Modelling and analysis of real-time coordination patterns

Kemper, S.

Appendix A

Proofs

A.1 Correctness of Representation

In this Section, we prove that the formula representation $\varphi(\mathfrak{S})$ of a real-time system $\mathfrak{S}$, as presented in Definitions 3.1.1, 3.1.4 and 3.1.9 in Chapter 3, is correct, that means that $\varphi(\mathfrak{S})$ exhibits the same behaviour as $\mathfrak{S}$. For this, every model of $\varphi(\mathfrak{S})_k$ has to correspond to a run of length k, and vice versa. To prove this, we show that the diagram in Figure A.1 commutes.

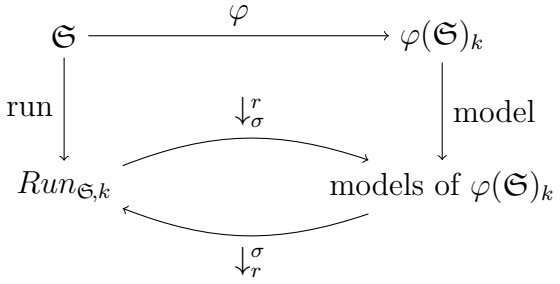

Figure A.1: Correctness of Representation

The commutative property expresses that models of $\varphi(\mathfrak{S})_k$ have a bijective correspondence to runs of the original system $\mathfrak{S}$, denoted by the maps $\downarrow^r_\sigma$ and $\downarrow^\sigma_r$: the run $\downarrow^\sigma_r(\downarrow^r_\sigma(r))$ of the model $\downarrow^r_\sigma(r)$ belonging to a run r again is r, and the model $\downarrow^r_\sigma(\downarrow^\sigma_r(\sigma))$ of a run $\downarrow^\sigma_r(\sigma)$ belonging to a model σ again is σ.

Remark A.1.1 (Notation). In the sequel, we use the notation of representation variables introduced in Section 3.1.1, and we use the symbol $\sim$ to refer to any arithmetic comparison (cf. Definitions 2.1.2 and 2.1.7).

As before, we use $\mathfrak{S}_\mathfrak{S}$ to refer to the associated LTS of a system $\mathfrak{S}$, with $\mathfrak{S}\in\{\mathfrak{A},\mathfrak{T},\mathfrak{N}\}$, and we use $Run_{\mathfrak{S},k}$ to refer to the set of all runs of $\mathfrak{S}_\mathfrak{S}$ up to length k.

Further, we use the symbols $\varphi(\mathfrak{S})_k$, σ and $\mathcal{V}(\varphi(\mathfrak{S})_k)$ to refer to the k-unfolding of $\mathfrak{S}$, a model of $\varphi(\mathfrak{S})_k$, and the set of all models of $\varphi(\mathfrak{S})_k$, respectively.

We first show that the formula representation is sound, i.e., that every model $\sigma{\in}\mathcal{V}(\varphi(\mathfrak{S})_k)$ yields a run $r{\in}Run_{\mathfrak{S},k}$.

Definition A.1.2 (Derived Run). For $\sigma{\in}\mathcal{V}(\varphi(\mathfrak{S})_k)$, the *derived run* r_σ is

$$\begin{array}{ll} r_\sigma{=}\langle l_0,\nu_0\rangle \xrightarrow{a_1} \langle l_1,\nu_1\rangle \xrightarrow{a_2} \ldots \xrightarrow{a_k} \langle l_k,\nu_k\rangle, & \text{if } \mathfrak{S}=\mathfrak{A}, \\ r_\sigma{=}\langle l_0,\delta_0,\nu_0\rangle \xrightarrow{P_1,\bar\delta_1,t_1} \langle l_1,\delta_1,\nu_1\rangle \xrightarrow{P_2,\bar\delta_2,t_2} \ldots \xrightarrow{P_k,\bar\delta_k,t_k} \langle l_k,\delta_k,\nu_k\rangle, & \text{if } \mathfrak{S}=\mathfrak{T}, \\ r_\sigma{=}\langle l_0,\delta_0,\nu_0\rangle \xrightarrow{\mathfrak{c}_1,\gamma_1} \langle l_1,\delta_1,\nu_1\rangle \xrightarrow{\mathfrak{c}_2,\gamma_2} \ldots \xrightarrow{\mathfrak{c}_k,\gamma_k} \langle l_k,\delta_k\nu_k\rangle, & \text{if } \mathfrak{S}=\mathfrak{N}, \end{array} \tag{i}$$

where we have for all $0{\le}k_0{\le}k$, $1{\le}k_1{\le}k$

$$l_{k_0} = s, \text{ iff } \sigma(\mathtt{s}_{\mathtt{k}_0}){=}\mathtt{tt}, \tag{ii}$$

$$\nu_{k_0}(x) = \sigma(\mathtt{z}_{\mathtt{k}_0}){-}\sigma(\mathtt{x}_{\mathtt{k}_0}) \tag{iii}$$

$$a_{k_1} = \begin{cases} \mathfrak{a}, \text{ iff } \sigma(\alpha_{\mathtt{k}_1}){=}\mathtt{tt}, \\ t, \text{ with } t{=}\sigma(\mathtt{z}_{\mathtt{k}_1}){-}\sigma(\mathtt{z}_{\mathtt{k}_1-1}), \text{ otherwise} \end{cases} \tag{iv}$$

$$P_{k_1} = \bigcup_{\sigma(\mathtt{p}_{\mathtt{k}_1})=\mathtt{tt}} p, \tag{v}$$

$$\delta_{k_0}(d) = \begin{cases} \Delta^{-1}(\mathtt{n}^{\mathtt{i}}), \text{ iff } \sigma(\mathtt{Dd}_{\mathtt{k}_0}){=}\mathtt{n}^{\mathtt{i}}{\neq}\mathtt{n}^{\perp} \\ \perp, \text{ otherwise} \end{cases}, d{\in}\mathcal{D} \tag{vi}$$

$$\bar\delta_{k_1}(q) = \begin{cases} \delta(m)_{k_1-1}, \text{ iff } q{=}s.m, m{\in}\mathcal{D} \\ \delta(m)_{k_1}, \text{ iff } q{=}t.m \text{ or } q{=}m, m{\in}\mathcal{D} \\ \Delta^{-1}(\mathtt{n}^{\mathtt{i}}), \text{ iff } q{=}p, p{\in}\mathcal{P}, \sigma(\mathtt{Dp}_{\mathtt{k}_1}){=}\mathtt{n}^{\mathtt{i}}{\neq}\mathtt{n}^{\perp} \\ \perp, \text{ otherwise} \end{cases} \tag{vii}$$

$$t_{k_1} = \sigma(\mathtt{z}_{\mathtt{k}_1}){-}\sigma(\mathtt{z}_{\mathtt{k}_1-1}) \tag{viii}$$

$$\mathbb{c}_{k_1}(p) = \begin{cases} \text{——} \text{ iff } \sigma(\mathtt{p}_{\mathtt{k}_1}){=}\mathtt{tt}, \\ \text{-?-} \text{ iff } \sigma(\mathtt{p}_{\mathtt{k}_1}){=}\mathtt{ff} \text{ and } \sigma(\mathtt{cp}_{\mathtt{k}_1}){=}\mathtt{ff} \\ \text{-!-} \text{ iff } \sigma(\mathtt{p}_{\mathtt{k}_1}){=}\mathtt{ff} \text{ and } \sigma(\mathtt{cp}_{\mathtt{k}_1}){=}\mathtt{tt} \end{cases} \tag{ix}$$

$$\gamma_{k_1} = \begin{cases} \bar\delta_{k_1}, \text{ iff } \sigma(\mathtt{z}_{\mathtt{k}_1}){=}\sigma(\mathtt{z}_{\mathtt{k}_1-1}) \\ t_{k_1}, \text{ otherwise} \end{cases} \tag{x}$$

The derived run for a products, that means for $\sigma{\in}\mathcal{V}(\varphi(\mathfrak{S}_1{\bowtie}\mathfrak{S}_2)_k)$ is defined in the same way, except for replacing l_i in (i) by $(l_{i,1}, l_{i,2})$, and rewriting (ii) to

$$l_{k_0,i} = s, \qquad \text{iff } \sigma(\mathtt{s}_{\mathtt{k}_0}) = \mathtt{tt} \text{ for } s{\in}S_i, i{=}1,2, \tag{ii'}$$

Remark A.1.3 (Derived Run). Note that in (ii), for each k_0, there exists exactly one location s such that $\sigma(\mathtt{s}_{\mathtt{k}_0}){=}\mathtt{tt}$, cf. (3.5), (3.14) and (3.24). Similarly, in (iv), there exists at most one event $\mathfrak{a}$ such that $\sigma(\alpha_{\mathtt{k}_1}){=}\mathtt{tt}$, cf. (3.6).

Since Δ is injective (cf. Section 3.1.1.3), there exists a $d_i{\in}\mathcal{D}ata$ with $\Delta(d_i){=}\mathtt{n}^{\mathtt{i}}$, such that $\Delta^{-1}(\mathtt{n}^{\mathtt{i}})$ is well-defined in (vii), (vi) and (x).

Lemma A.1.4 (Soundness). For $\sigma{\in}\mathcal{V}(\varphi(\mathfrak{S})_k)$, the derived run r_σ is a run of $\mathfrak{S}_{\mathfrak{S}}$ of length k, i.e., $r_\sigma{\in}Run_{\mathfrak{S},k}$.

Proof. Induction on k.

IA $\sigma{\models}\varphi(\mathfrak{S})_0$: $\sigma(\bar{\mathtt{s}}_0)\overset{(3.1),(3.10),(3.20)}{=}\mathtt{tt}$ for the initial location $\bar{s}$, thus $l_0\overset{\text{(ii)}}{=}\bar{s}$. For all clocks x, $\nu_0(x)\overset{\text{(iii)}}{=}\sigma(\mathtt{z}_0){-}\sigma(\mathtt{x}_0)\overset{(3.1),(3.10),(3.20)}{=}0$, therefore in particular $\nu_0{\models}I(\bar{s})$, and thus $r_\sigma{=}\langle\bar{s},\mathbf{0}\rangle{\in}Run_{\mathfrak{S},0}$ for $\mathfrak{S}{=}\mathfrak{A}$.

For $\mathfrak{S}{\neq}\mathfrak{A}$, in addition we have $\sigma(\mathtt{Dd}_0)\overset{(3.10),(3.20)}{=}\mathtt{n}^\perp$ for $d{\in}\mathcal{D}$, and $\sigma(\mathtt{Dp}_0)\overset{(3.10),(3.20)}{=}\mathtt{n}^\perp$ for $p{\in}\mathcal{P}$, thus $r_\sigma{=}\langle\bar{s},\mathbf{0},\mathbf{0}\rangle{\in}Run_{\mathfrak{S},0}$ for $\mathfrak{S}{\in}\{\mathfrak{T},\mathfrak{M}\}$.

IH $\sigma{\models}\varphi(\mathfrak{S})_k$: $r_\sigma{\in}Run_{\mathfrak{S},k}$ for some $k{\geq}0$.

IS $\sigma{\models}\varphi(\mathfrak{S})_{k+1}$: we consider the different systems separately

$\mathfrak{S}=\mathfrak{A}$: $r_\sigma{=}\langle l_0,\nu_0\rangle\xrightarrow{a_1}\ldots\xrightarrow{a_k}\langle l_k,\nu_k\rangle\xrightarrow{a_{k+1}}\langle l_{k+1},\nu_{k+1}\rangle$, and either for some $e{\in}E$ $\sigma{\models}\varphi^{action}(e)_{k+1/t}$, or $\sigma{\models}\varphi^{delay}(s)_{k+1/t}$ for some $s{\in}S$ (cf. (3.7) and (3.29)).
Case $\sigma{\models}\varphi^{action}(e)_{k+1/t}$ $(*)$: let $e{=}(s,\mathfrak{a},cc,\lambda,s')$ (cf. (3.2)), then

- $l_k\overset{\text{IH}}{=}s$, $l_{k+1}\overset{\text{(ii)}}{=}s'$
- $a_{k+1}\overset{\text{(iv)}}{=}\mathfrak{a}$
- $\nu_{k+1}{=}\nu_k[\lambda]$: for all clocks x, we have
  $\lambda(x){=}id$: $\nu_{k+1}(x)\overset{\text{(iii)}}{=}\sigma(\mathtt{z}_{k+1}){-}\sigma(\mathtt{x}_{k+1})\overset{(*)}{=}\sigma(\mathtt{z}_k){-}\sigma(\mathtt{x}_k)\overset{\text{(iii)},\text{IH}}{=}\nu_k(x)$
  $\lambda(x){=}x'$: $\nu_{k+1}(x)\overset{\text{(iii)}}{=}\sigma(\mathtt{z}_{k+1}){-}\sigma(\mathtt{x}_{k+1})\overset{(*)}{=}\sigma(\mathtt{z}_{k+1}){-}\sigma(\mathtt{x}'_{k+1})\overset{\text{(iii)}}{=}\nu_{k+1}(x')$
  $\lambda(x){=}n$: $\nu_{k+1}(x)\overset{\text{(iii)}}{=}\sigma(\mathtt{z}_{k+1}){-}\sigma(\mathtt{x}_{k+1})\overset{(*)}{=}\sigma(\mathtt{z}_{k+1}){-}(\sigma(\mathtt{z}_{k+1}){-}n){=}n$
- $\nu_k{\models}cc$: for $cc{=}x{\sim}n$,[1] we have $\mathtt{cc}_k{=}(\mathtt{z}_k{-}\mathtt{x}_k){\sim}n$. Because of $(*)$, we have $\sigma{\models}\mathtt{cc}_k$, that means $(\sigma(\mathtt{z}_k){-}\sigma(\mathtt{x}_k)){\sim}n$ holds, and since $\nu_k(x)\overset{\text{IH},\text{(iii)}}{=}(\sigma(\mathtt{z}_k){-}\sigma(\mathtt{x}_k))$ for all clocks x, we have $\nu_k{\models}cc$. The argumentation for $\nu_{k+1}{\models}I(s')$ is similar

Thus, $\langle s,\nu_k\rangle\xrightarrow{\mathfrak{a}}\langle s',\nu_k[\lambda]\rangle$ is gained from e using (2.1)
Case $\sigma{\models}\varphi^{delay}(s)_{k+1/t}$ $(**)$: let $s{\in}S$ (cf. (3.3)), then

- $l_k\overset{\text{IH}}{=}s$, $l_{k+1}\overset{\text{(ii)}}{=}s$
- $a_{k+1}{=}t$, with $t\overset{\text{(iv)}}{=}\sigma(\mathtt{z}_{k+1}){-}\sigma(\mathtt{z}_k)$
- $\nu_{k+1}{=}\nu_k{+}t$: for all clocks x, we have
  $\nu_{k+1}(x)\overset{\text{(iv)}}{=}\sigma(\mathtt{z}_{k+1}){-}\sigma(\mathtt{x}_{k+1})\overset{(**)}{=}\sigma(\mathtt{z}_{k+1}){-}\sigma(\mathtt{x}_k){=}\sigma(\mathtt{z}_{k+1}){-}\sigma(\mathtt{x}_k){+}$
  $\sigma(\mathtt{z}_k){-}\sigma(\mathtt{z}_k){=}(\sigma(\mathtt{z}_k){-}\sigma(\mathtt{x}_k)){+}(\sigma(\mathtt{z}_{k+1}){-}\sigma(\mathtt{z}_k))\overset{\text{(iii)},\text{(iv)},\text{IH}}{=}\nu_k(x){+}t$
- $\nu_{k+1}{\models}I(s)$ as above, $\nu_k{+}t'{\models}I(s)$ for all $t'{\leq}t$ because of convexity (cf. Remark 2.1.6)

Thus, $\langle s,\nu_k\rangle\xrightarrow{t}\langle s,\nu_k{+}t\rangle$ is gained from s using (2.2).
Together, we get $r_\sigma{\in}Run_{\mathfrak{S},k+1}$ for $\mathfrak{S}{=}\mathfrak{A}$.

[1] Here and in the remainder of the proof, we only show the basic cases for simple clock constraints (without clock differences) and simple data constraints (without addition/subtraction), but the results directly carry over to constraints involving arithmetic operations, and to Boolean combinations of these.

$\mathfrak{S} = \mathfrak{T}$: $r_\sigma{=}\langle l_0,\delta_0,\nu_0\rangle \xrightarrow{P_1,\bar\delta_1,t_1} \ldots \xrightarrow{P_k,\bar\delta_k,t_k} \langle l_k,\delta_k,\nu_k\rangle \xrightarrow{P_{k+1},\bar\delta_{k+1},t_{k+1}} \langle l_{k+1},\delta_{k+1},\nu_{k+1}\rangle$, and either $\sigma{\models}\varphi^{visible}(e)_{k+1/t}$ or $\sigma{\models}\varphi^{invisible}(e)_{k+1/t}$ for some $e{\in}E$ (cf. (3.16) and (3.29)).

Case $\sigma{\models}\varphi^{visible}(e)_{k+1/t}$ (†): let $e{=}(s, P, dc, cc, \lambda, s')$ (cf. (3.11)), then

- $l_k \overset{\text{IH}}{=} s$, $l_{k+1} \overset{\text{(ii)}}{=} s'$, $t_{k+1} \overset{\text{(viii)}}{=} \sigma(\mathtt{z_{k+1}}){-}\sigma(\mathtt{z_k})$
- $\nu_{k+1}{=}\nu_k{+}t_{k+1}[\lambda]$: for all clocks x, we have
  $\lambda(x){=}id$: $\nu_{k+1}(x) \overset{\text{(iii)}}{=} \sigma(\mathtt{z_{k+1}}){-}\sigma(\mathtt{x_{k+1}}) \overset{(\dagger)}{=} \sigma(\mathtt{z_{k+1}}){-}\sigma(\mathtt{x_k}){=}$
  $\sigma(\mathtt{z_{k+1}}){-}\sigma(\mathtt{x_k}){+}\sigma(\mathtt{z_k}){-}\sigma(\mathtt{z_k}){=}$
  $(\sigma(\mathtt{z_k}){-}\sigma(\mathtt{x_k})){+}(\sigma(\mathtt{z_{k+1}}){-}\sigma(\mathtt{z_k})) \overset{\text{(iii)},\text{(viii)},\text{IH}}{=} \nu_k(x){+}t_{k+1}$
  $\lambda(x){=}x'$: $\nu_{k+1}(x) \overset{\text{(iii)}}{=} \sigma(\mathtt{z_{k+1}}){-}\sigma(\mathtt{x_{k+1}}) \overset{\dagger}{=} \sigma(\mathtt{z_{k+1}}){-}\sigma(\mathtt{x'_{k+1}}){=}$
  $\sigma(\mathtt{z_{k+1}}){-}\sigma(\mathtt{x'_{k+1}}){+}\sigma(\mathtt{z_k}){-}\sigma(\mathtt{z_k}){=}$
  $(\sigma(\mathtt{z_k}){-}\sigma(\mathtt{x'_{k+1}})){+}(\sigma(\mathtt{z_{k+1}}){-}\sigma(\mathtt{z_k})) \overset{\text{(iii)},\text{(viii)},\text{IH}}{=} \nu_{k+1}(x'){+}t_{k+1}$
  $\lambda(x){=}n$: $\nu_{k+1}(x) \overset{\text{(iii)}}{=} \sigma(\mathtt{z_{k+1}}){-}\sigma(\mathtt{x_{k+1}}) \overset{(\dagger)}{=} \sigma(\mathtt{z_{k+1}}){-}(\sigma(\mathtt{z_{k+1}}){-}n){=}n$
- $\nu_k{+}t_{k+1}{\models}cc$: for $cc{=}x{\sim}n$, we have $\mathtt{cc_k}{=}(\mathtt{z_k}{-}\mathtt{x_k}){\sim}n$. Because of (†), we have $\sigma{\models}\mathtt{cc_k}$, that means $(\sigma(\mathtt{z_k}){-}\sigma(\mathtt{x_k})){\sim}n$ holds, and since $\nu_k(x){+}t_{k+1} \overset{\text{IH},\text{(iii)},\text{(viii)}}{=} \sigma(\mathtt{z_k}){-}\sigma(\mathtt{x_k}){+}\sigma(\mathtt{z_{k+1}}){-}\sigma(\mathtt{z_k}){=}\sigma(\mathtt{z_{k+1}}){-}\sigma(\mathtt{x_k})$ for all clocks x, we have $\nu_k{+}t_{k+1}{\models}cc$.
- $P_{k+1} \overset{\text{(v)}}{=} \bigcup_{\sigma(\mathtt{p_{k+1}})=\mathtt{tt}} p \overset{(\dagger)}{=} P$
- $\bar\delta_{k+1}{\models}dc$: let $dc{=}(\mathbf{D}{\simeq}\mathbf{D}')$, ${\simeq}\in\{{=},{\leqslant}\}$ (cf. Section 3.1.1.5). For the constituents of dc, we have
  for $p{\in}\mathcal{P}|_{dc}$: $\bar\delta_{k+1}(p) \overset{\text{(vii)}}{=} \Delta^{-1}(\mathtt{n^i}) \overset{def.\Delta}{=} d_i$ if $\sigma(\mathtt{Dp_{k+1}}){=}d_i$
  for $s.m{\in}\mathcal{D}|_{dc}$: $\bar\delta_{k+1}(s.m) \overset{\text{(vii)}}{=} \Delta^{-1}(\mathtt{n^i}) \overset{def.\Delta}{=} d_i$ if $\sigma(\mathtt{Dm_k}){=}d_i$
  for $t.m{\in}\mathcal{D}|_{dc}$: $\bar\delta_{k+1}(t.m) \overset{\text{(vii)}}{=} \Delta^{-1}(\mathtt{n^i}) \overset{def.\Delta}{=} d_i$ if $\sigma(\mathtt{Dm_{k+1}}){=}d_i$
  for $m{\in}\mathcal{D}|_{dc}$: $\bar\delta_{k+1}(m) \overset{\text{(vii)}}{=} \Delta^{-1}(\mathtt{n^i}) \overset{def.\Delta}{=} d_i$ if $\sigma(\mathtt{Dm_{k+1}}){=}d_i$
  Because of (†), we have $\sigma{\models}\mathtt{dc_{k+1}}$, thus for all possible instantiations of $\mathbf{D}$ and $\mathbf{D}'$ with $\mathtt{Dp_{k+1}}$, $\mathtt{Dm_k}$, $\mathtt{Dm_{k+1}}$ or elements of $\mathcal{D}ata$ (cf. Section 3.1.1.5), we have $\bar\delta_{k+1}{\models}dc$
- $\delta_{k+1}(m){=}\perp$ if $m{\notin}\#(s')$: because of (†), in particular $\sigma{\models}(\mathtt{Dm_{t+1}}{=}\mathtt{n}^\perp)$ (cf. (3.11)), and thus $\delta_{k+1}(m) \overset{\text{(vi)}}{=} \perp$ for all $m{\notin}\#(s')$

Thus, $\langle s,\delta_k,\nu_k\rangle \xrightarrow{P,\bar\delta_{k+1},t_{k+1}} \langle s',\delta_{k+1},\nu_k{+}t_{k+1}[\lambda]\rangle$ is gained from e using (3.11).

Case $\sigma{\models}\varphi^{invisible}(e)_{k+1/t}$ (‡): let $e{=}(s, \emptyset, dc, cc, \lambda, s')$ (cf. (3.12)), then

- $l_k \overset{\text{IH}}{=} s$, $l_{k+1} \overset{\text{(ii)}}{=} s'$, $t_{k+1} \overset{\text{(viii)}}{=} \sigma(\mathtt{z_{k+1}}){-}\sigma(\mathtt{z_k})$,
- $\nu_{k+1}{=}\nu_k{+}t_{k+1}[\lambda]$, $\nu_k{+}t_{k+1}{\models}cc$, $\delta_{k+1}(m){=}\perp$ if $m{\notin}\#(s')$, and $\bar\delta_{k+1}{\models}dc$ as before
- $P_{k+1} \overset{\text{(v)}}{=} \bigcup_{\sigma(\mathtt{p_{k+1}})=\mathtt{tt}} p \overset{(\ddagger)}{=} \emptyset$
- $\delta_{k+1}(p) \overset{\text{(vii)}}{=} \perp$ for all $p{\in}\mathcal{P}$, because $\sigma(\mathtt{p_{k+1}}) \overset{(\ddagger)}{=} \mathtt{ff}$, and $\sigma(\mathtt{Dp_{k+1}}) \overset{(\ddagger)}{=} \mathtt{n}^\perp$ for all $p{\in}\mathcal{P}$

Thus, $\langle s,\delta_k,\nu_k\rangle \xrightarrow{\emptyset,\bar\delta_{k+1},t_{k+1}} \langle s',\delta_{k+1},\nu_k{+}t_{k+1}[\lambda]\rangle$ is gained from e using (3.11) in case $t_{k+1}{>}0$, and using (3.12) in case $t_{k+1}{=}0$.

Together, we get $r_\sigma \in Run_{\mathfrak{S},k+1}$ for $\mathfrak{S}=\mathfrak{T}$.

$\mathfrak{S} = \mathfrak{N}$: $r_\sigma = \langle l_0, \delta_0, \nu_0\rangle \xrightarrow{\mathbb{c}_1,\gamma_1} \dots \xrightarrow{\mathbb{c}_k,\gamma_k} \langle l_k, \delta_k, \nu_k\rangle \xrightarrow{\mathbb{c}_{k+1},\gamma_{k+1}} \langle l_{k+1}, \delta_{k+1}, \nu_{k+1}\rangle$, and either $\sigma \models \varphi^{commu}(e)_{k+1/t}$ or $\sigma \models \varphi^{delay}(e)_{k+1/t}$ for some $e \in E$ (cf. (3.26) and (3.29))

Case $\sigma \models \varphi^{commu}(e)_{k+1/t}$ ($\star$): let $e=(s, \mathbb{c}, dc, cc, \lambda, s')$ (cf. (3.21)), then

- $l_k \overset{\text{IH}}{=} s$, $l_{k+1} \overset{\text{(ii)}}{=} s'$
- $\nu_{k+1}=\nu_k[\lambda]$, $\nu_k \models cc$, $\nu_{k+1} \models I(s')$: equivalent to the respective cases for $\mathfrak{S}=\mathfrak{A}$ above
- $\delta_{k+1}(m)=\bot$ if $m \notin \#(s')$: equivalent to the respective case for $\mathfrak{S}=\mathfrak{T}$ above
- $\gamma_{k+1} \overset{\text{(x)},(\star)}{=} \bar{\delta}_{k+1}$
- $\bar{\delta}_{k+1} \models dc$: equivalent to the respective case for $\mathfrak{S}=\mathfrak{T}$ above
- $\bar{\delta}_{k+1}(p)=\bot$ iff $\mathbb{c}_{k+1}(p) \neq$ ——: because of ($\star$), in particular $\sigma \models \langle \mathtt{p}_{\mathbb{c}} \rangle_{\mathtt{k+1}}$ for all $p \in \mathcal{P}$. If $\sigma \models \neg \mathtt{p_{k+1}}$, then either $\mathbb{c}(p) \overset{\text{(ix)}}{=}$ -!- or $\mathbb{c}(p) \overset{\text{(ix)}}{=}$ -?-, and $\sigma(\mathtt{Dp_{k+1}}) \overset{(3.25),(\star)}{=} \mathtt{n}^{\bot}$

Thus, $\langle s, \delta_k, \nu_k\rangle \xrightarrow{\mathbb{c}_{k+1},\bar{\delta}_{k+1}} \langle s, \delta_{k+1}, \nu_k[\lambda]\rangle$ is gained from e using (2.15).

Case $\sigma \models \varphi^{delay}(e)_{k+1/t}$ ($\star\star$): let $e=(s, \mathbb{c}, dc, cc, id, s)$ (cf. (3.22)), then

- $l_k \overset{\text{IH}}{=} s$, $l_{k+1} \overset{\text{(ii)}}{=} s'$
- $\gamma_{k+1} \overset{\text{(x)},(\star\star)}{=} t_{k+1}$
- $\nu_{k+1}=\nu_k+t_{k+1}$: equivalent to the respective case for $\mathfrak{S}=\mathfrak{A}$ above
- $\nu_k \models cc$, $\nu_{k+1} \models I(s)$: equivalent to the respective cases for $\mathfrak{S}=\mathfrak{A}$ above, $\nu_k+t' \models cc$ and $\nu_k+t' \models I(s)$ for all $t' \leq t_{k+1}$ because of convexity (cf. Remark 2.1.6)
- $\delta_k \models dc$: equivalent to the respective case above, with δ_k instead of $\bar{\delta}_{k+1}$

Thus, $\langle s, \delta_k, \nu_k\rangle \xrightarrow{\mathbb{c}_{k+1},t_{k+1}} \langle s, \delta_k, \nu_k+t_{k+1}\rangle$ is gained from e using (2.16).

Together, we get $r_\sigma \in Run_{\mathfrak{S},k+1}$ for $\mathfrak{S}=\mathfrak{N}$.

Finally, we get $r_\sigma \in Run_{\mathfrak{S},k+1}$ for all systems $\mathfrak{S} \in \{\mathfrak{A}, \mathfrak{T}, \mathfrak{N}\}$, and we define the map $\downarrow_r^\sigma : \mathcal{V}(\varphi(\mathfrak{S})_k) \to Run_{\mathfrak{S},k}$ such that for every interpretation $\sigma \in \mathcal{V}(\varphi(\mathfrak{S})_k)$, we have that $\downarrow_r^\sigma(\sigma) = r_\sigma \in Run_{\mathfrak{S},k}$ is the derived run. □

Proposition A.1.5 (Derived Run, Product). For $\sigma \in \mathcal{V}(\varphi(\mathfrak{S}_1 \bowtie \mathfrak{S}_2)_k)$, the derived run r_σ is a run of $\mathfrak{S}_{\mathfrak{T}_1 \bowtie \mathfrak{T}_2}$ of length k, i.e., $r_\sigma \in Run_{\mathfrak{S}_1 \bowtie \mathfrak{S}_2,k}$.

Proof (Idea). The proof is along the same lines as the proof of Lemma A.1.4. In **IS**, we first show that for $i=1,2$, reducing a transition of the product $\mathfrak{S}_1 \bowtie \mathfrak{S}_2$ (of the form $\langle (l_{k,1}, l_{k,2}), \nu_k\rangle \xrightarrow{a_k} \langle (l_{k+1,1}, l_{k+1,2}\nu_{k+1}\rangle$ for $\mathfrak{S}=\mathfrak{A}$, for example) to the constituents of $\mathfrak{S}_i$ yields a transition e_i in $\mathfrak{S}_i$. We then argue that all possible combinations of e_1 and e_2 correspond to a valid execution in the product automaton (cf. Definitions 2.2.8, 2.3.9 and 2.4.14). In end, we get $r_\sigma \in Run_{\mathfrak{S}_1 \bowtie \mathfrak{S}_2,k}$. □

We now show that the formula representation is complete, i.e, for every run $r{\in}Run_{\mathfrak{S},k}$, we can find a model $\sigma{\in}\mathcal{V}(\varphi(\mathfrak{S})_k)$.

Definition A.1.6 (Derived Interpretation). For $r{\in}Run_{\mathfrak{S},k}$, the *derived interpretation σ_r over (the variables in) $\varphi(\mathfrak{S})_k$* is (we use the notation of (i))

$$\sigma_r(\mathtt{s}_{\mathtt{k}_0}) = \mathtt{tt},\ \text{iff } s{=}l_{k_0} \tag{xi}$$

$$\sigma_r(\mathtt{z}_{\mathtt{k}_0}) = \begin{cases} 0,\ \text{if } k_0{=}0 \\ \sigma_r(\mathtt{z}_{\mathtt{k}_0-1}){+}t,\ \text{if } \mathfrak{S}{=}\mathfrak{A} \text{ and } a_{k_0}{=}t \\ \sigma_r(\mathtt{z}_{\mathtt{k}_0-1}){+}t_{k_0},\ \text{if } (\mathfrak{S}{=}\mathfrak{T}) \text{ or } (\mathfrak{S}{=}\mathfrak{N} \text{ and } \gamma_{k_0}{=}t_{k_0}) \\ \sigma_r(\mathtt{z}_{\mathtt{k}_0-1}),\ \text{otherwise} \end{cases} \tag{xii}$$

$$\sigma_r(\mathtt{x}_{\mathtt{k}_0}) = \sigma_r(\mathtt{z}_{\mathtt{k}_0}){-}\nu_{k_0}(x) \tag{xiii}$$

$$\sigma_r(\alpha_{k_0}) = \begin{cases} \mathtt{ff},\ \text{if } k_0{=}0 \\ \mathtt{tt},\ \text{iff } a_{k_0}{=}\mathfrak{a} \\ \mathtt{ff},\ \text{otherwise} \end{cases} \tag{xiv}$$

$$\sigma_r(\mathtt{p}_{\mathtt{k}_0}) = \begin{cases} \mathtt{ff},\ \text{if } k_0{=}0 \\ \mathtt{tt},\ \text{if } (p{\in}P_{k_0} \text{ and } \mathfrak{S}{=}\mathfrak{T}) \text{ or } (\mathbb{c}_{k_0}(p){=}\text{—} \text{ and } \mathfrak{S}{=}\mathfrak{N}) \\ \mathtt{ff},\ \text{otherwise} \end{cases} \tag{xv}$$

$$\sigma_r(\mathtt{Dp}_{\mathtt{k}_0}) = \begin{cases} \mathtt{n}^{\perp},\ \text{if } k_0{=}0 \\ \Delta(\bar{\delta}_{k_0}(p)),\ \text{if } (p{\in}P_{k_0} \text{ and } \mathfrak{S}{=}\mathfrak{T}) \text{ or } (\mathbb{c}_{k_0}(p){=}\text{—} \text{ and } \mathfrak{S}{=}\mathfrak{N}) \\ \mathtt{n}^{\perp},\ \text{otherwise} \end{cases} \tag{xvi}$$

$$\sigma_r(\mathtt{Dd}_{\mathtt{k}_0}) = \begin{cases} \mathtt{n}^{\perp},\ \text{if } k_0{=}0 \\ \Delta(\delta_{k_0}(d)),\ \text{if } d{\in}\#(l_{k_0}) \\ \Delta(\bar{\delta}_{k_0}(d)),\ \text{otherwise} \end{cases} \tag{xvii}$$

$$\sigma_r(\mathtt{d}_{\mathtt{k}_0}) = \begin{cases} \mathtt{ff},\ \text{if } \sigma_r(\mathtt{Dd}_{\mathtt{k}_0}){=}\mathtt{n}^{\perp} \\ \mathtt{tt},\ \text{otherwise} \end{cases} \tag{xviii}$$

$$\sigma_r(\mathtt{cp}_{\mathtt{k}_0}) = \begin{cases} \mathtt{tt},\ \text{if } (k_0{=}0) \text{ or } (\mathbb{c}_{k_0}(p){=}\text{-!-}) \\ \mathtt{ff},\ \text{if } \mathbb{c}_{k_0}(p){=}\text{-?-} \\ \text{unspecified, otherwise} \end{cases} \tag{xix}$$

for all $0{\leq}k_0{\leq}k$. The derived interpretation for a run of the product, that means for $r{\in}Run_{\mathfrak{S}_1\bowtie\mathfrak{S}_2,k}$, is defined in the same way, except for rewriting (xi) to

$$\sigma_r(\mathtt{s}_{\mathtt{k}_0}) = \mathtt{tt}, \qquad \text{iff } s{=}l_{k_0,i} \text{ for } s{\in}S_i, i{=}1,2 \tag{xi'}$$

Lemma A.1.7 (Completeness). For $r{\in}Run_{\mathfrak{S},k}$, the derived interpretation σ_r is a model of the k-unfolding of $\mathfrak{S}$, that means $\sigma_r{\models}\varphi(\mathfrak{S})_k$.

Proof. Induction on k.

IA $r{=}\langle l_0,\nu_0\rangle$ respectively $r{=}\langle l_0,\delta_0,\nu_0\rangle$ (cf. (i) again). We have $\sigma_r(\bar{\mathtt{s}}_0)\overset{\text{(xi)}}{=}\mathtt{tt}$ for the initial location $\bar{s}{=}l_0$, and $\sigma_r(\mathtt{s}_0)\overset{\text{(xi)}}{=}\mathtt{ff}$ otherwise. For clocks, we have $\sigma_r(\mathtt{z}_0)\overset{\text{(xii)}}{=}0$,

and $\sigma_r(\mathtt{x_0}) \overset{(xiii)}{=} \sigma_r(\mathtt{z_0}) - \nu_0(x) = 0$ for all other clocks x. For $I(\bar{s}) = x \sim c$,[2] we have $\mathtt{I(\bar{s})_0} = \mathtt{z_0} - \mathtt{x_0} \sim n$. Because $\nu_0 \models I(\bar{s})$ (cf. Definitions 2.2.4, 2.3.5 and 2.4.6), in particular $0 = \nu_0(x) \sim n$, therefore $\sigma_r(\mathtt{x_0}) \sim n$ holds, and thus $\sigma_r \models \mathtt{I(\bar{s})_0}$.

We have $\sigma_r(\mathtt{p_0}) \overset{(xv)}{=} \mathtt{ff}$, $\sigma_r(\mathtt{Dp_0}) \overset{(xvi)}{=} \mathtt{n}^{\perp}$, and $\sigma_r(\mathtt{cp_0}) \overset{(xix)}{=} \mathtt{tt}$ for all ports p, and for all data variables d we have $\sigma_r(\mathtt{Dd_0}) \overset{(xvii)}{=} \mathtt{n}^{\perp}$ and $\sigma_r(\mathtt{d_0}) \overset{(xviii)}{=} \mathtt{ff}$.

Thus, $\sigma_r \models \varphi^{init}(\mathfrak{S})$ (cf. (3.1), (3.10) and (3.20)), and therefore $\sigma_r \models \varphi(\mathfrak{S})_0$.

IH $r \in Run_{\mathfrak{S},k}$: $\sigma_r \models \varphi(\mathfrak{S})_k$, for some $k \geq 0$.

IS $r \in Run_{\mathfrak{S},k+1}$: we again consider the different systems separately

$\mathfrak{S} = \mathfrak{A}$: $r = \langle l_0, \nu_0 \rangle \xrightarrow{a_1} \ldots \xrightarrow{a_k} \langle l_k, \nu_k \rangle \xrightarrow{a_{k+1}} \langle l_{k+1}, \nu_{k+1} \rangle \in Run_{\mathfrak{A},k+1}$, and either the last step $\langle l_k, \nu_k \rangle \xrightarrow{a_{k+1}} \langle l_{k+1}, \nu_{k+1} \rangle$ is an action transition (2.1) resulting from execution of a transition $e = (s, \mathfrak{a}, cc, \lambda, s')$, or it is a delay transition (2.2) in location s.

In case of an action transition, we have $l_k = s$, $l_{k+1} = s'$, $a_{k+1} = \mathfrak{a}$ for some $\mathfrak{a} \in \Sigma$, and $\nu_{k+1} = \nu_k$. Then

- $\sigma_r(\mathtt{s_{k+1}}) \overset{(xi)}{=} \mathtt{tt}$ for $s = l_{k+1}$, and $\mathtt{ff}$ otherwise
- $\sigma_r(\mathtt{z_{k+1}}) \overset{(xii)}{=} \sigma_r(\mathtt{z_k})$
- $\sigma_r(\mathtt{x_{k+1}}) \overset{(xiii)}{=} \sigma_r(\mathtt{z_{k+1}}) - \nu_{k+1}(x) = \begin{cases} \overset{\lambda(x)=id,(xii)}{=} \sigma_r(\mathtt{z_k}) - \nu_k(x) = \sigma_r(\mathtt{x_k}) \\ \overset{\lambda(x)=x'}{=} \sigma_r(\mathtt{z_{k+1}}) - \nu_{k+1}(x') \overset{(xiii)}{=} \sigma_r(\mathtt{x'_{k+1}}) \\ \overset{\lambda(x)=n}{=} \sigma_r(\mathtt{z_{k+1}}) - n \end{cases}$
- $\sigma_r(\alpha_{\mathtt{k+1}}) \overset{(xiv)}{=} \mathtt{tt}$ for $\mathfrak{a} = a_{k+1}$, and $\mathtt{ff}$ otherwise
- For $cc = x \sim n$, we have $\mathtt{cc_k} = (\mathtt{z_k} - \mathtt{x_k}) \sim n$. Because $\nu_k \models (x \sim n)$ (Definition 2.2.4), we have $\sigma_r(\mathtt{z_k}) - \sigma_r(\mathtt{x_k}) \overset{(xiii),(IH)}{\sim} n$, thus $\sigma_r \models \mathtt{cc_k}$. The argumentation for $\sigma_r \models \mathtt{I(s')_{k+1}}$ is similar.

From the above, we get $\sigma_r \models \varphi^{action}(e)_{k+1/t}$ (3.2) (so $\sigma_r \models \varphi^{trans}(\mathfrak{A})_{k+1/t}$ (3.4)), $\sigma_r \models \varphi^{location}(\mathfrak{A})_{k+1/t}$ (3.5), and $\sigma_r \models \varphi^{mutex}(\mathfrak{A})_{k+1/t}$ (3.6).

In case of a delay transition, we have $l_k = s = l_{k+1}$, $a_{k+1} = t$ for some $t \in \mathtt{Time}$, and $\nu_{k+1} = \nu_k + t$. Then

- $\sigma_r(\mathtt{s_{k+1}}) \overset{(xi)}{=} \mathtt{tt}$ for $s = l_{k+1}$, and $\mathtt{ff}$ otherwise
- $\sigma_r(\mathtt{z_{k+1}}) \overset{(xii)}{=} \sigma_r(\mathtt{z_k}) + t$
- $\sigma_r(\mathtt{x_{k+1}}) \overset{(xiii)}{=} \sigma_r(\mathtt{z_{k+1}}) - \nu_{k+1}(x) \overset{(xii)}{=} \sigma_r(\mathtt{z_k}) + t - (\nu_k(x) + t) =$ $\sigma_r(\mathtt{z_k}) - \nu_k(x) \overset{(xiii)}{=} \sigma_r(\mathtt{x_k})$
- $\sigma_r(\alpha_{\mathtt{k+1}}) \overset{(xiv)}{=} \mathtt{ff}$ for all $\mathfrak{a} \in \Sigma$
- $\sigma_r \models \mathtt{I(s)_{k+1}}$: similar to the argumentation for $\sigma_r \models \mathtt{cc_k}$ above

From the above, we get $\sigma_r \models \varphi^{delay}(e)_{k+1/t}$ (3.3) (so $\sigma_r \models \varphi^{trans}(\mathfrak{A})_{k+1/t}$ (3.4)), $\sigma_r \models \varphi^{location}(\mathfrak{A})_{k+1/t}$ (3.5), and $\sigma_r \models \varphi^{mutex}(\mathfrak{A})_{k+1/t}$ (3.6).

Together, we get $\sigma_r \models \varphi(\mathfrak{S})_{k+1}$ for $\mathfrak{S} = \mathfrak{A}$

[2] Here and in the remainder of the proof, again we only show the basic cases for simple clock constraints (without clock differences) and simple data constraints (without addition/subtraction).

$\mathfrak{S} = \mathfrak{T}$: $r = \langle l_0, \delta_0, \nu_0\rangle \xrightarrow{P_1,\bar{\delta}_1,t_1} \ldots \xrightarrow{P_k,\bar{\delta}_k,t_k} \langle l_k, \delta_k, \nu_k\rangle \xrightarrow{P_{k+1},\bar{\delta}_{k+1},t_{k+1}} \langle l_{k+1}, \delta_{k+1}, \nu_{k+1}\rangle \in Run_{\mathfrak{T},k+1}$, and the last step $\langle l_k, \delta_k, \nu_k\rangle \xrightarrow{P_{k+1},\bar{\delta}_{k+1},t_{k+1}} \langle l_{k+1}, \delta_{k+1}, \nu_{k+1}\rangle$ results from following either a visible transition (2.7) or an invisible transition transition (2.7), (2.8).

In case of a visible transition $e=(s, P, dc, cc, \lambda, s')$, we have $l_k=s$, $l_{k+1}=s'$, $P_{k+1}=P$, and $\nu_{k+1}=\nu_k+t_{k+1}$, with $t_{k+1}>0$. Then

- $\sigma_r(\mathtt{s}_{k+1})$, $\sigma_r(\mathtt{x}_{k+1})$: equivalent to the respective cases for $\mathfrak{S}=\mathfrak{A}$ above
- $\sigma_r(\mathtt{z}_{k+1}) \overset{\text{(xii)}}{=} \sigma_r(\mathtt{z}_k)+t_{k+1}$
- $\sigma_r(\mathtt{p}_{k+1}) \overset{\text{(xv)}}{=} \mathtt{tt}$ for $p\in P$, $\mathtt{ff}$ otherwise
- $\sigma_r(\mathtt{Dp}_{k+1}) \overset{\text{(xvi)}}{=} \Delta(\bar{\delta}_{k+1})$ for $p\in P$, $\mathtt{n}^\perp$ otherwise
- $\sigma_r(\mathtt{Dd}_{k+1}) \overset{\text{(xvii)}}{=} \Delta(\delta_{k+1}(d))$ if $d\in\#(s')$, $\Delta(\bar{\delta}_{k+1}(d))$ otherwise
- $\sigma_r(\mathtt{d}_{k+1}) \overset{\text{(xviii)}}{=} \mathtt{ff}$ if $\sigma_r(\mathtt{Dd}_{k+1})=\mathtt{n}^\perp$, $\mathtt{tt}$ otherwise
- For $I(s)=(x\sim n)$, we have $\mathtt{I(s)}_{k+1\Delta}=(\mathtt{z}_{k+1}-\mathtt{x}_k)\sim n$. Since $\nu_k+t\models I(s)$ for all $0\leq t\leq t_{k+1}$ (Definition 2.3.5), in particular $\nu_k(x)+t_{k+1}\models(x\sim n)$; so $\sigma_r(\mathtt{z}_{k+1})-\sigma_r(\mathtt{x}_k) \overset{\text{IH,(xiii)}}{=} (\sigma_r(\mathtt{z}_k)+t_{k+1})-(\sigma_r(\mathtt{z}_k)-\nu_k(x))=\nu_k(x)+t_{k+1}$, which means that $\sigma_r(\mathtt{z}_{k+1})-\sigma_r(\mathtt{x}_k)\sim n$ holds. Therefore $\sigma_r\models\mathtt{I(s)}_{k+1\Delta}$. The argumentation for $\sigma_r\models\mathtt{cc}_{k+1\Delta}$ is similar, and the argumentation for $\sigma_r\models\mathtt{I(s')}_{k+1}$ is equivalent to the respective case for $\mathfrak{S}=\mathfrak{A}$ above
- For $dc=(\mathbf{D}\simeq\mathbf{D}')$, $\simeq\in\{=,\leqslant\}$ (cf. Definition 2.1.7 and Section 3.1.1.5), we have $\mathtt{dc}_{k+1}=(\mathtt{D}\simeq\mathtt{D}')$, where $\mathtt{D}$ and $\mathtt{D}'$ are either port data variables $\mathtt{Dp}_{k+1}$ or data content variables $\mathtt{Dd}_k$, $\mathtt{Dd}_{k+1}$ (cf. Section 3.1.1.5). Because $\bar{\delta}_{k+1}\models dc$ (Definition 2.3.5), in particular $\bar{\delta}(\mathbf{D})$ and $\bar{\delta}(\mathbf{D}')$ such that $\bar{\delta}(\mathbf{D})\simeq\bar{\delta}(\mathbf{D}')$ holds. Therefore, $\sigma_r(\mathtt{D})\simeq\sigma_r(\mathtt{D}')$ holds as well ((xvi), (xvii)),[3] and thus $\sigma_r\models\mathtt{dc}_{k+1}$.
  The case where $\mathbf{D}\in\mathcal{D}ata$ and/or $\mathbf{D}'\in\mathcal{D}ata$, that means where $\mathtt{D}$ or $\mathtt{D}'$ are data element representations $\mathtt{n}^\mathtt{i}$, is a simplification of the above.

From the above, we get $\sigma_r\models\varphi^{visible}(e)_{k+1/t}$ (3.11) (so $\sigma_r\models\varphi^{trans}(\mathfrak{T})_{k+1/t}$ (3.16), $\sigma_r\models\varphi^{location}(\mathfrak{T})_{k+1/t}$ (3.14), and $\sigma_r\models\varphi^{mutex}(\mathfrak{T})_{k+1/t}$ (3.15).

In case of an invisible transition $e=(s,\emptyset,dc,cc,\lambda,s')$, we have $l_k=s$, $l_{k+1}=s'$, $P_{k+1}=\emptyset$, and $\nu_{k+1}=\nu_k+t_{k+1}$, with $t_{k+1}\geq 0$. Then

- $\sigma_r(\mathtt{s}_{k+1})$, $\sigma_r(\mathtt{z}_{k+1})$, $\sigma_r(\mathtt{x}_{k+1})$ as above
- $\sigma_r(\mathtt{p}_{k+1}) \overset{\text{(xv)}}{=} \mathtt{ff}$ for all $p\in\mathcal{P}$
- $\sigma_r(\mathtt{Dd}_{k+1})$, $\sigma_r(\mathtt{d}_{k+1})$, $\sigma_r\models\mathtt{I(s)}_{k+1\Delta}$, $\sigma_r\models\mathtt{cc}_{k+1\Delta}$, $\sigma_r\models\mathtt{I(s')}_{k+1}$ $\sigma_r\models\mathtt{dc}_{k+1}$: as above

From the above, we get $\sigma_r\models\varphi^{invisible}(e)_{k+1/t}$ (3.12) (so $\sigma_r\models\varphi^{trans}(\mathfrak{T})_{k+1/t}$ (3.16), $\sigma_r\models\varphi^{location}(\mathfrak{T})_{k+1/t}$ (3.14), and $\sigma_r\models\varphi^{mutex}(\mathfrak{T})_{k+1/t}$ (3.15).

Together, we get $\sigma_r\models\varphi(\mathfrak{S})_{k+1}$ for $\mathfrak{S}=\mathfrak{T}$

$\mathfrak{S} = \mathfrak{N}$: $r=\langle l_0, \delta_0, \nu_0\rangle \xrightarrow{e_1,\gamma_1} \ldots \xrightarrow{e_k,\gamma_k} \langle l_k, \delta_k, \nu_k\rangle \xrightarrow{e_{k+1},\gamma_{k+1}} \langle l_{k+1}, \delta_{k+1}, \nu_{k+1}\rangle$, where $r\in Run_{\mathfrak{N},k+1}$, and either the last step $\langle l_k, \delta_k, \nu_k\rangle \xrightarrow{e_{k+1},\gamma_{k+1}} \langle l_{k+1}, \delta_{k+1}, \nu_{k+1}\rangle$

[3] Note that (Definition 2.3.5) $\delta(d)$ and $\bar{\delta}(d)$ coincide in case $d\in\#(s)$ for any location s.

is an action transition (2.15) resulting from executing a communication, or it is a delayed action transition (2.16) resulting from executing a delay.

In case of a communication $e=(s, \mathbb{c}, dc, cc, \lambda, s')$, we have $l_k=s$, $l_{k+1}=s'$, and $\gamma_{k+1}=\bar{\delta}_{k+1}$. Then

- $\sigma_r(\mathtt{s}_{k+1})$, $\sigma_r(\mathtt{x}_{k+1})$, $\sigma_r(\mathtt{z}_{k+1})$, $\sigma_r(\mathtt{d}_{k+1})$: equivalent to the respective cases for $\mathfrak{S}=\mathfrak{T}$ above
- $\sigma_r(\mathtt{p}_{k+1}) \overset{(xv)}{=} \mathtt{tt}$ if $\mathbb{c}_{k+1}(p)=$——, $\mathtt{ff}$ otherwise
- $\sigma_r(\mathtt{cp}_{k+1}) \overset{(xix)}{=} \mathtt{tt}$ if $\mathbb{c}_{k+1}(p)=\mathtt{-!-}$, $\sigma_r(\mathtt{cp}_{k+1})=\mathtt{ff}$ if $\mathbb{c}_{k+1}(p)=\mathtt{-?-}$, unspecified otherwise
- $\sigma_r(\mathtt{Dp}_{k+1}) \overset{(xvi)}{=} \Delta(\bar{\delta}_{k+1}(p))$ if $\mathbb{c}_{k+1}(p)=$——, $\mathtt{n}^{\perp}$ otherwise
- $\sigma_r(\mathtt{Dd}_{k+1})$, $\sigma_r(\mathtt{d}_{k+1})$, $\sigma_r \models \mathtt{cc}_k$, $\sigma_r \models \mathtt{dc}_{k+1}$, $\sigma_r \models \mathtt{I}(\mathtt{s}')_{k+1}$: equivalent to the respective cases for $\mathfrak{S}=\mathfrak{T}$ above

From the above, we get $\sigma_r \models \varphi^{commu}(e)_{k+1/t}$ (3.21) (so $\sigma_r \models \varphi^{trans}(\mathfrak{N})_{k+1/t}$ (3.26)), $\sigma_r \models \varphi^{location}(\mathfrak{N})_{k+1/t}$ (3.24), and $\sigma_r \models \varphi^{mutex}(\mathfrak{N})_{k+1}$ (3.25).

The case of a delay $e=(s, \mathbb{c}, dc, cc, id, s)$ is essentially equivalent to the case of a communication, and needs not be considered separately. For a delay, we get $\sigma_r \models \varphi^{delay}(e)_{k+1/t}$ (3.22) (so $\sigma_r \models \varphi^{trans}(\mathfrak{N})_{k+1/t}$ (3.26)), $\sigma_r \models \varphi^{location}(\mathfrak{N})_{k+1/t}$ (3.24), and $\sigma_r \models \varphi^{mutex}(\mathfrak{N})_{k+1}$ (3.25).

Together, we get $\sigma_r \models \varphi(\mathfrak{S})_{k+1}$ for $\mathfrak{S}=\mathfrak{N}$

Finally, we get $\sigma_r \models \varphi(\mathfrak{S})_{k+1}$ for all systems $\mathfrak{S} \in \{\mathfrak{A}, \mathfrak{T}, \mathfrak{N}\}$, and we define the map $\downarrow^r_\sigma : Run_{\mathfrak{S},k} \to \mathcal{V}(\varphi(\mathfrak{S})_k)$ such that for every run $r \in Run_{\mathfrak{S},k}$, $\downarrow^r_\sigma(r)=\sigma_r \in \mathcal{V}(\varphi(\mathfrak{S})_k)$ is the derived interpretation. □

Proposition A.1.8 (Derived Interpretation, Product). For $r \in Run_{\mathfrak{S}_1 \bowtie \mathfrak{S}_2,k}$, the derived interpretation σ_r is a model of $\varphi(\mathfrak{S}_1 \bowtie \mathfrak{S}_2)_k$, i.e. $\sigma \in \mathcal{V}(\varphi(\mathfrak{S}_1 \bowtie \mathfrak{S}_2)_k)$.

Proof (Idea). The proof is along the same lines as the proof of Lemma A.1.7: in **IS**, we show that for $i=1, 2$, the derived interpretation σ_r for a run $r \in Run_{\mathfrak{S}_1 \bowtie \mathfrak{S}_2,k+1}$, reduced to the variables of $\varphi(\mathfrak{S}_i)_k$, is a model of $\varphi(\mathfrak{S}_i)_k$. □

Using the above, the proof of Theorem 3.2.4 (found on Page 61) is straightforward:

Proof of Theorem 3.2.4. This follows directly from Lemma A.1.4 and Lemma A.1.7. □

Theorem A.1.9 (Soundness, Completeness). The formula representation $\varphi(\mathfrak{S})$ of a real-time system $\mathfrak{S}$, as defined in Definitions 3.1.1, 3.1.4 and 3.1.9, is correct, that means $\varphi(\mathfrak{S})$ exhibits the same behaviour as $\mathfrak{S}$.

Proof. This follows directly from Lemma A.1.4 and Lemma A.1.7. □

A.2 Correctness of Abstraction

In this Section, we prove that the abstraction function α, as presented in Section 4.1, yields *a correct over-approximation*. To yield an over-approximation, every finite run of the concrete system $\mathfrak{S}$ (represented by a model of $\varphi(\mathfrak{S})_k$, see Theorem A.1.9) has to be reproducible in the abstract case.[4] This is captured in Lemma 4.1.7. Here, we prove an even stronger correctness result, which in particular emphasises the structural relationships between concrete and abstract formula. We show that the diagram in Figure A.2 commutes, which allows us to conclude the existence of a homomorphism h_R between concrete and abstract set of runs.

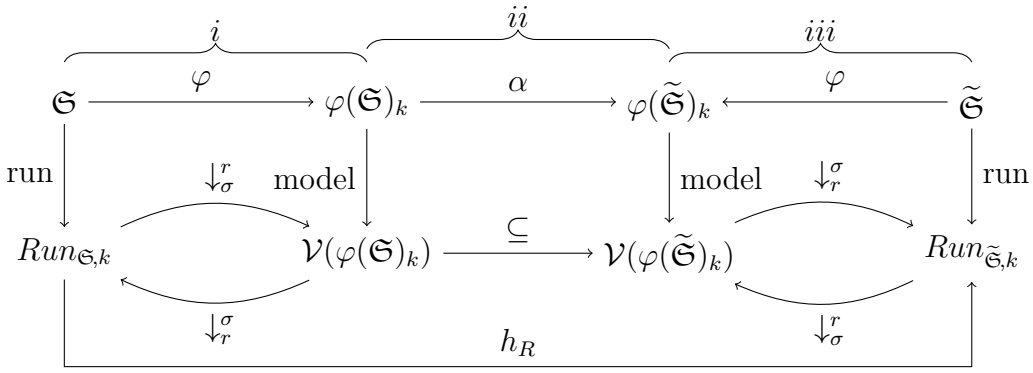

Figure A.2: Strong Correctness of Abstraction

The idea of the proof is as follows: since α works locally, it retains the formula structure of $\varphi(\mathfrak{S})$ if $\mathfrak{S}{=}\mathfrak{A}$ (cf. (3.7)), and it retains the formula structure of $\varphi(\mathfrak{S})$ up to data constraints if $\mathfrak{S}{=}\mathfrak{T}$ (cf. (3.16)).[5] Therefore, there exists some system $\widetilde{\mathfrak{S}}$ of the same representation $\varphi(\widetilde{\mathfrak{S}})_k = \alpha(\varphi(\mathfrak{S})_k)$ (up to logical equivalence and data constraints). With this, subdiagrams (i) and (iii) in Figure A.2 commute according to Theorem A.1.9. Moreover, subdiagram (ii) in Figure A.2 commutes according to Lemma 4.1.7 (since every model of $\varphi(\widetilde{\mathfrak{S}})_k$ is a model of $\alpha(\varphi(\mathfrak{S})_k)$), such that the whole diagram commutes.

Notation A.2.1 (Notation of Systems). If not stated otherwise, we shall assume the constituents of a TA $\mathfrak{A}$ to be denoted as $\mathfrak{A}{=}(S, s_0, \Sigma, \mathcal{X}, I, E)$, and of a TCA $\mathfrak{T}$ as $\mathfrak{T}{=}(S, s_0, \mathcal{P}, \mathcal{X}, I, \mathcal{D}, \#, E)$. We use the general notion $\mathfrak{S}$, with $\mathfrak{S}{\in}\{\mathfrak{A}, \mathfrak{T}\}$, whenever possible, and if applicable, we may refer to common constituents (i.e., S, s_0, $\mathcal{X}$, I, E) without explicitly mentioning $\mathfrak{A}$ or $\mathfrak{T}$. For a system with identifier $\widetilde{\mathfrak{S}}$, we add the symbol $\widetilde{\ }$ to all constituents, equivalently, for a system with identifier $\mathfrak{S}_i$, we add index i to all constituents.

We use the notation of representation variables introduced in Section 3.1.1.

[4] Note that unlike in Section A.1, where we had $\mathfrak{S}{\in}\{\mathfrak{A}, \mathfrak{T}, \mathfrak{N}\}$, here we only have $\mathfrak{S}{\in}\{\mathfrak{A}, \mathfrak{T}\}$, cf. Section 4.1.

[5] To guarantee that α yields an over-approximation, we may retain only those data constraints that reason about ports not merged by γ, cf. Definition 4.1.3 and the explanations thereafter. Therefore, we cannot expect that the formula structure of data constraints is preserved.

Definition A.2.2 (Homomorphism of Runs). Let $\mathfrak{A}$, $\widetilde{\mathfrak{A}}$ be TA, $\mathfrak{T}$, $\widetilde{\mathfrak{T}}$ be TCA, both with $\mathcal{X}\supseteq\widetilde{\mathcal{X}}$ and $|S|\geq|\widetilde{S}|$. Let $|\Sigma|\geq|\widetilde{\Sigma}|$, $|\mathcal{P}|\geq|\widetilde{\mathcal{P}}|$, and $|\mathcal{D}|\geq|\widetilde{\mathcal{D}}|$. Let $\mathfrak{S}_{\mathfrak{A}}$, $\mathfrak{S}_{\mathfrak{T}}$, $\mathfrak{S}_{\widetilde{\mathfrak{A}}}$ and $\mathfrak{S}_{\widetilde{\mathfrak{T}}}$ be the associated transition systems, and let $Run_{\mathfrak{A}}$, $Run_{\mathfrak{T}}$, $Run_{\widetilde{\mathfrak{A}}}$ and $Run_{\widetilde{\mathfrak{T}}}$ be the sets of runs. Let $\gamma_S{:}S\to\widetilde{S}$, $\gamma_\Sigma{:}\Sigma\to\widetilde{\Sigma}$, $\gamma_P{:}\mathcal{P}\to\widetilde{\mathcal{P}}$ and $\gamma_\mathcal{D}{:}\mathcal{D}\to\widetilde{\mathcal{D}}$ be total, surjective mappings.

A function $h_R{:}Run_{\mathfrak{A}}\to Run_{\widetilde{\mathfrak{A}}}$ is called a *homomorphism of runs (between $Run_{\mathfrak{A}}$ and $Run_{\widetilde{\mathfrak{A}}}$)* iff for each run

$$r{=}\langle l_0,\nu_0\rangle\xrightarrow{a_1}\langle l_1,\nu_1\rangle\xrightarrow{a_2}\langle l_2,\nu_2\rangle{\in}Run_{\mathfrak{A}},$$

there exists a run $h(r){=}\widetilde{r}$,

$$\widetilde{r}{=}\langle\widetilde{l_0},\widetilde{\nu_0}\rangle\xrightarrow{\widetilde{a_1}}\langle\widetilde{l_1},\widetilde{\nu_1}\rangle\xrightarrow{\widetilde{a_2}}\langle\widetilde{l_2},\widetilde{\nu_2}\rangle\ldots{\in}Run_{\widetilde{\mathfrak{A}}},$$

with $\gamma_S(l_i){=}\widetilde{l_i}$, $\widetilde{\nu_i}{=}\nu_i|_{\widetilde{\mathcal{X}}}$, and $\gamma_\Sigma(\gamma_i){=}\widetilde{\gamma_i}$ for all $i{\geq}0$.

A function $h_R{:}Run_{\mathfrak{T}}\to Run_{\widetilde{\mathfrak{T}}}$ is called a *homomorphism of runs (between $Run_{\mathfrak{T}}$ and $Run_{\widetilde{\mathfrak{T}}}$)* iff for each run

$$r{=}\langle l_0,\delta_0,\nu_0\rangle\xrightarrow{P_1,\bar{\delta}_1,t_1}\langle l_1,\delta_1,\nu_1\rangle\xrightarrow{P_2,\bar{\delta}_2,t_2}\langle l_2,\delta_2,\nu_2\rangle{\in}Run_{\mathfrak{T}},$$

there exists a run $h(r){=}\widetilde{r}$,

$$\widetilde{r}{=}\langle\widetilde{l_0},\widetilde{\delta_0},\widetilde{\nu_0}\rangle\xrightarrow{\widetilde{P_1},\widetilde{\bar{\delta}_1},\widetilde{t_1}}\langle\widetilde{l_1},\widetilde{\delta_1},\widetilde{\nu_1}\rangle\xrightarrow{\widetilde{P_2},\widetilde{\bar{\delta}_2},\widetilde{t_2}}\langle\widetilde{l_2},\widetilde{\delta_2},\widetilde{\nu_2}\rangle{\in}Run_{\widetilde{\mathfrak{T}}},$$

with $\gamma_S(l_i){=}\widetilde{l_i}$, $\widetilde{\nu_i}{=}\nu_i|_{\widetilde{\mathcal{X}}}$, $\gamma_P(P_i){=}\widetilde{P_i}$, $\widetilde{\bar{\delta}_i}(\widetilde{p}){=}n$ only if $\delta_i(p){=}n$ for some $p{\in}\gamma_P^{-1}(\widetilde{p})$, $\widetilde{\delta_i}(\widetilde{d}){=}n$ only if $\delta_i(d){=}n$ for some $d{\in}\gamma_\mathcal{D}^{-1}(\widetilde{d})$, and $\widetilde{\bar{\delta}_i}(\widetilde{d}){=}n$ only if $\bar{\delta}_i(d){=}n$ for some $d{\in}\gamma_\mathcal{D}^{-1}(\widetilde{d})$, for all $i{\geq}0$.

For the sets of finite runs $Run_{\mathfrak{A},k}$, $Run_{\mathfrak{T},k}$, $Run_{\widetilde{\mathfrak{A}},k}$ and $Run_{\widetilde{\mathfrak{T}},k}$, h_R is defined analogously.

Intuitively speaking, an abstraction is correct if the semantics of the abstract system is not reduced with respect to the semantics of the concrete system. That means, every behaviour that is possible in the concrete system has to be possible in the abstract system as well. Since we have defined the semantics of a real-time system $\mathfrak{S}$ via sets of runs (Definitions 2.2.4 and 2.3.5), an abstraction of $\mathfrak{S}$ is correct if for all systems $\mathfrak{S}$ and $\widetilde{\mathfrak{S}}$, such that $\widetilde{\mathfrak{S}}$ is obtained from $\mathfrak{S}$ by abstraction, there exists a homomorphism of runs $h_R{:}Run_{\mathfrak{S}}\to Run_{\widetilde{\mathfrak{S}}}$, as defined in definition A.2.2. To prove the existence of h_R, we show that Figure A.2 is a commuting diagram.

The general proof idea is shown in Figure A.3: let $\mathfrak{S}$ be a real-time system, with k-unfolding $\varphi(\mathfrak{S})_k$. The abstraction function α preserves the structure of $\varphi(\mathfrak{S})_k$, that means the abstraction $\alpha(\varphi(\mathfrak{S})_k)$ of $\varphi(\mathfrak{S})_k$ is the k-unfolding $\varphi(\widetilde{\mathfrak{S}})_k$ of some system $\widetilde{\mathfrak{S}}$. Though the abstraction function α is defined on formulas rather than on systems, the system $\widetilde{\mathfrak{S}}$ can be "derived" from the formula representation $\alpha(\varphi(\mathfrak{S})_k)$ (Proposition A.2.7). For $r{\in}Run_{\mathfrak{S},k}$ and $\widetilde{r}{\in}Run_{\widetilde{\mathfrak{S}},k}$, such that $h_R(r){=}\widetilde{r}$, there exists an interpretation $\sigma{\in}\mathcal{V}(\varphi(\widetilde{\mathfrak{S}})_k)$, such that $\widetilde{r}$ is the derived run r_σ of σ.

In other words, the commutative property can be summarised as follows: the possible behaviour of the abstract system $\widetilde{\mathfrak{S}}$, given by the set of runs $Run_{\widetilde{\mathfrak{S}},k}$, is

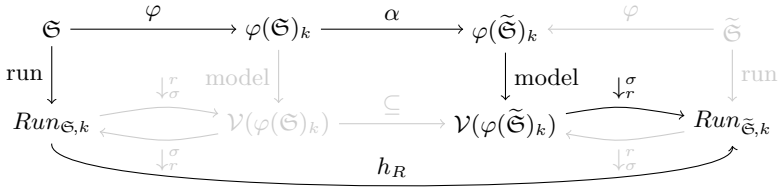

Figure A.3: Abstraction by Omission: Basic proof idea

obtained from the possible behaviour of the original system $Run_{\mathfrak{S},k}$ and the homomorphism of runs h_R (lower path in Figure A.3). $Run_{\widetilde{\mathfrak{S}},k}$ is also obtained from the k-unfolding $\varphi(\mathfrak{S})_k$ of $\mathfrak{S}$, the abstraction function α, the set of models of $\alpha(\varphi(\mathfrak{S})_k)$, and the set of derived runs for these interpretations (upper path in Figure A.3).

We can already state that

Proposition A.2.3 (Commuting Subdiagrams). The subdiagrams (*i*) and (*iii*) in Figure A.2 are a commuting diagram each.

Proof. This follows directly from Theorem A.1.9. □

The subdiagram (*iii*) in Figure A.2 is a commuting diagram when considered separately. Yet, with respect to the overall context of Figure A.2, the fact that MO is not defined on systems $\mathfrak{S}$ but on formulas has to be taken into account. However, Proposition A.2.7 below will show the existence of such an abstract system $\widetilde{\mathfrak{S}}$.

We first show that the abstract formula $\alpha(\varphi(\mathfrak{S}))$ is weaker than the concrete formula $\varphi(\mathfrak{S})$ (cf. Lemma 4.1.7 on Page 73).

Proof of Lemma 4.1.7. Let L be a literal. The proof is done inductively on the structure of the formula F:

IA: We need to consider the different cases in (4.1)

- If $F{=}L$, $Conts(L){\cap}^\bullet\alpha{=}\emptyset$, then $\alpha(F)\overset{(4.1a)}{=}L$. $(L \to L)$ holds trivially.
- If $F{=}L$, $Conts(L){\cap}^\bullet\alpha{\neq}\emptyset$, $L{=}p{\in}\mathtt{P}$, then $\alpha(F)\overset{(4.1b)}{=}\gamma(p)$. By definition of γ, $\gamma(p){=}q$ for some $q{\in}\mathtt{P}'$. By definition of γ_α (4.3), $(p \to q)$ holds.
- For all other literals L, $\alpha(L)\overset{(4.1e)}{=}\mathtt{true}$. $(L \to \mathtt{true})$ holds trivially.
- If $F{=}\neg p{\wedge}p'$, with $p, p'{\in}\mathtt{P}$ and $\gamma(p){=}\gamma(p'){=}q$ (basic case of (4.1c)), then $\alpha(F)\overset{(4.1c),(4.2a)}{=}\alpha(\neg p){\wedge}\alpha(p')\overset{(4.1c),(4.1b)}{=}q{\wedge}q{=}q$. By definition of γ_α, $((\neg p{\wedge}p'){\to}q)$ holds.
- If $F{=}\neg p{\wedge}\neg p''$, with $p, p''{\in}\mathtt{P}$ and $\gamma(p){=}\gamma(p''){=}q$ (basic case of (4.1d)), then $\alpha(F)\overset{(4.1d),(4.2a)}{=}\alpha(\neg p){\wedge}\alpha(\neg p'')\overset{(4.1d),(4.1b)}{=}\neg q{\wedge}\neg q = \neg q$. By definition of γ_α, $((\neg p{\wedge}\neg p'') \to \neg q)$ holds.

IH: For formulas F_1 and F_2, $(F_1 \to \alpha(F_1))$ and $(F_2 \to \alpha(F_2))$ holds.

IS:
- If $F{=}F_1{\wedge}F_2$, then $\alpha(F) \overset{(4.2a)}{=} \alpha(F_1){\wedge}\alpha(F_2)$. $((F_1{\wedge}F_2) \rightarrow (\alpha(F_1){\wedge}\alpha(F_2)))$ holds by IH and propositional logic.
- If $F{=}F_1{\vee}F_2$, then $\alpha(F) \overset{(4.2b)}{=} \alpha(F_1){\vee}\alpha(F_2)$. $((F_1{\vee}F_2) \rightarrow (\alpha(F_1){\vee}\alpha(F_2)))$ holds by IH and propositional logic.
- If $F{=}F_1{\wedge}\gamma_\alpha$, then $\alpha(F) \overset{(4.4)}{=} \alpha(F_1) \wedge \gamma_\alpha$. $((F_1{\wedge}\gamma_\alpha) \rightarrow (\alpha(F_1){\wedge}\gamma_\alpha))$ holds by IH and propositional logic.

□

We want to show that MO preserves the formula representation. Since MO is defined for formulas in NNF (cf. Definition 4.1.5), we first show that transformation to NNF preserves the formula representation.

Remark A.2.4 (NNF preserves the Formula Representation). Let $\mathfrak{S}$ be a real-time system, with formula representation $\varphi(\mathfrak{S})$ and k-unfolding $\varphi(\mathfrak{S})_k$. The *transformation to NNF of* $\varphi(\mathfrak{S})$ *and* $\varphi(\mathfrak{S})_k$ *preserves the formula structure*, that means, $NNF(\varphi(\mathfrak{S}))$ is a formula of the form (3.7) respectively (3.16), and similarly, $NNF(\varphi(\mathfrak{S})_k)$ is a formula of the form (3.29).

Proof. For $\mathfrak{S}{=}\mathfrak{A}$, the formulas $\varphi(\mathfrak{A})$ and $\varphi(\mathfrak{A})_k$ are in NNF already, so nothing needs to be shown.

For $\mathfrak{S}{=}\mathfrak{T}$, the only parts of $\varphi(\mathfrak{T})$ and $\varphi(\mathfrak{T})_k$ which are not yet in NNF are the representations of data constraints. It is easy to see that for a data constraint $dc{\in}DC(\mathcal{P},\mathcal{D})$, with representation $\mathtt{dc}{\in}DC(\mathtt{P_{DA}},\mathtt{D})$, the transformation $NNF(\mathtt{dc})$ to NNF is a well-formed data constraint according to Definition 2.1.7, too, that means $NNF(\mathtt{dc}){\in}DC(\mathtt{P_{DA}},\mathtt{D})$. □

Next, we show that MO preserves the structure of data and clock constraints.

Lemma A.2.5 (MO preserves Data and Clock Constraints). Let $\mathcal{P}$ be a set of ports, $\mathcal{D}$ a set of data variables, and $\mathcal{X}$ a set of clocks. Let $dc{\in}DC(\mathcal{P},\mathcal{D})$ be a data constraint (cf. Definition 2.1.7), $cc{\in}CC(\mathcal{X})$ a clock constraint (cf. Definition 2.1.2). Let $\mathtt{dc}{\in}DC(\mathtt{P_{DA}},\mathtt{D_{CO}})$ be the representation of dc, and $\mathtt{cc}{\in}CC(\mathtt{X})$ the representations of cc (cf. Section 3.1.1). The abstraction function α *preserves the structure of data and clock constraints*, that means $\alpha(\mathtt{dc})$ and $\alpha(\mathtt{cc})$ are also valid representations of data and clock constraints.

Proof. By definition, α changes only literals. Since `dc` and `cc` do not contain propositional variables, neither of (4.1b), (4.1c) or (4.1d) is applicable. Therefore, α preserves the logical structure,[6] and literals are either kept unchanged (4.1a) or mapped to `true` (4.1e). Thus, $\alpha(\mathtt{dc}){\in}DC(\mathtt{P_{DA}},\mathtt{D_{CO}})$, and $\alpha(\mathtt{cc}){\in}CC(\mathtt{X})$. □

[6]The logical structure of a formula is the order of its literals and the logical operators $\wedge$, $\vee$ and $\neg$. For example, for a formula $F = (p \vee \neg q) \wedge \neg(r \wedge \neg(x = 5))$, with $p, q, r \in \mathcal{P}$ being atomic propositions and $x \in \mathcal{V}$ being a variable, the logical structure is $F = (l_1 \vee l_2) \wedge \neg(l_3 \wedge l_4)$ (for literals l_i). Note that an occurrence of $\neg$ is part of the logical structure only if it is not part of a literal.

Remark A.2.6 (Lifting of MO). For argumentation purposes, we lift α in the straightforward way to reason about constituents of systems rather than formulas. For example, for a clock x with representation $\mathtt{x}$, we may write $x{\in}\mathcal{O}$ instead of $\mathtt{x}{\in}\mathcal{O}$.

Similarly, we lift α to reason about sets rather than single variables. For example, for the set of locations S and the set of clocks $\mathcal{X}$, we may write $\alpha(S)$ and $\alpha(\mathcal{X})$ to denote the set of locations respectively clocks in the abstract system, that means $\alpha(S){=}\{s' \mid s{\in}S, \gamma(s){=}s'\}$, and $\alpha(\mathcal{X}){=}\{x \mid x{\notin}\mathcal{O}\}{=}\mathcal{X}\backslash\mathcal{O}$. By $\alpha(\lambda)$, we denote the update map λ, reduced to the clocks of the abstract system. That is, $\alpha(\lambda){=}\lambda|_{\alpha(\mathcal{X})}$.

We are now ready to show that MO preserves the formula representation of TA, and preserves the formula representation of TCA up to data constraints.

Proposition A.2.7 (MO preserves the Formula Representation). Let $\mathfrak{A} = (S, s_0, \Sigma, \mathcal{X}, I, E)$ be a TA, $\mathfrak{T}{=}(S, s_0, \mathcal{P}, \mathcal{X}, I, \mathcal{D}, \#, E)$ a TCA, with formula representations $\varphi(\mathfrak{A})$, $\varphi(\mathfrak{T})$, and k-unfoldings $\varphi(\mathfrak{A})_k$, $\varphi(\mathfrak{T})_k$ in NNF (cf. Remark A.2.4). Let α be an abstraction function, with γ and $\mathcal{O}$ as in Definition 4.1.5.

The abstraction by merging omission *preserves the formula representation and k-unfolding of* $\mathfrak{A}$, and it *preserves the formula representation and k-unfolding of* $\mathfrak{T}$ *up to data constraints*. That means, there exists a TA $\widetilde{\mathfrak{A}}$, with formula representation $\varphi(\widetilde{\mathfrak{A}})$ and k-unfolding $\varphi(\widetilde{\mathfrak{A}})_k$, such that

$$\begin{aligned} \varphi(\widetilde{\mathfrak{A}}) &= \alpha(\varphi(\mathfrak{S})), \text{ and} \\ \varphi(\widetilde{\mathfrak{A}})_k &= \alpha(\varphi(A)_k), \end{aligned} \tag{xx}$$

and there exists a TCA $\mathfrak{T}$, with formula representation $\varphi(\widetilde{\mathfrak{T}})$ and k-unfolding $\varphi(\widetilde{\mathfrak{T}})_k$, such that

$$\begin{aligned} \varphi(\widetilde{\mathfrak{T}})\backslash_{dc} &= \alpha(\varphi(\mathfrak{T}))\backslash_{dc}, \text{ and} \\ \varphi(\widetilde{\mathfrak{T}})_k\backslash_{dc} &= \alpha(\varphi(T)_k)\backslash_{dc}, \end{aligned} \tag{xxi}$$

where $\backslash_{dc}$ is a function that replaces all literals of the form $(\mathtt{D}{\simeq}\mathtt{D}')$ or $\neg(\mathtt{D}{\simeq}\mathtt{D}')$, with $\simeq\, {\in}\{=, \leqslant\}$, and $\mathtt{D}$, $\mathtt{D}'$ either port data variables $\mathtt{Dp}_\mathtt{t}$, data content variables $\mathtt{Dd}_\mathtt{t}$, or data element representations $\mathtt{n}^\mathtt{i}$ (cf. Definition 2.1.7 and Section 3.1.1.5), and $t{\in}\mathbb{N}$, in a formula by $\mathtt{true}$.

Proof of (xx). (for the proof of (xxi), please refer to Page 143).
Let $\mathfrak{A}'{=}(S', s_0', \Sigma', \mathcal{X}', I', E')$ be a TA, with $S'{=}\alpha(S)$, $s_0'{=}\alpha(s_0)$, $\Sigma'{=}\alpha(\Sigma)$, $\mathcal{X}'{=}\alpha(\mathcal{X})$, $I'(s){=}\alpha(I(s))$ for all $s{\in}S'$, and $E'{=}\{(\alpha(s), \alpha(\mathfrak{a}), \alpha(cc), \alpha(\lambda), \alpha(s')) \mid (s, \mathfrak{a}, cc, \lambda, s') \in E\}$. Let $\varphi(\mathfrak{A}')$ and $\varphi(\mathfrak{A}')_k$ be the formula representation and k-unfolding of $\mathfrak{A}'$. Observe that we have

$$S'{=}\alpha(S){=}\{s|s{\in}S, \alpha(s){=}id\}{\cup}\{s'|s{\in}S, \alpha(s){=}s'\}, \text{and} \tag{*}$$
$$\Sigma'{=}\alpha(\Sigma){=}\{\mathfrak{a}|\mathfrak{a}{\in}\Sigma, \alpha(\mathfrak{a}){=}id\}{\cup}\{\mathfrak{a}'|\mathfrak{a}{\in}\Sigma, \alpha(\mathfrak{a}){=}\mathfrak{a}'\} \tag{**}$$

We first show that $\varphi(\mathfrak{A}'){=}\alpha(\varphi(\mathfrak{A}))$. By Definitions 3.1.1 and 4.1.5, we have

$$\begin{aligned} \alpha(\varphi(\mathfrak{A})) &= \alpha(\varphi^{init}(\mathfrak{A}) \wedge \varphi^{trans}(\mathfrak{A}) \wedge \varphi^{location}(\mathfrak{A}) \wedge \varphi^{mutex}(\mathfrak{A})) \\ &= \alpha(\varphi^{init}(\mathfrak{A})) \wedge \alpha(\varphi^{trans}(\mathfrak{A})) \wedge \alpha(\varphi^{location}(\mathfrak{A})) \wedge \alpha(\varphi^{mutex}(\mathfrak{A})) \\ \varphi(\mathfrak{A}') &= \varphi^{init}(\mathfrak{A}') \wedge \varphi^{trans}(\mathfrak{A}') \wedge \varphi^{location}(\mathfrak{A}') \wedge \varphi^{mutex}(\mathfrak{A}') \end{aligned}$$

Consider the corresponding parts in $\alpha(\varphi(\mathfrak{A}))$ and $\varphi(\mathfrak{A}')$ separately

1. Initial constraints φ^{init}:

$$\begin{aligned}\alpha(\varphi^{init}(\mathfrak{A})) &= \alpha\big(\bar{\mathtt{s}}_0 \wedge \bigwedge_{s\in S, s\neq\bar{s}} \neg \mathtt{s}_0 \wedge \mathtt{I}(\bar{\mathtt{s}})_0 \wedge \bigwedge_{\mathfrak{a}\in\Sigma} (\neg\alpha_0) \wedge (\mathtt{z}_0{=}0) \wedge \bigwedge_{x\in\mathcal{X}} (\mathtt{x}_0{=}0)\big)\\ &= \alpha(\bar{\mathtt{s}}_0) \wedge \bigwedge_{s\in S, s\neq\bar{s}} \alpha(\neg \mathtt{s}_0) \wedge \alpha(\mathtt{I}(\bar{\mathtt{s}})_0) \wedge \bigwedge_{\mathfrak{a}\in\Sigma} \alpha((\neg\alpha_0)) \wedge\\ &\quad \alpha((\mathtt{z}_0{=}0)) \wedge \bigwedge_{x\in\mathcal{X}} \alpha((\mathtt{x}_0{=}0))\\ &= \alpha(\bar{\mathtt{s}}_0) \wedge \bigwedge_{\substack{s\in S, s\neq\bar{s},\\ \alpha(s)=id}} \neg \mathtt{s}_0 \wedge \bigwedge_{\substack{s\in S, s\neq\bar{s},\\ \alpha(s)=s'\neq\alpha(\bar{s})}} \neg \mathtt{s}'_0 \wedge \alpha(\mathtt{I}(\bar{\mathtt{s}})_0) \wedge \bigwedge_{\substack{\mathfrak{a}\in\Sigma,\\ \alpha(\mathfrak{a})=id}} (\neg\mathfrak{a}_0) \wedge\\ &\quad \bigwedge_{\substack{\mathfrak{a}\in\Sigma,\\ \alpha(\mathfrak{a})=\mathfrak{a}'}} (\neg\alpha'_0) \wedge (\mathtt{z}_0{=}0) \wedge \bigwedge_{x\in\mathcal{X}\setminus\mathcal{O}} (\mathtt{x}_0{=}0)\end{aligned}$$

$$\varphi^{init}(\mathfrak{A}') = \bar{\mathtt{s}}'_0 \wedge \bigwedge_{s\in S', s\neq\bar{s}'} \neg \mathtt{s}_0 \wedge \mathtt{I}(\bar{\mathtt{s}}')_0 \wedge \bigwedge_{\mathfrak{a}\in\Sigma'} (\neg\alpha_0) \wedge (\mathtt{z}_0{=}0) \wedge \bigwedge_{x\in\mathcal{X}'} (\mathtt{x}_0{=}0)$$

By definition of $\mathfrak{A}'$, we have $\alpha(\bar{\mathtt{s}}_0){=}\bar{\mathtt{s}}'_0$, and $\alpha(\mathtt{I}(\bar{\mathtt{s}})_0){=}\mathtt{I}(\bar{\mathtt{s}}')_0$. Because of (*), (**), and the fact that $\mathcal{X}'{=}\mathcal{X}\setminus\mathcal{O}$, we finally get

$$\alpha(\varphi^{init}(\mathfrak{A})) = \varphi^{init}(\mathfrak{A}')$$

2. Transition relation φ^{trans}:

$$\begin{aligned}\alpha(\varphi^{trans}(\mathfrak{A})) &= \alpha(\bigvee_{e\in E} \varphi^{action}(e) \vee \bigvee_{s\in S} \varphi^{delay}(s))\\ &= \bigvee_{e\in E} \alpha(\varphi^{action}(e)) \vee \bigvee_{s\in S} \alpha(\varphi^{delay}(s))\end{aligned}$$

$$\varphi^{trans}(\mathfrak{A}') = \bigvee_{e\in E'} \alpha(\varphi^{action}(e)) \vee \bigvee_{s\in S'} \alpha(\varphi^{delay}(s))$$

Consider an action transition $e{=}(s, \mathfrak{a}, cc, \lambda, s'){\in}E$:

$$\begin{aligned}\alpha(\varphi^{action}(e)) &= \alpha(\mathtt{s}_\mathtt{t} \wedge \alpha_{\mathtt{t}+1} \wedge \mathtt{cc}_\mathtt{t} \wedge (\mathtt{z}_\mathtt{t}{=}\mathtt{z}_{\mathtt{t}+1}) \wedge \bigwedge_{\lambda(x)=id} (\mathtt{x}_{\mathtt{t}+1}{=}\mathtt{x}_\mathtt{t}) \wedge\\ &\quad \bigwedge_{\lambda(x)=x'} (\mathtt{x}_{\mathtt{t}+1}{=}\mathtt{x'}_{\mathtt{t}+1}) \wedge \bigwedge_{\lambda(x)=n} (\mathtt{x}_{\mathtt{t}+1}{=}\mathtt{z}_{\mathtt{t}+1}{-}n) \wedge \mathtt{s'}_{\mathtt{t}+1} \wedge \mathtt{I}(\mathtt{s}')_{\mathtt{t}+1})\\ &= \alpha(\mathtt{s}_\mathtt{t}) \wedge \alpha(\alpha_\mathtt{t}) \wedge \alpha(\mathtt{cc}_\mathtt{t}) \wedge \alpha(\mathtt{z}_\mathtt{t}{=}\mathtt{z}_{\mathtt{t}+1}) \wedge \bigwedge_{\lambda(x)=id} \alpha(\mathtt{x}_{\mathtt{t}+1}{=}\mathtt{x}_\mathtt{t}) \wedge\\ &\quad \bigwedge_{\lambda(x)=x'} \alpha(\mathtt{x}_{\mathtt{t}+1}{=}\mathtt{x'}_{\mathtt{t}+1}) \wedge \bigwedge_{\lambda(x)=n} \alpha(\mathtt{x}_{\mathtt{t}+1}{=}\mathtt{z}_{\mathtt{t}+1}{-}n) \wedge \alpha(\mathtt{s'}_{\mathtt{t}+1}) \wedge \mathtt{I}(\mathtt{s}')_{\mathtt{t}+1}\\ &= \alpha(\mathtt{s}_\mathtt{t}) \wedge \alpha(\alpha_{\mathtt{t}+1}) \wedge \alpha(\mathtt{cc}_\mathtt{t}) \wedge (\mathtt{z}_\mathtt{t}{=}\mathtt{z}_{\mathtt{t}+1}) \wedge \bigwedge_{\substack{\lambda(x)=id,\\ x\in\mathcal{X}\setminus\mathcal{O}}} (\mathtt{x}_{\mathtt{t}+1}{=}\mathtt{x}_\mathtt{t}) \wedge\\ &\quad \bigwedge_{\substack{\lambda(x)=x',\\ x,x'\in\mathcal{X}\setminus\mathcal{O}}} (\mathtt{x}_{\mathtt{t}+1}{=}\mathtt{x'}_{\mathtt{t}+1}) \wedge \bigwedge_{\substack{\lambda(x)=n,\\ x\in\mathcal{X}\setminus\mathcal{O}}} (\mathtt{x}_{\mathtt{t}+1}{=}\mathtt{z}_{\mathtt{t}+1}{-}n) \wedge \alpha(\mathtt{s'}_{\mathtt{t}+1}) \wedge \mathtt{I}(\mathtt{s}')_{\mathtt{t}+1}\end{aligned}$$

and its counterpart $e'{=}(\alpha(s), \alpha(\mathfrak{a}), \alpha(cc), \alpha(\lambda), \alpha(s')){\in}E'$:

$$\varphi^{action}(e') = \alpha(\mathtt{s}_t) \wedge \alpha(\alpha_{t+1}) \wedge \alpha(\mathtt{cc}_t) \wedge (\mathtt{z}_t{=}\mathtt{z}_{t+1}) \wedge \bigwedge_{\alpha(\lambda)(x)=id} (\mathtt{x}_{t+1}{=}\mathtt{x}_t) \wedge$$
$$\bigwedge_{\alpha(\lambda)(x)=x'} (\mathtt{x}_{t+1}{=}\mathtt{x'}_{t+1}) \wedge \bigwedge_{\alpha(\lambda)(x)=n} (\mathtt{x}_{t+1}{=}\mathtt{z}_{t+1}{-}n) \wedge \alpha(\mathtt{s'}_{t+1}) \wedge \alpha(\mathtt{I(s')}_t)$$

Because $\mathcal{X}'{=}\mathcal{X}\backslash\mathcal{O}$, we have

$$\alpha(\varphi^{action}(e)) = \varphi^{action}(e')$$

For a delay transition in s, we have

$$\begin{aligned}\alpha(\varphi^{delay}(s)) &= \alpha(\mathtt{s}_t \wedge \bigwedge_{\mathfrak{a}\in\Sigma} \neg\alpha_{t+1} \wedge (\mathtt{z}_t{\leq}\mathtt{z}_{t+1}) \wedge \bigwedge_{x\in\mathcal{X}} (\mathtt{x}_t{=}\mathtt{x}_{t+1}) \wedge \mathtt{s}_{t+1} \wedge \mathtt{I(s)}_{t+1}) \\ &= \alpha(\mathtt{s}_t) \wedge \bigwedge_{\mathfrak{a}\in\Sigma} \alpha(\neg\alpha_{t+1}) \wedge \alpha(\mathtt{z}_t{\leq}\mathtt{z}_{t+1}) \wedge \bigwedge_{x\in\mathcal{X}} \alpha(\mathtt{x}_t{=}\mathtt{x}_{t+1}) \wedge \alpha(\mathtt{s}_{t+1}) \wedge \alpha(\mathtt{I(s)}_{t+1}) \\ &= \alpha(\mathtt{s}_t) \wedge \bigwedge_{\substack{\mathfrak{a}\in\Sigma,\\ \alpha(\mathfrak{a})=id}} \neg\alpha_{t+1} \wedge \bigwedge_{\substack{\mathfrak{a}\in\Sigma,\\ \alpha(\mathfrak{a})=\mathfrak{a}'}} (\neg\alpha'_{t+1}) \wedge (\mathtt{z}_t{\leq}\mathtt{z}_{t+1}) \wedge \\ &\quad \bigwedge_{\substack{x\in\mathcal{X},\\ x\in\mathcal{X}\backslash\mathcal{O}}} (\mathtt{x}_t{=}\mathtt{x}_{t+1}) \wedge \alpha(\mathtt{s}_{t+1}) \wedge \alpha(\mathtt{I(s)}_{t+1})\end{aligned}$$

and for the corresponding delay transition in s', we have

$$\varphi^{delay}(s') = \alpha(\mathtt{s}_t) \wedge \bigwedge_{\mathfrak{a}\in\Sigma'} \neg\alpha_{t+1} \wedge (\mathtt{z}_t{\leq}\mathtt{z}_{t+1}) \wedge \bigwedge_{x\in\mathcal{X}'} (\mathtt{x}_t{=}\mathtt{x}_{t+1}) \wedge \alpha(\mathtt{s}_{t+1}) \wedge \alpha(\mathtt{I(s)}_{t+1})$$

Because of (**), we have

$$\alpha(\varphi^{delay}(s)) = \varphi^{delay}(s')$$

Since there is a one-to-one relation between transitions in E and E', and by definition of $\vee$, we finally have

$$\alpha(\varphi^{trans}(\mathfrak{A})) = \varphi^{trans}(\mathfrak{A}')$$

3. Mutual exclusion of locations $\varphi^{location}$:

$$\begin{aligned}\alpha(\varphi^{location}(\mathfrak{A})) &= \alpha(\bigvee_{s\in S} (\mathtt{s}_{t+1} \wedge \bigwedge_{s'\in S, s'\neq s} \neg\mathtt{s}'_{t+1})) \\ &= \bigvee_{s\in S} (\alpha(\mathtt{s}_{t+1}) \wedge \bigwedge_{s'\in S, s'\neq s} \alpha(\neg\mathtt{s}'_{t+1})) \\ &= \bigvee_{\substack{s\in S\\ \alpha(s)=id}} (\mathtt{s}_t \wedge \bigwedge_{\substack{s'\in S, s'\neq s\\ \alpha(s')=id}} \neg\mathtt{s'}_t \wedge \bigwedge_{\substack{s'\in S, s'\neq s\\ \alpha(s')=\bar{s}\neq s}} \neg\bar{\mathtt{s}}_t) \vee \\ &\quad \bigvee_{\substack{s\in S\\ \alpha(s)=\hat{s}}} (\hat{\mathtt{s}}_t \wedge \bigwedge_{\substack{s'\in S, s'\neq s\\ \alpha(s')=id}} \neg\mathtt{s'}_t \wedge \bigwedge_{\substack{s'\in S, s'\neq s\\ \alpha(s')=\bar{s}\neq s}} \neg\bar{\mathtt{s}}_t)\end{aligned}$$

$$\varphi^{location}(\mathfrak{A}')) = \bigvee_{s\in S'} (\mathtt{s}_{t+1} \wedge \bigwedge_{s'\in S', s'\neq s} \neg\mathtt{s}'_{t+1})$$

Because of (*), we have

$$\alpha(\varphi^{location}(\mathfrak{A})) = \varphi^{location}(\mathfrak{A}'))$$

4. Mutual exclusion of events φ^{mutex}:

$$\begin{aligned}
\alpha(\varphi^{mutex}(\mathfrak{A})) &= \alpha(\bigvee_{\mathfrak{a}\in\Sigma}(\boldsymbol{\alpha}_{t+1}\wedge\bigwedge_{\mathfrak{a}'\in\Sigma,\mathfrak{a}'\neq\mathfrak{a}}\neg\boldsymbol{\alpha}'_{t+1})\vee\bigwedge_{\mathfrak{a}\in\Sigma}(\neg\boldsymbol{\alpha}_{t+1}))\\
&= \bigvee_{\mathfrak{a}\in\Sigma}(\alpha(\boldsymbol{\alpha}_{t+1})\wedge\bigwedge_{\mathfrak{a}'\in\Sigma,\mathfrak{a}'\neq\mathfrak{a}}\alpha(\neg\boldsymbol{\alpha}'_{t+1}))\vee\bigwedge_{\mathfrak{a}\in\Sigma}\alpha(\neg\boldsymbol{\alpha}_{t+1})\\
&= \bigvee_{\substack{\mathfrak{a}\in\Sigma\\ \alpha(\mathfrak{a})=id}}(\boldsymbol{\alpha}_t\wedge\bigwedge_{\substack{\mathfrak{a}'\in\Sigma,\mathfrak{a}'\neq\mathfrak{a}\\ \alpha(\mathfrak{a}')=id}}\neg\boldsymbol{\alpha}'_t\wedge\bigwedge_{\substack{\mathfrak{a}'\in\Sigma,\mathfrak{a}'\neq\mathfrak{a}\\ \alpha(\mathfrak{a}')=\bar{\mathfrak{a}}\neq\mathfrak{a}}}\neg\bar{\boldsymbol{\alpha}}_t)\vee\\
&\quad\bigvee_{\substack{\mathfrak{a}\in\Sigma\\ \alpha(\mathfrak{a})=\hat{\mathfrak{a}}}}(\hat{\boldsymbol{\alpha}}_t\wedge\bigwedge_{\substack{\mathfrak{a}'\in\Sigma,\mathfrak{a}'\neq\mathfrak{a}\\ \alpha(\mathfrak{a}')=id}}\neg\boldsymbol{\alpha}'_t\wedge\bigwedge_{\substack{\mathfrak{a}'\in\Sigma,\mathfrak{a}'\neq\mathfrak{a}\\ \alpha(\mathfrak{a}')=\bar{\mathfrak{a}}\neq\mathfrak{a}}}\neg\bar{\boldsymbol{\alpha}}_t)\vee\\
&\quad\bigwedge_{\substack{\mathfrak{a}\in\Sigma,\\ \alpha(\mathfrak{a})=id}}(\neg\mathfrak{a})\wedge\bigwedge_{\substack{\mathfrak{a}\in\Sigma,\\ \alpha(\mathfrak{a})=\bar{\mathfrak{a}}}}(\neg\bar{\boldsymbol{\alpha}}_{t+1})\\
\varphi^{mutex}(\mathfrak{A}')) &= \bigvee_{\mathfrak{a}\in\Sigma'}(\boldsymbol{\alpha}_{t+1}\wedge\bigwedge_{\mathfrak{a}'\in\Sigma',\mathfrak{a}'\neq\mathfrak{a}}\neg\boldsymbol{\alpha}'_{t+1})\vee\bigwedge_{\mathfrak{a}\in\Sigma'}(\neg\boldsymbol{\alpha}_{t+1})
\end{aligned}$$

Because of (**), we have

$$\alpha(\varphi^{mutex}(\mathfrak{A})) = \varphi^{mutex}(\mathfrak{A}')),$$

From the four cases above, we get

$$\alpha(\varphi(\mathfrak{A})) = \varphi(\mathfrak{A}')$$

The argumentation for

$$\alpha(\varphi(\mathfrak{A})_k)=\varphi(\mathfrak{A}')_k$$

is similar.

Thus, the TA $\mathfrak{A}'$ satisfies the conditions (xx), and we have shown that MO preserves the formula representation and k-unfolding of TA. □

Proof of (xxi). Let $\mathfrak{T}' = (S', s'_0, \mathcal{P}, \mathcal{X}', I', \mathcal{D}', \#', E')$ be a TCA, with $S'=\alpha(S)$, $s'_0=\alpha(s_0), \mathcal{P}'=\alpha(\mathcal{P}), \mathcal{X}'=\alpha(\mathcal{X}), I'(s)=\alpha(I(s))$ for all $s\in S'$, $\mathcal{D}'=\alpha(\mathcal{D})$, $\#'(s)=\alpha(\#(s))$ for all $s\in S'$, and $E'=\{(\alpha(s),\alpha(P),\alpha(dc),\alpha(cc),\alpha(\lambda),\alpha(s'))|(s,P,dc,cc,\lambda,s')\in E\}$. Let $\varphi(\mathfrak{T}')$ and $\varphi(\mathfrak{T}')_k$ be the formula representation and k-unfolding of $\mathfrak{T}'$. Observe that we have

$$S'=\alpha(S)=\{s|s\in S,\alpha(s)=id\}\cup\{s'|s\in S,\alpha(s)=s'\}, \tag{†}$$
$$\Sigma'=\alpha(\Sigma)=\{\mathfrak{a}|\mathfrak{a}\in\Sigma,\alpha(\mathfrak{a})=id\}\cup\{\mathfrak{a}'|\mathfrak{a}\in\Sigma,\alpha(\mathfrak{a})=\mathfrak{a}'\},\text{ and} \tag{‡}$$
$$\mathcal{D}'=\alpha(\mathcal{D})=\{d|d\in\mathcal{D},\alpha(d)=id\}\cup\{d'|d\in\mathcal{D},\alpha(d)=d'\} \tag{††}$$

We first show that $\varphi(\mathfrak{T}')\backslash_{dc}{=}\alpha(\varphi(\mathfrak{T}))\backslash_{dc}$. By definition of $\backslash_{dc}$, and Definitions 3.1.4 and 4.1.5, we have

$$\begin{aligned}\alpha(\varphi(\mathfrak{T}))\backslash_{dc} &= \alpha(\varphi^{init}(\mathfrak{T}) \wedge \varphi^{trans}(\mathfrak{T}) \wedge \varphi^{location}(\mathfrak{T}) \wedge \varphi^{mutex}(\mathfrak{T}))\backslash_{dc}\\ &= \alpha(\varphi^{init}(\mathfrak{T}))\backslash_{dc} \wedge \alpha(\varphi^{trans}(\mathfrak{T}))\backslash_{dc} \wedge\\ &\quad \alpha(\varphi^{location}(\mathfrak{T}))\backslash_{dc} \wedge \alpha(\varphi^{mutex}(\mathfrak{T}))\backslash_{dc}\\ \varphi(\mathfrak{T}')\backslash_{dc} &= \varphi^{init}(\mathfrak{T}')\backslash_{dc} \wedge \varphi^{trans}(\mathfrak{T}')\backslash_{dc} \wedge \varphi^{location}(\mathfrak{T}')\backslash_{dc} \wedge \varphi^{mutex}(\mathfrak{T}')\backslash_{dc}\end{aligned}$$

Consider the corresponding parts in $\alpha(\varphi(\mathfrak{T}))\backslash_{dc}$ and $\varphi(\mathfrak{T}')\backslash_{dc}$ separately

1. Initial constraints φ^{init}:

$$\begin{aligned}\alpha(\varphi^{init}(\mathfrak{T}))\backslash_{dc} &= \alpha\big(\bar{\mathtt{s}}_0 \wedge \bigwedge_{s\in S, s\neq\bar{s}} \neg \mathtt{s}_0 \wedge \mathtt{I}(\bar{\mathtt{s}})_0 \wedge \bigwedge_{p\in\mathcal{P}} (\neg \mathtt{p}_0 \wedge (\mathtt{Dp}_0{=}\mathtt{n}^{\perp})) \wedge\\ &\qquad \bigwedge_{d\in\mathcal{D}} (\neg \mathtt{d}_0 \wedge (\mathtt{Dd}_0{=}\mathtt{n}^{\perp})) \wedge (\mathtt{z}_0{=}0) \wedge \bigwedge_{x\in\mathcal{X}} (\mathtt{x}_0{=}0)\big)\backslash_{dc}\\ &= \alpha(\bar{\mathtt{s}}_0)\backslash_{dc} \wedge \bigwedge_{s\in S, s\neq\bar{s}} \alpha(\neg \mathtt{s}_0)\backslash_{dc} \wedge \alpha(\mathtt{I}(\bar{\mathtt{s}})_0)\backslash_{dc} \wedge \bigwedge_{p\in\mathcal{P}} \alpha(\neg \mathtt{p}_0)\backslash_{dc} \wedge\\ &\qquad \bigwedge_{p\in\mathcal{P}} \alpha((\mathtt{Dp}_0{=}\mathtt{n}^{\perp}))\backslash_{dc} \wedge \bigwedge_{d\in\mathcal{D}} \alpha(\neg \mathtt{d}_0)\backslash_{dc} \wedge \bigwedge_{d\in\mathcal{D}} \alpha((\mathtt{Dd}_0{=}\mathtt{n}^{\perp}))\backslash_{dc} \wedge\\ &\qquad \alpha((\mathtt{z}_0{=}0))\backslash_{dc} \wedge \bigwedge_{x\in\mathcal{X}} \alpha((\mathtt{x}_0{=}0))\backslash_{dc}\\ &= \alpha(\bar{\mathtt{s}}_0) \wedge \bigwedge_{\substack{s\in S, s\neq\bar{s},\\ \alpha(s)=id}} \neg \mathtt{s}_0 \wedge \bigwedge_{\substack{s\in S, s\neq\bar{s},\\ \alpha(s)=s'\neq\alpha(\bar{s})}} \neg \mathtt{s}'_0 \wedge \alpha(\mathtt{I}(\bar{\mathtt{s}})_0) \wedge \bigwedge_{\substack{p\in\mathcal{P},\\ \alpha(p)=id}} (\neg \mathtt{p}_0) \wedge\\ &\qquad \bigwedge_{\substack{p\in\mathcal{P},\\ \alpha(p)=p'}} (\neg \mathtt{p}'_0) \wedge \bigwedge_{\substack{d\in\mathcal{D},\\ \alpha(d)=id}} (\neg \mathtt{d}_0) \wedge \bigwedge_{\substack{d\in\mathcal{D},\\ \alpha(d)=d'}} (\neg \mathtt{d}'_0) \wedge\\ &\qquad (\mathtt{z}_0{=}0) \wedge \bigwedge_{x\in\mathcal{X}\backslash\mathcal{O}} (\mathtt{x}_0{=}0)\end{aligned}$$

$$\begin{aligned}\varphi^{init}(\mathfrak{T}')\backslash_{dc} &= \bar{\mathtt{s}}'_0\backslash_{dc} \wedge \bigwedge_{s\in S', s\neq\bar{s}'} \neg \mathtt{s}_0\backslash_{dc} \wedge \mathtt{I}(\bar{\mathtt{s}}')_0\backslash_{dc} \wedge \bigwedge_{p\in\mathcal{P}'} (\neg \mathtt{p}_0 \wedge (\mathtt{Dp}_0{=}\mathtt{n}^{\perp}))\backslash_{dc} \wedge\\ &\qquad \bigwedge_{d\in\mathcal{D}'} (\neg \mathtt{d}_0 \wedge (\mathtt{Dd}_0{=}\mathtt{n}^{\perp}))\backslash_{dc} \wedge (\mathtt{z}_0{=}0)\backslash_{dc} \wedge \bigwedge_{x\in\mathcal{X}'} (\mathtt{x}_0{=}0)\backslash_{dc}\\ &= \bar{\mathtt{s}}'_0 \wedge \bigwedge_{s\in S', s\neq\bar{s}'} \neg \mathtt{s}_0 \wedge \mathtt{I}(\bar{\mathtt{s}}')_0 \wedge \bigwedge_{p\in\mathcal{P}'} (\neg \mathtt{p}_0) \wedge\\ &\qquad \bigwedge_{d\in\mathcal{D}'} (\neg \mathtt{d}_0) \wedge (\mathtt{z}_0{=}0) \wedge \bigwedge_{x\in\mathcal{X}'} (\mathtt{x}_0{=}0)\end{aligned}$$

By definition of $\mathfrak{T}'$, we have $\alpha(\bar{\mathtt{s}}_0) = \bar{\mathtt{s}}'_0$, and $\alpha(\mathtt{I}(\bar{\mathtt{s}})_0) = \mathtt{I}(\bar{\mathtt{s}}')_0$. Because of (†), (‡) and (††), and the fact that $\mathcal{X}'{=}\mathcal{X}\backslash\mathcal{O}$, we finally get

$$\alpha(\varphi^{init}(\mathfrak{T}))\backslash_{dc} = \varphi^{init}(\mathfrak{T}')\backslash_{dc}$$

2. Transition relation φ^{trans}:

$$\begin{aligned}\alpha(\varphi^{trans}(\mathfrak{T}))\backslash_{dc} &= \alpha\Big(\bigvee_{e\in E, e\text{ visible}} \varphi^{visible}(e) \vee \bigvee_{e\in E, e\text{ invisible}} \varphi^{invisible}(e)\Big)\backslash_{dc}\\ &= \bigvee_{e\in E, e\text{ visible}} \alpha(\varphi^{visible}(e))\backslash_{dc} \vee \bigvee_{e\in E, e\text{ invisible}} \alpha(\varphi^{invisible}(e))\backslash_{dc}\end{aligned}$$

$$\varphi^{trans}(\mathfrak{T}')\backslash_{dc} = \bigvee_{e\in E', e \text{ visible}} \varphi^{visible}(e)\backslash_{dc} \vee \bigvee_{e\in E', e \text{ invisible}} \varphi^{invisible}(e)\backslash_{dc}$$

Consider a visible transition $e{=}(s, P, dc, cc, \lambda, s') \in E$:

$$\begin{aligned}
\alpha(\varphi^{visible}(e))\backslash_{dc} = {} & \alpha(\mathtt{s}_t \wedge \mathtt{I(s)}_{t\Delta} \wedge \bigwedge_{p\in P} \mathtt{p}_{t+1} \wedge \bigwedge_{p\notin P} \neg\mathtt{p}_{t+1} \wedge \bigwedge_{d\notin \#(s')} \neg\mathtt{d}_{t+1} \wedge \mathtt{dc}_{t+1} \wedge \\
& \mathtt{cc}_{t\Delta} \wedge (\mathtt{z}_t{<}\mathtt{z}_{t+1}) \wedge \bigwedge_{\lambda(x)=id} (\mathtt{x}_{t+1}{=}\mathtt{x}_t) \wedge \bigwedge_{\lambda(x)=x'} (\mathtt{x}_{t+1}{=}\mathtt{x'}_{t+1}) \wedge \\
& \bigwedge_{\lambda(x)=n} (\mathtt{x}_{t+1}{=}\mathtt{z}_{t+1}{-}n) \wedge \mathtt{s'}_{t+1} \wedge \mathtt{I(s')}_{t+1})\backslash_{dc} \\
= {} & \alpha(\mathtt{s}_t) \wedge \alpha(\mathtt{I(s)}_{t\Delta}) \wedge \bigwedge_{\substack{p\in P,\\ \alpha(p)=id}} \mathtt{p}_{t+1} \wedge \bigwedge_{\substack{p\in P,\\ \alpha(p)=p'}} \mathtt{p'}_{t+1} \wedge \bigwedge_{\substack{p\notin P,\\ \alpha(p)=id}} \neg\mathtt{p}_{t+1} \wedge \\
& \bigwedge_{\substack{p\notin P,\\ \alpha(p)=p'\notin\alpha(P)}} \neg\mathtt{p'}_{t+1} \wedge \bigwedge_{\substack{d\notin\#(s'),\\ \alpha(d)=id}} (\neg\mathtt{d}_{t+1}) \wedge \bigwedge_{\substack{d\notin\#(s'),\\ \alpha(d)=d'}} (\neg\mathtt{d'}_{t+1}) \wedge \\
& \alpha(\mathtt{cc}_{t\Delta}) \wedge (\mathtt{z}_t{<}\mathtt{z}_{t+1}) \wedge \bigwedge_{\substack{\lambda(x)=id,\\ x\in\mathcal{X}\setminus\mathcal{O}}} (\mathtt{x}_{t+1}{=}\mathtt{x}_t) \wedge \\
& \bigwedge_{\substack{\lambda(x)=x',\\ x,x'\in\mathcal{X}\setminus\mathcal{O}}} (\mathtt{x}_{t+1}{=}\mathtt{x'}_{t+1}) \wedge \bigwedge_{\substack{\lambda(x)=n,\\ x\in\mathcal{X}\setminus\mathcal{O}}} (\mathtt{x}_{t+1}{=}\mathtt{z}_{t+1}{-}n) \wedge \alpha(\mathtt{s'}_{t+1}) \wedge \mathtt{I(s')}_{t+1}
\end{aligned}$$

and its counterpart $e'{=}(\alpha(s), \alpha(P), \alpha(dc), \alpha(cc), \alpha(\lambda), \alpha(s'))\in E'$:

$$\begin{aligned}
\varphi^{visible}(e')\backslash_{dc} = {} & \alpha(\mathtt{s}_t)\backslash_{dc} \wedge \alpha(\mathtt{I(s)}_{t\Delta})\backslash_{dc} \wedge \bigwedge_{p\in\alpha(P)} \mathtt{p}_{t+1}\backslash_{dc} \wedge \bigwedge_{p\notin\alpha(P)} (\neg\mathtt{p}_{t+1})\backslash_{dc} \wedge \\
& \bigwedge_{d\notin\#(\alpha(s'))} (\neg\mathtt{d}_{t+1})\backslash_{dc} \wedge \alpha(\mathtt{dc}_{t+1})\backslash_{dc} \wedge \alpha(\mathtt{cc}_{t\Delta})\backslash_{dc} \wedge \alpha(\mathtt{z}_t{<}\mathtt{z}_{t+1})\backslash_{dc} \wedge \\
& \bigwedge_{\alpha(\lambda)(x)=id} (\mathtt{x}_{t+1}{=}\mathtt{x}_t)\backslash_{dc} \wedge \bigwedge_{\alpha(\lambda)(x)=x'} (\mathtt{x}_{t+1}{=}\mathtt{x'}_{t+1})\backslash_{dc} \wedge \\
& \bigwedge_{\alpha(\lambda)(x)=n} (\mathtt{x}_{t+1}{=}\mathtt{z}_{t+1}{-}n)\backslash_{dc} \wedge \alpha(\mathtt{s'}_{t+1})\backslash_{dc} \wedge \alpha(\mathtt{I(s')}_{t+1})\backslash_{dc} \\
= {} & \alpha(\mathtt{s}_t) \wedge \alpha(\mathtt{I(s)}_{t\Delta}) \wedge \bigwedge_{p\in\alpha(P)} \mathtt{p}_{t+1} \wedge \bigwedge_{p\notin\alpha(P)} (\neg\mathtt{p}_{t+1}) \wedge \\
& \bigwedge_{d\notin\#(\alpha(s'))} (\neg\mathtt{d}_{t+1}) \wedge \alpha(\mathtt{cc}_{t\Delta}) \wedge (\mathtt{z}_t{<}\mathtt{z}_{t+1}) \wedge \\
& \bigwedge_{\alpha(\lambda)(x)=id} (\mathtt{x}_{t+1}{=}\mathtt{x}_t) \wedge \bigwedge_{\alpha(\lambda)(x)=x'} (\mathtt{x}_{t+1}{=}\mathtt{x'}_{t+1}) \wedge \\
& \bigwedge_{\alpha(\lambda)(x)=n} (\mathtt{x}_{t+1}{=}\mathtt{z}_{t+1}{-}n) \wedge \alpha(\mathtt{s'}_{t+1}) \wedge \alpha(\mathtt{I(s')}_{t+1})
\end{aligned}$$

Because of (‡) and (††), and the fact that $\mathcal{X}'{=}\mathcal{X}\setminus\mathcal{O}$, we have

$$\alpha(\varphi^{visible}(e))\backslash_{dc} = \varphi^{visible}(e')\backslash_{dc}$$

Equivalently, we can show for an invisible transition $e{=}(s, \emptyset, \texttt{true}, cc, \lambda, s')$ and its counterpart $e'{=}(\alpha(s), \emptyset, \texttt{true}, \alpha(cc), \alpha(\lambda), \alpha(s'))$ that

$$\alpha(\varphi^{invisible}(e)\backslash_{dc} = \varphi^{invisible}(e')\backslash_{dc}$$

Since there is a one-to-one relation between transitions in E and E', and by definition of $\vee$, we finally have

$$\alpha(\varphi^{trans}(\mathfrak{T}))\backslash_{dc} = \varphi^{trans}(\mathfrak{T}')\backslash_{dc}$$

3. Mutual exclusion of locations $\varphi^{location}$: because $\backslash_{dc}$ does not change $\varphi^{mutex}(\mathfrak{T}')$ or $\alpha(\varphi^{mutex}(\mathfrak{T}))$, equivalently to the case for TA above, we have

$$\alpha(\varphi^{location}(\mathfrak{T}))\backslash_{dc} = \varphi^{location}(\mathfrak{T}')\backslash_{dc}$$

4. Data consistency constraints φ^{mutex}: trivially,

$$\alpha(\varphi^{mutex}(\mathfrak{T})))\backslash_{dc} = \texttt{true} = \varphi^{mutex}(\mathfrak{T}')\backslash_{dc}$$

From the four cases above, we get

$$\alpha(\varphi(\mathfrak{T}))\backslash_{dc} = \varphi(\mathfrak{T}')\backslash_{dc}$$

With a similar argumentation, we get

$$\alpha(\varphi(\mathfrak{T})_k)\backslash_{dc}{=}\varphi(\mathfrak{T}')_k\backslash_{dc}$$

Thus, the TCA $\mathfrak{T}'$ satisfies the conditions (xxi), and we have shown that MO preserves the formula representation and k-unfolding of TCA, up to data constraints. □

Proposition A.2.8 (Commuting Subdiagram). The subdiagram (*ii*) of Figure A.2 is *a partially commuting diagram.*[7]

Proof. This follows directly from Proposition A.2.7. □

We now have all the results to give the proof of Theorem 4.1.8.

Proof of Theorem 4.1.8. For the abstraction by omission to be correct, every finite run in the original system $\mathfrak{S}$ has to be reproducible in the abstract system $\widetilde{\mathfrak{S}}$. We show this by defining a homomorphism h_R between original and abstract sets of runs $Run_{\mathfrak{S},k}$ and $Run_{\widetilde{\mathfrak{S}},k}$, such that Figure A.2 commutes.

[7]Here, "partially commuting" means that every model $\sigma{\in}\mathcal{V}(\varphi(\mathfrak{T})_k)$ is also a model of $\mathcal{V}(\varphi(\widetilde{\mathfrak{T}})_k)$, but not necessarily vice versa.

Let $\mathfrak{S}$ be a real-time system, with formula representation $\varphi(\mathfrak{T})$ and k-unfolding $\varphi(\mathfrak{S})_k$, let $\mathcal{V}(\varphi(\mathfrak{S})_k)$ be the set of models of $\varphi(\mathfrak{S})_k$, let $Run_{\mathfrak{S},k}$ be the set of finite runs of length k of $\mathfrak{S}$.

Let α be an abstraction function, with γ and $\mathcal{O}$ as in Definition 4.1.5, let $\widetilde{\mathfrak{S}}$ be the abstract system that results from applying α to the formula representation $\varphi(\mathfrak{S})$ and the k-unfolding $\varphi(\mathfrak{S})_k$, that means $\varphi(\widetilde{\mathfrak{S}})=\alpha(\varphi(\mathfrak{S}))$ and $\varphi(\widetilde{\mathfrak{S}})_k=\alpha(\varphi(\mathfrak{S})_k)$, cf. Proposition A.2.7, let $\mathcal{V}(\varphi(\widetilde{\mathfrak{S}})_k)$ be the set of models of $\varphi(\widetilde{\mathfrak{S}})_k$, and $Run_{\widetilde{\mathfrak{S}},k}$ the set of finite runs of length k of $\widetilde{\mathfrak{S}}$. Let $\xi{:}\mathcal{V}(\varphi(\mathfrak{S})_k)\to\mathcal{V}(\varphi(\widetilde{\mathfrak{S}})_k)$ be a mapping assigning to each interpretation $\sigma\in\mathcal{V}(\varphi(\mathfrak{S})_k)$ the interpretation $\widetilde{\sigma}\in\mathcal{V}(\varphi(\widetilde{\mathfrak{S}})_k)$, which is obtained from restricting σ to the variables in $\varphi(\widetilde{\mathfrak{S}})_k$.[8]

We define a homomorphism h_R (cf. Definition A.2.2) as

$$h_R{:}Run_{\mathfrak{S},k}\to Run_{\widetilde{\mathfrak{S}},k}$$
$$h_R(r) = \downarrow_r^{\sigma}(\xi(\downarrow_{\sigma}^{r}(r)))$$

(cf. Lemmas A.1.4 and A.1.7). That means, a run $r_{\widetilde{\sigma}}\in Run_{\widetilde{\mathfrak{S}},k}$ is obtained from a run $r\in Run_{\mathfrak{S},k}$ by mapping r to the derived interpretation $\downarrow_{\sigma}^{r}(r)=\sigma_r\in\mathcal{V}(\varphi(\mathfrak{S})_k)$ (Definition A.1.6), reducing it to the interpretation $\xi(\downarrow_{\sigma}^{r}(r))=\widetilde{\sigma}\in\mathcal{V}(\varphi(\widetilde{\mathfrak{S}})_k)$ over the variables in $\varphi(\widetilde{\mathfrak{S}})_k$, and mapping it to the derived run $\downarrow_r^{\sigma}(\xi(\downarrow_{\sigma}^{r}(r)))=r_{\widetilde{\sigma}}\in Run_{\widetilde{\mathfrak{S}},k}$ (Definition A.1.2).

We define $\gamma_S{:}S\to\widetilde{S}$, with $\gamma_S(s)=\widetilde{s}$ iff $\gamma(s)=\widetilde{s}$,[9] $\gamma_\Sigma{:}\Sigma\to\widetilde{\Sigma}$, with $\gamma_\Sigma(\mathfrak{a})=\widetilde{\mathfrak{a}}$ iff $\gamma(\mathfrak{a})=\widetilde{\mathfrak{a}}$, $\gamma_P{:}p\to\widetilde{p}$, with $\gamma_P(p)=\widetilde{p}$ iff $\gamma(p)=\widetilde{p}$, and $\gamma_{\mathcal{D}}{:}d\to\widetilde{d}$, with $\gamma_{\mathcal{D}}(d)=\widetilde{d}$ iff $\gamma(d)=\widetilde{d}$. With this, h_R is a homomorphism as defined in Definition A.2.2. Together with the results of Proposition A.2.3, Proposition A.2.7, and Proposition A.2.8, Figure A.2 is a commuting diagram, that means every run of the original system $\mathfrak{S}$ is reproducible in the abstract system $\widetilde{\mathfrak{S}}$, and therefore the abstraction by omission is correct. □

[8] $\widetilde{\sigma}$ is well-defined, as by definition of α: $Vars(\alpha(\varphi(\mathfrak{S})_k))\subseteq Vars(\varphi(\mathfrak{T})_k)$, cf. Proposition A.2.7.

[9] Remember that we lifted α to constituents of TCA, cf. Remark A.2.6.