1, \sigma, g^T, u^T, x_2)$ is an edge in T. Furthermore, denote with $\hat{v}'$ the valuation of variables from E (or F) extended with new variables introduced with T. As $g^E[\hat{v}] = \mathbf{T} = g^F[\hat{v}]$, it follows that $(g^E \wedge g^T)[\hat{v}'] = \mathbf{T} = (g^F \wedge g^T)[\hat{v}']$ if and only if $g^T[\hat{v}'] = \mathbf{T}$.

- $\sigma \in \Sigma \cap \Sigma_T$. In this case $((l_1, x_1), \sigma, g^E, u, (l_1, x_2))$ is an edge in $E \parallel T$ if and only if $(l_1, \sigma, g^E, u, l_2)$ is an edge in E and $x_1 = x_2$. Similarly, $((l_1, x_1), \sigma, g^F, u, (l_1, x_2))$ is an edge in $F \parallel T$ if and only if $(l_1, \sigma, g^F, u, l_2)$ is an edge in F and $x_1 = x_2$.

- $\sigma \in \Sigma_T \cap \Sigma$. In this case $((l_1, x_1), \sigma, g^T, u^T, (l_1, x_2))$ is an edge in $E \parallel T$ if and only if $(l_1, \sigma, g^T, u^T, l_2)$ is an edge in T and $x_1 = x_2$. Similarly, $((l_1, x_1), \sigma, g^T, u^T, (l_1, x_2))$ is an edge in $F \parallel T$ if and only if $(l_1, \sigma, g^T, u^T, l_2)$ is an edge in T and $x_1 = x_2$.

Combining the observations above, we can conclude that $((l_1, x_1), \sigma, g^{ET}, u \oplus u^T, (l_2, x_2))$ is an edge in $E \parallel T$ if and only if $((l_1, x_1), \sigma, g^{FT}, u \oplus u^T, (l_2, x_2))$ is an edge in $F \parallel T$, and that $((l_1, x_1, \hat{v}'), \sigma, (l_2, x_2, \hat{w}'))$ is an edge in $U(E \parallel T)$ if and only if $((l_1, x_1, \hat{v}), \sigma, (l_2, x_2, \hat{w}))$ is an edge in $U(F \parallel T)$. This concludes the proof. □

Lemma 29. *Let $\mathcal{E}$ be a deterministic normalized EFA and let $\Lambda \subseteq \Sigma_{\mathcal{E}}$, where for each $\lambda \in \Lambda$, any transition $(l_1, \lambda, g, u, l_2) \in \rightarrow_{\mathcal{E}}$ implies $l_1 = l_2$ and $\hat{v}_2(v) = \hat{v}_1(u(v)) = \hat{v}_1(v)$ for all $v \in V$ and $\hat{v}_1, \hat{v}_2 \in \mathrm{Val}(V)$. Furthermore, let $\xi \in \Xi$ be an refinement function. Then $\mathcal{L}(\xi(SSEFA(\mathcal{E})_{|\Sigma_{\mathcal{E}} \setminus \Lambda} \parallel \mathcal{E})) = \mathcal{L}(\xi(SSEFA(\mathcal{E}) \parallel \mathcal{E}))$.*

Proof. This lemma is proven by induction on the structure of ξ. Denote $\xi = \xi_m \circ \ldots \circ \xi_1$. From Lemma 27 it follows that $((l_1, \hat{v}), \sigma, (l_2, \hat{w}))$ is an edge in $U(\mathrm{SSEFA}(\mathcal{E})_{|\Sigma_{\mathcal{E}} \setminus \Lambda} \parallel \mathcal{E})$ if and only if $((l_1, \hat{v}), \sigma, (l_2, \hat{w}))$ is an edge in $U(\mathrm{SSEFA}(\mathcal{E}) \parallel \mathcal{E})$. Now assume that $((l_1, \hat{v}), \sigma, (l_2, \hat{w}))$ is an edge in $U(\xi_i \circ \ldots \circ \xi_1(\mathrm{SSEFA}(\mathcal{E})_{|\Sigma_{\mathcal{E}} \setminus \Lambda} \parallel \mathcal{E}))$ if and only if $((l_1, \hat{v}), \sigma, (l_2, \hat{w}))$ is an edge in $U(\xi_i \circ \ldots \circ \xi_1(\mathrm{SSEFA}(\mathcal{E}) \parallel \mathcal{E}))$ with $i \in [0 \ldots m-1]$. Consider the following four cases for ξ_{i+1}.

- ξ_{i+1} is the identity function. It follows immediately that $((l_1, \hat{v}), \sigma, (l_2, \hat{w}))$ is an edge in $U(\xi_{x+1} \circ \xi_i \circ \ldots \circ \xi_1(\mathrm{SSEFA}(\mathcal{E})_{|\Sigma_{\mathcal{E}} \setminus \Lambda} \parallel \mathcal{E}))$ if and only if $((l_1, \hat{v}), \sigma, (l_2, \hat{w}))$ is an edge in $U(\xi_{x+1} \circ \xi_i \circ \ldots \circ \xi_1(\mathrm{SSEFA}(\mathcal{E}) \parallel \mathcal{E}))$

- ξ_{i+1} is a renaming. From the definition of renaming it follows directly that $((l_1, \hat{v}), \xi_{i+1}(\sigma), (l_2, \hat{w}))$ is an edge in $U(\xi_{x+1} \circ \xi_i \circ \ldots \circ \xi_1(\mathrm{SSEFA}(\mathcal{E})_{|\Sigma_{\mathcal{E}} \setminus \Lambda} \parallel \mathcal{E}))$ if and only if $((l_1, \hat{v}), \xi_{i+1}(\sigma), (l_2, \hat{w}))$ is an edge in $U(\xi_{x+1} \circ \xi_i \circ \ldots \circ \xi_1(\mathrm{SSEFA}(\mathcal{E}) \parallel \mathcal{E}))$.

- ξ_{i+1} is a renaming ρ in synchronous composition with the previous original system. From the definition of renaming it follows directly that $((l_1, \hat{v}), \rho(\sigma), (l_2, \hat{w}))$ is an edge in $U(\rho(\xi_i \circ \ldots \circ \xi_1(\mathrm{SSEFA}(\mathcal{E})_{|\Sigma_{\mathcal{E}} \setminus \Lambda} \parallel \mathcal{E})))$ if and only if $((l_1, \hat{v}), \rho(\sigma), (l_2, \hat{w}))$ is an edge in $U(\rho(\xi_i \circ \ldots \circ \xi_1(\mathrm{SSEFA}(\mathcal{E}) \parallel \mathcal{E})))$. Combining this with Lemma 28, it follows that $((l_1, \hat{v}), \rho(\sigma), (l_2, \hat{w}))$ is an edge in $U(\xi_{x+1} \circ \xi_i \circ \ldots \circ \xi_1(\mathrm{SSEFA}(\mathcal{E})_{|\Sigma_{\mathcal{E}} \setminus \Lambda} \parallel \mathcal{E}))$ if and only if $((l_1, \hat{v}), \rho(\sigma), (l_2, \hat{w}))$ is an edge in $U(\xi_{x+1} \circ \xi_i \circ \ldots \circ \xi_1(\mathrm{SSEFA}(\mathcal{E}) \parallel \mathcal{E}))$.

- ξ_{i+1} is an inverse renaming ρ^{-1} in synchronous composition with the previous original system. From the definition of inverse renaming it follows for every $\mu \in \rho^{-1}$ that $((l_1, \hat{v}), \mu, (l_2, \hat{w}))$ is an edge in $U(\rho^{-1}(\xi_i \circ \ldots \circ \xi_1(\mathrm{SSEFA}(\mathcal{E})_{|\Sigma_{\mathcal{E}} \setminus \Lambda} \parallel \mathcal{E})))$ if and only if $((l_1, \hat{v}), \mu, (l_2, \hat{w}))$ is an edge in

$U(\rho^{-1}(\xi_i \circ \ldots \circ \xi_1(\mathrm{SSEFA}(\mathcal{E}) \parallel \mathcal{E})))$. Combining this with Lemma 28, it follows that $((l_1, \hat{v}), \rho^{-1}\sigma), (l_2, \hat{w}))$ is an edge in $U(\xi_{x+1} \circ \xi_i \circ \ldots \circ \xi_1(\mathrm{SSEFA}(\mathcal{E})_{|\Sigma_\mathcal{E} \setminus \Lambda} \parallel \mathcal{E}))$ if and only if $((l_1, \hat{v}), \rho^{-1}(\sigma), (l_2, \hat{w}))$ is an edge in $U(\xi_{x+1} \circ \xi_i \circ \ldots \circ \xi_1(\mathrm{SSEFA}(\mathcal{E}) \parallel \mathcal{E}))$.

Therefore, we can conclude that $((l_1, \hat{v}), \sigma, (l_2, \hat{w}))$ is an edge in $U(\xi_m \circ \ldots \circ \xi_1(\mathrm{SSEFA}(\mathcal{E})_{|\Sigma_\mathcal{E} \setminus \Lambda} \parallel \mathcal{E})) = U(\xi(\mathrm{SSEFA}(\mathcal{E})_{|\Sigma_\mathcal{E} \setminus \Lambda} \parallel \mathcal{E}))$ if and only if $((l_1, \hat{v}), \sigma, (l_2, \hat{w}))$ is an edge in $U(\xi_m \circ \ldots \circ \xi_1(\mathrm{SSEFA}(\mathcal{E}) \parallel \mathcal{E})) = U(\xi(\mathrm{SSEFA}(\mathcal{E}) \parallel \mathcal{E}))$. As the initial location and valuation of $\mathrm{SSEFA}(\mathcal{E}) \parallel \mathcal{E}$ and $\mathrm{SSEFA}(\mathcal{E}) \parallel \mathcal{E}$ are the same and that each refinement function ξ does not alter the initial location and initial valuation, it follows from the definition of languages that $\mathcal{L}(\xi(\mathrm{SSEFA}(\mathcal{E})_{|\Sigma_\mathcal{E} \setminus \Lambda} \parallel \mathcal{E})) = \mathcal{L}(\xi(\mathrm{SSEFA}(\mathcal{E}) \parallel \mathcal{E}))$. This concludes the proof. □

Lemma 30. *Let E and E' be two EFAs such that $E' \preceq E$. Then $E' \parallel E \underline{\leftrightarrow}_V E'$.*

Proof. Denote $E = (L, \Sigma, V, \to, l_0, \hat{v}_0, L_m)$ and $E' = (L, \Sigma, V, \to', l_0, \hat{v}_0, L_m)$. From the definitions of subautomaton and synchronous product it follows that the initial location of $E' \parallel E$ is (l_0, l_0) and of E' is (l_0), and that $((x, x), \sigma, g' \wedge g, u, (y, y))$ is an edge in $E' \parallel E$ if and only if (x, σ, g', u, y) is an edge in E'.

Let the relation R be defined as $R = \{((x, x), x, \hat{v}) \mid x \in L, \hat{v} \in \mathrm{Val}(V)\}$. We will show that this is a valuation bisimulation relation.

Consider the triple $((x, x), x, \hat{v}) \in R$.

- Let $((x, x), \hat{v}) \xrightarrow{\sigma} ((y, y), \hat{w})$ be an edge in $U(E' \parallel E)$ for some y and $\hat{w}$. It holds then that $(x, x) \xrightarrow{\sigma, g' \wedge g, u} (y, y)$ is an edge in $E' \parallel E$ with $(g' \wedge g)[\hat{w}] = \mathbf{T}$ and $\hat{w}(v) = \hat{v}(u(v))$. As $(g' \wedge g)[\hat{w}] = \mathbf{T}$, it holds that $g'[\hat{w}] = \mathbf{T}$ and $g[\hat{w}] = \mathbf{T}$. Furthermore, as $x \xrightarrow{\sigma, g', u} y$ is an edge in E', it follows that $(x, \hat{v}) \xrightarrow{\sigma} (y, \hat{w})$ is an edge in $U(E')$. By construction of R it follows that $((y, y), y, \hat{w}) \in R$.

- Let $(x, \hat{v}) \xrightarrow{\sigma} (y, \hat{w})$ be an edge in $U(E')$ for some y and $\hat{w}$. It holds that $x \xrightarrow{\sigma, g', u} y$ is an edge in E' with $g'[\hat{v}] = \mathbf{T}$ and $\hat{w}(v) = \hat{v}(u(v))$. As $g' \preceq g$, it holds that $g[\hat{v}] = \mathbf{T}$. Furthermore, as $(x, x) \xrightarrow{\sigma, g' \wedge g, u} (y, y)$ is an edge in $E' \parallel E$, it follows that $((x, x), \hat{v}) \xrightarrow{\sigma} ((y, y), \hat{w})$ is an edge in $U(E' \parallel E)$. By construction of R it follows that $((y, y), y, \hat{w}) \in R$.

- Let $(x, x) \in L_m \times L_m$ in $E' \parallel E$. From the definition of synchronous product it follows that $x \in L_m$ in E'.

- Let $x \in L_m$ in E'. As E' is a subautomaton of E, it follows that $x \in L_m$ in E. Therefore, $(x, x) \in L_m \times L_m$ in $E' \parallel E$.

This shows that R is a valuation bisimulation relation. As the initial locations and valuations are related, i.e., $((l_0, l_0), l_0, \hat{v}_0) \in R$, it follows that $E' \parallel E \underline{\leftrightarrow}_V E'$. This concludes the proof. □

Lemma 31. *Let E and E' be two EFAs such that $E' \preceq E$, and let $\xi \in \Xi$ be an refinement function. Then $\mathcal{L}(\xi(E' \parallel E)) = \mathcal{L}(\xi(E'))$.*

Proof. It follows from Lemma 30 that $E' \parallel E \leftrightarrow_V E'$. It then follows from Lemma 16 that $\mathcal{L}(\xi(E' \parallel E)) = \mathcal{L}(\xi(E'))$. □

Proof of Theorem 8. From the definition of Ξ and the construction of ξ, it follows directly that $\xi \in \Xi$. Therefore, $\xi_1 \circ \xi \in \Xi$ and $(\mathcal{E}_{|\Sigma\setminus\Lambda}, \xi_1 \circ \xi)$ is a coordinator tuple.

Now we show that the two coordinator tuples are coordinator equivalent. By rewriting, we can show the following.

$$\begin{aligned}\mathcal{L}(\xi_1(\xi(\sup CN(\mathcal{E}_{|\Sigma\setminus\Lambda})))) &= \mathcal{L}(\xi_1(\sup CN(\mathcal{E}_{|\Sigma\setminus\Lambda}) \parallel \mathcal{E}))\\ &= \mathcal{L}(\xi_1(\text{SSEFA}(\mathcal{E}_{|\Sigma\setminus\Lambda}) \parallel \mathcal{E}))\\ &= \mathcal{L}(\xi_1(\text{SSEFA}(\mathcal{E})_{|\Sigma\setminus\Lambda} \parallel \mathcal{E})) \text{ from Lemma 26}\\ &= \mathcal{L}(\xi_1(\text{SSEFA}(\mathcal{E}) \parallel \mathcal{E})) \text{ from Lemma 29}\\ &= \mathcal{L}(\xi_1(\text{SSEFA}(\mathcal{E}))) \text{ from Lemma 31}\\ &= \mathcal{L}(\xi_1(\sup CN(\mathcal{E}))).\end{aligned}$$

This concludes the proof. □

9 Event merging

Lemma 32. *Let E be a deterministic normalized EFA and let $\rho : \Sigma_E \to \Sigma'$ be a renaming. For each edge $e_E = (l_1, \sigma, g, u, l_2) \in \to_E$ in E, let $e_{\rho(E)} = (l_1, \rho(\sigma), g, u, l_2) \in \to_{\rho(E)}$ denote the edge in $\rho(E)$. Then the fixed point guards of SSEFA $g^*_{e_E} = g^*_{e_{\rho(E)}}$, in other words, $\rho(SSEFA(E)) = SSEFA(\rho(E))$.*

Proof. Clearly, $\rho(\text{SSEFA}(E))$ and $\text{SSEFA}(\rho(E))$ have the same location set, same alphabet, same set of variables, same initial location, same initial valuation, and same set of marked locations. It only remains to be proven that they have the same set of edges.

Assume $(l_1, \sigma, g^*, u, l_2)$ is an edge in $\rho(\text{SSEFA}(E))$. From the definition of renaming, it follows that there exists an event $\mu \in \Sigma_E$ such that $\rho(\mu) = \sigma$ and (l_1, μ, g^*, u, l_2) is an edge in $\text{SSEFA}(E)$. As the algorithm SSEFA only adjusts the guards of edges, it follows that $e = (l_1, \mu, g, u, l_2)$ with $g^* \preceq g$ is an edge in E. This means that $f = (l_1, \sigma, g, u, l_2)$ is an edge in $\rho(E)$. Observe that the only difference between edges e and f is the event name.

Now, consider SSEFA, Algorithm 1. As renaming by definition preserves the controllability status of an event, it holds for each iteration j that the nonblocking predicates and the bad location predicates are the same, as the only difference between E and $\rho(E)$, the event labels on the edges, is never used to calculate the fixed points. Therefore, the guards and updates of edges e and f are the same. As g^* is the fixed point guard of edge e, it must hold that g^* is also the fixed point guard of edge f. Therefore, $(l_1, \sigma, g^*, u, l_2)$ is an edge in $\text{SSEFA}(\rho(E))$.

Secondly, assume that $(l_1, \sigma, g^*, u, l_2)$ is an edge in SSEFA$(\rho(E))$. As the algorithm SSEFA only adjusts the guards of edges, it follows that $e = (l_1, \sigma, g, u, l_2)$ with $g^* \preceq g$ is an edge in $\rho(E)$. From the definition of renaming it follows directly that $f = (l_1, \mu, g, u, l_2)$ with $\rho(\mu) = \sigma$ is an edge in E. Observe that the only difference between edges e and f is the event name.

As before, we can conclude that if g^* is the fixed-point guard of edge e, it must hold that g^* is also the fixed-point guard of edge f. Therefore, (l_1, μ, g^*, u, l_2) is an edge in SSEFA(E). After applying renaming we obtain that $(l_1, \sigma, g^*, u, l_2)$ is an edge in ρ(SSEFA(E)). This concludes the proof. □

Lemma 33. *Let $\mathcal{E} = \{E^1, \dots, E^n\}$ a deterministic normalized EFA system. Let $E^k \in \mathcal{E}$ and let $\rho : \Sigma_{\mathcal{E}} \to \Sigma'$ be a renaming such that the following conditions hold for all $\sigma_1, \sigma_2 \in \Sigma_{\mathcal{E}}$ with $\rho(\sigma_1) = \rho(\sigma_2)$:*

1. *$g_{\sigma_1} = g_{\sigma_2}$ and $u_{\sigma_1} = u_{\sigma_2}$,*
2. *for all $i \neq k$, it holds that $\sigma_1 \in \Sigma^i$ if and only if $\sigma_2 \in \Sigma^i$, and for all $l_1, l_2 \in L^i$ it holds that $l_1 \xrightarrow{\sigma_1, g_{\sigma_1}, u_{\sigma_1}} l_2$ in E^i if and only if $l_1 \xrightarrow{\sigma_2, g_{\sigma_2}, u_{\sigma_2}} l_2$ in E^i,*
3. *$\sigma_1 \in \Sigma_c$ if and only if $\sigma_2 \in \Sigma_c$.*

Then $\rho^{-1}(\rho(\mathcal{E})) \parallel \mathcal{E} = \mathcal{E} \parallel \mathcal{E}$ if and only if $\rho(E^k)$ is deterministic.

Proof. As $\mathcal{E} = \{E^1, \dots, E^n\}$, we can rewrite $\mathcal{E} \parallel \mathcal{E} = (E^1 \parallel \dots \parallel E^n) \parallel (E^1 \parallel \dots \parallel E^n) = (E^1 \parallel E^1) \parallel \dots \parallel (E^n \parallel E^n)$, and $\rho^{-1}(\rho(\mathcal{E})) \parallel \mathcal{E} = (\rho^{-1}(\rho(E^1)) \parallel \dots \parallel \rho^{-1}(\rho(E^n))) \parallel (E^1 \parallel \dots \parallel E^n) = (\rho^{-1}(\rho(E^1)) \parallel E^1) \parallel \dots \parallel (\rho^{-1}(\rho(E^n)) \parallel E^n)$. We now show for each $i \in [1 \dots n]$ that $E^i \parallel E^i = \rho^{-1}(\rho(E^i)) \parallel E^i$.

From the definition of renaming and inverse renaming, it follows for each i that $E^i \parallel E^i$ and $\rho^{-1}(\rho(E^i)) \parallel E^i$ have the same location set, same alphabet, same variable set, same initial location, same initial valuation, and same marked location set. It only remains to be proven that they have the same set of edges. In order to do that, we need to make a distinction between $i = k$ and $i \neq k$.

First, let $i = k$. Consider edge $e = (l_1, \sigma, g, u, l_2)$ in E^k. Therefore, in the synchronous product $E^k \parallel E^k$ we have the edge $((l_1, l_1), \sigma, g, u, (l_2, l_2))$. Furthermore, after applying renaming on E^k, we know that there exists an edge $(l_1, \rho(\sigma), g, u, l_2)$ in $\rho(E^k)$. Continuing with applying inverse renaming, we obtain in $\rho^{-1}(\rho(E^k))$ a set of edges $\{(l_1, \sigma, g, u, l_j) \mid (l_1, \sigma', g, u, l_j)$ is an edge in $E^k, \sigma' \in \rho^{-1}(\rho(\sigma))\}$ that have the same event σ as edge e and also originate from the same location. If we now consider the synchronous product $\rho^{-1}(\rho(E^k)) \parallel E^k$, we get the set of edges $A(e) = \{((l_1, l_1), \sigma, g, u, (l_j, l_2)) \mid (l_1, \sigma', g, u, l_j)$ is an edge in $E^k, \sigma' \in \rho^{-1}(\rho(\sigma)))\}$ that are associated with edge e. Observe that in $E^k \parallel E^k$ we only have one edge labeled with σ from location l_1, while in $\rho^{-1}(\rho(E^k)) \parallel E^k$ we may have multiple edges labeled with σ from location l_1.

We will now show that $\rho^{-1}(\rho(E^k)) \parallel E^k$ only has a single edge labeled with σ from location l_1 if and only if $\rho(E^k)$ is deterministic. From the definition of determinism, it follows that $\rho(E^k)$ is deterministic if and only if for each location in $U(\rho(E^k))$ and event μ there is at most one outgoing edge labeled with event μ. This implies that $\rho(E^k)$ is deterministic if and only if for each location in $\rho(E^k)$, event μ and valuation $\hat{v}$ there is at most one edge labeled with event μ such that the guard of that edge evaluates to true for valuation $\hat{v}$. From condition 1 it follows that for each location in $\rho(E^k)$ and event μ it holds that all outgoing edges labeled with μ have the same guard. Therefore, it holds that for each location in $\rho(E^k)$ and event μ only a single outgoing edge is labeled with event μ if and only if $\rho(E^k)$ is deterministic. Subsequently, for each location in E^k and event μ there is only one outgoing edge labeled with one of the events from $\rho^{-1}(\mu)$ if and only if $\rho(E^k)$ is deterministic. This is enough to show that when we consider edge e in E^k, the set $A(e)$ reduced to the single edge $((l_1, l_1), \sigma, g, u, (l_2, l_2))$ if and only if $\rho(E^k)$ is deterministic.

Finally, as edge e is chosen arbitrarily, it follows that $E^k \parallel E^k$ and $\rho^{-1}(\rho(E^k))$ $\parallel E^k$ have the same set of edges if and only if $\rho(E^k)$ is deterministic.

Second, let $i \neq k$. Consider edge $e = (l_1, \sigma, g, u, l_2)$ in E^i. From the second condition it follows that for all σ' such that $\rho(\sigma) = \rho(\sigma') = \mu$ it holds that $(l_1, \sigma', g, u, l_2)$ is also an edge in E^i. Or stated slightly different, we have a set of edges $B(e) = \{(l_1, \sigma', g, u, l_2) \mid e = (l_1, \sigma, g, u, l_2) \text{ is an edge in } E^i, \sigma' \in \rho^{-1}(\rho(\sigma))\}$ that are associated with edge e. Therefore, in the synchronous product $E^i \parallel E^i$ we have the set of edges $\{((l_1, l_1), \sigma', g, u, (l_2, l_2)) \mid e \text{ is an edge in } E^i,$ $\sigma' \in \rho^{-1}(\rho(\sigma))\}$.

Furthermore, if e is an edge in E^i, then $(l_1, \rho(\sigma), g, u, l_2)$ is an edge in $\rho(E^i)$. After applying the inverse renaming on $\rho(E^i)$, we know that in $\rho^{-1}(\rho(E^i))$ there is a set of edges $\{(l_1, \sigma', g, u, l_2) \mid e \text{ is an edge in } E^i, \sigma' \in \rho^{-1}(\rho(\sigma))\}$ associated with edge e. If we now perform the synchronous product to obtain $\rho^{-1}(\rho(E^i)) \parallel$ E^i, we get the set of edges $\{((l_1, l_1), \sigma', g, u, (l_2, l_2)) \mid \text{ e is an edge in } E^i, \sigma' \in \rho^{-1}(\rho(\sigma))\}$, where we used the previous observation that in E^i we have the set of edges $B(e)$ associated with e. We now have established that $E^i \parallel E^i$ and $\rho^{-1}(\rho(E^i)) \parallel E^i$ have the same set of edges associated with edge e. As edge e is chosen arbitrarily, it follows that $E^i \parallel E^i$ and $\rho^{-1}(\rho(E^i)) \parallel E^i$ have the same set of edges. □

Lemma 34. *Let $\mathcal{E} = \{E^1, \ldots, E^n\}$ be a deterministic normalized EFA system. Let $E^k \in \mathcal{E}$ and let $\rho : \Sigma_{\mathcal{E}} \to \Sigma'$ be a renaming such that the following conditions hold for all $\sigma_1, \sigma_2 \in \Sigma_{\mathcal{E}}$ with $\rho(\sigma_1) = \rho(\sigma_2)$:*

1. *$g_{\sigma_1} = g_{\sigma_2}$ and $u_{\sigma_1} = u_{\sigma_2}$,*
2. *for all $i \neq k$, it holds that $\sigma_1 \in \Sigma^i$ if and only if $\sigma_2 \in \Sigma^i$, and for all $l_1, l_2 \in L^i$ it holds that $l_1 \xrightarrow{\sigma_1, g_{\sigma_1}, u_{\sigma_1}} l_2$ in E^i if and only if $l_1 \xrightarrow{\sigma_2, g_{\sigma_2}, u_{\sigma_2}} l_2$ in E^i,*
3. *$\sigma_1 \in \Sigma_c$ if and only if $\sigma_2 \in \Sigma_c$.*

Then $\rho^{-1}(\rho(SSEFA(\mathcal{E}))) \parallel \mathcal{E} = SSEFA(E) \parallel \mathcal{E}$ if and only if $\rho(E^k)$ is deterministic.

Proof. As $\mathcal{E} = \{E^1, \ldots, E^n\}$, we can rewrite SSEFA($\mathcal{E}$). Let $e = ((x_1, \ldots, x_n), \sigma, g, u, (y_1, \ldots, y_n))$ be an edge in $E^1 \parallel \ldots \parallel E^n$, and $e_i = (x_i, \sigma, g, u, y_i)$ the edge in E^i if $\sigma \in \Sigma^i$ or $x_i = y_i$ if $\sigma \notin \Sigma^i$. After applying Algorithm 1, we get for each edge e the edge $e^* = ((x_1, \ldots, x_n), \sigma, g^*, u, (y_1, \ldots, y_n))$ in SSEFA($\mathcal{E}$) where g^* is the fixed-point guard. Now we can rewrite SSEFA($\mathcal{E}$) $= E^{1*} \parallel \ldots \parallel E^{n*}$ where each edge e_i in E_i is replaced by $e_i^* = (x_i, \sigma, g^*, u, y_i)$.

Rewrite SSEFA($\mathcal{E}$) $\parallel \mathcal{E} = (E^{1*} \parallel \ldots \parallel E^{n*}) \parallel (E^1 \parallel \ldots \parallel E^n) = (E^{1*} \parallel E^1) \parallel \ldots \parallel (E^{n*} \parallel E^n)$ and $\rho^{-1}(\rho(\text{SSEFA}(\mathcal{E})) \parallel \mathcal{E} = (\rho^{-1}(\rho(E^{1*})) \parallel \ldots \parallel \rho^{-1}(\rho(E^{n*}))) \parallel (E^1 \parallel \ldots \parallel E^n) = (\rho^{-1}(\rho(E^{1*})) \parallel E^1) \parallel \ldots \parallel (\rho^{-1}(\rho(E^{n*})) \parallel E^n)$. Following the same reasoning as the proof in Lemma 33 and knowing that for edge e we have now guards g^* and g instead of g and g, we can show for each $i \in [1 \ldots n]$ that $E^{i*} \parallel E^i = \rho^{-1}(\rho(E^{i*}) \parallel E^i$ if and only if $\rho(E^k)$ is deterministic. This concludes the proof. □

Theorem 9. *Let $(\mathcal{E}, \xi_1)$ be a coordinator tuple with $\mathcal{E} = \{E^1, \ldots, E^n\}$ a deterministic normalized EFA system. Let $E^k \in \mathcal{E}$ and let $\rho : \Sigma_{\mathcal{E}} \to \Sigma'$ be a renaming such that the following conditions hold for all $\sigma_1, \sigma_2 \in \Sigma_{\mathcal{E}}$ with $\rho(\sigma_1) = \rho(\sigma_2)$:*

1. *$g_{\sigma_1} = g_{\sigma_2}$ and $u_{\sigma_1} = u_{\sigma_2}$,*
2. *for all $i \neq k$, it holds that $\sigma_1 \in \Sigma^i$ if and only if $\sigma_2 \in \Sigma^i$, and for all $l_1, l_2 \in L^i$ it holds that $l_1 \xrightarrow{\sigma_1, g_{\sigma_1}, u_{\sigma_1}} l_2$ in E^i if and only if $l_1 \xrightarrow{\sigma_2, g_{\sigma_2}, u_{\sigma_2}} l_2$ in E^i,*
3. *$\sigma_1 \in \Sigma_c$ if and only if $\sigma_2 \in \Sigma_c$.*

Then refinement function $\xi(\mathcal{G}) = \rho^{-1}(\mathcal{G}) \parallel \mathcal{E}$ for any EFA system $\mathcal{G}$ with alphabet Σ' ensures that $(\mathcal{E}, \xi_1) \simeq_{co} (\rho(\mathcal{E}), \xi_1 \circ \xi)$.

Proof. From the definition of Ξ and the construction of ξ, it follows directly that $\xi \in \Xi$. Therefore, $\xi_1 \circ \xi \in \Xi$ and $(\rho(\mathcal{E}), \xi_1 \circ \xi)$ is a coordinator tuple.

Now we show that the two coordinator tuples are coordinator equivalent. By rewriting, we can show the following.

$$\begin{aligned}
\mathcal{L}(\xi_1(\xi(\sup CN(\mathcal{F})))) &= \mathcal{L}(\xi_1(\xi(\text{SSEFA}(\mathcal{F})))) \\
&= \mathcal{L}(\xi_1(\xi(\text{SSEFA}(\rho(\mathcal{E}))))) \\
&= \mathcal{L}(\xi_1(\xi(\rho(\text{SSEFA}(\mathcal{E}))))) \text{ by Lemma 32} \\
&= \mathcal{L}(\xi_1(\rho(\rho(\text{SSEFA}(\mathcal{E}))) \parallel \mathcal{E})) \\
&= \mathcal{L}(\xi_1(\text{SSEFA}(\mathcal{E}) \parallel \mathcal{E})) \text{ by Lemma 34} \\
&= \mathcal{L}(\xi_1(\text{SSEFA}(\mathcal{E}))) \text{ by Lemma 31} \\
&= \mathcal{L}(\xi_1(\sup CN(\mathcal{E}))).
\end{aligned}$$

This concludes the proof. □

10 Update merging

10.1 Description of the abstraction

The final abstraction considered in this paper is called update merging. In the context of separate guards and updates, a more appropriate terminology may be guard merging. As introduced by Mohajerani et al. [2016], update merging merges events together if they always appear together on the same transitions in the EFA system and they have the same set of updated variables. This formulation allows events to be merged if they both update a variable, but update it to different valuations. In this case, the abstracted system becomes nondeterministic, which we avoid. For update merging, the general strategy of nondeterminism avoidance by first applying a renaming would not help. Therefore, we need to strengthen the conditions when update merging may be applied: besides appearing always together on the same transitions, the updates should be the same. Requiring that the updates are the same ensures that a deterministic system remains deterministic after update merging.

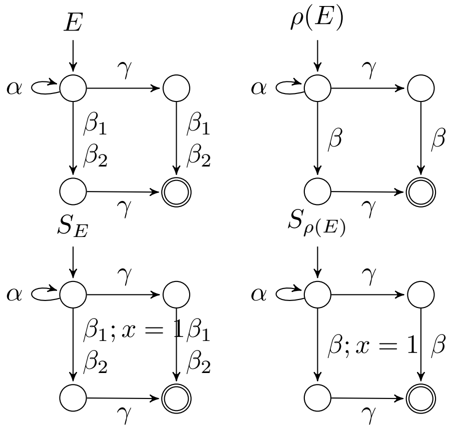
Similar to event merging, update merging applies a renaming function to get from multiple events to a single event. Refining update merging would require to apply the inverse renaming to get from a single event to multiple events. Unfortunately, this may introduce too many possible events in the coordinator. Therefore, the same solution as event merging may be applied: perform the synchronous composition of the inverse renamed abstracted coordinator with the original system. Theorem 10 expresses this formally. The proof of this theorem can be found in Section 10 of the supplementary material.

Theorem 10 (Update merging)**.** *Let $(\mathcal{E}, \xi_1)$ be a coordinator tuple with $\mathcal{E} = \{E^1, \ldots, E^n\}$ a deterministic normalized EFA system. Let $\rho : \Sigma_{\mathcal{E}} \to \Sigma'$ be a renaming such that the following conditions hold for all $\sigma_1, \sigma_2 \in \Sigma_{\mathcal{E}}$ with $\rho(\sigma_1) = \rho(\sigma_2)$:*

1. *$u_{\sigma_1} = u_{\sigma_2}$,*
2. *for all $i = 1, \ldots, n$, it holds that $\sigma_1 \in \Sigma^i$ if and only if $\sigma_2 \in \Sigma^i$, and for all $l_1, l_2 \in L_i$ it holds that $l_1 \xrightarrow{\sigma_1, g_{\sigma_1}, u_{\sigma_1}} l_2$ in E^i if and only if $l_1 \xrightarrow{\sigma_2, g_{\sigma_2}, u_{\sigma_2}} l_2$ in E^i,*
3. *$\sigma_1 \in \Sigma_c$ if and only if $\sigma_2 \in \Sigma_c$.*

Create the EFA system $\mathcal{F} = \{F^1, \ldots, F^n\}$ such that each $F^i = (L^i, V^i, \rho(\Sigma^i), \to^{i,F}, l_0^i, \hat{v}_0^i, L_m^i)$ with $\to^{i,F} = \{(x, \rho(\sigma), g^F_{\rho(\sigma)}, u_\sigma, y) \mid (x, \sigma, g^F_\sigma, u_\sigma, y) \in \to^{i,E}\}$ and $g^F_{\rho(\sigma)} = \bigvee_{\sigma' \in \rho^{-1}(\rho(\sigma))} g^E_{\sigma'}$. Then refinement function $\xi(\mathcal{G}) = \rho^{-1}(\mathcal{G}) \parallel \mathcal{E}$ for any EFA system $\mathcal{G}$ with alphabet Σ' ensures that $(\mathcal{E}, \xi_1) \simeq_{co} (\mathcal{F}, \xi_1 \circ \xi)$.

Example. Figure 5 shows an example where update merging is applied. In the original EFA system, events β_1 and β_2 have the same update and they always appear on the same transitions. Therefore, these events can be merged



Event	Guard and update
α	$x := 1$
β_1	$x = 0; x := x$
β_2	$x = 1; x := x$
γ	$x = 1$
α	$x := 1$
β	$x = 0 \vee x = 1; x := x$
γ	$x = 1$

Fig. 5: Example of update merging and coordinator refinement. Initially, $\hat{v}_0(x) = 0$. In the table, the top for events constitute the original alphabet, while the bottom three the one after event merging.

into, for example, β, which results in EFA $\rho(E)$. For the original and the abstracted system a coordinator is synthesized. The strengthened guards are shown directly in the automaton representation of the coordinators. If we would just apply inverse renaming, too much behavior is possible in $\rho^{-1}(S_{\rho(E)})$. In $S_{\rho(E)}$, the left transition labeled with β has guard $(x = 0 \vee x = 1) \wedge x = 1$, where $x = 1$ is added by the coordinator. Inverse renaming would result in two transitions, one labeled with β_1 and one labeled with β_2, and both having guard $(x = 0 \vee x = 1) \wedge x = 1$. Therefore, after performing event α, both β_1 and β_2 are possible in $\rho^{-1}(S_{\rho(E)})$, while only β_2 would be possible in S_E. Taking the synchronous composition of $\rho^{-1}(S_{\rho(E)})$ with E resolves this problem and ensures that $\mathcal{L}(\rho^{-1}(S_{\rho(E)}) \parallel E) = \mathcal{L}(S_E)$.

10.2 Proof

Lemma 35. *Let $\mathcal{E} = \{E^1, \dots, E^n\}$ a be deterministic normalized EFA system. Let $\rho : \Sigma_{\mathcal{E}} \to \Sigma'$ be a renaming such that the following conditions hold for all $\sigma_1, \sigma_2 \in \Sigma_{\mathcal{E}}$ with $\rho(\sigma_1) = \rho(\sigma_2)$:*

1. *$u_{\sigma_1} = u_{\sigma_2}$,*
2. *for all $i = 1, \dots, n$, it holds that $\sigma_1 \in \Sigma^i$ if and only if $\sigma_2 \in \Sigma^i$, and for all $l_1, l_2 \in L_i$ it holds that $l_1 \xrightarrow{\sigma_1, g_{\sigma_1}, u_{\sigma_1}} l_2$ in E^i if and only if $l_1 \xrightarrow{\sigma_2, g_{\sigma_2}, u_{\sigma_2}} l_2$ in E^i,*
3. *$\sigma_1 \in \Sigma_c$ if and only if $\sigma_2 \in \Sigma_c$.*

Then for all $\sigma_1, \sigma_2 \in \Sigma$ such that $\rho(\sigma_1) = \rho(\sigma_2)$ it holds that $(x, \sigma_1, g_{\sigma_1}, u_{\sigma_1}, y)$ is an edge in $\parallel \mathcal{E}$ if and only if $(x, \sigma_2, g_{\sigma_2}, u_{\sigma_2}, y)$ is an edge in $\parallel \mathcal{E}$.

Proof. Consider two cases for each E^i.

- $\sigma_1 \in \Sigma^i$ if and only if $\sigma_2 \in \Sigma^i$. Furthermore, it holds that $l_1 \xrightarrow{\sigma_1, g_{\sigma_1}, u_{\sigma_1}} l_2$ if and only if $l_1 \xrightarrow{\sigma_2, g_{\sigma_2}, u_{\sigma_2}} l_2$.
- $\sigma_1 \notin \Sigma^i$ if and only if $\sigma_2 \notin \Sigma^i$.

Combining the above observations with the definition of the synchronous product, it follows that $(x, \sigma_1, g_{\sigma_1}, u_{\sigma_1}, y)$ is an edge in $\parallel \mathcal{E}$ if and only if $(x, \sigma_2, g_{\sigma_2}, u_{\sigma_2}, y)$ is an edge in $\parallel \mathcal{E}$. □

Lemma 36. *Let $\mathcal{E} = \{E^1, \ldots, E^n\}$ a be deterministic normalized EFA system. Let $\rho : \Sigma_{\mathcal{E}} \to \Sigma'$ be a renaming such that the following conditions hold for all $\sigma_1, \sigma_2 \in \Sigma_{\mathcal{E}}$ with $\rho(\sigma_1) = \rho(\sigma_2)$:*

1. *$u_{\sigma_1} = u_{\sigma_2}$,*
2. *for all $i = 1, \ldots, n$, it holds that $\sigma_1 \in \Sigma^i$ if and only if $\sigma_2 \in \Sigma^i$, and for all $l_1, l_2 \in L_i$ it holds that $l_1 \xrightarrow{\sigma_1, g_{\sigma_1}, u_{\sigma_1}} l_2$ in E^i if and only if $l_1 \xrightarrow{\sigma_2, g_{\sigma_2}, u_{\sigma_2}} l_2$ in E^i,*
3. *$\sigma_1 \in \Sigma_c$ if and only if $\sigma_2 \in \Sigma_c$.*

Create the EFA system $\mathcal{F} = \{F^1, \ldots, F^n\}$ such that each $F^i = (L^i, V^i, \rho(\Sigma^i), \to^{i,F}, l_0^i, \hat{v}_0^i, L_m^i)$ with $\to^{i,F} = \{(x, \rho(\sigma), g^F_{\rho(\sigma)}, u_\sigma, y) \mid (x, \sigma, g^F_\sigma, u_\sigma, y) \in \to^{i,E}\}$ and $g^F_{\rho(\sigma)} = \bigvee_{\sigma' \in \rho^{-1}(\rho(\sigma))} g^E_{\sigma'}$. Then for each $\sigma \in \Sigma$ it holds that $(x, \sigma, g_\sigma, u_\sigma, y)$ is an edge in $\parallel \mathcal{E}$ if and only if $(x, \rho(\sigma), g_{\rho(\sigma)}, u_\sigma, y)$ is an edge in $\parallel \mathcal{F}$.

Proof. Consider two cases for each E^i.

- $\sigma \in \Sigma^i$ if and only if $\rho(\sigma) \in \rho(\Sigma^i) = \Sigma^{i,F}$. Furthermore, it holds that $l_1 \xrightarrow{\sigma, g_\sigma, u_\sigma} l_2$ in E^i if and only if $l_1 \xrightarrow{\rho(\sigma), g_{\rho(\sigma)}, u_\sigma} l_2$ in F^i.
- $\sigma \notin \Sigma^i$ if and only if $\rho(\sigma) \notin \rho(\Sigma^i) = \Sigma^{i,F}$.

Combining the above observations with the definition of the synchronous product, it follows that $(x, \sigma, g_\sigma, u_\sigma, y)$ is an edge in $\parallel \mathcal{E}$ if and only if $(x, \rho(\sigma), g_{\rho(\sigma)}, u_\sigma, y)$ is an edge in $\parallel \mathcal{F}$. □

Lemma 37. *Let $\mathcal{E} = \{E_1, \ldots, E^n\}$ a be deterministic normalized EFA system. Let $\rho : \Sigma_{\mathcal{E}} \to \Sigma'$ be a renaming such that the following conditions hold for all $\sigma_1, \sigma_2 \in \Sigma_{\mathcal{E}}$ with $\rho(\sigma_1) = \rho(\sigma_2)$:*

1. *$u_{\sigma_1} = u_{\sigma_2}$,*
2. *for all $i = 1, \ldots, n$, it holds that $\sigma_1 \in \Sigma^i$ if and only if $\sigma_2 \in \Sigma^i$, and for all $l_1, l_2 \in L_i$ it holds that $l_1 \xrightarrow{\sigma_1, g_{\sigma_1}, u_{\sigma_1}} l_2$ in E^i if and only if $l_1 \xrightarrow{\sigma_2, g_{\sigma_2}, u_{\sigma_2}} l_2$ in E^i,*
3. *$\sigma_1 \in \Sigma_c$ if and only if $\sigma_2 \in \Sigma_c$.*

Create the EFA system $\mathcal{F} = \{F^1, \ldots, F^n\}$ *such that each* $F^i = (L^i, V^i, \rho(\Sigma^i), \rightarrow^{i,F}, l_0^i, \hat{v}_0^i, L_m^i)$ *with* $\rightarrow^{i,F} = \{(x, \rho(\sigma), g^F_{\rho(\sigma)}, u_\sigma, y) \mid (x, \sigma, g^F_\sigma, u_\sigma, y) \in \rightarrow^{i,E}\}$ *and* $g^F_{\rho(\sigma)} = \bigvee_{\sigma' \in \rho^{-1}(\rho(\sigma))} g^E_{\sigma'}$*. Then for each edge* $e = (x, \mu, g^*_\mu, u_\mu, y)$ *in SSEFA*$(\mathcal{F})$ *there exists a set of edges* $A_e = \{(x, \sigma, g^*_\sigma, u_\sigma, y) \mid \sigma \in \rho^{-1}(\mu)\}$ *in SSEFA*$(\mathcal{E})$ *such that* $g^*_\mu \Leftrightarrow \bigvee_{e \in A} g^*_\sigma$ *and* $u_\sigma = u_\mu$*.*

Proof. From the construction of $\mathcal{F}$ it follows that $\parallel \mathcal{E}$ and $\parallel \mathcal{E}$ have the same set of locations, variables, initial location, initial valuation, and marked states. Furthermore, $\Sigma_\mathcal{F} = \rho(\Sigma_\mathcal{E})$ and $\rightarrow^\mathcal{F}$ can be constructed from $\rightarrow^\mathcal{E}$ according to Lemma 36. From Algorithm 1 it follows that SSEFA$(\mathcal{E})$ and SSEFA$(\mathcal{F})$ have the same set of locations, variables, initial location, initial valuation, and marked states. Furthermore, $\Sigma_{\text{SSEFA}(\mathcal{F})} = \Sigma_\mathcal{F} = \rho(\Sigma_\mathcal{E}) = \rho(\Sigma_{\text{SSEFA}(\mathcal{F})})$.

Combining Lemmas 35 and 36 we can construct the set of edges $A'_e = \{(x, \sigma, g_\sigma, u_\sigma, y) \mid \sigma \in \rho^{-1}(\mu)\}$ in $\mathcal{E}$ for each edge $e = (x, \mu, g_\mu, u_\mu, y)$ in $\mathcal{F}$ and $u_\sigma = u_\mu$. As Algorithm 1 does not change the updates, it holds $u_\sigma = u_\mu$ after applying SSEFA on both $\mathcal{E}$ and $\mathcal{F}$. It remains to be proven that $g^*_\mu \Leftrightarrow \bigvee_{e \in A} g^*_\sigma$. In the remainder of the proof, we use the notation ${}^\mathcal{E}x$ to refer to usage of some symbol x in EFA $\mathcal{E}$, while ${}^\mathcal{F}x$ refers to the usage of some symbol x in EFA $\mathcal{F}$.

Consider the first iteration of Algorithm 1, i.e., $j = 0$. After initializing the guards, it follows that $g^0_e = \bigvee_{e' \in A'} g^0_{e'}$. Observe that the initial nonblocking predicate for each location as defined in Line 3 does not depend on any guard. Therefore, these initial nonblocking predicates are the same for SSEFA$(\mathcal{E})$ and SSEFA$(\mathcal{F})$, i.e., ${}^\mathcal{E}N^{0,0}_l = {}^\mathcal{F}N^{0,0}_l$. The equation on Line 4 can be rewritten as

$$ {}^\mathcal{E}N^{0,k+1}_l = {}^\mathcal{E}N^{0,k}_l \vee \bigvee_{\mu \in \rho(\Sigma)} \left[\bigvee_{\{e' \mid \rho(\sigma_{e'}) = \mu\}} \left(g^0_{e'} \wedge {}^\mathcal{E}N^{0,k}_{t_{e'}}[u_{e'}] \right) \right] $$

Now, from Lemmas 35 and 36 it follows that for each $\mu \in \rho(\Sigma)$ and associated edge e in $\mathcal{F}$ with $\sigma_e = \mu$ and $o_e = l$ it holds for all edges e' in $\mathcal{E}$ with $\rho(\sigma_{e'}) = \mu$ and $o_{e'}$ that $t_e = t_{e'}$ and $u_e = u_{e'}$. Therefore, we can rewrite the above equation into

$$ \begin{aligned} {}^\mathcal{E}N^{0,k+1}_l &= {}^\mathcal{E}N^{0,k}_l \vee \bigvee_{\{e \in \rightarrow^\mathcal{F} \mid o_e = l\}} \left[{}^\mathcal{E}N^{0,k}_{t_e}[u_e] \wedge \bigvee_{e' \in A_e} g^0_{e'} \right] \\ &= {}^\mathcal{E}N^{0,k}_l \vee \bigvee_{\{e \in \rightarrow^\mathcal{F} \mid o_e = l\}} \left[{}^\mathcal{E}N^{0,k}_{t_e}[u_e] \wedge g^0_e \right]. \end{aligned} $$

As initially ${}^\mathcal{E}N^{0,0}_l = {}^\mathcal{F}N^{0,0}_l$ and using the above equation, we can show by induction on k that ${}^\mathcal{E}N^{0,k}_l = {}^\mathcal{F}N^{0,k}_l$. Therefore, we can conclude in Line 9 that ${}^\mathcal{E}N^0_l = {}^\mathcal{F}N^0_l$.

Moving to Line 12, we observe that the initial bad location predicates do not depend directly on any guard. Therefore, the initial bad location predicates are the same for SSEFA$(\mathcal{E})$ and SSEFA$(\mathcal{F})$, i.e., ${}^\mathcal{E}B^{0,0}_l = {}^\mathcal{F}B^{0,0}_l$. The equation on Line 13 can be rewritten as

$$^{\mathcal{E}}B_l^{0,k+1} = {}^{\mathcal{E}}B_l^{0,k} \vee \bigvee_{\mu \in \rho(\Sigma)} \left[\bigvee_{\{e' \mid \rho(\sigma_{e'})=\mu, \sigma_{e'} \in \Sigma_u\}} \left(g_{e'}^0 \wedge {}^{\mathcal{E}}B_{t_{e'}}^{0,k}[u_{e'}] \right) \right]$$

Now, using again that for each $\mu \in \rho(\Sigma)$ and associated edge e in $\mathcal{F}$ with $\sigma_e = \mu$ and $o_e = l$ it holds for all edges e' in $\mathcal{E}$ with $\rho(\sigma_{e'}) = \mu$ and $o_{e'}$ that $t_e = t_{e'}$, $u_e = u_{e'}$, and $\sigma_{e'} \in \Sigma_u$ if and only if $\mu \in \rho(\Sigma_u)$. Therefore, we can rewrite the above equation into

$$\begin{aligned} ^{\mathcal{E}}B_l^{0,k+1} &= {}^{\mathcal{E}}B_l^{0,k} \vee \bigvee_{\{e \in \rightarrow^{\mathcal{F}} \mid o_e = l, \sigma_e \in \rho(\Sigma_u)\}} \left[{}^{\mathcal{E}}B_{t_e}^{0,k}[u_e] \wedge \bigvee_{e' \in A_e} g_{e'}^0 \right] \\ &= {}^{\mathcal{E}}B_l^{0,k} \vee \bigvee_{\{e \in \rightarrow^{\mathcal{F}} \mid o_e = l, \sigma_e \in \rho(\Sigma_u)\}} \left[{}^{\mathcal{E}}B_{t_e}^{0,k}[u_e] \wedge g_e^0 \right]. \end{aligned}$$

As initially ${}^{\mathcal{E}}B_l^{0,0} = {}^{\mathcal{F}}B_l^{0,0}$ and using the above equation, we can show by induction on k that ${}^{\mathcal{E}}B_l^{0,k} = {}^{\mathcal{F}}B_l^{0,k}$. Therefore, we can conclude in Line 18 that ${}^{\mathcal{E}}B_l^0 = {}^{\mathcal{F}}B_l^0$.

Moving to Line 21, we can now conclude that for each $\mu \in \rho(\Sigma)$ and associated edge e in $\mathcal{F}$ with $\sigma_e = \mu$ and for all $e' \in A'_e$ it holds that $g_e^1 = g_e^0 \wedge \neg{}^{\mathcal{E}}B_l^0$ and $g_{e'}^1 = g_{e'}^0 \wedge \neg{}^{\mathcal{E}}B_l^0$ if $\mu \in \rho(\Sigma_c)$ and that $g_e^1 = g_e^0$ and $g_{e'}^1 = g_{e'}^0$ if $\mu \in \rho(\Sigma_u)$. Using the fact that $g_e^0 = \bigwedge_{e' \in A'_e} g_{e'}^0$, we can conclude that $g_e^1 \Leftrightarrow \bigwedge_{e' \in A'_e} g_{e'}^1$.

When the algorithm goes back to Line 3 for the next iteration, we can repeat the argumentation above for $j > 0$ to conclude after each iteration that ${}^{\mathcal{E}}N_l^j \Leftrightarrow {}^{\mathcal{F}}N_l^j$ and ${}^{\mathcal{E}}B_l^j \Leftrightarrow {}^{\mathcal{F}}B_l^j$. Therefore, for each $\mu \in \rho(\Sigma)$ and associated edge e in $\mathcal{F}$ with $\sigma_e = \mu$ and for all $e' \in A'_e$ it holds that $g_e^j \Leftrightarrow \bigwedge_{e' \in A'_e} g_{e'}^j$.

Finally, when we reach the fixed-point, i.e., $j = *$, we can conclude that for each edge $e = (x, \mu, g_\mu^*, u_\mu, y)$ in SSEFA($\mathcal{F}$) there exists a set of edges $A_e = \{(x, \sigma, g_\sigma^*, u_\sigma, y) \mid \sigma \in \rho^{-1}(\mu)\}$ in SSEFA($\mathcal{E}$) such that $g_\mu^* \Leftrightarrow \bigvee_{e \in A} g_\sigma^*$ and $u_\sigma = u_\mu$. □

Lemma 38. *Let $\mathcal{E} = \{E^1, \ldots, E^n\}$ a be deterministic normalized EFA system. Let $\rho : \Sigma_{\mathcal{E}} \to \Sigma'$ be a renaming such that the following conditions hold for all $\sigma_1, \sigma_2 \in \Sigma_{\mathcal{E}}$ with $\rho(\sigma_1) = \rho(\sigma_2)$:*

1. *$u_{\sigma_1} = u_{\sigma_2}$,*
2. *for all $i = 1, \ldots, n$, it holds that $\sigma_1 \in \Sigma^i$ if and only if $\sigma_2 \in \Sigma^i$, and for all $l_1, l_2 \in L_i$ it holds that $l_1 \xrightarrow{\sigma_1, g_{\sigma_1}, u_{\sigma_1}} l_2$ in E^i if and only if $l_1 \xrightarrow{\sigma_2, g_{\sigma_2}, u_{\sigma_2}} l_2$ in E^i,*
3. *$\sigma_1 \in \Sigma_c$ if and only if $\sigma_2 \in \Sigma_c$.*

Create the EFA system $\mathcal{F} = \{F^1, \ldots, F^n\}$ such that each $F^i = (L^i, V^i, \rho(\Sigma^i), \rightarrow^{i,F}, l_0^i, \hat{v}_0^i, L_m^i)$ with $\rightarrow^{i,F} = \{(x, \rho(\sigma), g_{\rho(\sigma)}^F, u_\sigma, y) \mid (x, \sigma, g_\sigma^F, u_\sigma, y) \in \rightarrow^{i,E}\}$ and $g_{\rho(\sigma)}^F = \bigvee_{\sigma' \in \rho^{-1}(\rho(\sigma))} g_{\sigma'}^E$. For each edge $e = (x, \mu, g_\mu^*, u_\mu, y)$ in SSEFA($\mathcal{F}$), *create the set of edges $A_e = \{(x, \sigma, g^*_\sigma, u_\sigma, y) \mid \sigma \in \rho^{-1}(\mu)\}$ in SSEFA($\mathcal{E}$) and denote each guard $g^*_f = g_f \wedge S_f$ where f is an edge, g_f the original guard of edge f before applying SSEFA, and S_f the final predicate added by SSEFA. Then $S_e \Leftrightarrow S_{e'}$ for any $e' \in A_e$.*

Proof. From Line 21 of Algorithm 1 it follows for any edge f that $g^*_f = g^0_f \wedge \bigwedge_{j=0\ldots(*-1)} \neg B^j_{t_f}$ if $\sigma_f \in \Sigma_c$, or $g^*_f = g^0_f$ if $\sigma_f \in \Sigma_u$. Therefore, $S_f = \bigwedge_{j=0\ldots(*-1)} \neg B^j_{t_f}$ if $\sigma_f \in \Sigma_c$, or $S_f = \mathbf{T}$ if $\sigma_f \in \Sigma_u$.

As renaming preserves controllability, we know that $\mu \in \rho(\Sigma_c)$ if and only if for all $\sigma \in \rho^{-1}$ it holds that $\sigma \in \Sigma_c$. Therefore, if $\mu \in \rho(\Sigma_u)$, it follows that $S_e = \mathbf{T}$ and for all $e' \in A_e$ that $S_{e'} = \mathbf{T}$. Thus $S_e \Leftrightarrow S_{e'}$.

Now consider that $\mu \in \rho(\Sigma_c)$. Therefore, $S_e = \bigwedge_{j=0\ldots(*-1)} \neg B^j_{t_e}$ and for all $e' \in A$ it holds that $S_{e'} = \bigwedge_{j=0\ldots(*-1)} \neg B^j_{t_{e'}}$. Observe that $t_e = t_{e'}$. Now, using the proof of Lemma 37 we know at each iteration j that ${}^{\mathcal{E}}B^j_l \Leftrightarrow {}^{\mathcal{F}}B^j_l$ for each location l. Therefore, it follows immediately that $S_e \Leftrightarrow S_{e'}$. This concludes the proof. □

Lemma 39. *Let $\mathcal{E} = \{E^1, \ldots, E^n\}$ a be deterministic normalized EFA system. Let $\rho : \Sigma_{\mathcal{E}} \to \Sigma'$ be a renaming such that the following conditions hold for all $\sigma_1, \sigma_2 \in \Sigma_{\mathcal{E}}$ with $\rho(\sigma_1) = \rho(\sigma_2)$:*

1. *$u_{\sigma_1} = u_{\sigma_2}$,*
2. *for all $i = 1, \ldots, n$, it holds that $\sigma_1 \in \Sigma^i$ if and only if $\sigma_2 \in \Sigma^i$, and for all $l_1, l_2 \in L_i$ it holds that $l_1 \xrightarrow{\sigma_1, g_{\sigma_1}, u_{\sigma_1}} l_2$ in E^i if and only if $l_1 \xrightarrow{\sigma_2, g_{\sigma_2}, u_{\sigma_2}} l_2$ in E^i,*
3. *$\sigma_1 \in \Sigma_c$ if and only if $\sigma_2 \in \Sigma_c$.*

Create the EFA system $\mathcal{F} = \{F^1, \ldots, F^n\}$ such that each $F^i = (L^i, V^i, \rho(\Sigma^i), \to^{i,F}, l^i_0, \hat{v}^i_0, L^i_m)$ with $\to^{i,F} = \{(x, \rho(\sigma), g^F_{\rho(\sigma)}, u_\sigma, y) \mid (x, \sigma, g^F_\sigma, u_\sigma, y) \in \to^{i,E}\}$ and $g^F_{\rho(\sigma)} = \bigvee_{\sigma' \in \rho^{-1}(\rho(\sigma))} g^E_{\sigma'}$. Then SSEFA($\mathcal{E}$) $\parallel \mathcal{E} \Leftrightarrow \rho^{-1}$(SSEFA($\mathcal{F}$)) $\parallel \mathcal{E}$.

Proof. From Lemma 37 we know that SSEFA($\mathcal{E}$) and SSEFA($\mathcal{F}$) have the same set of locations, variables, initial location, initial valuation, and marked states. Therefore, SSEFA($\mathcal{E}$) $\parallel \mathcal{E}$ and SSEFA($\mathcal{F}$) $\parallel \mathcal{F}$ also have the same set of locations, variables, initial location, initial valuation, and marked states. It remains to be proven that if $a = (l_1, \sigma, g_a, u, l_2)$ is an edge in SSEFA($\mathcal{E}$) $\parallel \mathcal{E}$ if and only if $b = (l_1, \sigma, g_b, u, l_2)$ is an edge in SSEFA($\mathcal{F}$) $\parallel \mathcal{F}$ and $g_a \Leftrightarrow g_b$.

Consider an edge $e = (x, \mu, g^*_\mu, u_\mu, y)$ in SSEFA($\mathcal{F}$). From Lemma 37 it follows that there exists a set of edges $A_e = \{(x, \sigma, g^*_\sigma, u_\mu, y) \mid \sigma \in \rho^{-1}(\mu)\}$ in SSEFA($\mathcal{E}$).

If we now apply reverse renaming on SSEFA($\mathcal{E}$), we get for edge e the set of edges $E_e = \{(x, \sigma, g^*_\mu, u_\mu, y) \mid \sigma \in \rho^{-1}(\mu)\}$. If we now perform the synchronous composition with the original plant model, the set E_e is transformed into $E'_e = \{((x, l_1), \sigma, g_\sigma \wedge g^*_\mu, u_\mu, (y, l_1)) \mid (x, \sigma, g^*_\mu, u_\mu, y) \in E_e\}$ and the set A_e into $A'_e = \{((x, l_1), \sigma, g_\sigma \wedge g^*_\sigma, u_\mu, (y, l_2)) \mid (x, \sigma, g^*_\sigma, u_\mu, y) \in A_e\}$

for some $l_1, l_2 \in L_{\mathcal{E}}$. Therefore, $((x,l_1),\sigma,g_\sigma \wedge g^*_\mu, u_\mu,(y,l_1))$ is an edge in $\rho^{-1}(\mathrm{SSEFA}(\mathcal{F})) \parallel \mathcal{E}$ if and only if $((x,l_1),\sigma,g_\sigma \wedge g^*_\sigma, u_\mu,(y,l_2))$ is an edge in $\mathrm{SSEFA}(E) \parallel \mathcal{E}$. It remains to be proven that the guards of these edges are logically equivalent.

From Algorithm 1 we can write $g^*_\mu = g_\mu \wedge S_e$ and for each edge $e' \in A$ $g^*_\sigma = g_\sigma \wedge S_{e'}$. Furthermore, from the construction of $\mathcal{F}$ it follows that $g_\mu = \bigvee_{\sigma' \in \rho^{-1}(\mu)} g_{\sigma'}$ and from Lemma 38 it follows that $S_e \Leftrightarrow S_{e'}$. Now we can state the following.

$$\begin{aligned} g_\sigma \wedge g^*_\mu &= g_\sigma \wedge g_\mu \wedge S_e \\ &= g_\sigma \wedge (\bigvee_{\sigma' \in \rho^{-1}(\rho(\sigma))} g_{\sigma'}) \wedge S_e \\ &\Leftrightarrow g_\sigma \wedge S_e \text{ as } p \wedge (p \vee q) \Leftrightarrow p \text{ for any predicates } p \text{ and } q \\ &\Leftrightarrow g_\sigma \wedge S_{e'} \\ &\Leftrightarrow g_\sigma \wedge g_\sigma \wedge S_{e'} \\ &= g_\sigma \wedge g^*_\sigma. \end{aligned}$$

This concludes the proof. □

Proof of Theorem 10. From the definition of Ξ and the construction of ξ, it follows directly that $\xi \in \Xi$. Therefore, $\xi_1 \circ \xi \in \Xi$ and $(\mathcal{F}, \xi_1 \circ \xi)$ is a coordinator tuple.

Now we show that the two coordinator tuples are coordinator equivalent. By rewriting, we can show the following.

$$\begin{aligned} \mathcal{L}(\xi_1(\xi(\sup CN(\mathcal{F})))) &= \mathcal{L}(\xi_1(\xi(\mathrm{SSEFA}(\mathcal{F})))) \\ &= \mathcal{L}(\xi_1(\rho(\mathrm{SSEFA}(\mathcal{F})) \parallel \mathcal{E})) \\ &= \mathcal{L}(\xi_1(\mathrm{SSEFA}(\mathcal{E}) \parallel \mathcal{E})) \text{ from Lemmas 8 and 39} \\ &= \mathcal{L}(\xi_1(\mathrm{SSEFA}(\mathcal{E}))) \text{ from Lemma 31} \\ &= \mathcal{L}(\xi_1(\sup CN(\mathcal{E}))). \end{aligned}$$

This concludes the proof. □

References

H. Flordal and R. Malik. Compositional Verification in Supervisory Control. *SIAM Journal on Control and Optimization*, 48(3):1914–1938, January 2009. ISSN 0363-0129.

S. Mohajerani, R. Malik, S. Ware, and M. Fabian. Compositional synthesis of discrete event systems using synthesis abstraction. In *2011 Chinese Control and Decision Conference (CCDC)*, pages 1549–1554, May 2011. doi: 10.1109/CCDC.2011.5968439.

S. Mohajerani, R. Malik, and M. Fabian. A Framework for Compositional Synthesis of Modular Nonblocking Supervisors. *IEEE Transactions on Automatic Control*, 59(1):150–162, January 2014a. ISSN 0018-9286.

Sahar Mohajerani, Robi Malik, and Martin Fabian. Synthesis Equivalence of Triples. Technical Report R004/2013, Chalmers University of Technology, Göteborg, January 2014b. URL `http://publications.lib.chalmers.se/records/fulltext/192916/local_192916.pdf`.

Sahar Mohajerani, Robi Malik, and Martin Fabian. A framework for compositional nonblocking verification of extended finite-state machines. *Discrete Event Dynamic Systems*, 26(1):33–84, March 2016. ISSN 0924-6703, 1573-7594.

L. Ouedraogo, R. Kumar, R. Malik, and K. Åkesson. Nonblocking and safe control of discrete-event systems modeled as extended finite automata. *IEEE Trans. on Automat. Sci. and Eng.*, 8(3):560–569, July 2011. ISSN 1545-5955. doi: 10.1109/TASE.2011.2124457.