Uniform Substitution for Dynamic Logic with Communicating Hybrid Programs

Marvin Brieger[1][0000−0001−9656−2830], Stefan Mitsch[2][0000−0002−3194−9759],
André Platzer[2,3][0000−0001−7238−5710]

[1] LMU Munich, Germany
[2] Carnegie Mellon University, Pittsburgh, USA
[3] Karlsruhe Institute of Technology, Germany
marvin.brieger@sosy.ifi.lmu.de smitsch@cs.cmu.edu platzer@kit.edu

Abstract. This paper introduces a uniform substitution calculus for dL_{CHP}, the dynamic logic of communicating hybrid programs. Uniform substitution enables parsimonious prover kernels by using axioms instead of axiom schemata. Instantiations can be recovered from a single proof rule responsible for soundness-critical instantiation checks rather than being spread across axiom schemata in side conditions. Even though communication and parallelism reasoning are notorious for necessitating subtle soundness-critical side conditions, uniform substitution when generalized to dL_{CHP} manages to limit and isolate their conceptual overhead. Since uniform substitution has proven to simplify the implementation of hybrid systems provers substantially, uniform substitution for dL_{CHP} paves the way for a parsimonious implementation of theorem provers for hybrid systems with communication and parallelism.

Keywords: Uniform substitution · Parallel programs · Differential dynamic logic · Assumption-commitment reasoning · CSP

1 Introduction

Hybrid systems and parallel systems are notoriously subtle to analyze. Combining both not only culminates these subtleties but is further complicated because parallel hybrid systems are interlocked by synchronization in a shared global time. The *dynamic logic of communicating hybrid programs* dL_{CHP} [6] tames the complexity of parallel hybrid systems providing a compositional proof calculus that disentangles reasoning into purely discrete, continuous, and communication pieces. However, the calculus is subject to schematic side conditions whose implementation is generally error-prone causing large soundness-critical code bases [29]. In particular, compositional reasoning about parallelism as in the idealized proof rule in Fig. 1 holds the challenge to exhaustively characterize *all* side conditions required to make *all* instances of

$$\frac{[\alpha]\varphi \qquad [\beta]\psi}{[\alpha \parallel \beta](\varphi \wedge \psi)}\ (\star\star)$$

Fig. 1: The proof rule is only sound under subtle side conditions $(\star\star)$.

this proof rule sound. Proof systems for discrete parallelism [1, 18, 26, 35, 44, 46] already have complicated side conditions, but complexity only increases with continuous interactions in shared global time.

In order to compositionally support compositional reasoning for parallel hybrid systems, this paper generalizes Church's uniform substitution [7] and develops a uniform substitution calculus [29–31] for dL_{CHP}. Uniform substitution modularizes the calculus itself enabling its parsimonious implementation. Although applicable to discrete parallelism, the dL_{CHP} development resolves the inherent challenge that parallel hybrid systems always synchronize in time.

Uniform substitution adopts a finite list of concrete formulas as axioms instead of an infinite set of formulas via axiom schemata with side conditions. This enables theorem provers without the extensive algorithmic checks otherwise required for each schema to sort out unsound instances. Thanks to the proof rule US for uniform substitution, only sound instances derive from the axioms such that the parallel composition rule in dL_{CHP} could be adopted almost literally as above, but with all the soundness-critical checking encapsulated solely in rule US. Thanks to US's checking, parallel systems reasoning even reduces to a single parallel injection axiom $[\alpha]\psi \rightarrow [\alpha \parallel \beta]\psi$ that merely describes the preservation of property ψ of one parallel component α in the parallel system $\alpha \parallel \beta$. Proofs about $\alpha \parallel \beta$ reduce to a sequence of property embeddings with this axiom from local abstractions of the subcomponents, which combine soundly due to US.

Soundness checks in uniform substitution are ultimately determined by the binding structures as identified in the static semantics. The development of uniform substitution for dL_{CHP} is, therefore, grounded in the following key observation: Communication and parallelism both cause additional binding structure that needs attention in the substitution process performed by rule US:

(B I) Expressions depend on communication along (co)finite channel sets (besides finitely many free variables), which, by the core substitution principle [7], must not be introduced free into contexts where they are written.

(B II) Subprograms in a parallel context need to be restricted in the variables and channels written as compositional proof rules for parallelism require local abstractions of subprograms not depending on the internals of the context [35].

Grounded in the need for abstraction (B II), $[\alpha]\psi \rightarrow [\alpha \parallel \beta]\psi$ can only be adopted as a sound axiom schema if α and β do not share state, and if program β does not interfere with the contract ψ, i.e., (i) ψ has no free variables bound by β (with exceptions), and (ii) ψ does not depend on communication channels written by β (except for channels joint with α). This extensive side condition would need nontrivial soundness-critical implementations of dL_{CHP} axiom schemata. Still, uniform substitution can be lifted with only small changes locally checking for clashes with written channels, and prohibited variables or channels.

The modularity of uniform substitution is the key to the parsimonious implementation [22] of the theorem prover KeYmaera X [10] for differential dynamic logic dL and differential game logic dGL [28], thus paving the way for a straightforward theorem prover implementation of dL_{CHP}. Since dL_{CHP} conservatively generalizes dL [6], its uniform substitution calculus inherits the complete [32] axiomatic treatment of differential equation invariants [29].

2 Dynamic Logic of Communicating Hybrid Programs

This section briefly recaps dL_{CHP} [6], the dynamic logic of communicating hybrid programs (CHPs). It combines hybrid programs [27] with CSP-style communication and parallelism [14]. By assumption-commitment (ac) reasoning [21,46,47], dL_{CHP} allows compositional verification of parallelism in dL. For uniform substitution, function and predicate symbols, and program constants are added.

2.1 Syntax

The set of variables $V = V_\mathbb{R} \cup V_\mathbb{N} \cup V_\mathcal{T}$ has real ($V_\mathbb{R}$), integer ($V_\mathbb{N}$), and trace ($V_\mathcal{T}$) variables. For each $x \in V_\mathbb{R}$, the differential symbol x' is in $V_\mathbb{R}$, too. The designated variable $\mu \in V_\mathbb{R}$ represents the shared global time. The set of channel names is Ω. By convention $x, y \in V_\mathbb{R}$, $n \in V_\mathbb{N}$, $h \in V_\mathcal{T}$, $\mathrm{ch} \in \Omega$, and $z \in V$. Channel set $Y \subseteq \Omega$ is (co)finite. Vectorial expressions are denoted $\bar{e}$. Moreover, $f^\mathbb{M}$, $g^\mathbb{M}$ are $\mathbb{M}$-valued function symbols and p, q, r are predicate symbols, where argument sorts are annotated by $\_ : \mathbb{M}_1, \dots, \mathbb{M}_k$. Finally, a, b are program constants.

Definition 1 (Terms). *Terms consist of real* ($\mathrm{Trm}_\mathbb{R}$)*, integer* ($\mathrm{Trm}_\mathbb{N}$)*, channel* (Trm_Ω)*, and trace* ($\mathrm{Trm}_\mathcal{T}$) *terms, and are defined by the grammar below, where* $\theta, \theta_1, \theta_2 \in \mathbb{Q}[V_\mathbb{R}] \subset \mathrm{Trm}_\mathbb{R}$ *are polynomials in* $V_\mathbb{R}$*:*

$$\begin{aligned}
\mathrm{Trm}_\mathbb{R} :&\quad \eta_1, \eta_2 ::= x \mid f^\mathbb{R}(Y, \bar{e}) \mid \eta_1 + \eta_2 \mid \eta_1 \cdot \eta_2 \mid (\theta)' \mid \mathtt{val}(te) \mid \mathtt{time}(te) \\
\mathrm{Trm}_\mathbb{N} :&\quad ie_1, ie_2 ::= n \mid f^\mathbb{N}(Y, \bar{e}) \mid ie_1 + ie_2 \mid |te| \\
\mathrm{Trm}_\Omega :&\quad ce_1, ce_2 ::= f^\Omega(Y, \bar{e}) \mid \mathtt{chan}(te) \\
\mathrm{Trm}_\mathcal{T} :&\quad te_1, te_2 ::= h \mid f^\mathcal{T}(Y, \bar{e}) \mid \langle ch, \theta_1, \theta_2 \rangle \mid te_1 \cdot te_2 \mid te \downarrow Y \mid te[ie]
\end{aligned}$$

Real terms are polynomials in $V_\mathbb{R}$ enriched with function symbols $f^\mathbb{R}(Y, \bar{e})$ (including constants $c \in \mathbb{Q}$) only depending on communication along channels Y and terms $\bar{e}$, differential terms $(\theta)'$, and $\mathtt{val}(te)$ and $\mathtt{time}(te)$, which access the value and the timestamp of the last communication in te, respectively. By convention, $\theta \in \mathbb{Q}[V_\mathbb{R}]$ denotes a pure polynomial in $V_\mathbb{R}$ without $(\cdot)'$, $\mathtt{val}(\cdot)$, and $\mathtt{time}(\cdot)$ as they occur in programs. For simplicity, we do not define $\mathbb{Q}[V_\mathbb{R}] \subset \mathrm{Trm}_\mathbb{R}$ as a fifth term sort but use the convention that function symbols $g^\mathbb{R}$ can only be replaced with $\mathbb{Q}[V_\mathbb{R}]$-terms. Integer terms are variables n, function symbols $f^\mathbb{N}(Y, \bar{e})$ (including constants 0, 1), addition, and length $|te|$ of trace term te.[4] The function symbol $f^\Omega(Y, \bar{e})$ includes constants $\mathrm{ch} \in \Omega$, and $\mathtt{chan}(te)$ is channel access. Trace terms record the communication history of programs. They encompass variables h, function symbols $f^\mathcal{T}(Y, \bar{e})$ (including the empty trace ϵ), communication items $\langle \mathrm{ch}, \theta_1, \theta_2 \rangle$ with value θ_1 and timestamp θ_2, projection

[4] Omitting multiplication results in decidable Presburger arithmetic [33].

$te \downarrow Y$ onto channels Y, and access $te[ie]$ of the ie-th item in te. Where useful, $\mathtt{op}(\bar{e})$ denotes built-in function symbols of fixed interpretation, e.g., $\cdot + \cdot$.

$\mathsf{dL}_{\mathrm{CHP}}$'s context-sensitive program and formula syntax presumes notions of free and bound variables (Section 2.3) defined on the context-free syntax:

Definition 2 (Programs). Communicating hybrid programs *are defined by the following grammar, where $\theta \in \mathbb{Q}[V_{\mathbb{R}}]$ is a polynomial in $V_{\mathbb{R}}$ and $\chi \in \mathrm{FOL}_{\mathbb{R}}$ is a formula of first-order real-arithmetic. In $\alpha \parallel \beta$, the subprograms must not share state but can share time and history, i.e., $\mathsf{BV}(\alpha) \cap \mathsf{BV}(\beta) \subseteq \{\mu, \mu'\} \cup V_{\mathcal{T}}$.*[5]

$$\begin{aligned}\alpha, \beta ::= {} & a\llparenthesis Y, \bar{z} \rrparenthesis \mid x := \theta \mid x := * \mid ?\chi \mid \{x' = \theta \,\&\, \chi\} \mid \alpha; \beta \mid \alpha \cup \beta \mid \alpha^* \mid \\ & ch(h)!\theta \mid ch(h)?x \mid \alpha \parallel \beta\end{aligned}$$

The program constant $a\llparenthesis Y, \bar{z} \rrparenthesis$ restricts the written channels to $Y \subseteq \Omega$ and the bound variables to $\bar{z} \subseteq V_{\mathbb{R}} \cup V_{\mathcal{T}}$, where Y and $\bar{z}$ are (co)finite. Instead of $a\llparenthesis Y, \bar{z} \rrparenthesis$, write a if Y and $\bar{z}$ can be arbitrary. Assignment $x := \theta$ updates x to θ, nondeterministic assignment $x := *$ assigns an arbitrary real value to x, and the test $?\chi$ does nothing if χ holds and aborts the computation otherwise. The continuous evolution $\{x' = \theta \& \chi\}$ follows the ODE $x' = \theta$ for any duration as long as formula χ is not violated. The global time μ evolves with every continuous evolution according to ODE $\mu' = 1$. Sequential composition $\alpha; \beta$ executes β after α, choice $\alpha \cup \beta$ executes α or β nondeterministically, α^* repeats α zero or more times, $\mathrm{ch}(h)!\theta$ sends θ along channel ch, and $\mathrm{ch}(h)?x$ receives a value into variable x along channel ch. The trace variable h records communication. Finally, $\alpha \parallel \beta$ executes α and β in parallel synchronized in global time μ.

Example 3. The program $\mathtt{ct}^* \parallel \mathtt{ve}^*$ models a simplified cruise control [23]. The vehicle $\mathtt{ve}$ repeatedly receives a target velocity $v_{\mathtt{ve}}^{\mathrm{tr}}$ from the controller $\mathtt{ct}$ along channel tar. The target $v_{\mathtt{ct}}^{\mathrm{tr}}$ sent by $\mathtt{ct}$ is in range $[0, V]$. Hence, $\mathtt{ve}$'s velocity $v_{\mathtt{ve}}$ stays in range $[0, V]$ within the $\epsilon > 0$ time units till the next communication if $v_{\mathtt{ve}} \in [0, V]$ held initially. The evolution $\{t' = 1\}$ allows passage of time in $\mathtt{ct}$.

$$\mathtt{ct} \equiv v_{\mathtt{ct}}^{\mathrm{tr}} := *; ?(0 \leq v_{\mathtt{ct}}^{\mathrm{tr}} \leq V); \mathrm{tar}(h)!v_{\mathtt{ct}}^{\mathrm{tr}}; \{t' = 1\}$$

$$\mathtt{ve} \equiv \mathrm{tar}(h)?v_{\mathtt{ve}}^{\mathrm{tr}}; a_{\mathtt{ve}} := \frac{v_{\mathtt{ve}}^{\mathrm{tr}} - v_{\mathtt{ve}}}{\epsilon}; t_0 := \mu; \{v_{\mathtt{ve}}' = a_{\mathtt{ve}} \,\&\, \mu - t_0 \leq \epsilon\}$$

Definition 4 (Formulas). *Formulas are defined by the grammar below for relations $\sim$, terms $e_1, e_2 \in \mathrm{Trm}$ of equal sort, and $z \in V$. Moreover, the ac-formulas are unaffected by state change in α, i.e., $(\mathsf{FV}(\mathsf{A}) \cup \mathsf{FV}(\mathsf{C})) \cap \mathsf{BV}(\alpha) \subseteq V_{\mathcal{T}}$.*

$$\varphi, \psi, \mathsf{A}, \mathsf{C} ::= e_1 \sim e_2 \mid p(Y, \bar{e}) \mid \neg\varphi \mid \varphi \wedge \psi \mid \forall z\, \varphi \mid [\alpha]\psi \mid [\alpha]_{\{\mathsf{A},\mathsf{C}\}}\psi$$

The formulas combine first-order dynamic logic with ac-reasoning. Predicate symbols $p(Y, \bar{e})$ depend on channels Y and terms $\bar{e}$. The ac-box $[\alpha]_{\{\mathsf{A},\mathsf{C}\}}\psi$ expresses that C holds after each communication event and ψ in the final state, for

[5] Previous work [6] disallows reading of variables bound in parallel as their change is not observable. This restriction is conceptually desirable but not soundness-critical. Here we drop it for simplicity, but it could be maintained by US as well.

all runs of α whose incoming communication satisfies A. Other connectives $\vee$, $\rightarrow$, $\leftrightarrow$ and quantifiers $\exists z\, \varphi \equiv \neg\forall z\, \neg\varphi$ can be derived. The relations $\sim$ include $=$ for all term sorts, $\geq$ on real and integer terms, and prefixing $\preceq$ on trace terms.

By convention, the predicate symbol $q_{\mathbb{R}}$ can only be replaced with formulas of first-order real arithmetic. It serves as placeholder for tests χ in CHPs.

Example 5. The cruise control from Example 3 is safe if its velocity stays in range $[0, V]$. This can be expressed with the formula $\varphi \rightarrow [\mathtt{ct}^* \parallel \mathtt{ve}^*]\psi_{\text{safe}}$, where $\psi_{\text{safe}} \equiv 0 \leq v_{\mathtt{ve}} \leq V$ and $\varphi \equiv \psi_{\text{safe}} \wedge \epsilon > 0 \wedge V > 0$.

2.2 Semantics

A *trace* $\tau = (\tau_1, ..., \tau_k)$ is a finite chronological sequence of communication events $\tau_i = \langle \text{ch}_i, d_i, s_i \rangle$, where $\text{ch}_i \in \Omega$, and $d_i \in \mathbb{R}$ is the communicated value, and $s_i \in \mathbb{R}$ is a timestamp such that $s_i \leq s_j$ for $1 \leq i < j \leq k$. A *recorded trace* $\tau = (\tau_1, ..., \tau_k)$ additionally carries a trace variable $h_i \in V_{\mathcal{T}}$ with each event, i.e., $\tau_i = \langle h_i, \text{ch}_i, d_i, s_i \rangle$. For variable $z \in V_{\mathbb{M}}$ and $\mathbb{M} \in \{\mathbb{R}, \mathbb{N}, \mathcal{T}\}$, let $\mathit{type}(z) = \mathbb{M}$. A *state* v maps each $z \in V$ to a value $v(z) \in \mathit{type}(z)$. The sets of traces, recorded traces, and states are denoted $\mathcal{T}$, $\mathcal{T}_{\text{rec}}$, and $\mathcal{S}$, respectively.

For $d \in \mathit{type}(z)$, the state v_z^d is the modification of v at z to d. For $\tau \in \mathcal{T}_{\text{rec}}$, the trace $\tau(h) \in \mathcal{T}$ is obtained from the subsequence of τ carrying $h \in V_{\mathcal{T}}$ by removing the carried variable. *State-trace concatenation* $v \cdot \tau \in \mathcal{S}$ for $\tau \in \mathcal{T}_{\text{rec}}$, appends $\tau(h)$ to v at h for all $h \in V_{\mathcal{T}}$. The *projection* $\tau \downarrow Y$ of (recorded) trace τ is the subsequence of all communication events in τ whose channel is in $Y \subseteq \Omega$. The *state projection* $v \downarrow Y \in \mathcal{S}$ modifies v at h to $v(h) \downarrow Y$ for all $h \in V_{\mathcal{T}}$.

An *interpretation* I assigns a function $I(f^{\mathbb{M}} : \mathbb{M}_1, \ldots, \mathbb{M}_k) : \bigtimes_{i=1}^{k} \mathbb{M}_i \rightarrow \mathbb{M}$ to each function symbol $f^{\mathbb{M}}$ that is smooth in all real-valued arguments if $\mathbb{M} = \mathbb{R}$, and a relation $I(p : \mathbb{M}_1, \ldots, \mathbb{M}_k) \subseteq \bigtimes_{i=1}^{k} \mathbb{M}_i$ to each k-ary predicate symbol p.

Definition 6 (Term semantics). *The* valuation $Iv[\![e]\!] \in \mathbb{R} \cup \mathbb{N} \cup \Omega \cup \mathcal{T}$ *of term e in interpretation I and state v is defined as follows:*

$$\begin{aligned}
Iv[\![z]\!] &= v(z) \\
Iv[\![f(Y, e_1, ..., e_k)]\!] &= I(f)(I\tilde{v}[\![e_1]\!], ..., I\tilde{v}[\![e_k]\!]) && \text{where } \tilde{v} = v \downarrow Y \\
Iv[\![\mathsf{op}(e_1, \ldots, e_k)]\!] &= \mathsf{op}(Iv[\![e_1]\!], \ldots, Iv[\![e_k]\!]) && \text{for builtin } \mathsf{op} \in \{\cdot + \cdot, \cdot \downarrow Y, \ldots\} \\
Iv[\![(\theta)']\!] &= \sum_{x \in V_{\mathbb{R}}} v(x') \frac{\partial Iv[\![\theta]\!]}{\partial x}
\end{aligned}$$

The projection $\tilde{v} = v \downarrow Y$ ensures that $f(Y, \bar{e})$ only depends on Y, i.e., the communication in v along channels $Y^{\complement}$ does not matter. The differentials $(\theta)'$ have a semantics describing the local rate of change of θ [29].

The denotational semantics of CHPs [6] combines dL's Kripke semantics [29] with a linear history semantics [47] and a global notion of time. Denotations are subsets of $\mathcal{D} = \mathcal{S} \times \mathcal{T}_{\text{rec}} \times \mathcal{S}_{\perp}$ with $\mathcal{S}_{\perp} = \mathcal{S} \cup \{\perp\}$. Final state $\perp$ marks an unfinished computation, i.e., it still can be continued or was aborted due to a failing test. If ($w' = \bot$ and $\tau' \preceq \tau$), where $\preceq$ is the prefix relation on traces, or $(\tau', w') = (\tau, w)$, then (τ', w') is a prefix of (τ, w) written $(\tau', w') \preceq (\tau, w)$. Since (even empty) communication of unfinished computations is still observable, denotations $D \subseteq \mathcal{D}$ of CHPs are prefix-closed and total, i.e., $(v, \tau, w) \in D$ and $(\tau', w') \preceq (\tau, w)$ implies $(v, \tau', w') \in D$, and $\bot_{\mathcal{D}} \subseteq D$ with $\bot_{\mathcal{D}} = \mathcal{S} \times \{\epsilon\} \times \{\bot\}$. Moreover, all $(v, \tau, w) \in D$ are chronological, i.e., $v(\mu) \leq w(\mu)$ and when $\tau = (\tau_1, \ldots, \tau_k) \neq \epsilon$ and let $\tau_i(\mu) = (\langle h_i, \mathrm{ch}_i, d_i, s_i \rangle)(\mu) = s_i$, then $v(\mu) \leq \tau_1(\mu)$ and if $w \neq \bot$, then $\tau_k(\mu) \leq w(\mu)$. Note that τ is chronological as all traces are.

The interpretation $I(a(\!|Y, \bar{z}|\!)) \subseteq \mathcal{D}$ of a program constant $a(\!|Y, \bar{z}|\!)$ is a prefix-closed and total set of chronological computations that (i) only communicate along (write) channels Y and (ii) only bind variables $\bar{z}$. More precisely, for all $(v, \tau, w) \in I(a(\!|Y, \bar{z}|\!))$, we have (i) $\tau \downarrow Y^{\complement} = \epsilon$, and (ii) $v = w$ on $V_{\mathcal{T}}$ and $w \cdot \tau = v$ on $\bar{z}^{\complement}$. For $D, M \subseteq \mathcal{D}$, we define $D_\bot = \{(v, \tau, \bot) \mid (v, \tau, w) \in D\}$, and $(v, \tau, w) \in D \triangleright M$ if $(v, \tau_1, u) \in D$ and $(u, \tau_2, w) \in M$ exist with $\tau = \tau_1 \cdot \tau_2$. For states w_α, w_β, the merged state $w_\alpha \oplus w_\beta$ is $\bot$ if one of the substates w_α or w_β is $\bot$. Otherwise, $w_\alpha \oplus w_\beta = w_\alpha$ on $\mathsf{BV}(\alpha)$ and $w_\alpha \oplus w_\beta = w_\beta$ on $\mathsf{BV}(\alpha)^{\complement}$ (or, equivalently by syntactic well-formedness, on $\mathsf{BV}(\beta)^{\complement}$ and $\mathsf{BV}(\beta)$, respectively). If Y is the set of all channel names occurring in α, we write $\tau \downarrow \alpha$ for $\tau \downarrow Y$.

Definition 7 (Program semantics). *Given an interpretation I, the* semantics *$I[\![\alpha]\!] \subseteq \mathcal{D}$ of a CHP α is defined as follows, where $\bot_{\mathcal{D}} = \mathcal{S} \times \{\epsilon\} \times \{\bot\}$ and $\vDash$ denotes the satisfaction relation (Def. 8):*

$$I[\![a(\!|Y, \bar{z}|\!)]\!] = I(a(\!|Y, \bar{z}|\!))$$

$$I[\![x := \theta]\!] = \bot_{\mathcal{D}} \cup \{(v, \epsilon, w) \mid w = v_x^d \textit{ where } d = Iv[\![\theta]\!]\}$$

$$I[\![x := *]\!] = \bot_{\mathcal{D}} \cup \{(v, \epsilon, w) \mid w = v_x^d \textit{ where } d \in \mathbb{R}\}$$

$$I[\![?\chi]\!] = \bot_{\mathcal{D}} \cup \{(v, \epsilon, v) \mid Iv \vDash \chi\}$$

$$\begin{aligned} I[\![\{x' = \theta \,\&\, \chi\}]\!] = \bot_{\mathcal{D}} \cup \big\{&(v, \epsilon, \varphi(s)) \mid v = \varphi(0) \textit{ on } \{\mu', x'\}^{\complement}, \textit{ and } \varphi(\zeta) = \varphi(0) \\ &\textit{on } \{x, x', \mu, \mu'\}^{\complement}, \textit{ and } I\varphi(\zeta) \vDash \mu' = 1 \wedge x' = \theta \wedge \chi \textit{ for all } \zeta \in [0, s] \textit{ and} \\ &\textit{a solution } \varphi : [0, s] \to \mathcal{S} \textit{ with } \varphi(\zeta)(z') = \frac{d\varphi(t)(z)}{dt}(\zeta) \textit{ for } z \in \{x, \mu\}\big\} \end{aligned}$$

$$I[\![ch(h)!\theta]\!] = \{(v, \tau, w) \mid (\tau, w) \preceq (\langle h, ch, d, v(\mu)\rangle, v) \textit{ where } d = Iv[\![\theta]\!]\}$$

$$I[\![ch(h)?x]\!] = \{(v, \tau, w) \mid (\tau, w) \preceq (\langle h, ch, d, v(\mu)\rangle, v_x^d) \textit{ where } d \in \mathbb{R}\}$$

$$I[\![\alpha \cup \beta]\!] = I[\![\alpha]\!] \cup I[\![\beta]\!]$$

$$I[\![\alpha ; \beta]\!] = I[\![\alpha]\!] \mathbin{\hat{\circ}} I[\![\beta]\!] \overset{\text{def}}{=} (I[\![\alpha]\!])_\bot \cup (I[\![\alpha]\!] \triangleright I[\![\beta]\!])$$

$$I[\![\alpha^*]\!] = \bigcup_{n \in \mathbb{N}} (I[\![\alpha]\!])^n = \bigcup_{n \in \mathbb{N}} I[\![\alpha^n]\!] \quad \text{where } \alpha^0 \equiv ?\mathsf{T} \text{ and } \alpha^{n+1} = \alpha ; \alpha^n$$

$$I[\![\alpha_1 \parallel \alpha_2]\!] = \left\{ (v, \tau, w_{\alpha_1} \oplus w_{\alpha_2}) \;\middle|\; \begin{array}{l} (v, \tau \downarrow \alpha_j, w_{\alpha_j}) \in I[\![\alpha_j]\!] \textit{ for } j = 1, 2, \textit{ and} \\ w_{\alpha_1} = w_{\alpha_2} \textit{ on } \{\mu, \mu'\}, \textit{ and } \tau = \tau \downarrow (\alpha_1 \| \alpha_2) \end{array} \right\}$$

The semantics is indeed constructed prefix-closed, total, and chronological. Communication τ of $\alpha_1 \parallel \alpha_2$ is implicitly characterized via its subsequences for the subprograms. By $\tau = \tau \downarrow (\alpha_1 \parallel \alpha_2)$, there is no non-causal communication. Joint communication and the whole computation are synchronized in global time by the projections and by $w_{\alpha_1} = w_{\alpha_2}$ on $\{\mu, \mu'\}$, respectively. Likewise, by projection, communication is synchronously recorded by trace variables.

Definition 8 (Formula semantics). *The* satisfaction $Iv \vDash \phi$ *of a* dL_{CHP} *formula* ϕ *in interpretation* I *and state* v *is inductively defined as follows:*

1. $Iv \vDash e_1{\sim}e_2$ *if* $Iv[\![e_1]\!] \sim Iv[\![e_2]\!]$ where $\sim$ is any relation symbol
2. $Iv \vDash p(Y, e_1, \dots, e_k)$ *if* $(I\tilde{v}[\![e_1]\!], \dots, I\tilde{v}[\![e_k]\!]) \in I(p)$ where $\tilde{v} = v \downarrow Y$
3. $Iv \vDash \varphi \wedge \psi$ *if* $Iv \vDash \varphi$ *and* $Iv \vDash \psi$
4. $Iv \vDash \neg\varphi$ *if* $Iv \nvDash \varphi$*, i.e., it is not the case that* $Iv \vDash \varphi$
5. $Iv \vDash \forall z\, \varphi$ *if* $Iv_z^d \vDash \varphi$ *for all* $d \in type(z)$
6. $Iv \vDash [\alpha]\psi$ *if* $Iw \cdot \tau \vDash \psi$ *for all* $(v, \tau, w) \in I[\![\alpha]\!]$ *with* $w \neq \bot$
7. $Iv \vDash [\alpha]_{\{\mathsf{A},\mathsf{C}\}}\psi$ *if for all* $(v, \tau, w) \in I[\![\alpha]\!]$ *the following conditions hold:*

$$\{Iv \cdot \tau' \mid \tau' \prec \tau\} \vDash \mathsf{A} \textit{ implies } Iv \cdot \tau \vDash \mathsf{C} \tag{commit}$$

$$\big(\{Iv \cdot \tau' \mid \tau' \preceq \tau\} \vDash \mathsf{A} \textit{ and } w \neq \bot\big) \textit{ implies } Iw \cdot \tau \vDash \psi \tag{post}$$

Where $U \vDash \varphi$ *for a set of interpretation-state pairs* U *and any formula* φ *if* $Iv \vDash \varphi$ *for all* $Iv \in U$*. In particular,* $\emptyset \vDash \varphi$*.*

In item 6 and 7, reachable worlds are built from states v and w, and communication τ, as change of state *and* communication are observable. The strict prefix $\prec$ for the assumption in case (commit) in item 6 excludes (when $\mathsf{A} \equiv \mathsf{C}$) the circularity that commitment C can be shown in states where it is assumed.

2.3 Static Semantics

In the uniform substitution process, checks of free and bound variables, as well as accessed and written channels, separate sound from unsound axiom instantiations. As parallelism requires fine-grained control over channels, the static semantics for dL [29] is lifted to a communication-aware static semantics for dL_{CHP}. It uses accessed channels to characterize the subsequence of a communication trace influencing truth of a formula even more precisely than free variables.

To precisely grasp free and bound variables, and accessed and written channels, Def. 9 gives a semantic characterization. In this section, formulas are considered truth-valued, i.e., $Iv[\![\phi]\!] = \mathbf{tt}$ if $Iv \vDash \phi$ and $Iv[\![\phi]\!] = \mathbf{ff}$ if $Iv \nvDash \phi$.

Definition 9 (Static semantics). *For term or formula* e*, and program* α*, free variables* $\mathsf{FV}(e)$ *and* $\mathsf{FV}(\alpha)$*, bound variables* $\mathsf{BV}(\alpha)$*, accessed channels* $\mathsf{CN}(e)$*, and written channels* $\mathsf{CN}(\alpha)$ *form the static semantics.*

$$\mathsf{FV}(e) = \{z \in V \mid \exists I, v, \tilde{v} \textit{ such that } v = \tilde{v} \textit{ on } \{z\}^{\complement} \textit{ and } Iv[\![e]\!] \neq I\tilde{v}[\![e]\!]\}$$

$$\mathsf{CN}(e) = \{ch \in \varOmega \mid \exists I, v, \tilde{v} \textit{ such that } v \downarrow \{ch\}^{\complement} = \tilde{v} \downarrow \{ch\}^{\complement} \textit{ and } Iv[\![e]\!] \neq I\tilde{v}[\![e]\!]\}$$

$$\begin{aligned}\mathsf{FV}(\alpha) = \{z \in V \mid {} & \exists I, v, \tilde{v}, \tau, w \textit{ such that } v = \tilde{v} \textit{ on } \{z\}^{\complement} \textit{ and } (v, \tau, w) \in I[\![\alpha]\!], \\ & \textit{and there is no } (\tilde{v}, \tilde{\tau}, \tilde{w}) \in I[\![\alpha]\!] \textit{ such that } \tilde{\tau} = \tau \textit{ and } w = \tilde{w} \textit{ on } \{z\}^{\complement}\}\end{aligned}$$

$$\mathsf{BV}(\alpha) = \{z \in V \mid \exists I, (v, \tau, w) \in I[\![\alpha]\!] \textit{ such that } w \neq \bot \textit{ and } (w \cdot \tau)(z) \neq v(z)\}$$

$$\mathsf{CN}(\alpha) = \{ch \in \varOmega \mid \exists I, (v, \tau, w) \in I[\![\alpha]\!] \textit{ such that } \tau \downarrow \{ch\} \neq \epsilon\}$$

The already subtle static semantics of hybrid systems [29] becomes even more subtle with communication and parallelism. For example, CHPs (silently) synchronize with the global time μ, which is free and bound in ODEs, and the differential μ' is bound, i.e., $\mu \in \mathsf{FV}(\{x' = \theta \,\&\, \chi\})$ and $\mu, \mu' \in \mathsf{BV}(\{x' = \theta \,\&\, \chi\})$ if the evolution has a run of non-zero duration, regardless of whether μ occurs in x. Since reachable worlds of CHPs consist of communication *and* state, bound variables $\mathsf{BV}(\alpha)$ of program α compare v with the state-trace concatenation $w \cdot \tau$ instead of missing τ. Consequently, $h \in \mathsf{BV}(\mathrm{ch}(h)!\theta) \subseteq \mathsf{FV}(\mathrm{ch}(h)!\theta)$, which also reflects that the initial communication never gets lost. All proofs for this section and computable overapproximations of the static semantics are in Appendix A.

Lemma 10 (Bound effect property). *The sets* $\mathsf{BV}(\alpha)$ *and* $\mathsf{CN}(\alpha)$ *are the smallest sets with the* bound effect property for program α*. That is,* $v = w$ *on* $V_{\mathcal{T}}$ *and* $v = w \cdot \tau$ *on* $\mathsf{BV}(\alpha)^{\complement}$ *if* $w \neq \bot$*, and* $\tau \downarrow \mathsf{CN}(\alpha)^{\complement} = \epsilon$ *for all* $(v, \tau, w) \in I[\![\alpha]\!]$*.*

By the following *communication-aware* coincidence property, terms and formulas only depend on their free variables, which for trace variables can be further refined to the subtraces whose channels are accessed. This subtrace-level precision is crucial in the soundness proof of the parallel injection axiom as it allows to drop β from $[\alpha \parallel \beta]\psi$ only if β does not write channels of ψ that are not also written by α. The signature $\Sigma(\cdot)$ of an expression denotes all occurring symbols.

Lemma 11 (Coincidence for terms and formulas). *The sets* $\mathsf{FV}(e)$ *and* $\mathsf{CN}(e)$ *are the smallest sets with the* communication-aware coincidence property for term or formula e*. That is, if* $v \downarrow \mathsf{CN}(e) = \tilde{v} \downarrow \mathsf{CN}(e)$ *on* $\mathsf{FV}(e)$ *and* $I = J$ *on* $\Sigma(e)$*, then* $Iv[\![e]\!] = J\tilde{v}[\![e]\!]$*. In particular, for formula* ϕ*:* $Iv \vDash \phi$ *iff* $J\tilde{v} \vDash \phi$*.*

Programs communicate but do *not* depend on the recorded history, thus the coincidence property for programs is not communication-aware. However, programs can produce the same communication starting from coinciding states.

Lemma 12 (Coincidence for programs). *The set* $\mathsf{FV}(\alpha)$ *is the smallest set with the* coincidence property for program α*. That is, if* $v = \tilde{v}$ *on* $X \supseteq \mathsf{FV}(\alpha)$*, and* $I = J$ *on* $\Sigma(\alpha)$*, and* $(v, \tau, w) \in I[\![\alpha]\!]$*, then* $(\tilde{v}, \tilde{\tau}, \tilde{w}) \in J[\![\alpha]\!]$ *exists such that* $w = \tilde{w}$ *on* X*, and* $\tau = \tilde{\tau}$*, and (*$w = \bot$ *iff* $\tilde{w} = \bot$*).*

3 Uniform Substitution for $\mathsf{dL}_{\mathrm{CHP}}$

In $\mathsf{dL}_{\mathrm{CHP}}$, a uniform substitution [29] σ maps function and predicate symbols to terms (of equal sort) and formulas, respectively, while substituting the arguments of the symbol for their placeholders in the replacement, and program constants are mapped to CHPs. For example, $\sigma = \{f(\cdot) \mapsto \cdot + 1, a \mapsto \mathrm{ch}(h)?v; \{x' = v\}\}$ replaces all occurrences of function symbol f with $\cdot + 1$ while the reserved 0-ary function symbol $\cdot$ marks the positions for the parameter of f in the replacement. Moreover, σ replaces the program constant a with the program $\mathrm{ch}(h)?v; \{x' = v\}$.

The key to sound uniform substitution is that new free variables must not be introduced into a context where they are bound [7]. In the presence of communication, likewise, *new channel access must not be introduced into contexts* where the channel is written (B I). For parallelism, substitution *must not reveal internals* of the parallel context to the local abstraction of a subprogram (B II), and must not violate state disjointness. The one-pass approach [31] used for $\mathsf{dL}_{\mathrm{CHP}}$ postpones these checks *and* simply applies the substitution recursively while collecting written variables and channels as taboo set, thus operates linearly in the input. Clashes between the taboo, and new free variables and channel access are only checked locally at the replacement site. Likewise, clashes between the permitted channels and variables of a program constant, and its replacement program are checked locally. All proofs for this section are in Appendix B.

The substitution operator $\sigma_Z^{U,W}(\alpha)$ for program α takes an input taboo $U \subseteq V \cup \Omega$ and a parallel context $W \subseteq V$, and returns, if defined, the substitution result and a set of output taboos $Z \subseteq V \cup \Omega$. For terms and formulas, the substitution operator σ^U only takes a taboo $U \subseteq V \cup \Omega$ as input. The substitution process clashes, i.e., prevents unsound instantiation, if it were to introduce a free variable or accessed channel into a context where it is bound (B I) *or* if it were to write variables and channels violating abstraction (B II). Moreover, substitution preserves well-formedness of programs and formulas, i.e., substitution clashes if replacements were to violate well-formedness.

$$
\begin{array}{rll}
\sigma^U(z) & \equiv z & \text{for } z \in V \\
\sigma^U(f(Y,e)) & \equiv \{\cdot \mapsto \sigma^U(e \downarrow Y)\}^{\emptyset}(\sigma f(\cdot)) & \text{if } (\mathsf{FV}(\sigma f(\cdot)) \cup \mathsf{CN}(\sigma f(\cdot))) \cap U = \emptyset \\
\sigma^U(\mathsf{op}(e_1, \ldots, e_k)) & \equiv \mathsf{op}(\sigma^U(e_1), \ldots, \sigma^U(e_k)) & \text{for built-in } \mathsf{op} \in \{\cdot + \cdot, \cdot \downarrow Y, \ldots\} \\
\sigma^U((\theta)') & \equiv (\sigma^{V \cup \Omega}(\theta))' & \\
\hline
\sigma^U(e_1 \sim e_2) & \equiv \sigma^U(e_1) \sim \sigma^U(e_2) & \\
\sigma^U(p(Y,e)) & \equiv \{\cdot \mapsto \sigma^U(e \downarrow Y)\}^{\emptyset}(\sigma p(\cdot)) & \text{if } (\mathsf{FV}(\sigma p(\cdot)) \cup \mathsf{CN}(\sigma p(\cdot))) \cap U = \emptyset \\
\sigma^U(\neg\varphi) & \equiv \neg\sigma^U(\varphi) & \\
\sigma^U(\varphi \wedge \psi) & \equiv \sigma^U(\varphi) \wedge \sigma^U(\psi) & \\
\sigma^U(\forall z\, \varphi) & \equiv \forall z\, \sigma^{U \cup \{z\}}(\varphi) & \\
\sigma^U([\alpha]\psi) & \equiv [\sigma_Z^{U,\emptyset}(\alpha)]\sigma^Z(\psi) & \\
\sigma^U([\alpha]_{\{\mathsf{A},\mathsf{C}\}}\psi) & \equiv [\sigma_Z^{U,\emptyset}(\alpha)]_{\{\sigma^Z(\mathsf{A}),\sigma^Z(\mathsf{C})\}}\sigma^Z(\psi) & \\
\hline
\sigma_{U \cup \mathsf{BV}(\sigma a) \cup \mathsf{CN}(\sigma a)}^{U,W}(a\llparenthesis Y, \bar{z}\rrparenthesis) & \equiv \sigma a & \text{if } \mathsf{BV}(\sigma a) \subseteq \bar{z} \text{ and } \mathsf{CN}(\sigma a) = Y \\
\sigma_{U \cup \{x\}}^{U,W}(x := \theta) & \equiv x := \sigma^{U \cup W}(\theta) & \\
\sigma_{U \cup \{x\}}^{U,W}(x := *) & \equiv x := * & \\
\sigma_U^{U,W}(?\chi) & \equiv \; ?\sigma^{U \cup W}(\chi) & \\
\sigma_Z^{U,W}(\{x' = \theta \,\&\, \chi\}) & \equiv \{x' = \sigma^{U \cup W}(\theta) \,\&\, \sigma^{U \cup W}(\chi)\} & \text{with } Z = U \cup \{x, x', \mu, \mu'\} \\
\sigma_{U \cup \{\mathrm{ch}, h\}}^{U,W}(\mathrm{ch}(h)!\theta) & \equiv \mathrm{ch}(h)!\sigma^{U \cup W}(\theta) & \\
\sigma_{U \cup \{\mathrm{ch}, h, x\}}^{U,W}(\mathrm{ch}(h)?x) & \equiv \mathrm{ch}(h)?x & \\
\sigma_{Z_1 \cup Z_2}^{U,W}(\alpha \cup \beta) & \equiv \sigma_{Z_1}^{U,W}(\alpha) \cup \sigma_{Z_2}^{U,W}(\beta) & \\
\sigma_{Z_2}^{U,W}(\alpha; \beta) & \equiv \sigma_{Z_1}^{U,W}(\alpha); \sigma_{Z_2}^{Z_1,W}(\beta) & \\
\sigma_Z^{U,W}(\alpha^*) & \equiv (\sigma_Z^{Z,W}(\alpha))^* & \text{when } \sigma_Z^{U,W}(\alpha) \text{ is defined} \\
\sigma_{Z_1 \cup Z_2}^{U,W}(\alpha \parallel \beta) & \equiv \sigma_{Z_1}^{U,W_{U,\beta}}(\alpha) \parallel \sigma_{Z_2}^{U,W_{U,\alpha}}(\beta) &
\end{array}
$$

Fig. 2: Application of uniform substitution for taboo U and parallel context W, where $W_{U,\gamma} \equiv W \cup (\mathsf{BV}(\sigma^{U,W}(\gamma)) \setminus (\{\mu, \mu'\} \cup V_{\mathcal{T}}))$ for any program γ, and $e{\downarrow}Y$ for term e is recursive push down of projection ${\downarrow}Y$, where $p(Y_0, e){\downarrow}Y \equiv p(Y_0 \cap Y, e)$.

The side condition $(\mathsf{FV}(\sigma f(\cdot)) \cup \mathsf{CN}(\sigma f(\cdot))) \cap U = \emptyset$ implements locally that the replacement for f must not introduce free parameters that are tabooed by U (B I). The substitution $\{\cdot \mapsto \sigma^U(e \downarrow Y)\}^\emptyset$ is responsible for the argument e,[6] where $\emptyset$ suffices as the taboo U is already checked on $e \downarrow Y$. By the projection, $e{\downarrow}Y$ only depends on channels Y. Quantification $\forall z$ taboos the bound variable z. Program α in a box or ac-box has an empty parallel context $\emptyset$.

The substitution $\sigma^{U,W}_Z(\alpha)$ computes the output taboo Z by adding the written variables and channels of program α to U, e.g., real variable x for assignment $x := \theta$ and for receiving $\mathrm{ch}(h)?x$ additionally channel ch and trace variable h. The output taboo Z is passed to ac-formulas and postconditions of boxes and ac-boxes for recursive checks for clashes w.r.t. (B I). Crucially for soundness, Lemma 13 below proves that $\sigma^{U,W}_Z(\cdot)$ correctly computes the output taboo Z.

The taboo $U \cup W$ passed to nested expressions contains the parallel context W to prevent free variables in replacements of function and predicate symbols that are bound in parallel. This prepares the substitution process to preserve the syntax restrictions for parallel composition from previous work [6].[7] Substitution for evolution $\{x' = \theta \;\&\; \chi\}$ considers that the global time μ, μ' is always implicitly bound regardless of whether it occurs in x, x'. The fixpoint notation $\sigma^{Z,W}_Z(\alpha)$ for the replacement of repetition α^* ensures that the output taboo of the first iteration is tabooed in the subsequent iterations [31]. Computing the parallel context of α and β in case $\alpha \parallel \beta$ requires one additional pass for both subprograms because what they potentially bind after substitution adds to the parallel context of the respective other subprogram.

Lemma 13 (Correct output taboo). *Application $\sigma^{U,W}_Z(\alpha)$ of uniform substitution retains input taboo U and correctly adds the bound variables and written channels of program α, i.e., $Z \supseteq U \cup \mathsf{BV}(\sigma^{U,W}_Z(\alpha)) \cup \mathsf{CN}(\sigma^{U,W}_Z(\alpha))$.*

The side condition of $\sigma^{U,W}_Z(a(\!|Y, \bar{z}|\!))$ maintains local abstraction of subprograms (B II) because the replacement cannot bind more than $a(\!|Y, \bar{z}|\!)$, thus cannot bind variables and channels of an abstraction that is independent of $a(\!|Y, \bar{z}|\!)$. This also preserves state-disjointness (well-formedness) of parallel programs.

3.1 Semantic Effect of Uniform Substitution

The key ingredients for proving soundness of uniform substitution are Lemma 16 and 17 below. They prove that the effect of the syntactic transformation applied by uniform substitution can be equally mimicked by semantically modifying the interpretation of function and predicate symbols, and program constants. This adjoint interpretation $\sigma^*_w I$ for interpretation I and state w changes how symbols are interpreted according to their syntactic replacements in the substitution σ.

[6] Extension to vectorial arguments is straightforward.

[7] For $\alpha \parallel \beta$, the restriction is $(\mathsf{V}(\alpha) \cap \mathsf{BV}(\beta)) \cup (\mathsf{V}(\beta) \cap \mathsf{BV}(\alpha)) \subseteq \{\mu, \mu'\} \cup V_{\mathcal{T}}$ [6]. However, in this paper, programs obey a less restrictive syntax for simplicity.

Definition 14 (Adjoint substitution). *For interpretation I and state w, the* adjoint interpretation $\sigma_w^* I$ *changes the meaning of function and predicate symbols, and program constants according to the substitution σ evaluated in state w:*

$$\begin{aligned}
&\sigma_w^* I(f^{\mathbb{M}} : \mathbb{M}_{\text{arg}}) : \mathbb{M}_{\text{arg}} \to \mathbb{M}; d \mapsto I_{\cdot}^{d} w[\![\sigma f(\cdot)]\!] && \text{where } \mathbb{M}, \mathbb{M}_{\text{arg}} \in \{\mathbb{R}, \mathbb{N}, \varOmega, \mathcal{T}\} \\
&\sigma_w^* I(p : \mathbb{M}_{\text{arg}}) = \{d \in \mathbb{M}_{\text{arg}} \mid I_{\cdot}^{d} w \vDash \sigma p(\cdot)\} && \text{where } \mathbb{M}_{\text{arg}} \in \{\mathbb{R}, \mathbb{N}, \varOmega, \mathcal{T}\} \\
&\sigma_w^* I(a(\!|Y, \bar{z}|\!)) = I[\![\sigma a]\!]
\end{aligned}$$

We follow the observation for dGL [31] that the more liberal one-pass substitution requires stronger coincidence between the substitution and the adjoint on neighborhoods of the original state. Where the dGL soundness proof has succeeded by a neighborhood semantics of state on taboos, the dL_{CHP} proof succeeds with a generalization to a neighborhood semantics of state and communication on taboos. The neighborhood of a state consists of its variations:

Definition 15 (Variation). *For a set $U \subseteq V \cup \varOmega$, a state v is a U*-variation of state *w if v and w only differ on variables or projections onto channels in U, i.e., $v \downarrow (U^{\complement} \cap \varOmega) = w \downarrow (U^{\complement} \cap \varOmega)$ on $U^{\complement} \cap V$.*

The proofs of Lemma 16 and 17 follow a lexicographic induction on the structure of substitution, and term, formula, or program. In Lemma 17, the induction is mutual for formulas and programs.

Lemma 16 (Semantic uniform substitution). *The term e evaluates equally over U-variations under uniform substitution σ^U and adjoint interpretation $\sigma_w^* I$, i.e., $Iv[\![\sigma^U(e)]\!] = \sigma_w^* Iv[\![e]\!]$ for all U-variations v of w.*

Lemma 17 (Semantic uniform substitution). *The formula ϕ and the program α have equal truth value and semantics, respectively, over U-variations under uniform substitution σ^U and adjoint interpretation $\sigma_w^* I$, i.e.,*

1. *for all U-variations v of w: $Iv \vDash \sigma^U(\phi)$ iff $\sigma_w^* Iv \vDash \phi$*
2. *for all $(U \cup W)$-variations v of w: $(v, \tau, o) \in I[\![\sigma_Z^{U,W}(\alpha)]\!]$ iff $(v, \tau, o) \in \sigma_w^* I[\![\alpha]\!]$*

3.2 Uniform Substitution Proof Rule

The proof rule US for uniform substitution is the single point of truth for the sound instantiation of axioms (plus renaming of bound variables [29] and written channels, e.g., $[x := \theta]\psi(x)$ to $[y := \theta]\psi(y)$ and $[\text{ch}(h)?x]\psi(\text{ch})$ to $[\text{dh}(h)?x]\psi(\text{dh})$. Soundness of the rule, i.e., that validity of its premise implies validity of the conclusion, immediately follows from Lemma 17. Since the substitution process starts with no taboos, $\sigma(\phi)$ is short for $\sigma^{\emptyset}(\phi)$. If the substitution clashes, i.e., $\sigma^{\emptyset}(\phi)$ is not defined, then rule US is not applicable.

Theorem 18 (US is sound). *The proof rule US is sound.*

$$\frac{\phi}{\sigma(\phi)}\ \text{US}$$

Unlike dL [29] and dGL [31], $\mathsf{dL}_{\mathrm{CHP}}$ has a context-sensitive syntax for programs and formulas (see Def. 2 and Def. 4). By Proposition 19, uniform substitution, however, preserves syntactic well-formedness. Since all axioms in Section 4 will be well-formed, only well-formed formulas can be derived in $\mathsf{dL}_{\mathrm{CHP}}$.

Proposition 19 (US preserves well-formedness). *The result $\sigma^U(\phi)$ (if defined) of applying uniform substitution to a well-formed formula ϕ is well-formed.*

4 Axiomatic Proof Calculus

Figure 3 presents a sound proof calculus for $\mathsf{dL}_{\mathrm{CHP}}$. The significant difference to $\mathsf{dL}_{\mathrm{CHP}}$'s schematic calculus [6] is that it completely abandons soundness-critical side conditions, internalizing them syntactically in the axioms. Only axiom $[]_{\mathrm{W}\mathsf{A}}$ was adjusted to obtain a symbolic representation and an ac-version K_{AC} of modal modus ponens is included. Now, distribution of ac-boxes over conjuncts $[]_{\mathsf{AC}}\wedge$ and ac-monotonicity $\mathrm{M}[\cdot]_{\mathsf{AC}}$ derive from K_{AC},thus are dropped. Except for the small changes soundness is inherited from the schematic axioms [6]. All proofs and supplementary material for this section are in Appendix C.

Algebraic laws for reasoning about traces [6] can be easily adapted to uniform substitution as well. Decidable first-order real arithmetic [41] and Presburger arithmetic [33] have corresponding oracle proof rules [6].

Remark 20. To obtain a truly finite list of axioms from Fig. 3, symbolic (co)finite sets can be finitely axiomatized as a boolean algebra together with extensionality, which can be unrolled to a finite disjunction for (co)finite sets (see Appendix C).

Parallel Composition. The parallel injection axiom $[\|\_]_{\mathsf{AC}}$ in Fig. 3 decomposes parallel CHPs by local abstraction (B II). Unlike $\mathsf{dL}_{\mathrm{CHP}}$'s [6] and Hoare-style [46, 47] schematic calculi for ac-reasoning, axiom $[\|\_]_{\mathsf{AC}}$ internalizes the noninterference property [6, Def. 7] that determines valid instances of formula

$$[\alpha]_{\{\mathsf{A},\mathsf{C}\}}\psi \rightarrow [\alpha \parallel \beta]_{\{\mathsf{A},\mathsf{C}\}}\psi \tag{1}$$

purely syntactically. To focus on noninterference, $a\llparenthesis Y_a, \bar{z}_a\rrparenthesis \parallel_{\mathrm{wf}} b\llparenthesis Y_b, \bar{z}_b\rrparenthesis$ abbreviates well-formed parallel composition $a\llparenthesis Y_a, \bar{z}_a\rrparenthesis \parallel b\llparenthesis Y_b, (\bar{z}_b \cap \bar{z}_a^{\complement}) \cup \{\mu, \mu'\} \cup V_{\mathcal{T}}\rrparenthesis$ using operator $\parallel_{\mathrm{wf}}$ for program constants $a\llparenthesis Y_a, \bar{z}_a\rrparenthesis$, $b\llparenthesis Y_b, \bar{z}_b\rrparenthesis$. This notation ensures disjoint parallel state except for the global time μ, μ' and recorder variables $V_{\mathcal{T}}$.

Intuitively, axiom $[\|\_]_{\mathsf{AC}}$ restricts β in equation (1) such that α overapproximates the behavior of $\alpha \parallel \beta$ influencing A, C, or ψ. For this purpose, noninterference internalized in $b\llparenthesis Y_b \cap (Y^{\complement} \cup Y_a), \bar{z}^{\complement}\rrparenthesis$ forbids b to bind variables $\bar{z}$ that are free in the postcondition $p(Y, \bar{z})$, and $Y^{\complement}$ forbids b to bind channels Y (except for channels Y_a written by a because joint parallel communication can already be observed from a, too). The cut with Y_b allows downscaling of the channels b has to bind. Since parallel programs always agree on the global time μ, μ' and the communication recorded by trace variables $V_{\mathcal{T}}$, the operator $\parallel_{\mathrm{wf}}$ allows their sharing even if $\bar{z}^{\complement}$ disallows it. Note that Y_a and Y, and $\bar{z}_a$ and $\bar{z}$ may overlap.

Despite its asymmetric shape, axiom $[\|\_]_{\mathsf{AC}}$ decomposes $[\alpha\|\beta](\phi\wedge\psi)$ into $[\alpha]\phi$ and $[\beta]\psi$ (if they mutually do not interfere) via independent proofs for $[\alpha\|\beta]\phi$ and $[\alpha\|\beta]\psi$, which drop either α or β by $[\|\_]_{\mathsf{AC}}$ modulo commutativity.

$$\text{[:=]}\quad [x := g^{\mathbb{R}}]p(x) \leftrightarrow p(g^{\mathbb{R}})^a \qquad [;]_{\mathsf{AC}}\quad [a;b]_{\{\mathrm{R},\mathrm{Q}\}}\mathrm{P} \leftrightarrow [a]_{\{\mathrm{R},\mathrm{Q}\}}[b]_{\{\mathrm{R},\mathrm{Q}\}}\mathrm{P}$$

$$\text{[:}{*}\text{]}\quad [x := *]p(x) \leftrightarrow \forall x\, p(x) \qquad [\cup]_{\mathsf{AC}}\quad [a \cup b]_{\{\mathrm{R},\mathrm{Q}\}}\mathrm{P} \leftrightarrow [a]_{\{\mathrm{R},\mathrm{Q}\}}\mathrm{P} \wedge [b]_{\{\mathrm{R},\mathrm{Q}\}}\mathrm{P}$$

$$[?]\quad [?q_{\mathbb{R}}]p \leftrightarrow (q_{\mathbb{R}} \rightarrow p)^a \qquad [^*]_{\mathsf{AC}}\quad [a^*]_{\{\mathrm{R},\mathrm{Q}\}}\mathrm{P} \leftrightarrow [a^0]_{\{\mathrm{R},\mathrm{Q}\}}\mathrm{P} \wedge [a]_{\{\mathrm{R},\mathrm{Q}\}}[a^*]_{\{\mathrm{R},\mathrm{Q}\}}\mathrm{P}^b$$

$$[]_{\top,\top}\quad [a]\mathrm{P} \leftrightarrow [a]_{\{\mathsf{T},\mathsf{T}\}}\mathrm{P} \qquad []_{\mathrm{W}_{\mathsf{A}}}\quad [a]_{\{\mathsf{T},\mathrm{W}_{\mathsf{A}}\}}\mathsf{T} \wedge [a]_{\{\mathrm{R}_1\wedge\mathrm{R}_2,\mathrm{Q}_1\wedge\mathrm{Q}_2\}}\mathrm{P} \rightarrow [a]_{\{\mathrm{R},\mathrm{Q}_1\wedge\mathrm{Q}_2\}}\mathrm{P}^c$$

$$[\|\_]_{\mathsf{AC}}\quad [a(\!|Y_a, \bar{z}_a|\!)]_{\{\mathrm{R},\mathrm{Q}\}}p(Y,\bar{z}) \rightarrow [a(\!|Y_a, \bar{z}_a|\!) \parallel_{\mathrm{wf}} b(\!|Y_b \cap (Y^{\complement} \cup Y_a), \bar{z}^{\complement}|\!)]_{\{\mathrm{R},\mathrm{Q}\}}p(Y,\bar{z})^d$$

$$[\mu]\quad [\{\bar{x}' = g^{\mathbb{R}}(\bar{x},\mu) \,\&\, q_{\mathbb{R}}(\bar{x},\mu)\}]p(\bar{x},\mu) \leftrightarrow [\{\mu' = 1, \bar{x}' = g^{\mathbb{R}}(\bar{x},\mu) \,\&\, q_{\mathbb{R}}(\bar{x},\mu)\}]p(\bar{x},\mu)^a$$

$$[\mathrm{ch}!]\quad [\mathrm{ch}(h)!g^{\mathbb{R}}]p(\mathrm{ch},h) \leftrightarrow \forall h_0\, \big(h_0 = h \cdot \langle \mathrm{ch}, g^{\mathbb{R}}, \mu\rangle \rightarrow p(\mathrm{ch},h_0)\big)$$

$$[\mathrm{ch}!]_{\mathsf{AC}}\quad [\mathrm{ch}(h)!g^{\mathbb{R}}]_{\{\widehat{r},\widehat{q}\}}\widehat{p} \leftrightarrow \widehat{q} \wedge \Big(\widehat{r} \rightarrow [\mathrm{ch}(h)!g^{\mathbb{R}}]\big(\widehat{q} \wedge (\widehat{r} \rightarrow \widehat{p})\big)\Big)$$

$$[\mathrm{ch}?]_{\mathsf{AC}}\quad [\mathrm{ch}(h)?x]_{\{\widehat{r},\widehat{q}\}}p(\mathrm{ch},h,x) \leftrightarrow [x := *][\mathrm{ch}(h)!x]_{\{\widehat{r},\widehat{q}\}}p(\mathrm{ch},h,x)$$

$$[\epsilon]_{\mathsf{AC}}\quad [a(\!|\emptyset, V_{\mathbb{R}}|\!)]_{\{\mathrm{R},\mathrm{Q}\}}\mathrm{P} \leftrightarrow \mathrm{Q} \wedge (\mathrm{R} \rightarrow [a(\!|\emptyset, V_{\mathbb{R}}|\!)]\mathrm{P})$$

$$\mathrm{W}[]_{\mathsf{AC}}\quad [a]_{\{\mathrm{R},\mathrm{Q}\}}\mathrm{P} \leftrightarrow \mathrm{Q} \wedge [a]_{\{\mathrm{R},\mathrm{Q}\}}(\mathrm{Q} \wedge (\mathrm{R} \rightarrow \mathrm{P}))$$

$$\mathrm{I}_{\mathsf{AC}}\quad [a^*]_{\{\mathrm{R},\mathrm{Q}\}}\mathrm{P} \leftrightarrow [a^0]_{\{\mathrm{R},\mathrm{Q}\}}\mathrm{P} \wedge [a^*]_{\{\mathrm{R},\mathsf{T}\}}(\mathrm{P} \rightarrow [a]_{\{\mathrm{R},\mathrm{Q}\}}\mathrm{P})$$

$$\mathrm{K}_{\mathsf{AC}}\quad [a]_{\{\mathrm{R},\mathrm{Q}_1\rightarrow\mathrm{Q}_2\}}(\mathrm{P}_1 \rightarrow \mathrm{P}_2) \rightarrow ([a]_{\{\mathrm{R},\mathrm{Q}_1\}}\mathrm{P}_1 \rightarrow [a]_{\{\mathrm{R},\mathrm{Q}_2\}}\mathrm{P}_2)$$

$$\mathrm{MP}\ \frac{p \rightarrow q \quad p}{q} \qquad \mathrm{G}_{\mathsf{AC}}\ \frac{\mathrm{Q} \wedge \mathrm{P}}{[a]_{\{\mathrm{R},\mathrm{Q}\}}\mathrm{P}} \qquad \forall\ \frac{p(x)}{\forall x\, p(x)} \qquad \mathrm{CE}\ \frac{\mathrm{P}_1 \leftrightarrow \mathrm{P}_2}{C(\mathrm{P}_1) \leftrightarrow C(\mathrm{P}_2)}$$

$\mathrm{P}_j \equiv p_j(Y,\bar{z})$, and $\mathrm{R}_j \equiv r_j(Y,\bar{h})$, and $\mathrm{Q}_j \equiv q_j(Y,\bar{h})$, and $\widehat{\chi} \equiv \chi(\mathrm{ch},h)$, where j may be blank, and $Y \subseteq \Omega$, $\bar{z} \subseteq V_{\mathbb{R}} \cup V_{\mathcal{T}}$, and $\bar{h} \subseteq V_{\mathcal{T}}$ are (co)finite.

[a] Replacements for function symbol $g^{\mathbb{R}}$ and predicate symbol $q_{\mathbb{R}}$ are restricted to polynomials in $V_{\mathbb{R}}$ and first-order real arithmetic, respectively.

[b] Recall that $[\alpha^0]_{\{\mathrm{R},\mathrm{Q}\}}\mathrm{P} \leftrightarrow \mathrm{Q} \wedge (\mathrm{R} \rightarrow \mathrm{P})$ by $[\epsilon]_{\mathsf{AC}}$ and [?] since $\alpha^0 \equiv\ ?\mathsf{T}$.

[c] W_{A} is the compositionality condition $(\mathrm{R} \wedge \mathrm{Q}_1 \rightarrow \mathrm{R}_2) \wedge (\mathrm{R} \wedge \mathrm{Q}_2 \rightarrow \mathrm{R}_1)$.

[d] The operator $\|_{\mathrm{wf}}$ abbreviates well-formed parallel composition (see above).

Fig. 3: $\mathsf{dL}_{\mathrm{CHP}}$ proof calculus

Axiom System. For each program statement, there is either a dynamic or an ac-axiom because the respective other version derives by axiom $[]_{\top,\top}$ or $[\epsilon]_{\mathsf{AC}}$. Axioms [:=], [:∗], and [?] are as in dL [29]. Axioms $[;]_{\mathsf{AC}}$, $[\cup]_{\mathsf{AC}}$, and $[^*]_{\mathsf{AC}}$ for decomposition, and I_{AC} for induction carefully generalize their versions in differential [29] dynamic [13] logic to ac-reasoning. Sending is handled step-wise via flattening the assumption-commitments by axiom $[\mathrm{ch}!]_{\mathsf{AC}}$ and axiom [ch!] that executes the effect onto the recorder h. The duality $[\mathrm{ch}?]_{\mathsf{AC}}$ turns receiving into arbitrary sending, which only synchronizes if it agrees with the parallel context on the value. Usage of axiom $\mathrm{W}[]_{\mathsf{AC}}$ is for convenience. Axiom $[\mu]$ materializes the flow of global time μ such that dL's axiomatization of continuous evolution [29] gets applicable, which requires ODE shape $\bar{x}' = f^{\mathbb{R}}(\bar{x})$. The axiomatic proof rules G_{AC}, MP, ∀, and CE are an ac-version of Gödels generalization rule, modus ponens, quantifier elimination, and contextual equivalence, respectively.

The axiom $[]_{\mathrm{W}_{\mathsf{A}}}$ can weaken assumptions. Its slight change compared to $\mathsf{dL}_{\mathrm{CHP}}$'s schematic calculus [6] exploits that the compositionality condition W_{A}

is only required for a's reachable worlds. Interestingly, dL_{CHP}'s monotonicity rule $\text{M}[\cdot]_{\mathsf{AC}}$ [6] does not derive from modal modus ponens K_{AC} and Gödel generalization G_{AC} in analogy to dL [29] but needs $\text{W}[]_{\mathsf{AC}}$ handling monotonicity of assumptions, which does not fit into G_{AC} because necessitating the assumption in G_{AC} would render the derivation of $[\alpha]_{\{\bot,\mathsf{T}\}}\mathsf{T}$ by G_{AC} impossible.

Axioms using postcondition $\text{P} \equiv p(Y, \bar{z})$, e.g., in $[;]_{\mathsf{AC}}$, allow any replacement of P since accessed channels $Y \subseteq \Omega$ and free variables $\bar{z} \subseteq V_{\mathbb{R}} \cup V_{\mathcal{T}}$ can be arbitrary. Replacements of assumptions $\text{R} \equiv r(Y, \bar{h})$ and commitments $\text{Q} \equiv q(Y, \bar{h})$ can instead only mention trace variables $\bar{h} \subseteq V_{\mathcal{T}}$ bound in their context. This reflects that trace variables are the only interface between the program α and the ac-formulas A and C in an ac-box $[\alpha]_{\{\mathsf{A},\mathsf{C}\}}\psi$ (well-formedness).

Theorem 21 (Soundness). *The proof calculus for dL_{CHP} presented in Fig. 3 is sound as an instantiation of the schematic calculus [6].*

Clashes. Clashes sort out unsound instantiations of axioms. Unlike in dL and dGL [29, 31] whose clashes are solely due to tabooed variables in terms and formulas, clashes in dL_{CHP} can also be due to tabooed channels, and even due to taboos in programs. For example, the substitution $\sigma = \{a \mapsto \text{gh}(h)!1, b \mapsto \text{ch}(h)!2, p \mapsto \psi, r \mapsto \mathsf{T}, q \mapsto \mathsf{T}\}$ with $\psi \equiv |h \downarrow \text{ch}| > 0 \wedge |h \downarrow \text{dh}| > 0 \wedge y < 0$ clashes below, where $Y = \{\text{ch}, \text{dh}\}$, and $\bar{z} \equiv h, y$, and $\text{R} \equiv r(Y)$, and $\text{Q} \equiv q(Y)$. Writing channel ch in the replacement for b would break the local abstraction of a as ch is accessed in ψ but not written in the replacement for a, thus the clash indeed sorts out an unsound instantiation.

$$\frac{[a\llparenthesis\{\text{gh}\}, h\rrparenthesis]_{\{\text{R},\text{Q}\}}p(Y, \bar{z}) \rightarrow [a\llparenthesis\{\text{gh}\}, h\rrparenthesis \parallel_{\text{wf}} b\llparenthesis\{\text{ch}\}\cap(Y^{\complement}\cup\{\text{gh}\}), \bar{z}^{\complement}\rrparenthesis]_{\{\text{R},\text{Q}\}}p(Y, \bar{z})}{[\text{gh}(h)!1]_{\{\mathsf{T},\mathsf{T}\}}\psi \rightarrow [\text{gh}(h)!1 \parallel \text{ch}(h)!2]_{\{\mathsf{T},\mathsf{T}\}}\psi} \ \lightning\,\text{clash}$$

In contrast, $\sigma = \{a \mapsto \text{ch}(h)?x; \text{gh}(h)!1, b \mapsto \text{ch}(h)!2, p \mapsto \psi, r \mapsto \mathsf{T}, q \mapsto \mathsf{T}\}$ does not clash below, where $Y = \{\text{ch}, \text{dh}\}$, and $Y_a = \{\text{ch}, \text{gh}\}$, and other abbreviations are as above, because $\text{ch} \in Y^{\complement} \cup Y_a = \{\text{dh}\}^{\complement}$. Intuitively, the ch-communication of b remains observable after dropping b from the parallel composition as it is joint with a.

$$\frac{\dfrac{*}{[a\llparenthesis Y_a, h, x\rrparenthesis]_{\{\text{R},\text{Q}\}}p(Y, \bar{z}) \rightarrow [a\llparenthesis Y_a, h, x\rrparenthesis \parallel_{\text{wf}} b\llparenthesis\{\text{ch}\} \cap (Y^{\complement} \cup Y_a), \bar{z}^{\complement}\rrparenthesis]_{\{\text{R},\text{Q}\}}p(Y, \bar{z})}\ [\|\_]_{\mathsf{AC}}}{[\text{ch}(h)?x; \text{gh}(h)!1]_{\{\mathsf{T},\mathsf{T}\}}\psi \rightarrow [(\text{ch}(h)?x; \text{gh}(h)!1) \parallel \text{ch}(h)!2]_{\{\mathsf{T},\mathsf{T}\}}\psi}\ \text{US}$$

Also note that by the operator $\parallel_{\text{wf}}$ for well-formed parallel composition, the recorder variable h can be shared without causing a clash above. However, clashes prevent instantiation that would violate syntactic well-formedness of programs (Def. 2) by binding the same state variable in parallel:

$$\frac{[a\llparenthesis\emptyset, x\rrparenthesis]_{\{r,q\}}p(x, y) \rightarrow [a\llparenthesis\emptyset, x\rrparenthesis \parallel_{\text{wf}} b\llparenthesis\emptyset, \{x, y\}^{\complement}\rrparenthesis]_{\{r,q\}}p(x, y)}{[x := y]_{\{\mathsf{T},\mathsf{T}\}}y = x \rightarrow [x := y \parallel x := 0]_{\{\mathsf{T},\mathsf{T}\}}y = x} \ \lightning\,\text{clash}$$

Well-formedness of programs and formulas is ensured in the axioms by well-formed parallel composition $\parallel_{\text{wf}}$ and limitation to trace variables $\bar{h}$ in $\text{R}_j \equiv r_j(Y, \bar{h})$ and $\text{Q}_j \equiv q_j(Y, \bar{h})$ in ac-boxes $[\alpha]_{\{\text{R}_j,\text{Q}_j\}}\psi$ in Fig. 3, respectively. By Proposition 19, uniform substitution always preserves well-formedness.

Example 22. The proof tree below decomposes safety (Example 5) of cruise control (Example 3) into safety ① of controller ct and branch ② to be continued to safety of the vehicle ve. The lower subproof introduces the ac-formulas

$$\mathsf{A} \equiv \mathsf{C} \equiv \big(|h \downarrow \mathrm{tar}| > 0 \to 0 \leq \mathtt{val}(h \downarrow \mathrm{tar}) \leq V\big)$$

using axiom $[]_{\mathrm{W}\mathbf{A}}$ to abstract from the communication between ct and ve. The upper subproof uses the parallel injection axiom $[\|\_]_{\mathbf{AC}}$ to drop ve. Uniform substitution US does not clash as the commitment C only refers to joint communication of ct and ve. Other applications of US (e.g., for $[]_{\mathrm{W}\mathbf{A}}$) are omitted. Rule Prop denotes propositional reasoning. Abbreviations are as follows: $\alpha \equiv a(\!|\mathrm{tar}, v^{\mathrm{tr}}_{\mathsf{ct}}, t, t', \mu, \mu', h|\!)$, $\mathrm{R} \equiv r(\mathrm{tar}, h)$, $\mathrm{Q} \equiv q(\mathrm{tar}, h)$, $\mathrm{P} \equiv p(\mathrm{tar})$.

$$\dfrac{\dfrac{\dfrac{\textcircled{1}}{\varphi \to [\mathsf{ct}^*]_{\{\mathsf{T},\mathsf{C}\}}\mathsf{T}} \qquad \dfrac{\dfrac{*}{[\alpha]_{\{\mathrm{R},\mathrm{Q}\}}\mathrm{P} \to [\alpha \parallel_{\mathrm{wf}} b(\!|\mathrm{tar}, v^{\mathrm{tr}}_{\mathrm{ve}}, a_{\mathrm{ve}}, t_0, v_{\mathrm{ve}}, v'_{\mathrm{ve}}|\!)]_{\{\mathrm{R},\mathrm{Q}\}}\mathrm{P}}\,{\scriptstyle [\|\_]_{\mathbf{AC}}}}{[\mathsf{ct}^*]_{\{\mathsf{T},\mathsf{C}\}}\mathsf{T} \to [\mathsf{ct}^* \| \mathsf{ve}^*]_{\{\mathsf{T},\mathsf{C}\}}\mathsf{T}}\,{\scriptstyle \mathrm{US}}}{\varphi \to [\mathsf{ct}^* \| \mathsf{ve}^*]_{\{\mathsf{T},\mathsf{C}\}}\mathsf{T}}\,{\scriptstyle \mathrm{MP,\ CE}}}{\varphi \to [\mathsf{ct}^* \| \mathsf{ve}^*]_{\{\mathsf{T}\wedge\mathsf{A},\mathsf{C}\}}\mathsf{T}}\,{\scriptstyle \mathrm{M}[\cdot]_{\mathbf{AC}}}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{*}{(\mathsf{C} \to \mathsf{A}) \wedge \mathsf{T}}\,{\scriptstyle \mathrm{Prop}}}{\varphi \to [\mathsf{ct}^* \| \mathsf{ve}^*]_{\{\mathsf{T},\mathsf{C}\to\mathsf{A}\}}\mathsf{T}}\,{\scriptstyle \mathrm{G}_{\mathbf{AC}}} \qquad \dfrac{\dfrac{\longleftarrow \qquad \dfrac{\textcircled{2}}{\varphi \to [\mathsf{ct}^* \| \mathsf{ve}^*]_{\{\mathsf{T}\wedge\mathsf{A},\mathsf{T}\}}\psi_{\mathrm{safe}}}}{\varphi \to [\mathsf{ct}^* \| \mathsf{ve}^*]_{\{\mathsf{T}\wedge\mathsf{A},\mathsf{C}\}}\mathsf{T} \wedge [\mathsf{ct}^* \| \mathsf{ve}^*]_{\{\mathsf{T}\wedge\mathsf{A},\mathsf{T}\}}\psi_{\mathrm{safe}}}\,{\scriptstyle \wedge\mathrm{R}}}{\varphi \to [\mathsf{ct}^* \| \mathsf{ve}^*]_{\{\mathsf{T}\wedge\mathsf{A},\mathsf{C}\wedge\mathsf{T}\}}(\mathsf{T} \wedge \psi_{\mathrm{safe}})}\,{\scriptstyle []_{\mathbf{AC}}\wedge}}{\varphi \to [\mathsf{ct}^* \| \mathsf{ve}^*]_{\{\mathsf{T},\mathsf{C}\to\mathsf{A}\}}\mathsf{T} \wedge [\mathsf{ct}^* \| \mathsf{ve}^*]_{\{\mathsf{T}\wedge\mathsf{A},\mathsf{C}\wedge\mathsf{T}\}}(\mathsf{T} \wedge \psi_{\mathrm{safe}})}\,{\scriptstyle \wedge\mathrm{R}}}{\varphi \to [\mathsf{ct}^* \| \mathsf{ve}^*]_{\{\mathsf{T},\mathsf{C}\wedge\mathsf{T}\}}(\mathsf{T} \wedge \psi_{\mathrm{safe}})}\,{\scriptstyle []_{\mathrm{W}\mathbf{A}}}}{\varphi \to [\mathsf{ct}^* \| \mathsf{ve}^*]\psi_{\mathrm{safe}}}\,{\scriptstyle []_{\top,\top},\ \mathrm{M}[\cdot]_{\mathbf{AC}}}$$

5 Related Work

Uniform substitution for differential dynamic logic dL [29] generalizes Church's uniform substitution for first-order logic [7, §35, 40]. Unlike the lifting from dL to differential game logic dGL [30], $\mathsf{dL}_{\mathrm{CHP}}$ generalizes into the complementary direction of communication and parallelism. Unlike schematic calculi [2, 18, 26, 44, 46], whose treacherous schematic simplicity relies on encoding all subtlety of parallel systems in significant soundness-critical side conditions, our development builds upon a minimalistic non-schematic parallel injection axiom *and* sound instantiation encapsulated in uniform substitution. This provides a new, more atomic and more modular understanding of parallel systems overcoming the root cause for large soundness-critical prover kernels [5, 8, 11, 15, 17, 36]. Usage of uniform substitution reduced the kernel of the theorem prover KeYmaera from 105 kLOC to 2 kLOC in KeYmaera X [22]. We expect $\mathsf{dL}_{\mathrm{CHP}}$'s integration into KeYmaera X to stay in the same order of magnitude.

To the best of our knowledge, assumption-commitment reasoning [21, 46][8] has no tool support, which might be due to vast implementation effort. The latter can

[8] Assumption-commitment and rely-guarantee reasoning are specific patterns for message-passing and shared variables concurrency, respectively. The broader assume-guarantee principle has been used across diverse areas for various purposes.

be underpinned by analogy with tools [5, 8, 15, 17, 36] for verification of shared-variables concurrency, some of which use rely-guarantee reasoning [36,39]. Unlike uniform substitution for dL_{CHP} that enables a straightforward implementation of a small prover kernel, they all rely on large soundness-critical code bases. Unlike refinement checking for CSP [11] and discrete-time CSP [4], dL_{CHP} supports safety properties of dense-time hybrid systems. Contrary to our goal of small prover kernels, implementations of model checkers [11] are inherently large.

Beyond embeddings of concurrency reasoning for discrete systems into proof assistants [3,24,25,38], dL_{CHP} can verify parallel hybrid systems synchronizing in shared global time. The latter imposes even more complicated binding structures than parallel or hybrid systems alone but dL_{CHP}'s uniform substitution calculus continues to manage them in a modular way.

The recent tool HHLPy [37] for hybrid CSP (HCSP) [16] is limited to the sequential fragment. Unlike extending HHLPy to parallelism, which would require extensive soundness-critical side conditions and a treatment of the duration calculus, integrating dL_{CHP} into KeYmaera X [10] boils down to adding a finite list of concrete object level formulas as axioms and only small changes to the uniform substitution process. In contrast to dL_{CHP}'s compositional parallel systems calculus [6], HCSP calculi [12,19,42] are non-compositional [6] as they either unroll exponentially many interleavings from the operational semantics [12, 42] or can only decompose independent parallel components [19] causing limited ability to reason about complex systems. Former HCSP tools [43, 45] only implement a non-compositional calculus [19] reinforcing the significance of our approach for managing parallel hybrid systems reasoning. Other hybrid process algebras defer to model checkers for reasoning [9, 20, 40]. Further discussion of dL_{CHP} is in [6].

6 Conclusion

This paper introduced a sound one-pass uniform substitution calculus for the dynamic logic of communicating hybrid programs dL_{CHP} thereby mastering the significant challenge of developing simple sound proof calculi for parallel hybrid systems with communication. Uniform substitution can separate even notoriously complicated binding structures from parallelism with communication in multi-dynamical logics into axioms and their instantiation. In the case of dL_{CHP}, this applies to channel access in predicates and the need for local abstraction of subprograms in parallel statements, and it even turns out that uniform substitution can maintain a context-sensitive syntax along the way. Thanks to uniform substitution, parallel systems reasoning reduces to multiple uses of an asymmetric parallel injection axiom.

Now, with uniform substitution a straightforward implementation of dL_{CHP} in KeYmaera X is only one step away.

Acknowledgments. This project was funded in part by the Deutsche Forschungs-gemeinschaft (DFG) – 378803395 (ConVeY), an Alexander von Humboldt Professorship, and by the AFOSR under grant number FA9550-16-1-0288.

A Details of the Static Semantics

This appendix reports proofs of the bound effect property and coincidence lemmas given in Section 2.3. Moreover, sound syntactical overapproximations of the static semantics from previous work [6] are given and extended to function and predicate symbols, and program constants.

Proof (of Lemma 10). Let $(v, \tau, w) \in I[\![\alpha]\!]$ with $w \neq \bot$. Then $v = w$ on $V_{\mathcal{T}}$ can be easily proven by induction on α because no program ever changes a trace variable. To prove $v = w \cdot \tau$ on $\mathsf{BV}(\alpha)^{\complement}$, let $z \notin \mathsf{BV}(\alpha)$. Then $(w \cdot \tau)(z) = w(z)$ by definition of $\mathsf{BV}(\cdot)$. To prove $\tau \downarrow \mathsf{CN}(\alpha)^{\complement} = \epsilon$, let $(v, \tau, w) \in I[\![\alpha]\!]$ and $\mathrm{ch} \notin \mathsf{CN}(\alpha)$. Then $\tau \downarrow \{\mathrm{ch}\} = \epsilon$ by definition of $\mathsf{CN}(\cdot)$. Since this holds for all $\mathrm{ch} \in \mathsf{CN}(\alpha)^{\complement}$, we obtain $\tau \downarrow \mathsf{CN}(\alpha)^{\complement} = \epsilon$.

Suppose that $\mathsf{BV}(\alpha)$ and $\mathsf{CN}(\alpha)$ are not the smallest sets with the bound effect property but $X \subseteq V$ and $Y \subseteq \Omega$ with $X \not\supseteq \mathsf{BV}(\alpha)$ or $Y \not\supseteq \mathsf{CN}(\alpha)$ have it, too. Then there is $z \in \mathsf{BV}(\alpha)$ with $z \notin X$ or $\mathrm{ch} \in \mathsf{CN}(\alpha)$ with $\mathrm{ch} \notin Y$. If $z \in \mathsf{BV}(\alpha)$ and $z \notin X$, then I and $(v, \tau, w) \in I[\![\alpha]\!]$ exist such that $v(z) \neq (w \cdot \tau)(z)$. But then X does not have the bound effect property as z changed by α. If $\mathrm{ch} \in \mathsf{CN}(\alpha)$ and $\mathrm{ch} \notin Y$, then I and $(v, \tau, w) \in I[\![\alpha]\!]$ exist such that $\tau \downarrow \{\mathrm{ch}\} \neq \epsilon$. But then $\tau \downarrow Y^{\complement} \neq \epsilon$ such that Y does not have the bound effect property. □

The following lemma prepares the proof of the communication-aware coincidence property (Lemma 11) for terms and formulas:

Lemma 23. *Let $T_Y^{\tau} = \{\tau' \mid \tau' \downarrow Y = \tau \downarrow Y\}$ for $Y \subseteq \Omega$ and $Y_0 = Y \cup \{ch\}$. Then for all $\tau', \tau \in \mathcal{T}$, if $\tau' \in T_Y^{\tau}$, then $\tau'' \in T_{Y_0}^{\tau}$ exists such that $\tau'' \downarrow \{ch\}^{\complement} = \tau' \downarrow \{ch\}^{\complement}$.*

Proof. For $\rho = \langle \mathrm{ch}, a, s \rangle$, we define $\Omega(\rho) = \mathrm{ch}$. Moreover, we identify the item ch with the singleton $\{\mathrm{ch}\}$. Now, the proof is by induction on the structure of τ':

1. $\tau' = \epsilon$, then let $\tau' \in T_Y^{\tau}$. Since $\tau' \downarrow Y = \epsilon$, we obtain $\tau \downarrow Y = \epsilon$. We define $\tau'' = \tau \downarrow \mathrm{ch}$. Now, $\tau'' \downarrow Y_0 = \tau \downarrow (Y_0 \cap \mathrm{ch}) = \tau \downarrow \mathrm{ch}$, which equals $\tau \downarrow Y_0$ because $\tau \downarrow Y = \epsilon$. Hence, $\tau'' \in T_{Y_0}^{\tau}$. Finally, $\tau'' \downarrow \mathrm{ch}^{\complement} = \tau \downarrow (\mathrm{ch} \cap \mathrm{ch}^{\complement}) = \epsilon = \tau' \downarrow \mathrm{ch}^{\complement}$.
2. $\tau' = \rho \cdot \tau_0'$ with $|\rho| = 1$, then let $\tau' \in T_Y^{\tau}$. Hence, $\tau' \downarrow Y = \tau \downarrow Y$.
   If $\Omega(\rho) \in Y$, then $\tau' \downarrow Y = \rho \cdot \tau_0' \downarrow Y$ and $\tau = \tau_1 \cdot \rho \cdot \tau_2$ for some τ_1, τ_2 with $\tau_1 \downarrow Y = \epsilon$. Hence, $\tau_0' \downarrow Y = \tau_2 \downarrow Y$ such that $\tau_0' \in T_Y^{\tau_2}$. By IH, $\tau_0'' \in T_{Y_0}^{\tau_2}$ exists such that $\tau_0'' \downarrow \mathrm{ch}^{\complement} = \tau_0' \downarrow \mathrm{ch}^{\complement}$. We define $\tau'' = \tau_1 \downarrow \mathrm{ch} \cdot \rho \cdot \tau_0''$. Since $\tau_0'' \in T_{Y_0}^{\tau_2}$, we have $\tau_0'' \downarrow Y_0 = \tau_2 \downarrow Y_0$. Moreover, $\tau_1 \downarrow Y = \epsilon$ implies $\tau_1 \downarrow \mathrm{ch} = \tau_1 \downarrow Y_0$. Therefore, $\tau'' \downarrow Y_0 = \tau_1 \downarrow (\mathrm{ch} \cap Y_0) \cdot \rho \downarrow Y_0 \cdot \tau_0'' \downarrow Y_0 = \tau_1 \downarrow Y_0 \cdot \rho \downarrow Y_0 \cdot \tau_2 \downarrow Y_0 = \tau \downarrow Y_0$. Hence, $\tau'' \in T_{Y_0}^{\tau}$. Moreover, $\tau'' \downarrow \mathrm{ch}^{\complement} = \tau_1 \downarrow (\mathrm{ch} \cap \mathrm{ch}^{\complement}) \cdot \rho \downarrow \mathrm{ch}^{\complement} \cdot \tau_0'' \downarrow \mathrm{ch}^{\complement} = \rho \downarrow \mathrm{ch}^{\complement} \cdot \tau_0'' \downarrow \mathrm{ch}^{\complement} = \tau' \downarrow \mathrm{ch}^{\complement}$.
   Otherwise, if $\Omega(\rho) \notin Y$, then $\tau_0' \downarrow Y = \tau' \downarrow Y = \tau \downarrow Y$ such that $\tau_0' \in T_Y^{\tau}$. By IH, $\tau_0'' \in T_{Y_0}^{\tau}$ exists such that $\tau_0'' \downarrow \mathrm{ch}^{\complement} = \tau_0' \downarrow \mathrm{ch}^{\complement}$. Now, we define $\tau'' = \rho \downarrow \mathrm{ch}^{\complement} \cdot \tau_0''$. Since $\tau'' \downarrow Y_0 = \rho \downarrow (\mathrm{ch}^{\complement} \cap Y_0) \cdot \tau_0'' \downarrow Y_0 = \rho \downarrow Y \cdot \tau_0'' \downarrow Y_0 = \tau_0'' \downarrow Y_0$, we have $\tau'' \in T_{Y_0}^{\tau}$. Finally, $\tau'' \downarrow \mathrm{ch}^{\complement} = \rho \downarrow (\mathrm{ch}^{\complement} \cap \mathrm{ch}^{\complement}) \cdot \tau_0'' \downarrow \mathrm{ch}^{\complement} = \rho \downarrow \mathrm{ch}^{\complement} \cdot \tau_0'' \downarrow \mathrm{ch}^{\complement} = \rho \downarrow \mathrm{ch}^{\complement} \cdot \tau_0' \downarrow \mathrm{ch}^{\complement} = \tau' \downarrow \mathrm{ch}^{\complement}$.

□

Proof (of Lemma 11). The proof generalizes the coincidence property proofs of dL [29, Lemma 10] to communication-aware coincidence. Since $I\tilde{v}[\![e]\!] = J\tilde{v}[\![e]\!]$ if $I = J$ on $\Sigma(e)$ by an induction on the structure of e, it suffices to prove that $Iv[\![e]\!] = I\tilde{v}[\![e]\!]$ for all I. Let $S_{X,Y}$ be a set of states between v and $\tilde{v}$ according to variables $X \subseteq V$ and channels $Y \subseteq \Omega$ as follows:

$$S_{X,Y} = \{v' \mid v' \downarrow Y = v \downarrow Y \text{ on } X \text{ and } v' \downarrow Y^{\complement} = \tilde{v} \downarrow Y^{\complement} \text{ on } X^{\complement}\}$$

Fix an interpretation I and prove $Iv'[\![e]\!] = I\tilde{v}[\![e]\!]$ for all $X \subseteq \mathsf{FV}(e)^{\complement}$, and $Y \subseteq \mathsf{CN}(e)^{\complement}$, and $v' \in S_{X,Y}$. Therefore, we increase the sets X and Y starting from $\emptyset$ for both, where v' may differ from $\tilde{v}$, by lexicographic induction on X and Y till we reach $X = \mathsf{FV}(e)^{\complement}$ and $Y = \mathsf{CN}(e)^{\complement}$. This suffices for $Iv[\![e]\!] = I\tilde{v}[\![e]\!]$ because $v \in S_{\mathsf{FV}(e)^{\complement},\mathsf{CN}(e)^{\complement}}$ by the premise that $v \downarrow \mathsf{CN}(e) = \tilde{v} \downarrow \mathsf{CN}(e)$ on $\mathsf{FV}(e)$.

1. $X = \emptyset$ and $Y = \emptyset$, then $S_{X,Y} = \{\tilde{v}\}$ such that $Iv'[\![e]\!] = I\tilde{v}[\![e]\!]$ for all $v' \in S_{X,Y}$ holds trivially.
2. $X = X_0 \cup \{z\}$ with $z \notin X_0$ and $z \notin \mathsf{FV}(e)$, then let $v' \in S_{X,Y}$. We define $v'' = v'^{\tilde{v}(z)}_{z}$. By $v' \in S_{X,Y}$, we obtain $v'' \downarrow Y = v \downarrow Y$ on X_0 since $v' \downarrow Y = v \downarrow Y$ on X, and $v'' \downarrow Y^{\complement} = \tilde{v} \downarrow Y^{\complement}$ on $X^{\complement}$ since $v' \downarrow Y^{\complement} = \tilde{v} \downarrow Y^{\complement}$ on $X^{\complement}$. Moreover, $v''(z) = \tilde{v}(z)$ if $z \in V_{\mathbb{R}} \cup V_{\mathbb{N}}$, and $v''(z) \downarrow Y^{\complement} = \tilde{v}(h) \downarrow Y^{\complement}$ if $z \in V_{\mathcal{T}}$, respectively, by definition of v''. Therefore, $v'' \downarrow Y^{\complement} = \tilde{v} \downarrow Y^{\complement}$ on $X^{\complement} \cup \{z\} = X_0^{\complement}$ such that $v'' \in S_{X_0,Y}$.
   By definition of $\mathsf{FV}(e)$, we obtain $Iv'[\![e]\!] = Iv''[\![e]\!]$ because $v'' = v'$ on $\{z\}^{\complement}$ but $z \notin \mathsf{FV}(e)$. Finally, $Iv'[\![e]\!] = Iv''[\![e]\!] \overset{\text{IH}}{=} I\tilde{v}[\![e]\!]$ by IH using $v'' \in S_{X_0,Y}$.
3. $X = X_0$ and $Y = Y_0 \cup \{\text{ch}\}$ with $\text{ch} \notin Y_0$ and $\text{ch} \notin \mathsf{CN}(e)$, then let $v' \in S_{X,Y}$. Consider $z \in X^{\complement} \cap V_{\mathcal{T}}$. Then $v'(z) \downarrow Y^{\complement} = \tilde{v}(z) \downarrow Y^{\complement}$. Moreover, $Y_0^{\complement} = Y^{\complement} \cup \{\text{ch}\}$. Therefore, by Lemma 23, τ''_z with $\tau''_z \downarrow Y_0^{\complement} = \tilde{v}(z) \downarrow Y_0^{\complement}$ exists such that $\tau''_z \downarrow \{\text{ch}\}^{\complement} = v'(z) \downarrow \{\text{ch}\}^{\complement}$.
   Using one τ''_z for each $z \in X^{\complement} \cap V_{\mathcal{T}}$, we define a state v'' as follows:

   $$v''(z) = \begin{cases} v'(z) & \text{for } z \in V_{\mathbb{R}} \cup V_{\mathbb{N}} \\ v'(z) & \text{for } z \in X \cap V_{\mathcal{T}} \\ \tau''_z & \text{for } z \in X^{\complement} \cap V_{\mathcal{T}} \end{cases}$$

   For $z \in X \cap (V_{\mathbb{R}} \cup V_{\mathbb{N}})$, we have $v''(z) = v'(z) = v(z)$. Moreover, for $z \in X^{\complement} \cap (V_{\mathbb{R}} \cup V_{\mathbb{N}})$, we have $v''(z) = v'(z) = \tilde{v}(z)$. Further, for $z \in X \cap V_{\mathcal{T}}$, we have $v''(z) \downarrow Y_0 = v'(z) \downarrow Y_0 = v(z) \downarrow Y_0$ because $v'(z) \downarrow Y = v(z) \downarrow Y$ as $v' \in S_{X,Y}$. Finally, for $z \in X^{\complement} \cap V_{\mathcal{T}}$, we have $v''(z) \downarrow Y_0^{\complement} = \tau''_z \downarrow Y_0^{\complement} = \tilde{v}(z) \downarrow Y_0^{\complement}$ due to Lemma 23. Therefore, $v'' \in S_{X,Y_0}$ such that $Iv''[\![e]\!] = I\tilde{v}[\![e]\!]$ by IH.
   Observe that $v'' = v'$ on $V_{\mathbb{R}} \cup V_{\mathbb{N}}$. Moreover, for $z \in X \cap V_{\mathcal{T}}$, we have $v''(z) \downarrow \{\text{ch}\}^{\complement} = v'(z) \downarrow \{\text{ch}\}^{\complement}$. Finally, for $z \in X^{\complement} \cap V_{\mathcal{T}}$, we have $v''(z) \downarrow \{\text{ch}\}^{\complement} = \tau''_z \downarrow \{\text{ch}\}^{\complement} = v'(z) \downarrow \{\text{ch}\}^{\complement}$ due to Lemma 23. Overall, $v'' \downarrow \{\text{ch}\}^{\complement} = v' \downarrow \{\text{ch}\}^{\complement}$. Since $\text{ch} \notin \mathsf{CN}(e)$, we obtain $Iv''[\![e]\!] = Iv'[\![e]\!]$ from the definition of $\mathsf{CN}(e)$.
   Finally, $Iv'[\![e]\!] = Iv''[\![e]\!] = I\tilde{v}[\![e]\!]$.

Suppose that $\mathsf{FV}(e)$ and $\mathsf{CN}(e)$ are not the smallest sets with the coincidence property but $X \subseteq V$ and $Y \subseteq \Omega$ with $X \not\supseteq \mathsf{FV}(e)$ or $Y \not\supseteq \mathsf{CN}(e)$ have the coincidence property, too. Then there is $z \in \mathsf{FV}(e)$ with $z \notin X$ or $\mathrm{ch} \in \mathsf{CN}(e)$ with $\mathrm{ch} \notin Y$. If $z \in \mathsf{FV}(e)$ and $z \notin X$, then by definition of $\mathsf{FV}(e)$, states $v, \tilde{v}$ with $v = \tilde{v}$ on $\{z\}^{\complement}$ exist such that $Iv[\![e]\!] \neq I\tilde{v}[\![e]\!]$. But then (X, Y) does not have the coincidence property because $v{\downarrow}Y = \tilde{v}{\downarrow}Y$ on X but $Iv[\![e]\!] \neq I\tilde{v}[\![e]\!]$. If $\mathrm{ch} \in \mathsf{CN}(e)$ and $\mathrm{ch} \notin Y$, then by definition of $\mathsf{CN}(e)$, states $v, \tilde{v}$ with $v \downarrow \{\mathrm{ch}\}^{\complement} = \tilde{v} \downarrow \{\mathrm{ch}\}^{\complement}$ exist such that $Iv[\![e]\!] = I\tilde{v}[\![e]\!]$. But then (X, Y) does not have the coincidence property because $v \downarrow Y = \tilde{v} \downarrow Y$ but $Iv[\![e]\!] \neq I\tilde{v}[\![e]\!]$. □

Proof (of Lemma 12). The proof generalizes the coincidence property proofs of dL [29, Lemma 12] to a coincidence property for CHPs. Let v'_S be the state between v and $\tilde{v}$ according to the variables S, i.e., $v'_S = \tilde{v}$ on S and $v'_S = v$ on $S^{\complement}$. Then we prove by induction on $S \subseteq \mathsf{FV}(\alpha)^{\complement}$ that for all v'_S a computation $(v'_S, \tau', w') \in I[\![\alpha]\!]$ exists such that $w' = w$ on $S^{\complement}$, and $\tau' = \tau$, and ($w' = \bot$ iff $w = \bot$). This suffices to prove the lemma because first, $X^{\complement} \subseteq \mathsf{FV}(\alpha)^{\complement}$ such that $(v'_{X^{\complement}}, \tau', w') \in I[\![\alpha]\!]$, where $v'_{X^{\complement}} = \tilde{v}$ on $X^{\complement}$ and $v'_{X^{\complement}} = v = \tilde{v}$ on $(X^{\complement})^{\complement} = X$, and $\tau' = \tau$, and $w' = w$ on $(X^{\complement})^{\complement} = X$. Second, $I[\![\alpha]\!] = J[\![\alpha]\!]$ by an induction on α. Now, we proceed with the induction on S:

1. $S = \emptyset$, then $v'_S = v$. If we define $\tau' = \tau$ and $w' = w$, then $(v'_S, \tau', w') \in I[\![\alpha]\!]$ and fulfills the conditions.
2. $S = S_0 \cup \{z\}$ with $z \notin S_0$ and $z \notin \mathsf{FV}(\alpha)$, then let v'_S be between v and $\tilde{v}$ according to S. Moreover, let $v'' = (v'_S)_z^{v(z)}$. Since $v'' = \tilde{v}$ on S_0 and $v'' = v$ on $S_0^{\complement}$, we have $v'' = v'_{S_0}$ such that by IH, (v'', τ'', w'') exists with $w'' = w$ on $S_0^{\complement}$, and $\tau'' = \tau$, and ($w'' = \bot$ iff $w = \bot$). Since $v'' = v'_S$ on $\{z\}^{\complement}$ but $z \notin \mathsf{FV}(\alpha)$, there is $(v'_S, \tau', w') \in I[\![\alpha]\!]$ by the definition of $\mathsf{FV}(\alpha)$ such that $\tau' = \tau''$ and $w' = w''$ on $\{z\}^{\complement}$, which includes that ($w' = \bot$ iff $w'' = \bot$). Thus, $w' = w'' = w$ on $\{z\}^{\complement} \cap S_0^{\complement} = S^{\complement}$. Moreover, $\tau' = \tau'' = \tau$ and ($w' = \bot$ iff $w = \bot$).

Suppose that $\mathsf{FV}(\alpha)$ is not the smallest set with the coincidence property but $X \not\supseteq \mathsf{FV}(\alpha)$ has the property, too. Then there is $z \in \mathsf{FV}(\alpha)$ with $z \notin X$. By definition of $\mathsf{FV}(\alpha)$, interpretation I, and $(v, \tau, w) \in I[\![\alpha]\!]$, and $\tilde{v}$ exist such that $v = \tilde{v}$ on $\{z\}^{\complement}$ and $(v, \tau, w) \in I[\![\alpha]\!]$ but there are no $\tilde{\tau}, \tilde{w}$ such that $(\tilde{v}, \tilde{\tau}, \tilde{w}) \in I[\![\alpha]\!]$, and $\tilde{w} = w$ on $\{z\}^{\complement}$, and $\tilde{\tau} = \tau$, and ($\tilde{w} = \bot$ iff $w = \bot$). But then X does not have the coincidence property because $v = \tilde{v}$ on $\{z\}^{\complement} \supseteq X$ but no $\tilde{\tau}$ and $\tilde{w}$ exist such that $(\tilde{v}, \tilde{\tau}, \tilde{w}) \in I[\![\alpha]\!]$, and $\tilde{w} = w$ on X, and $\tilde{\tau} = \tau$, and ($\tilde{w} = \bot$ iff $w = \bot$). □

The static semantics of Def. 9 is not computable [34]. Def. 24–29 adapt sound overapproximations of the static semantics computed from the syntactical structure [6] to $\mathsf{dL}_{\mathrm{CHP}}$. The definitions add the cases for function and predicate symbols, and program constants, which were only introduced in this paper.

Crucially, the bound effect property and the coincidence lemmas apply for overapproximations of the static semantics as well. Thus, the overapproximations can be soundly used in an implementation of uniform substitution.

Definition 24 (Bound variables). *The set of (syntactically)* bound variables $\mathit{BV}(\alpha)$ *of a program* α *is inductively defined as follows, where* $\{\bar{z}\} = \cup_{z \in \bar{z}}\{z\}$*:*

$$\begin{aligned}
\mathit{BV}(a\llparenthesis Y, \bar{z} \rrparenthesis) &= \{\bar{z}\} \\
\mathit{BV}(x := e) = \mathit{BV}(x := *) &= \{x\} \\
\mathit{BV}(\{x' = \theta \,\&\, \chi\}) &= \{x, x', \mu, \mu'\} \\
\mathit{BV}(?\chi) &= \emptyset \\
\mathit{BV}(\mathit{ch}(h)!\theta) &= \{h\} \\
\mathit{BV}(\mathit{ch}(h)?x) &= \{h, x\} \\
\mathit{BV}(\alpha \cup \beta) = \mathit{BV}(\alpha;\beta) = \mathit{BV}(\alpha \parallel \beta) &= \mathit{BV}(\alpha) \cup \mathit{BV}(\beta) \\
\mathit{BV}(\alpha^*) &= \mathit{BV}(\alpha)
\end{aligned}$$

Definition 25 (Written channels). *The set of (syntactically)* written channels $\mathit{CN}(\alpha)$ *of a program* α *is inductively defined as follows:*

$$\begin{aligned}
\mathit{CN}(a\llparenthesis Y, \bar{z} \rrparenthesis) &= Y \\
\mathit{CN}(x := e) = \mathit{CN}(x := *) = \mathit{CN}(\{x' = \theta \,\&\, \chi\}) = \mathit{CN}(?\chi) &= \emptyset \\
\mathit{CN}(\mathit{ch}(h)!\theta) = \mathit{CN}(\mathit{ch}(h)?x) &= \{\mathit{ch}\} \\
\mathit{CN}(\alpha \cup \beta) = \mathit{CN}(\alpha;\beta) = \mathit{CN}(\alpha \parallel \beta) &= \mathit{CN}(\alpha) \cup \mathit{CN}(\beta) \\
\mathit{CN}(\alpha^*) &= \mathit{CN}(\alpha)
\end{aligned}$$

Definition 26 (Parameters of terms). *The sets of (syntactically) free variables* $\mathit{FV}(e)$ *and (syntactically) accessed channels* $\mathit{CN}(e)$ *of a term* e *are inductively defined below, where* $\mathtt{op}$ *is any built-in function symbol of fixed interpretation (see Def. 1), e.g.,* $\cdot + \cdot$*, except for projection* $\cdot \downarrow \cdot$*. Moreover, let* $\mathit{FV}(\bar{e}) = \cup_{e \in \bar{e}} \mathit{FV}(e)$ *and* $\mathit{CN}(\bar{e}) = \cup_{e \in \bar{e}} \mathit{CN}(e)$*.*

$$\begin{array}{rlrl}
\mathit{FV}(f(Y, \bar{e})) = \mathit{FV}(\bar{e}) & & \mathit{CN}(f(Y, \bar{e})) = Y \cap \mathit{CN}(\bar{e}) & \\
\mathit{FV}(z) = \{z\} & \text{for } z \in V & \mathit{CN}(h) = \Omega & \text{for } h \in V_{\mathcal{T}} \\
 & & \mathit{CN}(z) = \emptyset & \text{for } z \notin V_{\mathcal{T}} \\
\mathit{FV}(\mathtt{op}(\bar{e})) = \mathit{FV}(\bar{e}) & & \mathit{CN}(\mathtt{op}(\bar{e})) = \mathit{CN}(\bar{e}) & \\
\mathit{FV}(te \downarrow Y) = \mathit{FV}(te) & & \mathit{CN}(te \downarrow Y) = Y \cap \mathit{CN}(te) &
\end{array}$$

The must-bound variables $\mathit{MBV}(\alpha)$ (Def. 27) are those variables that are bound on all execution paths of a program α. In contrast to $\mathit{BV}(\alpha)$, they can be soundly used [29] in the cases for $\mathit{FV}(\alpha;\beta)$ in Def. 28 and $\mathit{FV}([\alpha]\psi)$ in Def. 29.

Definition 27 (Must-bound variables). *The set of* must-bound variables $\mathit{MBV}(\alpha)$ *of a program* α *is inductively defined as follows:*

$$\begin{aligned}
\mathit{MBV}(a\llparenthesis Y, \bar{z} \rrparenthesis) &= \emptyset \\
\mathit{MBV}(\alpha) &= \mathit{BV}(\alpha) \quad \text{for atomic CHPs } \alpha \text{ except for program constants} \\
\mathit{MBV}(\alpha \cup \beta) &= \mathit{MBV}(\alpha) \cap \mathit{MBV}(\beta) \\
\mathit{MBV}(\alpha;\beta) = \mathit{MBV}(\alpha \parallel \beta) &= \mathit{MBV}(\alpha) \cup \mathit{MBV}(\beta) \\
\mathit{MBV}(\alpha^*) &= \emptyset
\end{aligned}$$

Definition 28 (Free variables of programs). *The set of (syntactically) free variables $FV(\alpha)$ of a program α is inductively defined as follows:*

$$\begin{aligned}
FV(a(\!|Y, \bar{z}|\!)) &= V_{\mathbb{R}} \cup V_{\mathcal{T}} \\
FV(x := \theta) &= FV(\theta) \\
FV(x := *) &= \emptyset \\
FV(?\chi) &= FV(\chi) \\
FV(\{x' = \theta \;\&\; \chi\}) &= \{x, \mu\} \cup FV(\theta) \cup FV(\chi) \\
FV(ch(h)!\theta) &= \{\mu, h\} \cup FV(\theta) \\
FV(ch(h)?x) &= \{\mu, h\} \\
FV(\alpha; \beta) &= FV(\alpha) \cup (FV(\beta) \setminus MBV(\alpha)) \\
FV(\alpha \cup \beta) = FV(\alpha \parallel \beta) &= FV(\alpha) \cup FV(\beta) \\
FV(\alpha^*) &= FV(\alpha)
\end{aligned}$$

Definition 29 (Parameters of formulas). *The sets of (syntactically) free variables $FV(\phi)$ and (syntactically) accessed channels $CN(\phi)$ of a formula ϕ are inductively defined as follows, where $FV(\bar{e}) = \cup_{e \in \bar{e}} FV(e)$ and $CN(\bar{e}) = \cup_{e \in \bar{e}} CN(e)$:*

$$\begin{aligned}
FV(p(Y, \bar{e})) &= FV(\bar{e}) & CN(p(Y, \bar{e})) &= Y \cap CN(\bar{e}) \\
FV(e_1 \sim e_2) &= FV(e_1) \cup FV(e_2) & CN(e_1 \sim e_2) &= CN(e_1) \cup CN(e_2) \\
FV(\neg\varphi) &= FV(\varphi) & CN(\neg\varphi) &= CN(\varphi) \\
FV(\varphi \wedge \psi) &= FV(\varphi) \cup FV(\psi) & CN(\varphi \wedge \psi) &= CN(\varphi) \cup CN(\psi) \\
FV(\forall z\, \varphi) &= FV(\varphi) \setminus \{z\} & CN(\forall z\, \varphi) &= CN(\varphi) \\
FV([\alpha]\psi) &= FV(\alpha) \cup (FV(\psi) \setminus MBV(\alpha)) & CN([\alpha]\psi) &= CN(\psi) \\
FV([\alpha]_{\{\mathsf{A},\mathsf{C}\}}\psi) &= FV([\alpha]\psi) \cup FV(\mathsf{A}) \cup FV(\mathsf{C}) & CN([\alpha]_{\{\mathsf{A},\mathsf{C}\}}\psi) &= CN(\mathsf{A}) \cup CN(\mathsf{C}) \cup CN(\psi)
\end{aligned}$$

B Soundness of Uniform Substitution

This appendix reports the soundness proof of uniform substitution for dL_{CHP} (Theorem 21) and a proof that uniform substitution preserves the syntactic well-formedness of formulas (Proposition 19). Moreover, Theorem 32 given in this section enables the instantiation of axiomatic proof rules by uniform substitution.

Proof (of Lemma 13). The proof is by induction on the structure of program α and generalizes the corresponding proof for dGL [31, Lemma 13], where U_0 is short for $U \cup W$ and $\mathsf{BP}(\cdot) = \mathsf{BV}(\cdot) \cup \mathsf{CN}(\cdot)$ denotes all bound parameters:

1. $\alpha \equiv a(\!|Y, \bar{z}|\!)$, then $Z = U \cup \mathsf{BP}(\sigma a) = U \cup \mathsf{BP}(\sigma_Z^{U,W}(a(\!|Y, \bar{z}|\!)))$.
2. $\alpha \equiv x := \theta$, then $Z = U \cup \{x\}$ and $\{x\} \supseteq \mathsf{BV}(x := \sigma^{U_0}(\theta)) = \mathsf{BV}(\sigma_Z^{U,W}(\alpha))$. Moreover, $\emptyset = \mathsf{CN}(\sigma_Z^{U,W}(\alpha))$. Hence, $Z \supseteq U \cup \mathsf{BP}(\sigma_Z^{U,W}(\alpha))$.
3. $\alpha \equiv x := *$, then $Z = U \cup \{x\}$ and $\{x\} = \mathsf{BV}(\alpha) = \mathsf{BV}(\sigma_Z^{U,W}(\alpha))$. Moreover, $\emptyset = \mathsf{CN}(\sigma_Z^{U,W}(\alpha))$. Hence, $Z \supseteq U \cup \mathsf{BP}(\sigma_Z^{U,W}(\alpha))$.

4. $\alpha \equiv \{x' = \theta \,\&\, \chi\}$, then $Z = U \cup \{x, x', \mu, \mu'\}$. Moreover, $\{x, x', \mu, \mu'\} \supseteq \mathsf{BV}(\{x' = \sigma^{U_0}(\theta) \,\&\, \sigma^{U_0}(\chi)\}) = \mathsf{BV}(\sigma_Z^{U,W}(\alpha))$ and $\emptyset = \mathsf{CN}(\sigma_Z^{U,W}(\alpha))$. Thus, $Z \supseteq U \cup \mathsf{BP}(\sigma_Z^{U,W}(\alpha))$.
5. $\alpha \equiv ?\chi$, then $Z = U$ and $\emptyset = \mathsf{BV}(?\sigma^{U_0}(\chi)) = \mathsf{BV}(\sigma_Z^{U,W}(?\chi))$. Moreover, $\emptyset = \mathsf{CN}(\sigma_Z^{U,W}(?\chi))$. Thus, $Z \supseteq U \cup \mathsf{BP}(\sigma_Z^{U,W}(?\chi))$.
6. $\alpha \equiv \beta; \gamma$, then $\sigma_Z^{U,W}(\beta;\gamma) \equiv \sigma_{Z_0}^{U,W}(\beta); \sigma_Z^{Z_0,W}(\gamma)$. By IH, $Z_0 \overset{\text{IH}}{\supseteq} U \cup \mathsf{BP}(\sigma_{Z_0}^{U,W}(\beta))$ and $Z \overset{\text{IH}}{\supseteq} Z_0 \cup \mathsf{BP}(\sigma_Z^{Z_0,W}(\gamma))$. Thus, $Z \supseteq Z_0 \cup \mathsf{BP}(\sigma_Z^{Z_0,W}(\gamma)) \supseteq U \cup \mathsf{BP}(\sigma_{Z_0}^{U,W}(\beta)) \cup \mathsf{BP}(\sigma_Z^{Z_0,W}(\gamma)) \supseteq U \cup \mathsf{BP}(\sigma_{Z_0}^{U,W}(\beta); \sigma_Z^{Z_0,W}(\gamma)) \supseteq U \cup \mathsf{BP}(\sigma_Z^{U,W}(\beta;\gamma))$.
7. $\alpha \equiv \beta \cup \gamma$, then $\sigma_Z^{U,W}(\beta \cup \gamma) \equiv \sigma_{Z_1}^{U,W}(\beta) \cup \sigma_{Z_2}^{U,W}(\gamma)$ with $Z = Z_1 \cup Z_2$. Using IH, $Z = Z_1 \cup Z_2 \overset{\text{IH}}{\supseteq} U \cup \mathsf{BP}(\sigma_{Z_1}^{U,W}(\beta)) \cup \mathsf{BP}(\sigma_{Z_2}^{U,W}(\gamma))$, which equals $U \cup \mathsf{BP}(\sigma_{Z_1}^{U,W}(\beta) \cup \sigma_{Z_2}^{U,W}(\gamma)) = U \cup \mathsf{BP}(\sigma_Z^{U,W}(\beta \cup \gamma))$.
8. $\alpha \equiv \beta^*$, then by IH, $Z \supseteq U \cup \mathsf{BP}(\sigma_Z^{U,W}(\beta))$, i.e., the input taboo U is retained for β. Moreover, by IH, $Z \supseteq \mathsf{BP}(\sigma_Z^{Z,W}(\beta))$ the taboo set Z after one iteration is retained for β. Since $\mathsf{BV}(\beta) \supseteq \mathsf{BV}(\beta^*)$ and $\mathsf{CN}(\beta) \supseteq \mathsf{CN}(\beta^*)$, we obtain $Z \supseteq U \cup \mathsf{BP}(\sigma_Z^{Z,W}(\beta)) = U \cup \mathsf{BP}((\sigma_Z^{Z,W}(\beta))^*) = Z \cup \mathsf{BP}(\sigma_Z^{U,W}(\beta^*))$.
9. $\alpha \equiv \mathrm{ch}(h)!\theta$, then $Z = U \cup \{\mathrm{ch}, h\}$ and $\{\mathrm{ch}\} = \mathsf{CN}(\mathrm{ch}(h)!\sigma^{U_0}(\theta))$, which equals $\mathsf{CN}(\sigma_Z^{U,W}(\mathrm{ch}(h)!\theta))$ and $\{h\} = \mathsf{BV}(\mathrm{ch}(h)!\sigma^{U_0}(\theta)) = \mathsf{BV}(\sigma_Z^{U,W}(\mathrm{ch}(h)!\theta))$. Thus, $Z \supseteq U \cup \mathsf{BP}(\sigma_Z^{U,W}(\mathrm{ch}(h)!\theta))$.
10. $\alpha \equiv \mathrm{ch}(h)?x$, then $Z = U \cup \{\mathrm{ch}, h, x\}$. Now, observe $\{h, x\} \supseteq \mathsf{BV}(\mathrm{ch}(h)?x) = \mathsf{BV}(\sigma_Z^{U,W}(\mathrm{ch}(h)?x))$. Moreover, $\{\mathrm{ch}\} = \mathsf{CN}(\mathrm{ch}(h)?x) = \mathsf{CN}(\sigma_Z^{U,W}(\mathrm{ch}(h)?x))$. Thus, $Z \supseteq U \cup \mathsf{BP}(\sigma_Z^{U,W}(\mathrm{ch}(h)?x))$.
11. $\alpha \equiv \beta \parallel \gamma$, then $\sigma_Z^{U,W}(\beta \parallel \gamma) \equiv \sigma_{Z_1}^{U,W_{U,\gamma}}(\beta) \parallel \sigma_{Z_1}^{U,W_{U,\beta}}(\gamma)$ with $Z = Z_1 \cup Z_2$. Using IH, $Z = Z_1 \cup Z_2 \overset{\text{IH}}{\supseteq} U \cup \mathsf{BP}(\sigma_{Z_1}^{U,W_{U,\gamma}}(\beta)) \cup \mathsf{BP}(\sigma_{Z_1}^{U,W_{U,\beta}}(\gamma)) \supseteq U \cup \mathsf{BP}(\sigma_Z^{U,W_{U,\gamma}}(\beta) \parallel \sigma_Z^{U,W_{U,\beta}}(\gamma)) = U \cup \mathsf{BP}(\sigma_Z^{U,W}(\beta \parallel \gamma))$. □

Proof (of Lemma 16). The proof generalizes the substitution lemma proof for dGL [31, Lemma 15] to multi-sorted terms and taboos with channels. The proof is by induction along the lexicographical order $\sqsubset$ of substitution-term tuples (σ, e) defined by $(\sigma', e') \sqsubset (\sigma, e)$ if $\sigma' \sqsubset \sigma$ or ($\sigma' = \sigma$ but $e' \sqsubset e$), where $\sqsubset$ on substitutions and terms, respectively, denotes the structural order, simultaneously for all U, v, and w. In the following, let v be any U-variation of w, i.e., $v \downarrow Y = w \downarrow Y$ on $U^{\mathsf{C}} \cap V$ with $Y = U^{\mathsf{C}} \cap \Omega$:

1. For $z \in V$, simply $Iv[\![\sigma^U(z)]\!] = Iv[\![z]\!] = v(z) = \sigma_w^* Iv[\![z]\!]$.
2. In case $f(Y, e)$, let $d = Iv[\![\sigma^U(e \downarrow Y)]\!]$. Then by IH, $Iv[\![\sigma^U(f(e \downarrow Y))]\!] = Iv[\![\{\cdot \mapsto \sigma^U(e \downarrow Y)\}^\emptyset(\sigma f(\cdot))]\!] \overset{\text{IH}}{=} I_{\cdot}^d v[\![\sigma f(\cdot)]\!]$ because $\cdot$ has arity 0 and f has arity 1 such that $\{\cdot \mapsto \sigma^U(e \downarrow Y)\}^\emptyset \sqsubset \sigma$. Since $\sigma^U(f(e \downarrow Y))$ is defined, $\mathsf{FV}(\sigma f(\cdot)) \cap U = \emptyset$ and $\mathsf{CN}(\sigma f(\cdot)) \cap U = \emptyset$. By premise, v is a U-variation of w, i.e., $v \downarrow (U^{\mathsf{C}} \cap \Omega) = w \downarrow (U^{\mathsf{C}} \cap \Omega)$ on $U^{\mathsf{C}} \cap V$. Thus, $v \downarrow \mathsf{CN}(\sigma f(\cdot)) = w \downarrow \mathsf{CN}(\sigma f(\cdot))$ on $\mathsf{FV}(\sigma f(\cdot))$ such that $I_{\cdot}^d v[\![\sigma f(\cdot)]\!] = I_{\cdot}^d w[\![\sigma f(\cdot)]\!]$ by coincidence (see Lemma 11). Finally, by Def. 14, $I_{\cdot}^d w[\![\sigma f(\cdot)]\!] = (\sigma_w^* I(f))(d)$, which equals by IH, $(\sigma_w^* I(f))(\sigma_w^* Iv[\![e \downarrow Y]\!]) = \sigma_w^* Iv[\![f(e \downarrow Y)]\!]$ because $e \downarrow Y \sqsubset f(e \downarrow Y)$.

3. Let $\mathsf{op}(e_1, \dots, e_k)$ be a concrete function. For $1 \leq i \leq k$, by IH, $Iv\llbracket\sigma^U(e_i)\rrbracket = \sigma^*_w Iv\llbracket e_i\rrbracket$. Finally, $Iv\llbracket\sigma^U(\mathsf{op}(e_1, \dots, e_k))\rrbracket = Iv\llbracket\mathsf{op}(\sigma^U(e_1), \dots, \sigma^U(e_k))\rrbracket = \sigma^*_w Iv\llbracket\mathsf{op}(e_1, \dots, e_k)\rrbracket$ because $Iv\llbracket\mathsf{op}(e_1, \dots, e_k)\rrbracket$ is a function of $Iv\llbracket e_i\rrbracket$ for $1 \leq i \leq k$.
4. By IH, $Iv\llbracket\sigma^{V\cup\Omega}(\theta)\rrbracket = \sigma^*_w I\llbracket\theta\rrbracket v$ for all v, w since v is a $(V \cup \Omega)$-variation of any state. Hence, $Iv\llbracket\sigma^U((\theta)')\rrbracket = Iv\llbracket(\sigma^{V\cup\Omega}(\theta))'\rrbracket = \sum_x v(x')\frac{\partial Iv\llbracket\sigma^{V\cup\Omega}(\theta)\rrbracket}{\partial x} \overset{\text{IH}}{=} \sum_x v(x')\frac{\partial \sigma^*_w Iv\llbracket\theta\rrbracket}{\partial x} = \sigma^*_w Iv\llbracket(\theta)'\rrbracket$. □

Lemma 30. *Let $(v, \tau, o) \in \sigma^*_w I\llbracket\alpha\rrbracket$, the uniform substitution $\sigma^{U,\emptyset}_Z(\alpha)$ be defined, and v be a U-variation of w. Then $o \cdot \tau$ is a Z-variation of w.*

Proof. Since v is a U-variation of w (Def. 15), we have

$$v \downarrow Y = w \downarrow Y \text{ on } U^\complement \cap V \text{ with } Y = U^\complement \cap \Omega, \tag{2}$$

and by the bound effect property (Lemma 10), we obtain

$$v = w \text{ on } \mathsf{BV}(\sigma^{U,\emptyset}_Z(\alpha))^\complement \cup V_\mathcal{T} \text{ and } \tau \downarrow \mathsf{CN}(\sigma^{U,\emptyset}_Z(\alpha))^\complement = \epsilon. \tag{3}$$

If only $o \cdot \tau$ is a U_0-variation of w, where $U_0 = U \cup \mathsf{BV}(\sigma^{U,\emptyset}_Z(\alpha)) \cup \mathsf{CN}(\sigma^{U,\emptyset}_Z(\alpha))$, then $o \cdot \tau$ is a Z-variation of w by Lemma 13. To prove $o \cdot \tau$ is a U_0-variation of w, we handle the variable restriction $U_0^\complement \cap V$ separately for $V_\mathbb{R} \cup V_\mathbb{N}$ and $V_\mathcal{T}$: In case $V_\mathbb{R} \cup V_\mathbb{N}$, by equation (2), $v = w$ on $U^\complement \cap (V_\mathbb{R} \cup V_\mathbb{N})$, and by equation (3), $v = o \cdot \tau$ on $\mathsf{BV}(\sigma^{U,\emptyset}_Z(\alpha))^\complement \cap (V_\mathbb{R} \cup V_\mathbb{N})$ such that $o \cdot \tau = v = w$ on $\mathsf{BV}(\sigma^{U,\emptyset}_Z(\alpha))^\complement \cap U^\complement \cap (V_\mathbb{R} \cup V_\mathbb{N})$. Moreover, observe that $\mathsf{BV}(\sigma^{U,\emptyset}_Z(\alpha))^\complement \cap U^\complement \cap (V_\mathbb{R} \cup V_\mathbb{N}) = U_0^\complement \cap (V_\mathbb{R} \cup V_\mathbb{N})$ since $\mathsf{CN}(\sigma^{U,\emptyset}_Z(\alpha)) \cap (V_\mathbb{R} \cup V_\mathbb{N}) = \emptyset$. In case $V_\mathcal{T}$, let $h \in U_0^\complement \cap V_\mathcal{T} = \mathsf{BV}(\sigma^{U,\emptyset}_Z(\alpha))^\complement \cap U^\complement \cap V_\mathcal{T}$ and $Y_0 = U_0^\complement \cap \Omega = \mathsf{CN}(\sigma^{U,\emptyset}_Z(\alpha))^\complement \cap Y$. Then we have

$$(o \cdot \tau)(h) \downarrow Y_0 = o(h) \downarrow Y_0 \cdot \tau(h) \downarrow Y_0 \overset{(3)}{=} o(h) \downarrow Y_0 \overset{(3)}{=} v(h) \downarrow Y_0 \overset{(2)}{=} o(h) \downarrow Y_0$$

because $\mathsf{CN}(\sigma^{U,\emptyset}_Z(\alpha)) \subseteq Y_0$ and $Y \subseteq Y_0$. □

By mentioning Y, the program constant $a(\!|Y, \bar{z}|\!)$ signals that a synchronizes along all the channels Y.[9] Synchronization forces the parallel context to agree with the local program on the communication along the synchronized channels. Uniform substitution must preserve synchronization as otherwise the parallel context could unsoundly perform additional communication.

For example, the substitution $\sigma = \{a \mapsto ?\mathrm{T}, b \mapsto \mathrm{ch}(h)!\theta\}$ would turn the valid formula

$$[a(\!|\mathrm{ch}, \bar{z}|\!)]|h \downarrow \mathrm{ch}| = 1 \rightarrow [a(\!|\mathrm{ch}, \bar{z}|\!) \parallel b(\!|\mathrm{ch}, \bar{z}^\complement|\!)]|h \downarrow \mathrm{ch}| = 1$$

[9] Synchronization must not be confused with actually reading or writing all the channels since a program can simply not communicate along a channel, e.g., $\mathsf{skip} \cup \mathrm{ch}(h)!\theta$ synchronizes on ch but not all runs communicate along ch.

into the invalid formula

$$[?\mathsf{T}]|h \downarrow \mathrm{ch}| = 1 \rightarrow [?\mathsf{T} \parallel \mathrm{ch}(h)!\theta]|h \downarrow \mathrm{ch}| = 1$$

stating that if the initial history contains one ch-communication ($[?\mathsf{T}]|h{\downarrow}\mathrm{ch}| = 1$), there is still only one after sending along ch by $\mathrm{ch}(h)!\theta$. The problem is that the replacement for a no longer forces the replacement for b to synchronize along ch. Where b could only communicate along ch if it agreed on this communication with a, the replacement for b unsoundly can perform additional communication independent of the replacement for a.

Uniform substitution (see Fig. 2) preserves synchronization by the side condition $\mathbb{CN}(\sigma a) = Y$ for replacing program constants $a(\!|Y, \bar{z}|\!)$. The standard intuition for uniform substitution would suggest that $\mathbb{CN}(\sigma a) \subseteq Y$ suffices since this already prevents the local replacement of a to unsoundly bind accessed channels.

The additional inclusion $\mathbb{CN}(\sigma a) \supseteq Y$ ensures that uniform substitution $\sigma(\alpha)$ for program α preserves the synchronization along the channels $\mathit{CN}(\alpha)$ where α synchronizes on, i.e., $\mathit{CN}(\alpha) \subseteq \mathit{CN}(\sigma(\alpha))$ with syntactical channels $\mathit{CN}(\cdot)$ (see Appendix A). Since the upper bound $\mathbb{CN}(\sigma\alpha) \subseteq Y$ is defined semantically, the substitution $\sigma(\alpha)$ might introduce extra synchronization on the channels $C_\alpha = \mathit{CN}(\sigma(\alpha)) \setminus \mathit{CN}(\alpha)$. However, this is harmless as $\sigma(\alpha)$ still performs no actual communication on C_α, i.e., $C_\alpha \cap \mathbb{CN}(\sigma(\alpha)) = \emptyset$. Hence, potential extra synchronization in $\sigma(\alpha)$ at most leads to less behavior when parallel programs do not agree with empty communication along C_α. In summary, $\mathit{CN}(\alpha)$ is bound as follows: $\mathbb{CN}(\sigma(\alpha)) \subseteq \mathit{CN}(\alpha) \subseteq \mathit{CN}(\sigma(\alpha))$. Lemma 31 heavily relies on these bounds and is crucially used later in proving Lemma 17 about uniform substitution.

Lemma 31 (Uniform substitution preserves synchronization). *Let α and β be programs, the substitutions $\sigma(\gamma) = \sigma_Z^{U,W}(\gamma)$ for $\gamma \in \{\alpha, \beta\}$ be defined, and I an interpretation. Then for $\gamma \in \{\alpha, \beta\}$, the following are equivalent:*

1. $(v, \tau \downarrow \sigma(\gamma), o_\gamma) \in I[\![\sigma(\gamma)]\!]$ *for* $\gamma \in \{\alpha, \beta\}$ *and* $\tau \downarrow (\sigma(\alpha) \parallel \sigma(\beta)) = \tau$
2. $(v, \tau \downarrow \gamma, o_\gamma) \in I[\![\sigma(\gamma)]\!]$ *for* $\gamma \in \{\alpha, \beta\}$ *and* $\tau \downarrow (\alpha \parallel \beta) = \tau$

Proof. For $\gamma \in \{\alpha, \beta\}$, the following set inclusions can be proven by induction on the structure of γ:

$$\mathbb{CN}(\sigma(\gamma)) \subseteq \mathbb{CN}(\gamma) \subseteq \mathit{CN}(\gamma) \subseteq \mathit{CN}(\sigma(\gamma)) \tag{4}$$

First, $\mathbb{CN}(\sigma(\gamma)) \subseteq \mathbb{CN}(\gamma)$ because locally each replacement σa of a program constant $a(\!|Y, \bar{z}|\!)$ cannot write more channels than $a(\!|Y, \bar{z}|\!)$ due to the side condition $\mathbb{CN}(\sigma a) \subseteq Y$ in Fig. 2. The inclusion $\mathbb{CN}(\gamma) \subseteq \mathit{CN}(\gamma)$ holds as $\mathit{CN}(\gamma)$ is a sound overapproximation of $\mathbb{CN}(\gamma)$ (see Appendix A). In the induction for $\mathit{CN}(\gamma) \subseteq \mathit{CN}(\sigma(\gamma))$, the case $\mathit{CN}(a(\!|Y, \bar{z}|\!)) = Y \subseteq \mathbb{CN}(\sigma a) \subseteq \mathit{CN}(\sigma a)$ if $\gamma \equiv a(\!|Y, \bar{z}|\!)$ uses $\mathbb{CN}(\sigma a) \supseteq Y$ from Fig. 2 again.

First, we prove that item 1 implies item 2. Therefore, let $C_\gamma = \mathit{CN}(\sigma(\gamma)) \setminus \mathit{CN}(\gamma)$ be the extra channels introduced by substitution. Now, let $(v, \tau{\downarrow}\sigma(\gamma), o_\gamma) \in I[\![\sigma(\gamma)]\!]$ for $\gamma \in \{\alpha, \beta\}$. Then τ contains no communication along C_γ, i.e.,

$\tau \downarrow C_\gamma = \epsilon$, since $C_\gamma \cap \mathsf{CN}(\sigma(\gamma)) = \emptyset$ by equation (4). Hence, $\tau \downarrow \sigma(\gamma) = \tau \downarrow \gamma$ such that $(v, \tau\downarrow\gamma, o_\gamma) \in I[\![\sigma(\gamma)]\!]$. Moreover, let $\tau\downarrow(\sigma(\alpha) \parallel \sigma(\beta)) = \tau$. Intuitively, τ has no extra communication outside $\mathit{CN}(\alpha) \cup \mathit{CN}(\beta)$ because $\tau \downarrow (\sigma(\alpha) \parallel \sigma(\beta)) = \tau$ and in τ there is no communication on the difference $(\mathit{CN}(\sigma(\alpha)) \cup \mathit{CN}(\sigma(\beta))) \setminus (\mathit{CN}(\alpha) \cup \mathit{CN}(\beta))$. Formally, since $\tau \downarrow C_\gamma = \epsilon$ for $\gamma \in \{\alpha, \beta\}$, the trace τ does not contain communication on $(\mathit{CN}(\sigma(\alpha)) \cup \mathit{CN}(\sigma(\beta))) \setminus (\mathit{CN}(\alpha) \cup \mathit{CN}(\beta)) \subseteq C_\alpha \cup C_\beta$, which justifies the equality $(\star)$ in the following:

$$\tau \downarrow (\alpha \parallel \beta) = (\tau \downarrow (\sigma(\alpha) \parallel \sigma(\beta))) \downarrow (\alpha \parallel \beta) \overset{(\star)}{=} \tau \downarrow (\sigma(\alpha) \parallel \sigma(\beta)) = \tau.$$

Conversely, let $(v, \tau \downarrow \gamma, o_\gamma) \in I[\![\sigma(\gamma)]\!]$ for $\gamma \in \{\alpha, \beta\}$ and $\tau \downarrow (\alpha \parallel \beta) = \tau$. Moreover, let $\overline{\alpha} = \beta$ and $\overline{\beta} = \alpha$. Then

$$\tau \downarrow \sigma(\gamma) = (\tau \downarrow (\alpha \parallel \beta)) \downarrow \sigma(\gamma) = \tau \downarrow \Big(\big(\mathit{CN}(\alpha) \cup \mathit{CN}(\beta) \big) \cap \mathit{CN}(\sigma(\gamma)) \Big),$$

which equals $\tau \downarrow (\mathit{CN}(\gamma) \cup (\mathit{CN}(\overline{\gamma}) \cap \mathit{CN}(\sigma(\gamma))))$ because $\mathit{CN}(\gamma) \subseteq \mathit{CN}(\sigma(\gamma))$ by equation (4). Since there is no communication along $C_\gamma = \mathit{CN}(\sigma(\gamma)) \setminus \mathit{CN}(\gamma)$ by $C_\gamma \cap \mathsf{CN}(\sigma(\gamma)) = \emptyset$ again, there is also non along $(\mathit{CN}(\overline{\gamma}) \cap \mathit{CN}(\sigma(\gamma))) \setminus \mathit{CN}(\gamma) \subseteq C_\gamma$. Hence, $\tau \downarrow (\mathit{CN}(\gamma) \cup (\mathit{CN}(\overline{\gamma}) \cap \mathit{CN}(\sigma(\gamma)))) = \tau \downarrow \gamma$ such that $\tau \downarrow \sigma(\gamma) = \tau \downarrow \gamma$, which implies $(v, \tau \downarrow \sigma(\gamma), o_\gamma) \in I[\![\sigma(\gamma)]\!]$. Moreover, $\tau \downarrow (\alpha \parallel \beta) = \tau$ implies $\tau \downarrow (\sigma(\alpha) \parallel \sigma(\beta)) = (\tau \downarrow (\alpha \parallel \beta)) \downarrow (\sigma(\alpha) \parallel \sigma(\beta))$, which equals $\tau \downarrow (\alpha \parallel \beta) = \tau$ since $\mathit{CN}(\sigma(\alpha) \parallel \sigma(\beta)) \supseteq \mathit{CN}(\alpha \parallel \beta)$. □

Proof (of Lemma 17). The proof generalizes the substitution lemma proof for dGL [31, Lemma 16 and Lemma 17] to $\mathsf{dL}_{\mathrm{CHP}}$, where taboos contain channels and the parallel context needs to be respected. The proof is by lexicographic mutual structural induction, i.e., along the lexicographic order $\sqsubset$ on tuples (σ, π) of substitutions σ and formula-program expressions π defined by $(\sigma', \pi') \sqsubset (\sigma, \pi)$ if $\sigma' \sqsubset \sigma$ or ($\sigma' = \sigma$ but $\pi' \sqsubset \pi$), where $\sqsubset$ on substitutions and formula-program expressions, respectively, denotes the (mutual) structural order, simultaneously for all U, v, and w.

First, consider the formula cases. Therefore, let v be any U-variation of w.

1. $Iv \vDash \sigma^U(e_1 \sim e_2)$ iff $Iv \vDash \sigma^U(e_1) \sim \sigma^U(e_2)$ iff $Iv[\![\sigma^U(e_1)]\!] \sim Iv[\![\sigma^U(e_2)]\!]$, which is by Lemma 16 equivalent to $\sigma^*_w Iv[\![e_1]\!] \sim \sigma^*_w Iv[\![e_2]\!]$, iff $\sigma^*_w Iv \vDash e_1 \sim e_2$
2. In case $p(Y, e)$, let $d = Iv[\![\sigma^U(e \downarrow Y)]\!]$. Then $Iv \vDash \sigma^U(p(e \downarrow Y))$ iff $Iv \vDash \{\cdot \mapsto \sigma^U(e \downarrow Y)\}^\emptyset(\sigma p(\cdot))$, by IH, iff $I^d_{\cdot} v \vDash \sigma p(\cdot)$ because $\cdot$ has arity 0 and p has arity 1 such that $\{\cdot \mapsto \sigma^U(e \downarrow Y)\}^\emptyset \sqsubset \sigma$. Since $\sigma^U(p(e \downarrow Y))$ is defined, $\mathsf{FV}(\sigma p(\cdot)) \cap U = \emptyset$ and $\mathsf{CN}(\sigma p(\cdot)) \cap U = \emptyset$. By premise, v is a U-variation of w, i.e., $v \downarrow (U^{\complement} \cap \Omega) = w \downarrow (U^{\complement} \cap \Omega)$ on $U^{\complement} \cap V$. Thus, $v \downarrow \mathsf{CN}(\sigma p(\cdot)) = w \downarrow \mathsf{CN}(\sigma p(\cdot))$ on $\mathsf{FV}(\sigma p(\cdot))$ such that ($I^d_{\cdot} v \vDash \sigma p(\cdot)$ iff $I^d_{\cdot} w \vDash \sigma p(\cdot)$) by coincidence (see Lemma 11). Finally, $I^d_{\cdot} w \vDash \sigma p(\cdot)$, by Def. 14, iff $(d) \in \sigma^*_w I(p)$, by IH, iff $(\sigma^*_w Iv[\![e \downarrow Y]\!]) \in \sigma^*_w I(p)$ iff $\sigma^*_w Iv \vDash p(e \downarrow Y)$ because $e \downarrow Y \sqsubset p(e \downarrow Y)$.
3. $Iv \vDash \sigma^U(\neg\phi)$ iff $Iv \vDash \neg\sigma^U(\phi)$ iff $Iv \nvDash \sigma^U(\phi)$, by IH, iff $\sigma^*_w Iv \nvDash \phi$ iff $\sigma^*_w Iv \vDash \neg\phi$
4. $Iv \vDash \sigma^U(\varphi \wedge \psi)$ iff $Iv \vDash \sigma^U(\varphi) \wedge \sigma^U(\psi)$ iff ($Iv \vDash \sigma^U(\varphi)$ and $Iv \vDash \sigma^U(\psi)$), by IH, iff ($\sigma^*_w Iv \vDash \varphi$ and $\sigma^*_w Iv \vDash \psi$) iff $\sigma^*_w Iv \vDash \varphi \wedge \psi$

5. Observe that v_z^d is a $(U \cup \{z\})$-variation of w for all $d \in \mathit{type}(z)$ since v is a U-variation of w. Now, $Iv \vDash \sigma^U(\forall z\, \varphi)$ iff $Iv \vDash \forall z\, \sigma^{U \cup \{z\}}(\varphi)$ iff $Iv_z^d \vDash \sigma^{U \cup \{z\}}(\varphi)$ for all $d \in \mathit{type}(z)$, by IH, iff $\sigma_w^* I v_z^d \vDash \varphi$ for all $d \in \mathit{type}(z)$ iff $\sigma_w^* Iv \vDash \forall z\, \varphi$.
6. Let $Iv \vDash \sigma^U([\alpha]\psi)$. Then $Iv \vDash [\sigma_Z^{U,\emptyset}(\alpha)]\sigma^Z(\psi)$. To prove $\sigma_w^* Iv \vDash [\alpha]\psi$, let $(v, \tau, o) \in \sigma_w^* I\llbracket\alpha\rrbracket$ with $o \neq \bot$. By the mutual IH, $(v, \tau, o) \in I\llbracket\sigma_Z^{U,\emptyset}(\alpha)\rrbracket$ because v is a $(U \cup \emptyset)$-variation of w. Hence, $Io \cdot \tau \vDash \sigma^Z(\psi)$ by the premise. Finally, by IH, $\sigma_w^* Io \cdot \tau \vDash \psi$ because $o \cdot \tau$ is a Z-variation of w by Lemma 30.
   Conversely, let $\sigma_w^* Iv \vDash [\alpha]\psi$. To prove $Iv \vDash \sigma^U([\alpha]\psi)$, recall $\sigma^U([\alpha]\psi) = [\sigma_Z^{U,\emptyset}(\alpha)]\sigma^Z(\psi)$ and let $(v, \tau, o) \in I\llbracket\sigma_Z^{U,\emptyset}(\alpha)\rrbracket$ with $o \neq \bot$. By the mutual IH, $(v, \tau, o) \in \sigma_w^* I\llbracket\alpha\rrbracket$ because v is a $(U \cup \emptyset)$-variation of w. Therefore, $\sigma_w^* Io \cdot \tau \vDash \psi$ by the premise. Finally, by IH, $Io \cdot \tau \vDash \sigma^Z(\psi)$ because $o \cdot \tau$ is a Z-variation of w by Lemma 30.
7. Let $Iv \vDash \sigma^U([\alpha]_{\{\mathsf{A},\mathsf{C}\}}\psi)$. Then $Iv \vDash [\sigma_Z^{U,\emptyset}(\alpha)]_{\{\sigma^Z(\mathsf{A}),\sigma^Z(\mathsf{C})\}}\sigma^Z(\psi)$. To prove $\sigma_w^* Iv \vDash [\alpha]_{\{\mathsf{A},\mathsf{C}\}}\psi$, let $(v, \tau, o) \in \sigma_w^* I\llbracket\alpha\rrbracket$. By the mutual IH, $(v, \tau, o) \in I\llbracket\sigma_Z^{U,\emptyset}(\alpha)\rrbracket$ because v is a $(U \cup \emptyset)$-variation of w. Since variation is monotone in the variation set, v is a $(U \cup \mathsf{BV}(\sigma_Z^{U,\emptyset}(\alpha)) \cup \mathsf{CN}(\sigma_Z^{U,\emptyset}(\alpha)))$-variation of w. Moreover, by the bound effect property (Lemma 10), we have $\tau \downarrow \mathsf{CN}(\sigma_Z^{U,\emptyset}(\alpha))^{\complement} = \epsilon$, which implies that $v \cdot \tau'$ is a $(U \cup \mathsf{BV}(\sigma_Z^{U,\emptyset}(\alpha)) \cup \mathsf{CN}(\sigma_Z^{U,\emptyset}(\alpha)))$-variation of w for all $\tau' \preceq \tau$. Hence, by Lemma 13, $v \cdot \tau'$ is a Z-variation of w for all $\tau' \preceq \tau$. For (commit), assume $\{\sigma_w^* Iv \cdot \tau' \mid \tau' \prec \tau\} \vDash \mathsf{A}$. By IH, $\{Iv \cdot \tau' \mid \tau' \prec \tau\} \vDash \sigma^Z(\mathsf{A})$. Hence, $Iv \cdot \tau \vDash \sigma^Z(\mathsf{C})$ by premise, which implies $\sigma_w^* Iv \cdot \tau \vDash \mathsf{C}$ by IH again. For (post), assume $o \neq \bot$ and $\{\sigma_w^* Iv \cdot \tau' \mid \tau' \preceq \tau\} \vDash \mathsf{A}$. By IH, $\{Iv \cdot \tau' \mid \tau' \preceq \tau\} \vDash \sigma^Z(\mathsf{A})$. Hence, $Io \cdot \tau \vDash \sigma^Z(\psi)$ by premise, which implies $\sigma_w^* Io \cdot \tau \vDash \psi$ by IH again because $o \cdot \tau$ is a Z-variation of w by Lemma 30.
   Conversely, let $\sigma_w^* Iv \vDash [\alpha]_{\{\mathsf{A},\mathsf{C}\}}\psi$. To prove $Iv \vDash \sigma^U([\alpha]_{\{\mathsf{A},\mathsf{C}\}}\psi)$, recall that $\sigma^U([\alpha]_{\{\mathsf{A},\mathsf{C}\}}\psi) = [\sigma_Z^{U,\emptyset}(\alpha)]_{\{\sigma^Z(\mathsf{A}),\sigma^Z(\mathsf{C})\}}\sigma^Z(\psi)$ and let $(v, \tau, w) \in I\llbracket\sigma_Z^{U,\emptyset}(\alpha)\rrbracket$. By the mutual IH, $(v, \tau, w) \in \sigma_w^* I\llbracket\alpha\rrbracket$ because v is a $(U \cup \emptyset)$-variation of w. As conversely, $v \cdot \tau'$ is a Z-variation of w for all $\tau' \preceq \tau$. For (commit), assume $\{Iv \cdot \tau' \mid \tau' \prec \tau\} \vDash \sigma^Z(\mathsf{A})$. By IH $\{\sigma_w^* Iv \cdot \tau' \mid \tau' \prec \tau\} \vDash \mathsf{A}$. Hence, $\sigma_w^* Iv \cdot \tau \vDash \mathsf{C}$ by premise, which implies $Iv \cdot \tau \vDash \sigma^Z(\mathsf{C})$ by IH again. For (post), assume $o \neq \bot$ and $\{Iv \cdot \tau' \mid \tau' \preceq \mathsf{A}\} \vDash \sigma^Z(\mathsf{A})$. By IH $\{\sigma_w^* Iv \cdot \tau' \mid \tau' \preceq \tau\} \vDash \mathsf{A}$. Hence, $\sigma_w^* Io \cdot \tau \vDash \psi$ by premise, which implies $Io \cdot \tau \vDash \sigma^Z(\psi)$ by IH again because $o \cdot \tau$ is a Z-variation of w by Lemma 30.

Secondly, consider the program cases. Therefore, let v be any $(U \cup W)$-variation of w. W.l.o.g. $\tau = \epsilon$ and $o \neq \bot$ in the cases of non-communicating atomic programs α below because $\tau = \epsilon$ for all $(v, \tau, o) \in I\llbracket\alpha\rrbracket$ if α is an atomic program but not a communication primitive and $((v, \epsilon, \bot) \in I\llbracket\sigma_Z^{U,W}(\alpha)\rrbracket$ iff $(v, \epsilon, \bot) \in \sigma_w^* I\llbracket\alpha\rrbracket)$ by totality.

1. Since $\sigma_Z^{U,W}(a\llparenthesis Y, \bar{z}\rrparenthesis)$ is defined, $\mathsf{BV}(\sigma a) \subseteq \bar{z}$ and $\mathsf{CN}(\sigma a) \subseteq Y$. Hence, the equivalence $(v, \tau, o) \in I\llbracket\sigma_Z^{U,W}(a\llparenthesis Y, \bar{z}\rrparenthesis)\rrbracket$ iff $(v, \tau, o) \in I\llbracket\sigma a\rrbracket$ is defined, where the latter is by Def. 14 equivalent to $(v, \tau, o) \in \sigma_w^* I(a\llparenthesis Y, \bar{z}\rrparenthesis)$ iff $(v, \tau, o) \in \sigma_w^* I\llbracket a\llparenthesis Y, \bar{z}\rrparenthesis\rrbracket$.

2. $(v, \epsilon, o) \in I\llbracket\sigma_Z^{U,W}(x := \theta)\rrbracket$ iff $(v, \epsilon, o) \in I\llbracket x := \sigma^{U\cup W}(\theta)\rrbracket$ iff $o = v_x^d$, where $d = Iv\llbracket\sigma^{U\cup W}(\theta)\rrbracket$, by Lemma 16, iff $o = v_x^d$, where $d = \sigma_w^* Iv\llbracket\theta\rrbracket$ iff $(v, \epsilon, o) \in \sigma_w^* I\llbracket x := \theta\rrbracket$
3. $(v, \epsilon, o) \in I\llbracket\sigma^U(x := *)\rrbracket$ iff $o = v_x^d$, where $d \in \mathbb{R}$, iff $(v, \epsilon, o) \in \sigma_w^* I\llbracket x := *\rrbracket$
4. $(v, \epsilon, o) \in I\llbracket\sigma_Z^{U,W}(\{x' = \theta \& \chi\})\rrbracket$ iff $(v, \epsilon, o) \in I\llbracket\{x' = \sigma^{U\cup W}(\theta) \& \sigma^{U\cup W}(\chi)\}\rrbracket$ iff a solution $\varphi : [0, s] \to \mathcal{S}$ from v to o exists with $I\varphi(\zeta) \vDash \mu' = 1 \wedge x' = \sigma^{U\cup W}(\theta) \wedge \sigma^{U\cup W}(\chi)$ and $v = \varphi(\zeta)$ on $\{x, x', \mu, \mu'\}^\complement$ for all $\zeta \in [0, s]$ and if $\varphi(\zeta)(z') = \frac{d\varphi(t)(z)}{dt}(z)$ for $z \in \{\mu, x\}$. Since $\varphi(\zeta)$ is a $(U \cup \{x, x', \mu, \mu'\})$-variation of w, we obtain $\sigma_w^* I\varphi(\zeta) \vDash \mu' = 1 \wedge x' = \theta$ by Lemma 16 and $\sigma_w^* I\varphi(\zeta) \vDash \chi$ by the mutual IH. Thus, $(v, \epsilon, o) \in I\llbracket\sigma_Z^{U,W}(\{x' = \theta \,\&\, \chi\})\rrbracket$ iff $(v, \tau, o) \in \sigma_w^* I\llbracket\{x' = \theta \,\&\, \chi\}\rrbracket$.
5. $(v, \epsilon, o) \in I\llbracket\sigma_Z^{U,W}(?\chi)\rrbracket$ iff $(v, \epsilon, o) \in I\llbracket ?\sigma^{U\cup W}(\chi)\rrbracket$ iff $o = v$ and $Iv \vDash \sigma^{U\cup W}(\chi)$, by the mutual IH, iff $o = v$ and $\sigma_w^* Iv \vDash \chi$ iff $(v, \epsilon, o) \in \sigma_w^* I\llbracket ?\chi\rrbracket$
6. $(v, \tau, o) \in I\llbracket\sigma_Z^{U,W}(\mathrm{ch}(h)!\theta)\rrbracket$ iff $(v, \tau, o) \in I\llbracket\mathrm{ch}(h)!\sigma^{U\cup W}(\theta)\rrbracket$ iff $(\tau, o) \preceq (\langle h, \mathrm{ch}, d, v(\mu)\rangle, v)$ with $d = Iv\llbracket\sigma^{U\cup W}(\theta)\rrbracket$, which by Lemma 16, is equivalent to $(\tau, o) \preceq (\langle h, \mathrm{ch}, d^*, v(\mu)\rangle, v)$ with $d^* = \sigma_w^* Iv\llbracket\theta\rrbracket$ iff $(v, \tau, o) \in \sigma_w^* I\llbracket\mathrm{ch}(h)!\theta\rrbracket$
7. $(v, \tau, o) \in I\llbracket\sigma_Z^{U,W}(\mathrm{ch}(h)?x)\rrbracket$ iff $(v, \tau, o) \in I\llbracket\mathrm{ch}(h)?x\rrbracket$, which is equivalent to $(\tau, o) \preceq (\langle h, \mathrm{ch}, d, v(\mu)\rangle, v_x^d)$ with $d \in \mathbb{R}$ iff $(v, \tau, o) \in \sigma_w^* I\llbracket\mathrm{ch}(h)?x\rrbracket$
8. $(v, \tau, o) \in I\llbracket\sigma_Z^{U,W}(\alpha \cup \beta)\rrbracket$ iff $(v, \tau, o) \in I\llbracket\sigma_{Z_1}^{U,W}(\alpha) \cup \sigma_{Z_2}^{U,W}(\beta)\rrbracket$, where $Z = Z_1 \cup Z_2$, iff $(v, \tau, o) \in I\llbracket\sigma_{Z_1}^{U,W}(\alpha)\rrbracket$ or $(v, \tau, o) \in I\llbracket\sigma_{Z_2}^{U,W}(\beta)\rrbracket$, which by IH, is equivalent to $(v, \tau, o) \in \sigma_w^* I\llbracket\alpha\rrbracket$ or $(v, \tau, o) \in \sigma_w^* I\llbracket\beta\rrbracket$ iff $(v, \tau, o) \in \sigma_w^* I\llbracket\alpha \cup \beta\rrbracket$
9. Let $(v, \tau, o) \in I\llbracket\sigma_Z^{U,W}(\alpha; \beta)\rrbracket$. Since $\sigma_Z^{U,W}(\alpha; \beta) \equiv \sigma_{Z_0}^{U,W}(\alpha); \sigma_Z^{Z_0,W}(\beta)$, there is $(v, \tau, o) \in (I\llbracket\sigma_{Z_0}^{U,W}(\alpha)\rrbracket)_\perp$ or $(v, \tau, o) \in I\llbracket\sigma_{Z_0}^{U,W}(\alpha)\rrbracket \triangleright I\llbracket\sigma_Z^{Z_0,W}(\beta)\rrbracket$. In case $(v, \tau, o) \in (I\llbracket\sigma_{Z_0}^{U,W}(\alpha)\rrbracket)_\perp$, there is $(v, \tau, o) \in (\sigma_w^* I\llbracket\alpha\rrbracket)_\perp \subseteq \sigma_w^* I\llbracket\alpha; \beta\rrbracket$ by IH. Otherwise, $(v, \tau, o) \in I\llbracket\sigma_{Z_0}^{U,W}(\alpha)\rrbracket \triangleright I\llbracket\sigma_Z^{Z_0,W}(\beta)\rrbracket$, so computations $(v, \tau_1, u) \in I\llbracket\sigma_{Z_0}^{U,W}(\alpha)\rrbracket$ and $(u, \tau_2, o) \in I\llbracket\sigma_Z^{Z_0,W}(\beta)\rrbracket$ exist with $\tau = \tau_1 \cdot \tau_2$. By IH, $(v, \tau_1, u) \in \sigma_w^* I\llbracket\alpha\rrbracket$. Since v is a $(U \cup W)$-variation of w and $v = u$ on $\mathsf{BV}(\sigma_{Z_0}^{U,W}(\alpha))^\complement$ by the bound effect property, we obtain $w = v = u$ on $(U \cup W)^\complement \cap \mathsf{BV}(\sigma_{Z_0}^{U,W}(\alpha))^\complement = W^\complement \cap (U \cup \mathsf{BV}(\sigma_{Z_0}^{U,W}(\alpha)))^\complement$. By Lemma 13, $Z_0 \supseteq U \cup \mathsf{BV}(\sigma_{Z_0}^{U,W}(\alpha))$ such that $w = u$ on $W^\complement \cap Z_0^\complement = (W \cup Z_0)^\complement$, i.e., u is a $(Z_0 \cup W)$-variation of w such that IH is applicable on $(u, \tau_2, o) \in I\llbracket\sigma^{Z_0 \cup W}(\beta)\rrbracket$. Therefore, by IH, $(u, \tau_2, o) \in \sigma_w^* I\llbracket\beta\rrbracket$. Finally, $(v, \tau, o) \in \sigma_w^* I\llbracket\alpha\rrbracket \triangleright \sigma_w^* I\llbracket\beta\rrbracket \subseteq \sigma_w^* I\llbracket\alpha; \beta\rrbracket$.
   Conversely, let $(v, \tau, o) \in \sigma_w^* I\llbracket\alpha; \beta\rrbracket$. Then $(v, \tau, o) \in (\sigma_w^* I\llbracket\alpha\rrbracket)_\perp$ or $(v, \tau, o) \in \sigma_w^* I\llbracket\alpha\rrbracket \triangleright \sigma_w^* I\llbracket\beta\rrbracket$. If $(v, \tau, o) \in (\sigma_w^* I\llbracket\alpha\rrbracket)_\perp$, there is $(v, \tau, o) \in (I\llbracket\sigma_{Z_0}^{U,W}(\alpha)\rrbracket)_\perp \subseteq I\llbracket\sigma_{Z_0}^{U,W}(\alpha); \sigma_Z^{Z_0,W}(\beta)\rrbracket = I\llbracket\sigma_Z^{U,W}(\alpha; \beta)\rrbracket$ by IH. Otherwise, $(v, \tau, o) \in \sigma_w^* I\llbracket\alpha\rrbracket \triangleright \sigma_w^* I\llbracket\beta\rrbracket$, so computations $(v, \tau_1, u) \in \sigma_w^* I\llbracket\alpha\rrbracket$ and $(u, \tau_2, o) \in \sigma_w^* I\llbracket\beta\rrbracket$ exist such that $\tau = \tau_1 \cdot \tau_2$. By IH, $(v, \tau_1, u) \in I\llbracket\sigma_{Z_0}^{U,W}(\alpha)\rrbracket$. As conversely, u is a $(Z_0 \cup W)$-variation of w such that IH is applicable on $(u, \tau_2, o) \in \sigma_w^* I\llbracket\beta\rrbracket$. Therefore, by IH, $(u, \tau_2, o) \in I\llbracket\sigma_Z^{Z_0,W}(\beta)\rrbracket$. Finally, $(v, \tau, o) \in I\llbracket\sigma_{Z_0}^{U,W}(\alpha)\rrbracket \triangleright I\llbracket\sigma_Z^{Z_0,W}(\beta)\rrbracket \subseteq I\llbracket\sigma_{Z_0}^{U,W}(\alpha); \sigma_Z^{Z_0,W}(\beta)\rrbracket = I\llbracket\sigma_Z^{U,W}(\alpha; \beta)\rrbracket$.
10. In case $\sigma_Z^{U,W}(\alpha^*)$, the output taboo $Z \supseteq U$ of $\sigma_Z^{U,W}(\alpha)$ is added to U during an additional first pass over α as indicated by the fixpoint notation

$\sigma_Z^{Z,W}(\alpha)$. Since v is a $(U \cup W)$-variation of w and variation is monotone in the variation set, v is a $(Z \cup W)$-variation of w. Now, $(v, \tau, o) \in I[\![\sigma_Z^{U,W}(\alpha^*)]\!]$ iff $(v, \tau, o) \in I[\![\sigma_Z^{Z,W}(\alpha)^*]\!]$ iff $(v, \tau, o) \in I[\![(\sigma_Z^{Z,W}(\alpha))^n]\!]$ for some $n \in \mathbb{N}$ iff $(v, \tau, o) \in I[\![\sigma_Z^{Z,W}(\alpha^n)]\!]$ for some $n \in \mathbb{N}$, by IH, iff $(v, \tau, o) \in \sigma_w^* I[\![\alpha^n]\!]$ for some $n \in \mathbb{N}$ iff $(v, \tau, o) \in \sigma_w^* I[\![\alpha^*]\!]$.

11. In case $\alpha \parallel \beta$, we have $\sigma_Z^{U,W}(\alpha \parallel \beta) \equiv \sigma(\alpha) \parallel \sigma(\beta)$, where $\sigma(\alpha) \equiv \sigma_{Z_1}^{U,W_{U,\beta}}(\alpha)$, and $\sigma(\beta) \equiv \sigma_{Z_2}^{U,W_{U,\alpha}}(\beta)$, and $W_{U,\gamma} \equiv W \cup (\mathsf{BV}(\sigma^{U,W}(\gamma)) \setminus (\{\mu, \mu'\} \cup V_{\mathcal{T}}))$ for any program γ, and $Z = Z_1 \cup Z_2$. Since v is a $(U \cup W)$-variation of w, by monotony in the variation set, v is a $(U \cup W_{U,\gamma})$-variation of w for any γ. We define explicit merging $o_1 \oplus_\gamma o_2$ to be o_1 on $\mathsf{BV}(\gamma)$ and o_2 elsewhere.

    Now, $(v, \tau, o) \in I[\![\sigma_Z^{U,W}(\alpha \parallel \beta)]\!]$ iff $(v, \tau \downarrow \sigma(\gamma), o_\gamma) \in I[\![\sigma(\gamma)]\!]$ for $\gamma \in \{\alpha, \beta\}$, and $o = o_\alpha \oplus_{\sigma(\alpha)} o_\beta$, and $o_\alpha = o_\beta$ on $\{\mu, \mu'\}$, and $\tau = \tau \downarrow (\sigma(\alpha) \parallel \sigma(\beta))$, iff, by Lemma 31 and the argument about merging $\oplus$ below, $(v, \tau \downarrow \gamma, o_\gamma) \in \sigma_w^* I[\![\sigma(\gamma)]\!]$ for $\gamma \in \{\alpha, \beta\}$, and $o = o_\alpha \oplus_\alpha o_\beta$, and $o_\alpha = o_\beta$ on $\{\mu, \mu'\}$, and $\tau = \tau \downarrow (\alpha \parallel \beta)$ iff, by IH, $(v, \tau \downarrow \gamma, o_\gamma) \in \sigma_w^* I[\![\gamma]\!]$ for $\gamma \in \{\alpha, \beta\}$, and $o = o_\alpha \oplus_\alpha o_\beta$, and $o_\alpha = o_\beta$ on $\{\mu, \mu'\}$, and $\tau = \tau \downarrow (\alpha \parallel \beta)$ iff $(v, \tau, o) \in \sigma_w^* I[\![\alpha \parallel \beta]\!]$.

    Finally, $o_\alpha \oplus_\alpha o_\beta = o_\alpha \oplus_{\sigma(\alpha)} o_\beta$ has been left open above. On $\mathsf{BV}(\sigma(\alpha))$, we have $o_\alpha \oplus_\alpha o_\beta = o_\alpha = o_\alpha \oplus_{\sigma(\alpha)} o_\beta$ since $\mathsf{BV}(\sigma(\alpha)) \subseteq \mathsf{BV}(\alpha)$. On $\mathsf{BV}(\sigma(\alpha))^{\mathsf{C}}$, first consider $\mathsf{BV}(\sigma(\alpha))^{\mathsf{C}} \cap \mathsf{BV}(\alpha)^{\mathsf{C}}$, where $o_\alpha \oplus_\alpha o_\beta = o_\beta = o_\alpha \oplus_{\sigma(\alpha)} o_\beta$. Now, consider $X = \mathsf{BV}(\sigma(\alpha))^{\mathsf{C}} \cap \mathsf{BV}(\alpha)$. Further, let $X_{\text{obs}} = \{z \mid v(z) \neq o_\alpha(z) \vee v(z) \neq o_\beta(z)\}$. On $X \cap X_{\text{obs}}^{\mathsf{C}}$, we have $o_\alpha \oplus_{\sigma(\alpha)} o_\beta = v = o_\alpha \oplus_\alpha o_\beta$. Otherwise, if $z \in X \cap X_{\text{obs}}$, then $z \in \mathsf{BV}(\alpha)$ but $z \notin V_{\mathcal{T}}$ by Lemma 10. Then either $z \in V_{\mathbb{R}} \setminus \{\mu, \mu'\}$ such that $z \notin \mathsf{BV}(\beta)$ by well-formedness of $\alpha \parallel \beta$ or $z \in \{\mu, \mu'\}$ such that the programs must agree upon the value of z in their final states. If $z \notin \mathsf{BV}(\beta)$, then $v(z) \neq o_\alpha(z)$ since $z \in X_{\text{obs}}$ such that $z \in \mathsf{BV}(\sigma(\alpha))$ in contradiction to $z \in X$. Therefore, $z \notin X \cap X_{\text{obs}}$ such that $X \cap X_{\text{obs}} = \emptyset$. Otherwise, if $z \in \{\mu, \mu'\}$, the final states agree upon the value of z. Thus, $o_\alpha \oplus_\alpha o_\beta = o_\alpha \oplus_{\sigma(\alpha)} o_\beta$ on $\{\mu, \mu'\}$. □

Once the uniform substitution lemmas from Section 3.1 are proven, the soundness proof of uniform substitution (Theorem 18) is easy:

Proof (of Theorem 18). Let the premise ϕ be valid, i.e., $J\tilde{v} \vDash \phi$ for all pairs $J\tilde{v}$ of interpretation and state. For proving the conclusion, let Iv be any pair of interpretation and state. By validity of the premise, thus $\sigma_v^* Iv \vDash \phi$. Since v is a $\emptyset$-variation of v, Lemma 17 implies $Iv \vDash \sigma^{\emptyset}(\phi)$. □

Besides the instantiation of axioms by US, Theorem 32 even allows instantiation of axiomatic proof rules using uniform substitution. The rule must be locally sound, i.e., validity of the premises in any interpretation implies validity of the conclusion under this interpretation.

Theorem 32 (Sound uniform substitution for rules). *If the inference* INF *is locally sound, so is the inference* US-INF*:*

$$\frac{\phi_1 \ \dots \ \phi_k}{\psi} \ (\text{INF}) \qquad\qquad \frac{\sigma^{V \cup \Omega}(\phi_1) \ \dots \ \sigma^{V \cup \Omega}(\phi_k)}{\sigma^{V \cup \Omega}(\psi)} \ (\text{US-INF})$$

Proof. Assume that the inference INF is locally sound. To prove that the inference US-INF is locally sound, let I be any interpretation such that $I \vDash \sigma^{V \cup \Omega}(\phi_j)$ for $1 \leq j \leq k$, i.e., $Iv \vDash \sigma^{V \cup \Omega}(\phi_j)$ for all v and j. Since v is a $(V \cup \Omega)$-variation of any w, by Lemma 17, $\sigma^*_w I \vDash \phi_j$ for all j. Thus, $\sigma^*_w I \vDash \psi$ by local soundness of INF. Finally, $I \vDash \sigma^{V \cup \Omega}(\psi)$ by Lemma 17 again. □

Lemma 34 prepares the proof of Proposition 19 that uniform substitution preserves well-formedness proving that it respects the parallel context. As well-formedness is defined in terms of the static semantics, the proofs of Lemma 34 and Proposition 19 make use of the uniform substitution lemmas via Lemma 33.

Lemma 33. *For formula ϕ and program α, we have*

1. $\mathsf{FV}(\sigma^U(\phi)) \subseteq \mathsf{FV}(\phi) \cup U^\complement$*, and*
2. $\mathsf{FV}(\sigma^{U,W}_Z(\alpha)) \subseteq \mathsf{FV}(\alpha) \cup (U \cup W)^\complement$*, and*
3. $\mathsf{BV}(\sigma^{U,W}_Z(\alpha)) \subseteq \mathsf{BV}(\alpha)$

if the substitutions $\sigma^U(\phi)$ and $\sigma^{U,W}_Z(\alpha)$ are defined.

Proof. For 1, let $z \in \mathsf{FV}(\sigma^U(\phi))$ but assume $z \notin U^\complement$. By definition of $\mathsf{FV}(\phi)$, there are I, v, and $\tilde{v}$ with $v = \tilde{v}$ on $\{z\}^\complement$ and $Iv[\![\sigma^U(\phi)]\!] \neq I\tilde{v}[\![\sigma^U(\phi)]\!]$. Since $z \in U$, state $\tilde{v}$ is a U-variation of v. By Lemma 17, $\sigma^*_v Iv[\![\phi]\!] \neq \sigma^*_v I\tilde{v}[\![\phi]\!]$. Therefore, $z \in \mathsf{FV}(\phi)$.

For item 2, let $z \in \mathsf{FV}(\sigma^{U,W}_Z(\alpha))$ but assume $z \notin (U \cup W)^\complement$. By definition of $\mathsf{FV}(\alpha)$, there are $I, v, \tilde{v}, \tau, w$ such that $v = \tilde{v}$ on $\{z\}^\complement$ and $(v, \tau, w) \in I[\![\sigma^{U,W}_Z(\alpha)]\!]$ but there is no $(\tilde{v}, \tilde{\tau}, \tilde{w}) \in I[\![\sigma^{U,W}_Z(\alpha)]\!]$ such that $\tilde{\tau} = \tau$ and $\tilde{w} = w$ on $\{z\}^\complement$. By Lemma 17, $(v, \tau, w) \in \sigma^*_v I[\![\alpha]\!]$. Since $x \in U \cup W$, state $\tilde{v}$ is a $(U \cup W)$-variation of v. Hence, there is no $(\tilde{v}, \tilde{\tau}, \tilde{w}) \in \sigma^*_v I[\![\alpha]\!]$ such that $\tilde{\tau} = \tau$ and $\tilde{w} = w$ on $\{z\}^\complement$ because otherwise, $(\tilde{v}, \tilde{\tau}, \tilde{w}) \in I[\![\sigma^{U,W}_Z(\alpha)]\!]$ by Lemma 17. In summary, $z \in \mathsf{FV}(\alpha)$.

For item 3, let $z \in \mathsf{BV}(\sigma^{U,W}_Z(\alpha))$. Then I and $(v, \tau, w) \in I[\![\sigma^{U,W}_Z(\alpha)]\!]$ exist such that $(w \cdot \tau)(z) \neq v(z)$. By Lemma 17, $(v, \tau, w) \in \sigma^*_v I[\![\alpha]\!]$. Hence, $z \in \mathsf{BV}(\alpha)$. □

Lemma 34. *If a program α respects the parallel context W, i.e., $\mathsf{V}(\alpha) \cap W \subseteq \{\mu, \mu'\} \cup V_\mathcal{T}$, the result of substitution $\sigma^{U,W}_Z(\alpha)$, if defined, respects the context W, too, i.e., $\mathsf{V}(\sigma^{U,W}_Z(\alpha)) \cap W \subseteq \{\mu, \mu'\} \cup V_\mathcal{T}$.*

Proof. Let $\mathsf{V}(\alpha) \cap W \subseteq \{\mu, \mu'\} \cup V_\mathcal{T}$. Then $\mathsf{V}(\sigma^{U,W}_Z(\alpha)) \cap W = (\mathsf{FV}(\sigma^{U,W}_Z(\alpha)) \cup \mathsf{BV}(\sigma^{U,W}_Z(\alpha))) \cap W$, which is by Lemma 33 smaller or equal to $(\mathsf{FV}(\alpha) \cup (U \cup W)^\complement \cup \mathsf{BV}(\alpha)) \cap W = (\mathsf{FV}(\alpha) \cup \mathsf{BV}(\alpha)) \cap W = \mathsf{V}(\alpha) \cap W$, which is by the premise smaller or equal to $\{\mu, \mu'\} \cup V_\mathcal{T}$. □

The proof of Proposition 19 shows that uniform substitution (see Fig. 2) already preserves the stronger well-formedness condition $(\mathsf{V}(\alpha) \cap \mathsf{BV}(\beta)) \cup (\mathsf{V}(\beta) \cap \mathsf{BV}(\alpha)) \subseteq \{\mu, \mu'\} \cup V_\mathcal{T}$ for programs α, β from previous work [6]. Importantly, note that the weaker well-formedness condition $\mathsf{BV}(\alpha) \cap \mathsf{BV}(\beta) \subseteq \{\mu, \mu'\} \cup V_\mathcal{T}$ as imposed on programs in this paper, is still sufficient to preserve itself and that the proofs of Lemma 34 and Proposition 19 can be adjusted accordingly.

Proof (of Proposition 19). The proof is by simultaneous induction on the structure of programs and formulas. It uses the abbreviation $V_G = \{\mu, \mu'\} \cup V_{\mathcal{T}}$.

The only program with context-sensitive syntax is parallel composition, thus atomic programs are trivially well-formed and other compound programs are well-formed since by IH, their subprograms are well-formed.

Now, let $\alpha \parallel \beta$ be well-formed. Then $(\mathsf{V}(\alpha) \cap \mathsf{BV}(\beta)) \cup (\mathsf{V}(\beta) \cap \mathsf{BV}(\alpha)) \subseteq V_G$. Moreover, let $\sigma_Z^{U,W}(\alpha \parallel \beta)$ be defined. Thus, $\sigma_{Z_1}^{U,W_{U,\beta}}(\alpha)$ and $\sigma_{Z_2}^{U,W_{U,\alpha}}(\beta)$, where $W_{U,\alpha} = W \cup (\mathsf{BV}(\sigma^{U,W}(\alpha)) \setminus V_G)$ and $W_{U,\beta} = W \cup (\mathsf{BV}(\sigma^{U,W}(\beta)) \setminus V_G)$, are defined, and by IH, they are well-formed. By Lemma 34, $V_G \supseteq \mathsf{V}(\sigma_{Z_1}^{U,W_{U,\beta}}(\alpha)) \cap W_{U,\beta} \supseteq \mathsf{V}(\sigma_{Z_1}^{U,W_{U,\beta}}(\alpha)) \cap (\mathsf{BV}(\sigma^{U,W}(\beta)) \setminus V_G)$, which equals $\mathsf{V}(\sigma_{Z_1}^{U,W_{U,\beta}}(\alpha)) \cap (\mathsf{BV}(\sigma^{U,W_{U,\alpha}}(\beta)) \setminus V_G)$ since the parallel context does not influence the substitution result if it is defined. Thus, $V_G \supseteq \mathsf{V}(\sigma_{Z_1}^{U,W_{U,\beta}}(\alpha)) \cap \mathsf{BV}(\sigma^{U,W_{U,\alpha}}(\beta))$. Accordingly, $V_G \supseteq \mathsf{V}(\sigma_{Z_2}^{U,W_{U,\alpha}}(\beta)) \cap \mathsf{BV}(\sigma^{U,W_{U,\beta}}(\alpha))$ by Lemma 34. Since the context-sensitive side conditions are respected, $\sigma_Z^{U,W}(\alpha \parallel \beta)$ is well-formed as parallel composition of well-formed programs.

For formulas, the ac-box is the only interesting case. Let $[\alpha]_{\{\mathsf{A},\mathsf{C}\}}\psi$ be well-formed. Then $(\mathsf{FV}(\mathsf{A}) \cup \mathsf{FV}(\mathsf{C})) \cap \mathsf{BV}(\alpha) \subseteq V_{\mathcal{T}}$. Moreover, let $\sigma^U([\alpha]_{\{\mathsf{A},\mathsf{C}\}}\psi)$ be defined. For $\chi \in \{\mathsf{A}, \mathsf{C}\}$, we obtain by Lemma 33 that $\mathsf{FV}(\sigma^Z(\chi)) \cap \mathsf{BV}(\sigma_Z^{U,\emptyset}(\alpha)) \subseteq (\mathsf{FV}(\chi) \cup Z^{\complement}) \cap \mathsf{BV}(\sigma_Z^{U,\emptyset}(\alpha))$, which is by Lemma 13 smaller or equal to

$$\Big(\mathsf{FV}(\chi) \cup \big(U \cup \mathsf{BV}(\sigma_Z^{U,\emptyset}(\alpha)) \cup \mathsf{CN}(\sigma_Z^{U,\emptyset}(\alpha))\big)^{\complement}\Big) \cap \mathsf{BV}(\sigma_Z^{U,\emptyset}(\alpha)),$$

which equals $\mathsf{FV}(\chi) \cap \mathsf{BV}(\sigma_Z^{U,\emptyset}(\alpha))$, which is by Lemma 33 smaller or equal to $\mathsf{FV}(\chi) \cap \mathsf{BV}(\alpha)$, which is smaller or equal to $V_{\mathcal{T}}$ by the premise. □

C Details of the Axiomatic Calculus

This appendix reports a soundness proof for $\mathsf{dL}_{\mathrm{CHP}}$'s axiomatization (see Fig. 3). Moreover, Corollary 35 gives derivations of ac-monotonicity $\mathrm{M}[\cdot]_{\mathsf{AC}}$ and distribution of boxes over conjuncts $[]_{\mathsf{AC}}\wedge$. Finally, algebraic laws for reasoning about trace terms [6] are lifted to uniform substitution.

Proof (of Theorem 21). Since the axioms of Fig. 3 are instances of their schematic versions (except for K_{AC} and $[]_{\mathrm{W}\mathsf{A}}$), they are sound as the schematic axioms are sound [6]. In particular, note that axioms $[\epsilon]_{\mathsf{AC}}$ and $[\|\_]_{\mathsf{AC}}$ internalize the side conditions of the schematic calculus correctly. For the newly added axiom K_{AC} and the changed axiom $[]_{\mathrm{W}\mathsf{A}}$, soundness proofs are provided below. Recall that $\mathrm{R}_j \equiv r_j(Y, \bar{h})$, and $\mathrm{Q}_j \equiv q_j(Y, \bar{h})$, and $\mathrm{P}_j \equiv p_j(Y, \bar{z})$.

K_{AC}: Let $Iv \vDash [a]_{\{\mathrm{R},\mathrm{Q}_1 \rightarrow \mathrm{Q}_2\}}(\mathrm{P}_1 \rightarrow \mathrm{P}_2)$, and $Iv \vDash [a]_{\{\mathrm{R},\mathrm{Q}_1\}}\mathrm{P}_1$, and let $(v, \tau, w) \in I[\![a]\!]$. For (commit), assume $\{Iv \cdot \tau' \mid \tau' \prec \tau\} \vDash \mathrm{R}$. Then $Iv \cdot \tau \vDash \mathrm{Q}_1 \rightarrow \mathrm{Q}_2$ and $Iv \cdot \tau \vDash \mathrm{Q}_1$, thus $Iv \cdot \tau \vDash \mathrm{Q}_2$. For (post), assume $w \neq \bot$ and $\{Iv \cdot \tau' \mid \tau' \preceq \tau\} \vDash \mathrm{R}$. Then $Iv \cdot \tau \vDash \mathrm{P}_1 \rightarrow \mathrm{P}_2$ and $Iv \cdot \tau \vDash \mathrm{P}_1$ such that $Iv \cdot \tau \vDash \mathrm{P}_2$.

$[]\mathrm{W}_{\mathsf{A}}$: Let $Iv \vDash [a]_{\{\mathsf{T},\mathrm{W}_{\mathsf{A}}\}}\mathsf{T}$, where $\mathrm{W}_{\mathsf{A}} \equiv (\mathrm{R} \land \mathrm{Q}_1 \to \mathrm{R}_2) \land (\mathrm{R} \land \mathrm{Q}_2 \to \mathrm{R}_1)$, and let $Iv \vDash \Phi$ for $\Phi \equiv [a]_{\{\mathrm{R}_1 \land \mathrm{R}_2, \mathrm{Q}_1 \land \mathrm{Q}_2\}}\mathrm{P}$ and $(v, \tau, w) \in I\llbracket a \rrbracket$. By induction on the length of τ, we simultaneously prove $Iv \vDash [a]_{\{\mathrm{R}, \mathrm{Q}_1 \land \mathrm{Q}_2\}}\mathrm{P}$, and that $\{Iv \cdot \tau' \mid \tau' \prec \tau\} \vDash \mathrm{R}$ implies $\{Iv \cdot \tau' \mid \tau' \prec \tau\} \vDash \mathrm{R}_1 \land \mathrm{R}_2$:

1. $|\tau| = 0$, then $Iv \vDash \Phi$ implies $Iv \vDash \mathrm{Q}_1 \land \mathrm{Q}_2$ by axiom $\mathrm{W}[]_{\mathsf{AC}}$. Hence, (commit) holds because $\tau = \epsilon$. For (post), assume $w \neq \bot$ and $\{Iv \cdot \tau' \mid \tau' \preceq \tau\} \vDash \mathrm{R}$. By $Iv \vDash [a]_{\{\mathsf{T},\mathrm{W}_{\mathsf{A}}\}}\mathsf{T}$ and $\mathrm{W}[]_{\mathsf{AC}}$ again, we obtain $Iv \vDash \mathrm{W}_{\mathsf{A}}$. Then $\{Iv \cdot \tau' \mid \tau' \preceq \tau\} \vDash \mathrm{R}_1 \land \mathrm{R}_2$ because $Iv \vDash \mathrm{W}_{\mathsf{A}}$, and $Iv \vDash \mathrm{R}$, and $Iv \vDash \mathrm{Q}_1 \land \mathrm{Q}_2$. Thus, $Iw \cdot \tau \vDash \mathrm{P}$ by the premise $Iv \vDash \Phi$. Note that $\{Iv \cdot \tau' \mid \tau' \prec \tau\} \vDash \mathrm{R}_1 \land \mathrm{R}_2$ is trivially fulfilled since $\{Iv \cdot \tau' \mid \tau' \prec \tau\} = \emptyset$.

2. $|\tau| > 0$, then $\tau = \tau_0 \cdot \rho$ with $|\rho| = 1$. For (commit), assume $\{Iv \cdot \tau' \mid \tau' \prec \tau\} \vDash \mathrm{R}$. Then $\{Iv \cdot \tau' \mid \tau' \preceq \tau_0\} \vDash \mathrm{R}$, which implies $\{Iv \cdot \tau' \mid \tau' \prec \tau_0\} \vDash \mathrm{R}_1 \land \mathrm{R}_2$ by IH. Since $(v, \tau_0, \bot) \in I\llbracket a \rrbracket$ by prefix-closedness of the program semantics, we obtain $Iv \cdot \tau_0 \vDash \mathrm{Q}_1 \land \mathrm{Q}_2$ by $Iv \vDash \Phi$. Moreover, $Iv \cdot \tau_0 \vDash \mathrm{W}_{\mathsf{A}}$ by $Iv \vDash [a]_{\{\mathsf{T},\mathrm{W}_{\mathsf{A}}\}}\mathsf{T}$. Hence, $Iv \cdot \tau_0 \vDash \mathrm{R}_1 \land \mathrm{R}_2$ since $Iv \cdot \tau_0 \vDash \mathrm{W}_{\mathsf{A}}$, and $Iv \cdot \tau_0 \vDash \mathrm{R}$, and $Iv \cdot \tau_0 \vDash \mathrm{Q}_1 \land \mathrm{Q}_2$. Thus, $\{Iv \cdot \tau' \mid \tau' \prec \tau\} \vDash \mathrm{R}_1 \land \mathrm{R}_2$. Finally, $Iv \cdot \tau \vDash \mathrm{Q}_1 \land \mathrm{Q}_2$ by $Iv \vDash \Phi$ again. For (post), assume $w \neq \bot$ and $\{Iv \cdot \tau' \mid \tau' \preceq \tau\} \vDash \mathrm{R}$. Then $\{Iv \cdot \tau' \mid \tau' \prec \tau\} \vDash \mathrm{R}$, which implies $\{Iv \cdot \tau' \mid \tau' \prec \tau\} \vDash \mathrm{R}_1 \land \mathrm{R}_2$ and $Iv \cdot \tau \vDash \mathrm{Q}_1 \land \mathrm{Q}_2$ as in case (commit). By $(v, \tau, w) \in I\llbracket a \rrbracket$ and $Iv \vDash [a]_{\{\mathsf{T},\mathrm{W}_{\mathsf{A}}\}}\mathsf{T}$, we obtain $Iv \cdot \tau \vDash \mathrm{W}_{\mathsf{A}}$, which implies using $Iv \cdot \tau \vDash \mathrm{R}$ and $Iv \cdot \tau \vDash \mathrm{Q}_1 \land \mathrm{Q}_2$ that $Iv \cdot \tau \vDash \mathrm{R}_1 \land \mathrm{R}_2$. In summary, $\{Iv \cdot \tau' \mid \tau' \preceq \tau\} \vDash \mathrm{R}_1 \land \mathrm{R}_2$, which implies $Iw \cdot \tau \vDash \mathrm{P}$ by $Iv \vDash \Phi$.

Corollary 35. *The proof rule* $\mathrm{M}[\cdot]_{\mathsf{AC}}$ *of ac-monotonicity and the axiom* $[]_{\mathsf{AC}}\land$ *of distribution of conjuncts in commitments and postconditions over boxes [6] can be derived. Let* $\mathrm{R}_j \equiv r_j(Y, \bar{h})$*, and* $\mathrm{Q}_j \equiv q_j(Y, \bar{h})$*, and* $\mathrm{P}_j \equiv p_j(Y, \bar{z})$*.*

$$\dfrac{\mathrm{R}_2 \to \mathrm{R}_1 \qquad \mathrm{Q}_1 \to \mathrm{Q}_2 \qquad \mathrm{P}_1 \to \mathrm{P}_2}{[a]_{\{\mathrm{R}_1,\mathrm{Q}_1\}}\mathrm{P}_1 \to [a]_{\{\mathrm{R}_2,\mathrm{Q}_2\}}\mathrm{P}_2}\,\mathrm{M}[\cdot]_{\mathsf{AC}}$$

$$[a]_{\{\mathrm{R},\mathrm{Q}_1 \land \mathrm{Q}_2\}}(\mathrm{P}_1 \land \mathrm{P}_2) \leftrightarrow [a]_{\{\mathrm{R},\mathrm{Q}_1\}}\mathrm{P}_1 \land [a]_{\{\mathrm{R},\mathrm{Q}_2\}}\mathrm{P}_2 \qquad []_{\mathsf{AC}}\land$$

Proof. The proof is by derivation in the calculus. The sequent-style deduction is justified since sequent-style rules can be derived in a Hoare-style calculus.

$\mathrm{M}[\cdot]_{\mathsf{AC}}$: Let $\psi \equiv (\mathrm{R}_2 \land \mathrm{Q}_2 \to \mathsf{T}) \land (\mathrm{R}_2 \land \mathsf{T} \to \mathrm{R}_1)$ in the following, where Prop marks propositional reasoning.

$$\dfrac{\dfrac{\text{WL}\;\dfrac{\mathrm{G}_{\mathsf{AC}}\;\dfrac{\text{Prop}\;\dfrac{\mathrm{R}_2 \to \mathrm{R}_1}{\mathsf{T} \land \psi}}{[a]_{\{\mathsf{T},\psi\}}\mathsf{T}}}{[a]_{\{\mathrm{R}_1,\mathrm{Q}_1\}}\mathrm{P}_1 \to [a]_{\{\mathsf{T},\psi\}}\mathsf{T}} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\mathrm{Q}_1 \to \mathrm{Q}_2 \qquad \mathrm{P}_1 \to \mathrm{P}_2}{(\mathrm{Q}_1 \to \mathrm{Q}_2) \land (\mathrm{P}_1 \to \mathrm{P}_2)}\,\land\mathrm{R}}{[a]_{\{\mathrm{R}_1,\mathrm{Q}_1 \to \mathrm{Q}_2\}}(\mathrm{P}_1 \to \mathrm{P}_2)}\,\mathrm{G}_{\mathsf{AC}}}{[a]_{\{\mathrm{R}_1,\mathrm{Q}_1\}}\mathrm{P}_1 \to [a]_{\{\mathrm{R}_1,\mathrm{Q}_2\}}\mathrm{P}_2}\,\mathrm{K}_{\mathsf{AC}}}{[a]_{\{\mathrm{R}_1,\mathrm{Q}_1\}}\mathrm{P}_1 \to [a]_{\{\mathrm{R}_1 \land \mathsf{T},\mathrm{Q}_2 \land \mathsf{T}\}}\mathrm{P}_2}\,\land\mathsf{T}}{\dfrac{[a]_{\{\mathrm{R}_1,\mathrm{Q}_1\}}\mathrm{P}_1 \to [a]_{\{\mathsf{T},\psi\}}\mathsf{T} \land [a]_{\{\mathrm{R}_1 \land \mathsf{T},\mathrm{Q}_2 \land \mathsf{T}\}}\mathrm{P}_2}{\dfrac{[a]_{\{\mathrm{R}_1,\mathrm{Q}_1\}}\mathrm{P}_1 \to [a]_{\{\mathrm{R}_2,\mathrm{Q}_2 \land \mathsf{T}\}}\mathrm{P}_2}{[a]_{\{\mathrm{R}_1,\mathrm{Q}_1\}}\mathrm{P}_1 \to [a]_{\{\mathrm{R}_2,\mathrm{Q}_2\}}\mathrm{P}_2}\,\land\mathsf{T}}\,[]\mathrm{W}_{\mathsf{A}}}\,\land\mathrm{R}}{}$$

$[]_{\mathsf{AC}}\wedge$: The implication ($\to$) can be easily derived using rule $\mathrm{M}[\cdot]_{\mathsf{AC}}$. The other direction is derived below, where Prop marks propositional reasoning. The proof uses currying Curry, which can can be easily derived by Prop.

$$\dfrac{\dfrac{\dfrac{\mathrm{Prop}\,\dfrac{*}{\mathrm{Q}_1 \to (\mathrm{Q}_2 \to \mathrm{Q}_1 \wedge \mathrm{Q}_2)} \qquad \dfrac{*}{\mathrm{P}_1 \to (\mathrm{P}_2 \to \mathrm{P}_1 \wedge \mathrm{P}_2)}\,\mathrm{Prop}}{[a]_{\{\mathrm{R},\mathrm{Q}_1\}}\mathrm{P}_1 \to [a]_{\{\mathrm{R},\mathrm{Q}_2 \to \mathrm{Q}_1 \wedge \mathrm{Q}_2\}}(\mathrm{P}_2 \to \mathrm{P}_1 \wedge \mathrm{P}_2)}\,\mathrm{M}[\cdot]_{\mathsf{AC}}}{[a]_{\{\mathrm{R},\mathrm{Q}_1\}}\mathrm{P}_1 \to ([a]_{\{\mathrm{R},\mathrm{Q}_2\}}\mathrm{P}_2 \to [a]_{\{\mathrm{R},\mathrm{Q}_1 \wedge \mathrm{Q}_2\}}(\mathrm{P}_1 \wedge \mathrm{P}_2))}\,\mathrm{K}_{\mathsf{AC}}}{[a]_{\{\mathrm{R},\mathrm{Q}_1\}}\mathrm{P}_1 \wedge [a]_{\{\mathrm{R},\mathrm{Q}_2\}}\mathrm{P}_2 \to [a]_{\{\mathrm{R},\mathrm{Q}_1 \wedge \mathrm{Q}_2\}}(\mathrm{P}_1 \wedge \mathrm{P}_2)}\,\mathrm{Curry}$$

□

Algebra of Traces. Fig. 4 gives simple algebraic laws for step-wise simplification of trace terms. In contrast to the schematic algebra of traces in our previous report [6], the laws in Fig. 4 are flat axioms without side conditions. The axioms use that a symbolic representation of (co)finite sets can be given and finitely axiomatized, especially in axiom $\downarrow\cap$, axiom $\downarrow\in$, and axiom $\downarrow\notin$.

$\cdot\downarrow$	$(f^{\mathcal{T}} \cdot g^{\mathcal{T}}) \downarrow Y = f^{\mathcal{T}} \downarrow Y \cdot g^{\mathcal{T}} \downarrow Y$	$\cdot\mathrm{A}$	$(f_1^{\mathcal{T}} \cdot f_2^{\mathcal{T}}) \cdot f_3^{\mathcal{T}} = f_1^{\mathcal{T}} \cdot (f_2^{\mathcal{T}} \cdot f_3^{\mathcal{T}})$
$\downarrow\cap$	$(f^{\mathcal{T}} \downarrow Y') \downarrow Y = f^{\mathcal{T}} \downarrow (Y' \cap Y)$	$\cdot\mathrm{N}$	$f^{\mathcal{T}} \cdot \epsilon = f^{\mathcal{T}} = \epsilon \cdot f^{\mathcal{T}}$
$\epsilon\downarrow$	$\epsilon \downarrow Y = \epsilon$	$\downarrow\in$	$\mathrm{ch} \in Y \to \langle \mathrm{ch}, f^{\mathbb{R}}, g^{\mathbb{R}} \rangle \downarrow Y = \langle \mathrm{ch}, f^{\mathbb{R}}, g^{\mathbb{R}} \rangle$
val	$\mathtt{val}(\langle \mathrm{ch}, f^{\mathbb{R}}, g^{\mathbb{R}} \rangle) = f^{\mathbb{R}}$	$\downarrow\notin$	$ch \notin Y \to \langle \mathrm{ch}, f^{\mathbb{R}}, g^{\mathbb{R}} \rangle \downarrow Y = \epsilon$
time	$\mathtt{time}(\langle \mathrm{ch}, f^{\mathbb{R}}, g^{\mathbb{R}} \rangle) = g^{\mathbb{R}}$	$\lvert\cdot\rvert\geq\mathbf{0}$	$\lvert f^{\mathcal{T}} \rvert \geq 0$
chan	$\mathtt{chan}(\langle \mathrm{ch}, f^{\mathbb{R}}, g^{\mathbb{R}} \rangle) = \mathrm{ch}$	$\lvert\cdot\rvert\mathbf{+1}$	$\lvert f^{\mathcal{T}} \cdot \langle \mathrm{ch}, f^{\mathbb{R}}, g^{\mathbb{R}} \rangle \rvert = \lvert f^{\mathcal{T}} \rvert + 1$

$\mathrm{acc}_{=}$ $\lvert f^{\mathcal{T}} \rvert = f^{\mathbb{N}} \to (f^{\mathcal{T}} \cdot \langle \mathrm{ch}, f^{\mathbb{R}}, g^{\mathbb{R}} \rangle)[f^{\mathbb{N}}] = \langle \mathrm{ch}, f^{\mathbb{R}}, g^{\mathbb{R}} \rangle$

$\mathrm{acc}_{>}$ $\lvert f^{\mathcal{T}} \rvert > f^{\mathbb{N}} \to (f^{\mathcal{T}} \cdot \langle \mathrm{ch}, f^{\mathbb{R}}, g^{\mathbb{R}} \rangle)[f^{\mathbb{N}}] = f^{\mathcal{T}}[f^{\mathbb{N}}]$

Fig. 4: Axiomatic algebra of traces

Axiomatization of (Co)finite Sets. Strictly speaking, the calculus in Fig. 3 still has schematic occurrences of (co)finite sets. As suggested by Remark 20, this is easily fixed using symbolic (co)finite sets together with a non-schematic axiomatization. The class $C(E_1, \dots, E_n)$ of (co)finite sets over the (co)finite sets $E_1, \dots, E_n$ of atoms, has the following syntax

$$C_1, C_2 ::= \{e_i\} \mid \bot_{\mathrm{Set}} \mid \top^1_{\mathrm{Set}} \mid \dots \mid \top^n_{\mathrm{Set}} \mid C_1 \cap C_2 \mid C_1 \setminus C_2,$$

where $e_i \in E_i$ is any atom for any $1 \leq i \leq n$, symbol $\bot_{\mathrm{Set}}$ represents the empty set, $\top^i_{\mathrm{Set}}$ represents all atoms of E_i for $1 \leq i \leq n$, and $\cap$ and $\setminus$ are set intersection and set difference, respectively. Other operators like union $\cup$ can be defined. The class occurring in $\mathsf{dL}_{\mathrm{CHP}}$ is $C(\Omega, V_{\mathbb{R}}, V_{\mathbb{N}}, V_{\mathcal{T}})$.

For $C(E_1, \dots, E_n)$, any $C(S_1, \dots, S_k)$ with $\{S_1, \dots, S_k\} \subseteq \{E_1, \dots E_n\}$ forms a boolean algebra with binary operator $\cap$, neutral element $\top_{\mathrm{Set}} = \top^1_{\mathrm{Set}} \cup \dots \cup \top^k_{\mathrm{Set}}$

w.r.t. $\cap$, where $\top^i_{\mathrm{Set}}$ represents S_i, and the unary operation $\top_{\mathrm{Set}} \setminus \cdot$ of parameter $\cdot$. Laws for the boolean algebra can be adopted to axiomatize $C(S_1, \ldots, S_k)$.

In formulas, (co)finite sets can be compared $C_1 = C_2$, and we include the element relation $e \in Y$. Axioms for the element relation over (co)finite sets unroll the relation into a finite conjunction as follows:

$$\begin{array}{ll} \neg e \in \bot_{\mathrm{Set}} & e \in (\{\tilde{e}\} \cap C) \leftrightarrow e = \tilde{e} \wedge e \in C \\ e_i \in \top^i_{\mathrm{Set}} & e \in (C_1 \setminus C_2) \leftrightarrow e \in C_1 \wedge \neg e \in C_2 \\ e_i \notin \top^j_{\mathrm{Set}} \text{ for } i \neq j & \end{array}$$

Equality is axiomatized in terms of the extensionality principle as usual:

$$C_1 = C_2 \leftrightarrow \forall e\, (e \in C_1 \leftrightarrow e \in C_2)$$

References

[1] Apt, K.R., de Boer, F.S., Olderog, E.R.: Verification of Sequential and Concurrent Programs. Springer, 3rd edn. (2010). doi: 10.1007/978-1-84882-745-5

[2] Apt, K.R., Francez, N., de Roever, W.P.: A proof system for communicating sequential processes. ACM Transactions on Programming Languages and Systems **2**(3), 359–385 (1980). doi: 10.1145/357103.357110

[3] Armstrong, A., Gomes, V.B.F., Struth, G.: Algebras for program correctness in Isabelle/HOL. In: Höfner, P., Jipsen, P., Kahl, W., Müller, M.E. (eds.) Proc. 14th Intl. Conf. Relational and Algebraic Methods in Computer Science (RAMiCS). LNCS, vol. 8428, pp. 49–64. Springer (2014). doi: 10.1007/978-3-319-06251-8_4

[4] Armstrong, P.J., Lowe, G., Ouaknine, J., Roscoe, B.: Model checking timed CSP. In: Voronkov, A., Korovina, M.V. (eds.) HOWARD-60: A Festschrift on the Occasion of Howard Barringer's 60th Birthday, EPiC Series in Computing, vol. 42, pp. 13–33. EasyChair (2014). doi: 10.29007/6fqk

[5] Blom, S., Darabi, S., Huisman, M., Oortwijn, W.: The VerCors tool set: Verification of parallel and concurrent software. In: Polikarpova, N., Schneider, S.A. (eds.) Proc. 13th Intl. Conf. Integrated Formal Methods (IFM). LNCS, vol. 10510, pp. 102–110. Springer (2017). doi: 10.1007/978-3-319-66845-1_7

[6] Brieger, M., Mitsch, S., Platzer, A.: Dynamic logic of communicating hybrid programs. CoRR **abs/2302.14546** (2023). doi: 10.48550/arXiv.2302.14546

[7] Church, A.: Introduction to Mathematical Logic. Princeton University Press, Princeton (1956)

[8] Cohen, E., Dahlweid, M., Hillebrand, M.A., Leinenbach, D., Moskal, M., Santen, T., Schulte, W., Tobies, S.: VCC: A practical system for verifying concurrent C. In: Berghofer, S., Nipkow, T., Urban, C., Wenzel, M. (eds.) Proc. 22nd Intl. Conf. Theorem Proving in Higher Order Logics (TPHOLs). LNCS, vol. 5674, pp. 23–42. Springer (2009). doi: 10.1007/978-3-642-03359-9_2

[9] Cong, X., Yu, H., Xu, X.: Verification of hybrid chi model for cyber-physical systems using PHAVer. In: Barolli, L., You, I., Xhafa, F., Leu, F., Chen, H. (eds.) Proc. 7th Intl. Conf. Innovative Mobile and Internet Services in Ubiquitous Computing, (IMIS). pp. 122–128. IEEE Computer Society (2013). doi: 10.1109/IMIS.2013.29

[10] Fulton, N., Mitsch, S., Quesel, J.D., Völp, M., Platzer, A.: KeYmaera X: An axiomatic tactical theorem prover for hybrid systems. In: Felty, A., Middeldorp, A. (eds.) CADE. LNCS, vol. 9195, pp. 527–538. Springer, Berlin (2015). doi: 10.1007/978-3-319-21401-6_36

[11] Gibson-Robinson, T., Armstrong, P.J., Boulgakov, A., Roscoe, A.W.: FDR3 - A modern refinement checker for CSP. In: Ábrahám, E., Havelund, K. (eds.) Proc. 20th Intl. Conf. Tools and Algorithms for the Construction and Analysis of Systems (TACAS). LNCS, vol. 8413, pp. 187–201. Springer (2014). doi: 10.1007/978-3-642-54862-8_13

[12] Guelev, D.P., Wang, S., Zhan, N.: Compositional Hoare-style reasoning about hybrid CSP in the duration calculus. In: Larsen, K.G., Sokolsky, O., Wang, J. (eds.) Proc. 3rd Intl. Symp. Dependable Software Engineering. Theories, Tools, and Applications (SETTA). LNCS, vol. 10606, pp. 110–127. Springer (2017). doi: 10.1007/978-3-319-69483-2_7

[13] Harel, D.: First-Order Dynamic Logic, LNCS, vol. 68. Springer (1979). doi: 10.1007/3-540-09237-4

[14] Hoare, C.A.R.: Communicating sequential processes. Communications of the ACM **21**(8), 666–677 (1978). doi: 10.1145/359576.359585

[15] Jacobs, B., Smans, J., Philippaerts, P., Vogels, F., Penninckx, W., Piessens, F.: VeriFast: A powerful, sound, predictable, fast verifier for C and Java. In: Bobaru, M.G., Havelund, K., Holzmann, G.J., Joshi, R. (eds.) Proc. 3rd Intl. Symp. NASA Formal Methods (NFM). LNCS, vol. 6617, pp. 41–55. Springer (2011). doi: 10.1007/978-3-642-20398-5_4

[16] Jifeng, H.: A classical mind: essays in honour of C. A. R. Hoare, chap. From CSP to Hybrid Systems, pp. 171–189. Prentice Hall International (1994)

[17] Kirchner, F., Kosmatov, N., Prevosto, V., Signoles, J., Yakobowski, B.: Frama-C: A software analysis perspective. Formal Aspects Comput. **27**(3), 573–609 (2015). doi: 10.1007/s00165-014-0326-7

[18] Levin, G., Gries, D.: A proof technique for communicating sequential processes. Acta Informatica **15**(3), 281–302 (1981). doi: 10.1007/BF00289266

[19] Liu, J., Lv, J., Quan, Z., Zhan, N., Zhao, H., Zhou, C., Zou, L.: A calculus for hybrid CSP. In: Ueda, K. (ed.) Proc. 8th Asian Symp. Programming Languages and Systems (APLAS). LNCS, vol. 6461, pp. 1–15. Springer (2010). doi: 10.1007/978-3-642-17164-2_1

[20] Man, K.L., Reniers, M.A., Cuijpers, P.J.L.: Case studies in the hybrid process algebra HyPA. Int. J. Softw. Eng. Knowl. Eng. **15**(2), 299–306 (2005). doi: 10.1142/S0218194005002385

[21] Misra, J., Chandy, K.M.: Proofs of networks of processes. IEEE Transactions on Software Engineering **7**(4), 417–426 (1981). doi: 10.1109/TSE.1981.230844

[22] Mitsch, S., Platzer, A.: A retrospective on developing hybrid systems provers in the KeYmaera family - A tale of three provers. In: Ahrendt, W., Beckert, B., Bubel, R., Hähnle, R., Ulbrich, M. (eds.) Deductive Software Verification: Future Perspectives - Reflections on the Occasion of 20 Years of KeY, LNCS, vol. 12345, pp. 21–64. Springer (2020). doi: 10.1007/978-3-030-64354-6_2

[23] Müller, A., Mitsch, S., Retschitzegger, W., Schwinger, W., Platzer, A.: A component-based approach to hybrid systems safety verification. In: Abraham, E., Huisman, M. (eds.) IFM. LNCS, vol. 9681, pp. 441–456. Springer (2016). doi: 10.1007/978-3-319-33693-0_28

[24] Nieto, L.P.: Verification of parallel programs with the Owicki-Gries and Rely-Guarantee methods in Isabelle/HOL. Ph.D. thesis, Technical University Mu-

nich, Germany (2002), http://tumb1.biblio.tu-muenchen.de/publ/diss/in/2002/prensa_nieto.html

[25] Nipkow, T., Nieto, L.P.: Owicki/Gries in Isabelle/HOL. In: Finance, J. (ed.) Proc. 2nd Intl. Conf. Fundamental Approaches to Software Engineering (FASE). LNCS, vol. 1577, pp. 188–203. Springer (1999). doi: 10.1007/978-3-540-49020-3_13

[26] Owicki, S.S., Gries, D.: An axiomatic proof technique for parallel programs I. Acta Informatica **6**, 319–340 (1976). doi: 10.1007/BF00268134

[27] Platzer, A.: Differential dynamic logic for hybrid systems. J. Autom. Reas. **41**(2), 143–189 (2008). doi: 10.1007/s10817-008-9103-8

[28] Platzer, A.: Differential game logic. ACM Trans. Comput. Log. **17**(1), 1:1–1:51 (2015). doi: 10.1145/2817824

[29] Platzer, A.: A complete uniform substitution calculus for differential dynamic logic. J. Autom. Reas. **59**(2), 219–265 (2017). doi: 10.1007/s10817-016-9385-1

[30] Platzer, A.: Uniform substitution for differential game logic. In: Galmiche, D., Schulz, S., Sebastiani, R. (eds.) IJCAR. LNCS, vol. 10900, pp. 211–227. Springer (2018). doi: 10.1007/978-3-319-94205-6_15

[31] Platzer, A.: Uniform substitution at one fell swoop. In: Fontaine, P. (ed.) CADE. LNCS, vol. 11716, pp. 425–441. Springer (2019). doi: 10.1007/978-3-030-29436-6_25

[32] Platzer, A., Tan, Y.K.: Differential equation invariance axiomatization. J. ACM **67**(1), 6:1–6:66 (2020). doi: 10.1145/3380825

[33] Presburger, M.: Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. In: Comptes Rendus du I congrès de Mathématiciens des Pays Slaves (1931)

[34] Rice, H.G.: Classes of recursively enumerable sets and their decision problems. Transactions of the American Mathematical Society **74**(2), 358–366 (1953). doi: 10.2307/1990888

[35] de Roever, W.P., de Boer, F.S., Hannemann, U., Hooman, J.J.M., Lakhnech, Y., Poel, M., Zwiers, J.: Concurrency Verification: Introduction to Compositional and Noncompositional Methods, Cambridge Tracts in Theoretical Computer Science, vol. 54. Cambridge University Press (2001)

[36] Schellhorn, G., Bodenmüller, S., Bitterlich, M., Reif, W.: Software & system verification with KIV. LNCS **13360**, 408 – 436 (2022). doi: 10.1007/978-3-031-08166-8_20

[37] Sheng, H., Bentkamp, A., Zhan, B.: HHLPy: Practical verification of hybrid systems using hoare logic. In: Chechik, M., Katoen, J., Leucker, M. (eds.) Proc. 25th Intl. Symp. Formal Methods (FM). LNCS, vol. 14000, pp. 160–178. Springer (2023). doi: 10.1007/978-3-031-27481-7_11

[38] Shi, L., Zhao, Y., Liu, Y., Sun, J., Dong, J.S., Qin, S.: A UTP semantics for communicating processes with shared variables and its formal encoding in PVS. Formal Aspects Comput. **30**(3-4), 351–380 (2018). doi: 10.1007/s00165-018-0453-7

[39] Smans, J., Vanoverberghe, D., Devriese, D., Jacobs, B., Piessens, F.: Shared boxes: Rely-guarantee reasoning in VeriFast. Tech. rep., Katholieke Universiteit Leuven, Netherlands (2014), https://lirias.kuleuven.be/handle/123456789/456819

[40] Song, H., Compton, K.J., Rounds, W.C.: SPHIN: A model checker for reconfigurable hybrid systems based on SPIN. In: Lazic, R., Nagarajan, R. (eds.) Proc. 5th Intl. Workshop Automated Verification of Critical Systems (AVoCS). ENTCS, vol. 145, pp. 167–183. Elsevier (2005). doi: 10.1016/j.entcs.2005.10.011

[41] Tarski, A.: A Decision Method for Elementary Algebra and Geometry. University of California Press, Berkeley, 2nd edn. (1951). doi: 10.1525/9780520348097

[42] Wang, S., Zhan, N., Guelev, D.P.: An assume/guarantee based compositional calculus for hybrid CSP. In: Agrawal, M., Cooper, S.B., Li, A. (eds.) Proc. 9th Conf. Theory and Applications of Models of Computation (TAMC). LNCS, vol. 7287, pp. 72–83. Springer (2012). doi: 10.1007/978-3-642-29952-0_13

[43] Wang, S., Zhan, N., Zou, L.: An improved HHL prover: An interactive theorem prover for hybrid systems. In: Butler, M.J., Conchon, S., Zaïdi, F. (eds.) Proc. 17th Intl. Conf. Formal Engineering Methods (ICFEM). LNCS, vol. 9407, pp. 382–399. Springer (2015). doi: 10.1007/978-3-319-25423-4_25

[44] Xu, Q., de Roever, W.P., He, J.: The rely-guarantee method for verifying shared variable concurrent programs. Formal Aspects of Computing **9**(2), 149–174 (1997). doi: 10.1007/BF01211617

[45] Zou, L., Lv, J., Wang, S., Zhan, N., Tang, T., Yuan, L., Liu, Y.: Verifying Chinese train control system under a combined scenario by theorem proving. In: Cohen, E., Rybalchenko, A. (eds.) Proc. 5th Intl. Conf. Verified Software: Theories, Tools, Experiments (VSTTE). LNCS, vol. 8164, pp. 262–280. Springer (2013). doi: 10.1007/978-3-642-54108-7_14

[46] Zwiers, J., de Bruin, A., de Roever, W.P.: A proof system for partial correctness of dynamic networks of processes (extended abstract). In: Clarke, E.M., Kozen, D. (eds.) Proc. Carnegie Mellon Workshop Logics of Programs 1983. LNCS, vol. 164, pp. 513–527. Springer (1983). doi: 10.1007/3-540-12896-4_384

[47] Zwiers, J., de Roever, W.P., van Emde Boas, P.: Compositionality and concurrent networks: Soundness and completeness of a proofsystem. In: Brauer, W. (ed.) Proc. 12th Intl. Coll. Automata, Languages and Programming (ICALP). LNCS, vol. 194, pp. 509–519. Springer (1985). doi: 10.1007/BFb0015776