Supplementary Materials: Proofs

Proof of Proposition 4.1:

($\Rightarrow$) Let $\alpha' \in Im_t(A)$, i.e. $\exists \alpha \in A : (M, \alpha) \xrightarrow{t} (M', \alpha')$. Since $\phi[\alpha]$ is true and $guard(t)[\langle \alpha, \alpha' \rangle]$ is true, $(\phi[v/v^r] \wedge guard(t))[\langle \alpha, \alpha' \rangle]$ is also true. Let $Disjuncts$ be a set of formulas of $\Phi(V^r \cup V^w)$ s.t. $\bigvee Disjuncts \sim (\phi[v/v^r] \wedge guard(t))$. Then $\exists disjunct \in Disjunct$, s.t. $disjunct[\langle \alpha, \alpha' \rangle]$ is true. Let $write(disjunct) \in V^r$ be a set of read variables that are prescribed to be updated by $disjunct$. In this case, $(\exists write(disjunct) \in V^r : disjunct)[\langle \alpha', \alpha' \rangle]$ is true; and, consequently, $(\exists write(disjunct) \in V^r : disjunct)[v^r/v][v^w/v][\alpha']$ is also true. For each disjunct $disjunct_\oplus$ of DNF-formula $\phi[v/v^r] \wedge guard(t)$, procedure $\phi \oplus guard(t)$ computes $(\exists write(disjunct_\oplus) \in V^r : disjunct_\oplus)$. Result expressions for each $disjunct_\oplus$ are combined through disjunction. In the result expression, each symbol v^r and v^w is replaced with v. Thus, if $(\exists write(disjunct) \in V^r : disjunct)[v^r/v][v^w/v][\alpha']$ is true for some $disjunct$, then $(\phi \oplus guard(t))[\alpha']$ must be true. Thus, if $\alpha' \in Im_\alpha(t)$, then $\alpha' \in [[\phi']]$.

($\Leftarrow$) Let $\alpha' \in [[\phi']]$. Let $Disjunct_\oplus$ be a set of formulas s.t. $\bigvee Disjunct_\oplus \sim \phi'$. Then, $\exists disjunct_\oplus \in Disjunct_\oplus$, s.t. $disjunct_\oplus[\alpha']$. Let $\phi_a[v^r/v][v^w/v] \sim disjunct_\oplus$. If $disjunct_\oplus[\alpha']$ is true, then $\phi_a[\langle \alpha', \alpha' \rangle]$ is true. Let $(\exists write(\phi_b) \in V^r : \phi_b) \sim \phi_a$. Then, $\exists \beta$, s.t. $\phi_b[\langle \beta, \alpha' \rangle]$ is true. If $\phi_b[\langle \beta, \alpha' \rangle]$ is true for some β, then $(\phi[v/v^r] \wedge guard(t))[\langle \beta, \alpha' \rangle]$ is also true for β, which holds iff $\phi[\beta]$ and $guard(t)[\langle \beta, \alpha' \rangle]$ are both true. Since $[[\phi]] = A$, then $\beta \in A$. Thus, if $\alpha' \in [[\phi']]$, then $\alpha' \in Im_t(A)$.

Proof of Proposition 4.2:

($\Rightarrow$) Let $\alpha \in \Delta_t(A)$, i.e. $\neg(\exists \alpha' \in \mathcal{A} : (M, \alpha) \xrightarrow{t} (M', \alpha'))$. Since $M \xrightarrow{t} M'$, $\forall \alpha' \in \mathcal{A} : guard(t)[\langle \alpha, \alpha' \rangle])$ is false. Let $write(t) \subseteq V^w$ be a set of written variables that are updated by firing t. In this case, $(\exists write(t) \subseteq V^w : guard(t))[v^r/v][\alpha]$ is also false. Recall that $\phi[\alpha]$ is true. Then, $(\phi[v/v^r] \wedge \neg(\exists write(t) \subseteq V^w : guard(t))[v^r/v])[\alpha]$ is true. Note that $(\phi[v/v^r] \wedge \neg(\exists write(t) \subseteq V^w : guard(t))[v^r/v]) \sim (\phi \oplus \neg(\exists write(t) : guard(t)))$ according to implementation of procedure $\oplus$. Thus, $(\phi \oplus \neg(\exists write(t) : guard(t)))[\alpha]$ is true and, therefore, $\phi''[\alpha]$ is true. Consequently, if $\alpha \in \Delta_t(A)$, then $\alpha \in [[\phi'']]$.

($\Leftarrow$) Let $\alpha \in [[\phi'']]$. Recall that $(\phi \oplus \neg(\exists write(t) : guard(t))) \sim (\phi \wedge \neg(\exists write(t) \subseteq V^w : guard(t))[v^r/v])$. Then, $\phi[v/v^r] \wedge \neg(\exists write(t) \subseteq V^w : guard(t))[v^r/v][\alpha]$ is true. Thus, $(\exists write(t) \subseteq V^w : guard(t))[v^r/v][\alpha]$ must be false, which means that $(\exists write(t) \subseteq V^w : guard(t))[\langle \alpha, \alpha' \rangle]$ is false for any $\alpha' \in \mathcal{A}$, and, therefore, $guard(t)[\langle \alpha, \beta \rangle]$ is false for any $\beta \in \mathcal{A}$ due to existential quantifier $\exists write(t) \subseteq V^w$. Since $guard(t)[\langle \alpha, \beta \rangle]$ is false for any $\beta \in \mathcal{A}$, transition t cannot fire at (M, α) and, consequently, $\neg(\exists \alpha' \in \mathcal{A} : (M, \alpha) \xrightarrow{t} (M', \alpha'))$. Thus, if $\alpha \in [[\phi'']]$, then $\alpha \in \Delta_t(A)$.

Proof of Proposition 4.5:

Let $\mathcal{A}_{cl}$ be finite. Let $\mathcal{A}_{LTS}$ be a set of sets of variable states occurring in $LTS_\mathcal{N}$. Then, $\mathcal{A}_{LTS}$ is finite since $\mathcal{A}_{LTS} \subseteq \mathcal{A}_{cl}$. Relation $\leq^{|P|}$ defined on markings $\mathcal{M}$ is a wqo since $\leq$ on the set $\mathbb{Z}^+$ is a wqo. Let $\leq_\mathcal{M}$ be a relation on set S which holds for any $\langle M_i, A_i \rangle$ $\langle M_j, A_j \rangle \in S$ iff $M_i \leq^{|P|} M_j$. In this case, relation $\leq_\mathcal{M}$ is also a wqo. According to the properties of wqo, every infinite sequence $s_0, s_1, s_2, ...$ of elements from S contains an infinite increasing sequence $s_{i_0} \leq_\mathcal{M} s_{i_1} \leq_\mathcal{M} s_{i_2} ...$, where $i_0 < i_1 < i_2 < ....$ Quasi-ordering $\leq_\mathcal{S}$ narrows wqo $\leq_\mathcal{M}$ by an additional check for equality of sets of variable states. Since $\mathcal{A}_{LTS}$ is finite, for any infinite increasing sequence $s_{i_0} \leq_\mathcal{M} s_{i_1} \leq_\mathcal{M} s_{i_2} ...$ there can always be found two elements $s_{i_k}, s_{i_j} \in S$ with $i_k < i_j$ and $A_{i_k} = A_{i_j}$. Since $A_{i_k} = A_{i_j}$ and $M_{i_k} \leq^{|P|} M_{i_j}$, $s_{i_k} \leq_\mathcal{S} s_{i_j}$ holds. Thus, if $\mathcal{A}_{closure}$ is finite, then $\leq_\mathcal{S}$ is a wqo.

Proof of Proposition 4.7:

According to Proposition 4.4, $CT_{LTS_\mathcal{N}}$ can be effectively constructed if mapping $Succ$ is computable and wqo $\leq_\mathcal{S}$ is decidable. Consider computability of $Succ$. Based on Definition 3.2, set of transitions T and set of places P in $\mathcal{N}$ are finite. Then, set $Succ(s)$ for some state s is always finite. Let Φ be closed under $\{Im, \Delta\}$. Then, each set of variable states in $LTS_\mathcal{N}$ can be described by some formula from Φ. Given some state $\langle M, A\rangle$ in $LTS_{\mathcal{N}}$, a new state yielded by firing t at $\langle M, A\rangle$ can be computed effectively, since procedures $M(p) - F(p,t) + F(t,p)$ and $\phi \oplus guard(t)$ can be computed effectively. Thus, mapping $Succ$ is computable. Consider decidability of $\leq_{\mathcal{S}}$. $M_i \leq^{|P|} M_j$ is decidable due to finiteness of places in $\mathcal{N}$. $A_i = A_j$ is decidable if A_i, A_j can be represented as formulas of Φ. Let $[[\phi_i]] = A_i$ and $[[\phi_j]] = A_j$. Then, $A_i = A_j$ is identical to $\phi_i \sim \phi_j$ that can be checked effectively using formula (1). Hence, $\leq_{\mathcal{S}}$ is decidable. Thus, $CT_{LTS_{\mathcal{N}}}$ of $LTS_{\mathcal{N}}$ can be effectively constructed.

$$\phi_i \sim \phi_j \equiv \begin{cases} \neg\phi_i \wedge \phi_j \; is \; unsatisfiable \\ \phi_i \wedge \neg\phi_j \; is \; unsatisfiable \end{cases} . \quad (1)$$

Proof of Proposition 4.8:

($\Rightarrow$) Assume Algorithm 2 returns `true` for $\mathcal{N}$ and the closure of $A_I = \{\alpha_I\}$ under $\{Im, \Delta\}$ w.r.t. all $t \in T$ is finite. By construction, Algorithm 2 returns `true` iff coverability tree CT_{LTS_τ} for LTS_τ defined on $\mathcal{N}_\tau$ is finite and does not have any strictly covering nodes. If there is no strictly covering node in CT_{LTS_τ}, then $\mathcal{N}_\tau$ is bounded. Since $\mathcal{N}_\tau$ extends the behavior of $\mathcal{N}$, if $\mathcal{N}_\tau$ is bounded, then $\mathcal{N}$ is bounded. Thus, if Algorithm 2 returns `true`, $\mathcal{N}$ is bounded.

($\Leftarrow$) Assume $\mathcal{N}$ is bounded and the closure of $A_I = \{\alpha_I\}$ under $\{Im, \Delta\}$ w.r.t. all $t \in T$ is finite. Consider reachability graphs $RG_{\mathcal{N}}$, $RG_{\mathcal{N}_\tau}$ for $\mathcal{N}$ and $\mathcal{N}_\tau$. Markings that exist in $RG_{\mathcal{N}_\tau}$ must exist in $RG_{\mathcal{N}}$ since firing of a τ-transition does not update a DPN marking. Since $\mathcal{N}$ is bounded, a set of markings of $\mathcal{N}$ is finite. Thus, a set of markings of $\mathcal{N}_\tau$ is also finite. Since the closure of $A_I = \{\alpha_I\}$ under $\{Im, \Delta\}$ w.r.t. all $t \in T$ is finite and a set of markings of $\mathcal{N}_\tau$ is finite, CT_{LTS_τ} is finite and does not have strictly covering nodes; therefore, Algorithm 2 terminates and returns `true`. Thus, if $\mathcal{N}$ is bounded, Algorithm 2 returns `true`.

Proof of Proposition 4.10:

Let $D = \mathbb{R}$ and $\mathcal{P} = \{<, \leq, >, \geq, =, \neq\}$. Since Φ is closed under $\{Im, \Delta\}$, all sets of variable states generated based on functions $\{Im, \Delta\}$ w.r.t. to all transitions $t \in T$ can be represented using formulas of Φ. We prove that the closure of $A_I = \{\alpha_I\}$ under $\{Im, \Delta\}$ with respect to all $t \in T$ can be described with a finite set of formulas $L \in \Phi$ and, by that, we prove that the closure is finite. Note that the set of variables V is finite according to Definition 3.2. The set of predicates $\mathcal{P}$ is also finite. In what follows, we prove that there exists language L with a finite set of constants that can describe the closure of A_I under $\{Im, \Delta\}$.

Let ϕ_s be a formula describing some set of variable states and t be some transition. Then, $[[\phi_s \oplus guard(t)]] = Im_t(A)$ and $[[\phi_s \oplus \neg(\exists write(t) : guard(t))]] = \Delta_t(A)$. Let C_ϕ be a set of constants occurring in ϕ_s and C_t be a set of constants occurring in $guard(t)$. Note that both operations $\phi_s \oplus \neg(\exists write(t) : guard(t))$ and $\phi_s \oplus guard(t)$ are based on conjuncting two formulas of Φ, transforming the resultant expression to DNF, adding an existential quantifier for some variables to each disjunct and eliminating it. Conjunction of ϕ_s and $guard(t)$ as well as conjunction of ϕ_s and $\neg(\exists write(t) : guard(t))$, transformation of the resultant expression to DNF, and addition of an existential quantifier do not generate any new constants besides those in $(C_\phi \cup C_t)$. Let C_{res} be a set of constants in the result expression of the quantifier elimination. In what follows, we show that $C_{res} \subseteq (C_\phi \cup C_t)$ if $D = \mathbb{R}$.

Let $\phi_c = \phi_s \wedge guard(t)$ be a DNF-formula. Let $Disj$ be a set of disjuncts of ϕ_c. Let $disj \in Disj$. Let $V_{rem} \subseteq V$. Then, it is sufficient to prove that elimination of existence quantifier from $\exists V_{rem} : disj$ does not generate any new constants. Let $Atoms_{src}$ be a set of atomic formulas occurring in $disj$ and $Atoms_{res}$ be a set of atomic formulas occurring in a resultant formula of the quantifier elimination against $\exists V_{rem} : disj$. Let $Atoms_{sav} \subseteq Atoms_{src}$ be a set of atomic formulas that are not updated through the quantifier elimination. Let $Atoms_{impl}$ be a set of formulas that are added during the quantifier elimination. Then, $Atoms_{res} = Atoms_{sav} \cup Atoms_{impl}$. Let v_{sav} and v_{rem} be some variables, s.t. $v_{sav} \notin V_{rem}$ and $v_{rem} \in V_{rem}$. Then, formulas from $Atoms_{impl}$ are implications made based on formulas of the form $P(v_{rem}, v_{sav})$ from $Atoms_{src}$, since implications based on other types of formulas are redundant. In what follows, we describe how set $Atoms_{impl}$ is constructed for different types of formulas of the form $P(v_{rem}, v_{sav})$. We consider construction of $Atoms_{impl}$ for a single formula. To construct $Atoms_{impl}$ for multiple formulas of the form $P(v_{rem}, v_{sav})$, the same approach must be applied multiple times.

Let $\phi = (v_{rem} = v_{sav})$. Then, $Atoms_{impl} = \{\phi'[v_{rem}/v_{sav}] | (\phi' \in Atoms_{src}) \wedge (v_{rem} \in \phi') \wedge (v_{sav} \notin \phi')\}$.

Let $\phi = (v_{sav} \neq v_{rem})$. Let $A : V \to 2^D$ be a function mapping each variable $v \in V$ to a set of values that v can take according to constraints in $Atoms_{src}$. If $|A(v_{rem})| = 1$, then $Atoms_{impl} = \{v_{sav} \neq A(v_{rem})\}$; otherwise, $Atoms_{impl} = \emptyset$. For formulas over domain $D = \mathbb{R}$, $|A(v_{rem})| = 1$ holds only if a number that v_{rem} can take is present in any formula from the set $Atoms_{src}$.

Let $\phi = v_{sav} \geq v_{rem}$. Let $Const$ be a set of constants occurring in $Atoms_{src}$. If the minimal value of v_{rem} is defined, v_{rem} is either strictly greater some $const \in Const$ or greater than or equal to some $const \in Const$. If $v_{rem} \geq const$, then $Atoms_{impl} = \{v_{sav} \geq const\}$. If $v_{rem} > const$, then $Atoms_{impl} = \{v_{sav} > const\}$. If the minimal value of v_{rem} is not defined, $Atoms_{impl} = \emptyset$.

Let $\phi = v_{sav} \leq v_{rem}$. Let $Const$ be a set of constants occurring in $Atoms_{src}$. If the maximal value of v_{rem} is defined, v_{rem} is either strictly less some $const \in Const$ or less than or equal to some $const \in Const$. If $v_{rem} \leq const$, then $Atoms_{impl} = \{v_{sav} \leq const\}$. If $v_{rem} < const$, then $Atoms_{impl} = \{v_{sav} < const\}$. If the maximal value of v_{rem} is not defined, $Atoms_{impl} = \emptyset$.

Let $\phi = v_{sav} > v_{rem}$. Let $Const$ be a set of constants occurring in $Atoms_{src}$. If the minimal value of v_{rem} is defined, v_{rem} is either strictly greater some $const \in Const$ or greater than or equal to some $const \in Const$. If $v_{rem} \geq const$ or $v_{rem} > const$, $Atoms_{impl} = \{v_{sav} > const\}$. If the minimal value of v_{rem} is not defined, $Atoms_{impl} = \emptyset$.

Let $\phi = v_{sav} < v_{rem}$. Let $Const$ be a set of constants occurring in $Atoms_{src}$. If the maximal value of v_{rem} is defined, v_{rem} is either strictly less some $const \in Const$ or less than or equal to some $const \in Const$. If $v_{rem} \leq const$ or $v_{rem} < const$, $Atoms_{impl} = \{v_{sav} < const\}$. If the maximal value of v_{rem} is not defined, $Atoms_{impl} = \emptyset$.

All the mentioned above operations do not lead to appearance of any new constants. Thus, for the set of constraints $Atoms_{src}$, it is always possible to construct implications using constants only from $Atoms_{src}$ and, therefore, $C_{res} \subseteq (C_\phi \cup C_t)$, which means that the quantifier elimination does not lead to generation of any new constants for a DPN defined on the domain of real numbers. Thus, there is language L with a finite set of constants that describes the closure of A_I under $\{Im, \Delta\}$. Since in L the set of constants, the set of predicates and the set of variables are finite, the closure of A_I under $\{Im, \Delta\}$ is finite.

Proof of Lemma 4.4:

To prove that (M_I, A_I) O-simulates (M_I, α_I), it is sufficient to prove that for any $(M_I, \alpha_I) \xrightarrow{t} (M, \alpha)$ in $RG_\mathcal{N}$ there always exists transition $(M_I, A_I) \xrightarrow{t} (M, A)$ in $CG_\mathcal{N}$, s.t. (M, A) O-simulates (M, α). Note that each transition that may fire at (M_I, α_I) in $RG_\mathcal{N}$ may also fire at (M_I, A_I) in $CG_\mathcal{N}$. Let (M, α) and (M, A) be states yielded by firing some t at (M_I, α_I) and (M_I, A_I), respectively. Since $\alpha \in Im_t(\alpha_I)$ and $A_I = \{\alpha_I\}$, $\alpha \in A$. Thus, each transition that may fire at (M, α) may also fire at (M, A). Let (M', α') be a state yielded by firing some t' at (M, α). Then, $\alpha' \in Im_{t'}(\alpha)$ while $Im_{t'}(\alpha) \subseteq Im_{t'}(A)$. Thus, each transition that may fire at (M', α') may also fire at $(M', Im_{t'}(A))$ that is yielded by firing t' at (M, A). By repeating this inductive step, we prove that $CG_\mathcal{N}$ O-simulates $RG_\mathcal{N}$.

Proof of Theorem 4.5:

Consider property $P1$.

($\Rightarrow$) This direction follows by O-simulation. Let $P1$ hold for $RG_\mathcal{N}$. Assume to fix a state (M, α) reached by executing a trace σ', and for which property $P1$ must hold: there exists a trace

σ s.t. $(M, \alpha) \xrightarrow{\sigma} (M_F, \alpha')$ for some α'. Based on Definition 4.7 and Lemma 4.4, there exists at least one node (M, A) in $CG_{\mathcal{N}}$ reached by executing $O(\sigma')$, s.t. (M, A) O-simulates (M, α). Then by Lemma 4.4 there must also exist a run $(M, A) \xrightarrow{\sigma''} (M_F, A')$, for some A', with $\sigma'' = O(\sigma)$. Thus, if $P1$ holds for $RG_{\mathcal{N}}$, then $P1$ holds for $CG_{\mathcal{N}}$.

($\Leftarrow$) Let $P1$ hold for $CG_{\mathcal{N}}$. Assume that $P1$ does not hold for $RG_{\mathcal{N}}$. Then either $RG_{\mathcal{N}}$ has additional runs which do not correspond to runs of $CG_{\mathcal{N}}$ or there exists in $CG_{\mathcal{N}}$ at least one run, with trace σ, s.t. $O(\sigma)$ is not a trace of $RG_{\mathcal{N}}$. Based on Lemma 4.4, $RG_{\mathcal{N}}$ cannot have additional runs which do not correspond to runs of $CG_{\mathcal{N}}$. According to Definition 4.1, $CG_{\mathcal{N}}$ is a generalization of $RG_{\mathcal{N}}$. Then, by construction, there cannot exist a run in $CG_{\mathcal{N}}$ with trace σ, s.t. $O(\sigma)$ is not a trace of $RG_{\mathcal{N}}$. For instance, consider $(M, A) \xrightarrow{\sigma} (M', A')$ to be one-step with $O(\sigma) = t$. Then, there must exist some state (M, α) in $RG_{\mathcal{N}}$, s.t. $\alpha \in A$, (M, α) is O-simulated by (M, A) and $Im_t(\alpha) \neq \emptyset$. Otherwise, $(M, A) \xrightarrow{\sigma} (M', A')$ cannot exist in $CG_{\mathcal{N}}$. Thus, if $P1$ holds for $CG_{\mathcal{N}}$, then $P1$ holds for $RG_{\mathcal{N}}$.

For $P2$ and $P3$ we follow the similar reasoning that also comes from Definition 4.1 and Definition 4.7.