Technical Appendix

This document is structured as follows. First, we provide proofs for the propositions of the paper. Second, we outline the missing steps of the proof of equivalence between the Boolean FVP definition and the simple FVP definition of $meeting(P_1, P_2) = interacting$ (see section 'Equivalence Proof for Running Example' of the paper).

1 Proofs of Propositions

Proposition 1 (Equivalence between Statically Determined and Boolean FVP Definitions)**.** For each statically determined FVP definition, there is an equivalent Boolean FVP definition, and vice versa. ⧫

Proof. **From Statically Determined to Boolean FVP Definition.** We prove that every statically determined FVP definition can be translated into a Boolean FVP definition. Consider an arbitrary statically determined FVP definition for FVP $F = V$, abiding by the following rule-schema:

$$\begin{aligned}&\mathsf{holdsFor}(F = V,\ I_{n+m}) \leftarrow\\ &\quad \mathsf{holdsFor}(F_1 = V_1,\ I_1)[[, \mathsf{holdsFor}(F_2 = V_2,\ I_2),\ \dots\\ &\quad \mathsf{holdsFor}(F_n = V_n,\ I_n), \mathsf{intervalConstruct}(L_1,\ I_{n+1}),\ \dots\\ &\quad \mathsf{intervalConstruct}(L_m,\ I_{n+m})]].\end{aligned} \tag{1}$$

Provided that $\mathsf{holdsFor}(F = V, I_{n+m})$, $F = V$ holds at time-point T iff there is an interval i in list I_{n+m} that contains time-point T. In other words:

$$\mathsf{holdsAt}(F = V, T) \Leftrightarrow \exists i \in I_{n+m} : T \in i. \tag{2}$$

Following the definitions of the interval manipulation construct of RTEC, we transform expression $\exists i \in I_{n+k} : T \in i$, where $k \in [m]$[1], into a Boolean combination of conditions concerning time-point T and the lists of intervals of the FVPs in the body of rule (1). In the case of $\mathsf{union\_all}([I_{j_1}, \dots, I_{j_l}], I_{n+k})$, where $1 \leq q < l$ and $1 \leq j_q < n{+}k$, we have:

$$\exists i \in I_{n+k} : T \in i \Leftrightarrow \exists i_{j_1} \in I_{j_1} : T \in i_{j_1} \vee \dots \vee \exists i_{j_l} \in I_{j_l} : T \in i_{j_l} \tag{3}$$

In the case of $\mathsf{intersect\_all}([I_{j_1}, \dots, I_{j_l}], I_{n+k})$, we have:

$$\exists i \in I_{n+k} : T \in i \Leftrightarrow \exists i_{j_1} \in I_{j_1} : T \in i_{j_1} \wedge \dots \wedge \exists i_{j_l} \in I_{j_l} : T \in i_{j_l} \tag{4}$$

[1] We use $[k]$ to denote the set of the first k positive integers.

In the case of $\mathsf{relative\_complement\_all}(I_{j_0}, [I_{j_1}, \ldots, I_{j_l}], I_{n+k})$, we have:

$$\begin{aligned} &\exists i \in I_{n+k} : T \in i \Leftrightarrow \\ &\exists i_{j_0} \in I_{j_0} : T \in i_{j_0} \wedge \nexists i_{j_1} \in I_{j_1} : T \in i_{j_1} \wedge \cdots \wedge \nexists i_{j_l} \in I_{j_l} : T \in i_{j_l} \end{aligned} \tag{5}$$

We will demonstrate that $\exists i \in I_{n+k} : T \in i$, where $1 \leq k \leq m$, holds iff a Boolean combination of $\mathsf{holdsAt}(F_j = V_j, T)$ conditions, where $j \in [n]$, is satisfied using an inductive proof on k.

Base case: $k = 1$. List I_{n+1} is the output of the first interval manipulation construct in the conditions of rule (1). Therefore, according to equivalences (3)–(5), $\exists i \in I_{n+1} : T \in i$ is equivalent with a Boolean combination of expressions $\exists i \in I_j : T \in i$, where $j \in [n]$. Since there is no interval construct before the condition defining I_{n+1} in the body of rule (1), each list of intervals I_j is defined by a $\mathsf{holdsFor}(F_j = V_j, I_j)$ condition. Therefore, we have:

$$\exists i \in I_j : T \in i \Leftrightarrow \mathsf{holdsAt}(F_j = V_j, T) \tag{6}$$

By substituting each $\exists i \in I_j : T \in i$ expression in the right-hand side of equivalences (3)–(5) with $\mathsf{holdsAt}(F_j = V_j, T)$, based on equivalence (6), we prove that $\exists i \in I_{n+1} : T \in i$ is equivalent with a Boolean combination of $\mathsf{holdsAt}(F_j = V_j, T)$ conditions, where $j \in [n]$.

Inductive Step: Suppose that expression $\exists i \in I_{n+k} : T \in i$, where $1 \leq k < m$, is equivalent with a Boolean combination of $\mathsf{holdsAt}(F_j = V_j, T)$ conditions, where $j \in [n]$. We prove that expression $\exists i \in I_{n+k+1} : T \in i$ is equivalent with a Boolean combination of $\mathsf{holdsAt}(F_j = V_j, T)$ conditions. List I_{n+k+1} is the output of an interval manipulation construct in the conditions of rule (1). According to equivalences (3)–(5), $\exists i \in I_{n+k+1} : T \in i$ is equivalent with a Boolean combination of expressions $\exists i \in I_{n'} : T \in i$, where $n' \in [n{+}k]$. Based on our inductive assumption, each one of these expressions can be written as an equivalent Boolean combination of $\mathsf{holdsAt}(F_j = V_j, T)$ conditions, where $j \in [n]$. Therefore, expression $\exists i \in I_{n+k+1} : T \in i$ is equivalent with a Boolean combination of $\mathsf{holdsAt}(F_j = V_j, T)$, where $j \in [n]$.

By induction, $\exists i \in I_{n+m} : T \in i$ is equivalent with a Boolean combination C of $\mathsf{holdsAt}(F_j = V_j, T)$ conditions, where $j \in [n]$. Therefore, based on equivalence (2), $\mathsf{holdsAt}(F = V, T)$ is equivalent with expression C.

In order to generate a Boolean FVP definition p for $F = V$, we write expression C in disjunctive normal form (DNF), i.e., C^{DNF}. The Boolean FVP definition p that is equivalent with C^{DNF}, and thus equivalent with the arbitrary statically determined FVP definition we started with, is: $\bigvee_{d \in C^{DNF}} \wedge_{l \in d} l'$, where, for each literal l in a disjunct d of C^{DNF}, l' is $F{=}V$, if l is $\mathsf{holdsAt}(F = V, T)$, and l' is $\neg(F = V)$, if l is $\neg$ $\mathsf{holdsAt}(F = V, T)$. Given an arbitrary statically determined FVP definition r, we constructed a Boolean FVP definition that is equivalent with r. Therefore, for each statically determined FVP definition, there is an equivalent Boolean FVP definition.

From Boolean to Statically Determined FVP Definition. We prove that every Boolean FVP definition can be translated into a statically determined FVP definition. Consider an arbitrary Boolean FVP definition p for FVP $F{=}V$, containing m disjuncts. Each disjunct d_i, where $1 \leq i \leq m$, of p contains n_i FVP literals. FVP literal l_{ij} of d_i, where $1 \leq j \leq n_i$, is $[\neg](F_{ij} = V_{ij})$, where '[ ]' denotes that $\neg$ is optional. We construct a statically determined FVP definition

r that is equivalent with definition p. r has head $\mathsf{holdsFor}(F = V, I)$ and follows schema (1).

Based on definition p, $\mathsf{holdsAt}(F = V, T)$ is equivalent with the following Boolean expression:

$$\begin{aligned}&([\neg]\mathsf{holdsAt}(F_{11} = V_{11}, T) \land \dots \land [\neg]\mathsf{holdsAt}(F_{1n_1} = V_{1n_1}, T))\lor\\&([\neg]\mathsf{holdsAt}(F_{21} = V_{21}, T) \land \dots \land [\neg]\mathsf{holdsAt}(F_{2n_2} = V_{2n_2}, T)) \lor \dots \lor\\&([\neg]\mathsf{holdsAt}(F_{m1} = V_{m1}, T) \land \dots \land [\neg]\mathsf{holdsAt}(F_{mn_m} = V_{mn_m}, T))\end{aligned} \tag{7}$$

For each $\mathsf{holdsAt}(F_{jl_j} = V_{jl_j}, T)$ atom, where $j \in [m]$ and $l_j \in [n_j]$, in expression (7), we add a $\mathsf{holdsFor}(F_{jl_j} = V_{jl_j}, I_{jl_j})$ condition in the body of r. Moreover, for each $\neg\mathsf{holdsAt}(F_{jl_j} = V_{jl_j}, T)$ atom in expression (7), we add conditions $\mathsf{holdsFor}(F_{jl_j} = V_{jl_j}, I'_{jl_j})$ and $\mathsf{relative\_complement\_all}([i_w], [I'_{jl_j}], I_{jl_j})$ in the body of r, where i_w denotes the window of RTEC. As a result, based on equivalence (2), we have:

$$\mathsf{holdsAt}(F_{jl_j} = V_{jl_j}, T) \Leftrightarrow \exists i \in I_{jl_j} : T \in i \tag{8}$$

$$\neg\mathsf{holdsAt}(F_{jl_j} = V_{jl_j}, T) \Leftrightarrow \nexists i \in I'_{jl_j} : T \in i \Leftrightarrow \exists i \in I_{jl_j} : T \in i \tag{9}$$

Based on equivalences (8)–(9), expression (7) can be rewritten as follows:

$$\begin{aligned}&(\exists i_{11} \in I_{11} : T \in i_{11} \land \dots \land \exists i_{1n_1} \in I_{1n_1} : T \in i_{1n_1})\lor\\&(\exists i_{21} \in I_{21} : T \in i_{21} \land \dots \land \exists i_{2n_2} \in I_{2n_2} : T \in i_{2n_2}) \lor \dots \lor\\&(\exists i_{m1} \in I_{m1} : T \in i_{m1} \land \dots \land \exists i_{mn_m} \in I_{mn_m} : T \in i_{mn_m})\end{aligned} \tag{10}$$

For each disjunct in expression (10), we add an $\mathsf{intersect\_all}([I_{j1}, \dots, I_{jn_j}], I_j)$ condition in the body of r. List I_j satisfies the following condition:

$$\exists i \in I_j : T \in i \Leftrightarrow \exists i_{j1} \in I_{j1} : T \in i_{j1} \land \dots \land \exists i_{jn_j} \in I_{jn_j} : T \in i_{jn_j} \tag{11}$$

Based on equivalence (11), expression (10) can be rewritten as follows:

$$\exists i_1 \in I_1 : T \in i_1 \lor \dots \lor \exists i_m \in I_m : T \in i_m \tag{12}$$

Following expression (12), we add condition $\mathsf{union\_all}([I_1, \dots, I_m], I)$ in the body of r. List I satisfies the following condition:

$$\exists i \in I : T \in i \Leftrightarrow \exists i_1 \in I_1 : T \in i_1 \lor \dots \lor \exists i_m \in I_m : T \in i_m \tag{13}$$

According to definition r, with head $\mathsf{holdsFor}(F = V, I)$, we have $\mathsf{holdsAt}(F = V, T)$ iff $i \in I : T \in i$ (see equivalence (2)). Moreover, following equivalences (13), (11) and (8)–(9), $i \in I : T \in i$ is satisfied iff $F{=}V$ holds at T based on definition p. Therefore, definitions r and p are equivalent.

Given an arbitrary Boolean FVP definition p, we constructed a statically determined FVP definition that is equivalent with p. Therefore, for each Boolean FVP definition, there is an equivalent statically determined FVP definition. □

Before demonstrating the proof for Proposition 2 of the paper, we present an auxiliary definition and two lemmas that are useful for the proof.

We employ the notion of a guard condition symmetric rule-set that is *fully-guarded*, i.e., contains all the guard conditions that are possible according to Definition 8 of the paper.

Definition 1 (Fully-Guarded Rule-Set)**.** Consider a guard condition symmetric rule-set R, where $G^{1}, G^{2}, \dots, G^{n}$ are the guard condition sets of the sets $R^{1}, R^{2}, \dots, R^{n}$ in the complete partition of R. We say that R is fully guarded if $\forall i \in \{1, \dots, n\}$, $\forall c \in G^{j}$, where $j \neq i$, $\exists r \in R^{i}$, such that r contains guard condition c. ■

Definition 1 expresses that a fully-guarded rule-set R is maximal in terms of guard conditions, i.e., each set in the complete partition of R contains every possible guard condition in the guard condition sets of the remaining sets in the partition. The role of guard conditions is to prevent redundant initiations/terminations. For this reason, the corollary below follows directly from Definition 1:

Corollary 1. Consider a Boolean representation symmetric definition (R^{s}, R^{e}) for FVP $F{=}V$, where sets R^{s} and R^{e} are fully-guarded. A rule in R^{s} (resp. R^{e}) may only fire when $F{=}V$ does not hold (holds). △

The following lemma establishes a relationship between the initiations and the terminations induced by a Boolean representation symmetric definition (R^{s}, R^{e}) with fully-guarded sets R^{s} and R^{e}, and the Boolean representations of R^{s} and R^{e}.

Lemma 1. Consider a Boolean representation symmetric definition (R^{s}, R^{e}) for FVP $F{=}V$, where sets R^{s} and R^{e} are fully-guarded, and the Boolean representations p and $\overline{p}$ of R^{s} and R^{e}. $F{=}V$ is initiated (resp. terminated) at time-point $T{-}1$ according to R^{s} (R^{e}) iff Boolean FVP definition $\overline{p}$ (p) holds at $T{-}1$, and p ($\overline{p}$) holds at T. ▲

Proof. Suppose that $R^{s1}, R^{s2}, \dots, R^{sm}$ and $R^{e1}, R^{e2}, \dots, R^{em'}$ are the complete partitions of R^{s} and R^{e}, and that U^{i} (resp. U'^{i}) contains the singed FVP literals in the inertial conditions of R^{si} (R^{ei}). Based on the construction of a Boolean representation (Definition 9 of the paper), p (resp. $\overline{p}$) is in DNF and, for each R^{si} (R^{ei}), contains one disjunct p^{i}, denoting the conjunction of the FVP literals in U^{i} (U'^{i}). Moreover, based on Boolean representation symmetry, $\overline{p}$ is $\neg p$. We demonstrate that $\mathsf{initiatedAt}(F = V, T{-}1)$ iff p does not hold at $T{-}1$, i.e., $\neg p(T{-}1)$, and p holds at T, i.e., $p(T)$. We have:

$$\mathsf{initiatedAt}(F = V, T{-}1) \xLeftrightarrow{\text{complete partition}}$$

$$\exists i \in [m] : (\exists r \in R^{si} : r \text{ fires at } T{-}1) \xLeftrightarrow{r \text{ is fully-guarded} \rightarrow F{=}V \text{ does not hold at } T{-}1}$$

$$(\exists i \in [m] : (\exists r \in R^{si} : r \text{ fires at } T{-}1)) \wedge \overline{p}(T{-}1) \xLeftrightarrow{\forall l \in U^{i}: l \text{ holds at } T \text{ due to } r}$$

$$(\exists i \in [m] : (\forall l \in U^{i} : l \text{ holds at } T)) \wedge \overline{p}(T{-}1) \xLeftrightarrow{\text{Boolean symmetry}}$$

$$(\exists i \in [m] : (\forall l \in U^{i} : l \text{ holds at } T)) \wedge (\neg\, p(T{-}1)) \xLeftrightarrow{\forall p^{i} \in p, \forall l \in p^{i}: l \text{ holds at } T}$$

$$p(T) \wedge \neg\, p(T{-}1)$$

The proof showing that $F{=}V$ is terminated at time-point $T{-}1$ according to R^{e} iff Boolean FVP definition p holds at $T{-}1$ and $\neg p$ holds at T is similar, and thus omitted. □

Lemma 2. Consider a guard condition symmetric and Boolean representation symmetric definition (R^s, R^e) for FVP $F{=}V$ and the Boolean representation symmetric definition (R^{gs}, R^{ge}), where R^{gs} and R^{ge} are fully-guarded. R^{gs} and R^{ge} are constructed by adding the appropriate rules and guard conditions in R^s and R^e, following Definition 1, in order to get fully-guarded sets. Definitions (R^s, R^e) and (R^{gs}, R^{ge}) are equivalent. ▲

Proof. Consider the complete partitions $R^{s1}, \dots, R^{sm}$ and $R^{gs1}, \dots, R^{gsm}$ of R^s and R^{gs}, and the complete partitions $R^{e1}, \dots, R^{em'}$ and $R^{ge1}, \dots, R^{gem'}$ of R^e and R^{ge}. First, we show that definition (R^s, R^e) is equivalent to definition (R^{gs}, R^e). Since R^{gsi}, where $i \in [m]$, extends R^{si} with only guard conditions, R^{gsi} computes an initiation T only if R^{si} computes T, i.e., the initiations computed by R^{gsi} is a subset of the initiations computed by R^{si}. Consider an initiation T that is computed by R^{si} and not by R^{gsi}. We show that T is a redundant initiation, i.e., $F{=}V$ holds at T. Since the guard conditions in R^{gsi} prevented the computation of T, there is a set R^{gsj} whose guard conditions are all satisfied at T. Thus, we have:

$$\begin{aligned}
&\exists T' < T : (\mathsf{initiatedAt}(F = V,\, T') \text{ is computed by } R^{gsj}, \text{where } j \neq i) \wedge \\
&(\forall T'' : T' < T'' < T, \nexists k \in [m'] : R^{ek} \text{ computes } \mathsf{terminatedAt}(F = V,\, T'')) \leftrightarrow \\
&\exists T' < T : (\mathsf{initiatedAt}(F = V,\, T') \text{ is computed by } R^{sj}, \text{where } j \neq i) \wedge \\
&(\forall T'' : T' < T'' < T, \nexists k \in [m'] : R^{ek} \text{ computes } \mathsf{terminatedAt}(F = V,\, T'')) \leftrightarrow \\
&(R^s, R^e) \models \mathsf{holdsAt}(F = V,\, T)
\end{aligned}$$

The first equivalence is because R^{sj} computes strictly more initiations than R^{gsj}, while the second equivalence is due to the law of inertia. We have proven that every initiation that is computed by (R^s, R^e) and not by (R^{gs}, R^e) is redundant. Thus, (R^s, R^e) and (R^{gs}, R^e) are equivalent.

We work similarly to show that definition (R^{gs}, R^e) is equivalent with (R^{gs}, R^{ge}). Thus, we have proven that definitions (R^s, R^e) and (R^{gs}, R^{ge}) are equivalent. □

Proposition 2 (Translatable Simple FVP Definition)**.** A simple FVP definition (R^s, R^e) is translatable iff:
1. R^s and R^e are guard condition symmetric, and
2. (R^s, R^e) is Boolean representation symmetric.

When (R^s, R^e) is translatable, it is equivalent with the definition induced by the Boolean representation of R^s. ♦

Proof. **'if' direction.** Suppose that (R^s, R^e) is Boolean representation symmetric, and sets R^s and R^e are guard condition symmetric. p and $\overline{p}$ are the Boolean representations of R^s and R^e, respectively. p^i denotes the disjunct of p corresponding to set R^{si} of the complete partition of R^s. Based on Boolean representation symmetry, $\overline{p}$ is $\neg\, p$. Consider also the Boolean representation symmetric definition (R^{gs}, R^{ge}), where R^{gs} and R^{ge} are fully-guarded, and are constructed by adding the appropriate rules and guard conditions in R^s and R^e, following Definition 1, in order to get fully-guarded sets. $R^{s1}, \dots, R^{sm}$ and $R^{gs1}, \dots, R^{gsm}$ are the complete partitions of R^s and R^{gs}, and $R^{e1}, \dots, R^{em'}$ and $R^{ge1}, \dots, R^{gem'}$ are the complete partitions of R^e and R^{ge}. Based on Lemma 2, definitions (R^s, R^e) and (R^{gs}, R^{ge}) are equivalent.

We prove that (R^s, R^e) implies $\mathsf{holdsAt}(F=V,\,T)$ iff p implies $\mathsf{holdsAt}(F=V,\,T)$. We use an inductive proof on T. The base of the induction is time-point $\mathit{0}$, where without loss of generality, we assume that all definitions imply $\neg\ \mathsf{holdsAt}(F=V,\,T)$. Therefore, in the base case, (R^s, R^e) and p imply the same $\mathsf{holdsAt}$ atoms.

Below, we outline the proof for time-point T, given that (R^s, R^e) implies $\mathsf{holdsAt}(F=V,\,T{-}\mathit{1})$ iff p implies $\mathsf{holdsAt}(F=V,\,T{-}\mathit{1})$. $q(T_{\mathit{0}})$ denotes that Boolean FVP definition q holds at time-point T_{0}.

$$(R^s, R^e) \models \mathsf{holdsAt}(F=V,\,T) \xLeftrightarrow{\text{inertia}}$$

$$(\exists r \in R^s : r \text{ computes } \mathsf{initiatedAt}(F=V,\,T{-}\mathit{1}) \vee$$
$$(\mathsf{holdsAt}(F=V,\,T{-}\mathit{1}) \wedge$$
$$\neg\ (\exists r \in R^e : r \text{ computes } \mathsf{terminatedAt}(F=V,\,T{-}\mathit{1})))) \xLeftrightarrow{\text{complete partitions}}$$

$$(\exists i \in [m] : \exists r \in R^{si} : r \text{ computes } \mathsf{initiatedAt}(F=V,\,T{-}\mathit{1}) \vee$$
$$(\mathsf{holdsAt}(F=V,\,T{-}\mathit{1}) \wedge$$
$$\neg\ (\exists j \in [m'] : \exists r \in R^{ej} : r \text{ computes } \mathsf{terminatedAt}(F=V,\,T{-}\mathit{1})))) \xLeftrightarrow{\text{Lemma 2}}$$

$$(\exists i \in [m] : \exists r \in R^{gsi} : r \text{ computes } \mathsf{initiatedAt}(F=V,\,T{-}\mathit{1}) \vee$$
$$(\mathsf{holdsAt}(F=V,\,T{-}\mathit{1}) \wedge$$
$$\neg\ (\exists j \in [m'] : \exists r \in R^{gej} : r \text{ computes } \mathsf{terminatedAt}(F=V,\,T{-}\mathit{1})))) \xLeftrightarrow{\text{Lemma 1}}$$

$$((\neg\ p(T{-}\mathit{1}) \wedge p(T)) \vee$$
$$(\mathsf{holdsAt}(F=V,\,T{-}\mathit{1}) \wedge (\neg\ (p(T{-}\mathit{1}) \wedge \neg\ p(T))))) \xLeftrightarrow{\text{Inductive assumption, } \neg \text{ distribution}}$$

$$((\neg\ p(T{-}\mathit{1}) \wedge p(T)) \vee (p(T{-}\mathit{1}) \wedge (\neg\ p(T{-}\mathit{1}) \vee p(T)))) \xLeftrightarrow{\wedge \text{ distribution}}$$

$$((\neg\ p(T{-}\mathit{1}) \wedge p(T)) \vee (p(T{-}\mathit{1}) \wedge p(T))) \xLeftrightarrow{\text{Simplification}}$$

$$p(T) \xLeftrightarrow{\text{Boolean FVP definition}}$$

$$p \models \mathsf{holdsAt}(F=V,\,T)$$

‘only if’ direction. Suppose that definitions (R^s, R^e) and p are equivalent, i.e., (R^s, R^e) implies $\mathsf{holdsAt}(F=V,\,T)$ iff p implies $\mathsf{holdsAt}(F=V,\,T)$. Suppose that R^s is not guard condition symmetric. Then, following Definition 8 of the paper, we have:

$$\exists i \in [m] : \exists r \in R^{si} : \exists c \in G^j : ((c \in r) \wedge (\exists c' \in G^j : (c' \neq c) \wedge$$
$$\nexists (r' \in R^{si} : r' \text{ differs from } r \text{ only by having } c' \text{ instead of } c)))$$

Consider a scenario where $F{=}V$ does not hold at $T{-}\mathit{1}$, $\forall l \in U^i$, i.e., the set of FVPs in the inertial conditions of R^{si}, l holds at T, and c' is the only condition in G^j that holds at $T{-}\mathit{1}$. In this case, there is no rule r' in R^s that specifies the initiation of $F{=}V$ at $T{-}\mathit{1}$, based on the satisfaction of the FVP literals in U^i, when the literals in U^j do not hold, due to the satisfaction of c'. Thus, R^s does not initiate $F{=}V$ at $T{-}\mathit{1}$. On the contrary, p is satisfied at T, because all literals in the i-th disjunct of p hold at T. Therefore, p implies that $F{=}V$ holds at T, while (R^s, R^e) implies that $F{=}V$ does not hold at T, based on the law of inertia. This is a contradiction. Thus, R^s is guard condition symmetric. We may prove similarly that R^e is guard condition symmetric.

Suppose that (R^s, R^e) is not Boolean representation symmetric. Then, p and $\overline{p}$, i.e., the Boolean representations of R^s and R^e are not complementary. Thus, there are scenarios where both p and $\overline{p}$ hold or scenarios where none of these formulas hold. In the former case, there exists set R^{si} and R^{ej} for which the FVP literals in their inertial conditions hold at the same time; this leads to the concurrent initiation and termination of $F{=}V$, which is a contradiction. For the latter case, consider a scenario where $F{=}V$ holds at $T{-}1$, we have $\neg p$ and $\neg\overline{p}$ at T. Since p (resp. $\overline{p}$) does not hold at T, there is no set of FVP literals U^i (U'^i) for which all FVP literals in U^i (U'^i) hold at T. Therefore, R^s (R^e) does not imply an initiation (termination) of $F{=}V$ at $T{-}1$ (see the leftward direction of the proof of Lemma 1; this direction holds for all inertial condition symmetric sets). As a result, $F{=}V$ is not initiated at $T{-}1$, and (R^s, R^e) implies that $F{=}V$ holds at T, based on the law of inertia. On the contrary, p is not satisfied at T, and therefore implies that $F{=}V$ does not hold at T. This is a contradiction. Thus, (R^s, R^e) is Boolean representation symmetric.

We have proven that if definitions (R^s, R^e) and p are equivalent, then R^s and R^e are guard condition symmetric and (R^s, R^e) is Boolean representation symmetric. □

Proposition 3 (Compiler Correctness)**.** Given a simple FVP definition (R^s, R^e), if (R^s, R^e) is translatable, then Algorithm 1 of the paper returns a statically determined FVP definition that is equivalent with (R^s, R^e). Otherwise, Algorithm 1 returns false. ◆

Proof. Algorithm 1 of the paper decides whether (R^s, R^e) is translatable in lines 1–4, by checking if (R^s, R^e) satisfies the symmetry conditions of Proposition 2 of the paper, following directly the definitions of these symmetries. Since we have proven that Proposition 2 of the paper is correct, i.e., a simple FVP definition is translatable iff it satisfies guard condition symmetry and Boolean representation symmetry. It follows, that our compiler detects translatable simple FVP definitions correctly. Thus, it returns false iff the input definition (R^s, R^e) is not translatable.

Given a Boolean FVP definition that is equivalent with (R^s, R^e), lines 5–14 of Algorithm 1 of the paper construct a statically determined FVP definition r that is equivalent with R^h. These steps of the algorithm follow the translation steps we outlined in the second part of the proof of Proposition 1, where we proved their correctness. Therefore, if (R^s, R^e) is translatable, then Algorithm 1 returns a statically determined FVP definition that is equivalent with (R^s, R^e). □

2 Complete 'Equivalence Proof for Running Example'

We present all substitutions that take place in equivalences (12)–(16) of the paper. The base of the induction is time-point 0, where without loss of generality, we assume that not FVPs hold, and thus (R^s_m, R^e_m) and p_m imply the same $\mathsf{holdsAt}$ atoms. The complete proof for the inductive step is presented below:

$$
\begin{aligned}
&(R^s_m, R^e_m) \models \mathsf{holdsAt}(\mathit{meeting}(P_1, P_2) = \mathit{interacting},\, T)\\
\Leftrightarrow &(\mathsf{initiatedAt}(\mathit{meeting}(P_1, P_2) = \mathit{interacting}, T-1)\\
&\quad \vee (\mathsf{holdsAt}(\mathit{meeting}(P_1, P_2) = \mathit{interacting},\, T-1) \wedge\\
&\quad \neg\mathsf{terminatedAt}(\mathit{meeting}(P_1, P_2) = \mathit{interacting}, T-1)))\\
\Leftrightarrow &(\mathrm{b}_1@T-1 \vee \mathrm{b}_2@T-1 \vee \mathrm{b}_3@T-1 \vee\\
&\quad (\mathsf{holdsAt}(\mathit{active}(P_1) = \mathsf{true},\, T-1) \wedge \neg\mathrm{b}_4@T-1 \wedge\\
&\quad \mathsf{holdsAt}(\mathit{close}(P_1, P_2) = \mathsf{true},\, T-1) \wedge \neg\mathrm{b}_5@T-1))\\
\Leftrightarrow &((\mathsf{happensAt}(\mathsf{start}(\mathit{active}(P_1) = \mathsf{true}),\, T-1) \wedge\\
&\qquad \mathsf{holdsAt}(\mathit{close}(P_1, P_2) = \mathsf{true},\, T-1) \wedge\\
&\qquad \neg\mathsf{happensAt}(\mathsf{end}(\mathit{close}(P_1, P_2) = \mathsf{true}),\, T-1)) \vee\\
&\quad (\mathsf{happensAt}(\mathsf{start}(\mathit{close}(P_1, P_2) = \mathsf{true}),\, T-1) \wedge\\
&\qquad \mathsf{holdsAt}(\mathit{active}(P_1) = \mathsf{true},\, T) \wedge\\
&\qquad \neg\ \mathsf{happensAt}(\mathsf{end}(\mathit{active}(P_1) = \mathsf{true}),\, T)) \vee\\
&\quad (\mathsf{happensAt}(\mathsf{start}(\mathit{active}(P_1) = \mathsf{true}),\, T-1) \wedge\\
&\qquad \mathsf{happensAt}(\mathsf{start}(\mathit{close}(P_1, P_2) = \mathsf{true}),\, T-1)) \vee\\
&\quad (\mathsf{holdsAt}(\mathit{active}(P_1) = \mathsf{true},\, T-1) \wedge \neg\ \mathsf{happensAt}(\mathsf{end}(\mathit{active}(P_1) = \mathsf{true}),\, T-1) \wedge\\
&\quad \mathsf{holdsAt}(\mathit{close}(P_1, P_2) = \mathsf{true},\, T-1) \wedge \neg\ \mathsf{happensAt}(\mathsf{end}(\mathit{close}(P_1, P_2) = \mathsf{true}),\, T-1)\\
\Leftrightarrow &((\mathsf{happensAt}(\mathsf{start}(\mathit{active}(P_1) = \mathsf{true}),\, T-1) \vee\\
&\quad (\mathsf{holdsAt}(\mathit{active}(P_1) = \mathsf{true},\, T-1) \wedge\\
&\quad \neg\mathsf{happensAt}(\mathsf{end}(\mathit{active}(P_1) = \mathsf{true}),\, T-1))) \wedge\\
&\quad (\mathsf{happensAt}(\mathsf{start}(\mathit{close}(P_1, P_2) = \mathsf{true}),\, T-1) \vee\\
&\quad (\mathsf{holdsAt}(\mathit{close}(P_1, P_2) = \mathsf{true},\, T-1) \wedge\\
&\quad \neg\mathsf{happensAt}(\mathsf{end}(\mathit{close}(P_1, P_2) = \mathsf{true}),\, T-1))))\\
\Leftrightarrow &((\mathsf{initiatedAt}(\mathit{active}(P_1) = \mathsf{true},\, T-1) \vee\\
&\quad (\mathsf{holdsAt}(\mathit{active}(P_1) = \mathsf{true},\, T-1) \wedge\\
&\quad \neg\mathsf{terminatedAt}(\mathit{active}(P_1) = \mathsf{true},\, T-1))) \wedge\\
&\quad (\mathsf{initiatedAt}(\mathit{close}(P_1, P_2) = \mathsf{true},\, T-1) \vee\\
&\quad (\mathsf{holdsAt}(\mathit{close}(P_1, P_2) = \mathsf{true},\, T-1) \wedge\\
&\quad \neg\mathsf{terminatedAt}(\mathit{close}(P_1, P_2) = \mathsf{true},\, T-1))))\\
\Leftrightarrow &\ (\mathsf{holdsAt}(\mathit{active}(P_1) = \mathsf{true},\, T) \wedge \mathsf{holdsAt}(\mathit{close}(P_1, P_2) = \mathsf{true},\, T))\\
\Leftrightarrow &\ p_m \models \mathsf{holdsAt}(\mathit{meeting}(P_1, P_2) = \mathit{interacting}, T)
\end{aligned}
$$

Below, we prove the third to last equivalence, i.e., the substitution of predicates $\mathsf{happensAt}(\mathsf{start}(F = V),\, T)$ and $\mathsf{happensAt}(\mathsf{end}(F = V),\, T)$ with predicates $\mathsf{initiatedAt}(F = V,\, T)$ and $\mathsf{terminatedAt}(F = V,\, T)$, respectively, in this particular context.

$$
\begin{aligned}
&(\dots \wedge \mathsf{initiatedAt}(F=V,T) \wedge \dots) \vee\\
&(\dots \wedge \mathsf{holdsAt}(F=V,T) \wedge \neg\ \mathsf{terminatedAt}(F=V,T) \wedge \dots) \Leftrightarrow\\
&(\dots \wedge \mathsf{initiatedAt}(F=V,T) \wedge (\neg\ \mathsf{holdsAt}(F=V,T) \vee \mathsf{holdsAt}(F=V,T)) \wedge \dots) \vee\\
&(\dots \wedge \mathsf{holdsAt}(F=V,T) \wedge \neg\ \mathsf{terminatedAt}(F=V,T) \wedge \dots) \Leftrightarrow\\
&(\dots \wedge \mathsf{initiatedAt}(F=V,T) \wedge \neg\ \mathsf{holdsAt}(F=V,T) \wedge \dots) \vee\\
&(\dots \wedge \mathsf{initiatedAt}(F=V,T) \wedge \mathsf{holdsAt}(F=V,T) \wedge \dots) \vee\\
&(\dots \wedge \mathsf{holdsAt}(F=V,T) \wedge \neg\ \mathsf{terminatedAt}(F=V,T) \wedge \dots) \xLeftrightarrow{\mathsf{start}\ \text{definition} \wedge \text{no concurr init term}}\\
&(\dots \wedge \mathsf{happensAt}(\mathsf{start}(F=V),T) \wedge \dots) \vee\\
&(\dots \wedge \mathsf{holdsAt}(F=V,T) \wedge \neg\ \mathsf{terminatedAt}(F=V,T) \wedge \dots) \xLeftrightarrow{\mathsf{holdsAt} \rightarrow \neg\ \mathsf{terminatedAt} \leftrightarrow \neg\ \mathsf{end}}\\
&(\dots \wedge \mathsf{happensAt}(\mathsf{start}(F=V),T) \wedge \dots) \vee\\
&(\dots \wedge \mathsf{holdsAt}(F=V,T) \wedge \neg\ \mathsf{happensAt}(\mathsf{end}(F=V),T) \wedge \dots)
\end{aligned}
$$

Similarly, we may prove the following equivalence:

$$
\begin{aligned}
&(\dots \wedge \mathsf{terminatedAt}(F=V,T) \wedge \dots) \vee\\
&(\dots \wedge \neg\ \mathsf{holdsAt}(F=V,T) \wedge \neg\ \mathsf{initiatedAt}(F=V,T) \wedge \dots) \Leftrightarrow\\
&(\dots \wedge \mathsf{happensAt}(\mathsf{end}(F=V),T) \wedge \dots) \vee\\
&(\dots \wedge \neg\ \mathsf{holdsAt}(F=V,T) \wedge \neg\ \mathsf{happensAt}(\mathsf{start}(F=V),T) \wedge \dots)
\end{aligned}
$$